

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Новиков Денис Владимирович  
Должность: Директор филиала  
Дата подписания: 11.11.2024 11:28:14  
Уникальный идентификатор:  
3357c68ce48ec4f695c95289ac7a9678e502be60

Федеральное агентство морского и речного транспорта  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«Волжский государственный университет водного транспорта»

Кафедра информатики, систем управления и телекоммуникаций

Серия «Информационные технологии  
в системах управления и телекоммуникации»

Выпуск 20

Р.Н. Гурьяшова

# Лабораторный практикум по информатике

Учебное пособие  
для студентов очного и заочного обучения  
технических специальностей

Рекомендовано

федеральным государственным бюджетным образовательным учреждением высшего профессионального образования «Московский государственный технический университет имени Н.Э. Баумана» в качестве учебного пособия для студентов высших учебных заведений, обучающихся по специальности 180407.65 «Эксплуатация судового оборудования и средств автоматики» и направлениям подготовки 180100.62 «Кораблестроение, океанотехника и системотехника объектов морской инфраструктуры»; 190600.62 «Эксплуатация транспортно-технологических машин и комплексов»

Рег. номер рецензии 2973 от 02.02.2015 г. МГУП

Нижний Новгород  
Издательство ФГБОУ ВО «ВГУВТ»  
2015

**УДК 004.42**

**Г95**

Редакционная коллегия серии «Информационные технологии в системах управления и телекоммуникаций»:

д. т. н., профессор *Ю.С. Федосенко* (отв. редактор),

д. т. н., профессор *М.М. Чиркова*,

к. т. н., доцент *В.И. Логинов*,

к. т. н., доцент *А.В. Преображенский*

**Гурьяшова, Р.Н.**

Лабораторный практикум по информатике : учеб. пособие для студ. оч. и заоч. обуч. технич. специальностей / Р.Н. Гурьяшова. – Н. Новгород : Изд-во ФГБОУ ВО «ВГУВТ», 2015. – 144 с.

Цель пособия – формирование у студентов начальных навыков алгоритмизации и программирования, составляющих основу профессиональной деятельности современного инженера. Методическое изложение материала основано на многолетнем опыте преподавания дисциплины «Информатика» студентам технических специальностей в ВГАВТ.

Излагаются основные типовые приёмы решения задач и структуры базовых алгоритмов. Алгоритмы приводятся как в словесном описании, так и в виде схем. Приводятся также листинги программ, иллюстрирующие выполнение лабораторных работ на императивных языках программирования высокого уровня – С, Fortran и Basic.

*Для студентов, обучающихся по техническим специальностям и направлениям подготовки в высших учебных заведениях водного транспорта.*

Работа рекомендована к изданию кафедрой информатики, систем управления и телекоммуникации (протокол № 5 от 15.05.2014).

© ФГБОУ ВО «ВГУВТ», 2015

# Введение

*«Единственный способ изучать новый язык программирования – писать на нём программы»*

*Брайэн Керниган*

В инженерной практике необходимо умение ставить задачи, разрабатывать алгоритмы их решения, выполнять расчёты, а также анализировать полученные результаты. Поэтому инженер должен грамотно применять персональный компьютер для решения различных возникающих в профессиональной деятельности задач.

В пособии рассматриваются типовые приёмы и структуры линейных, разветвлённых и циклических алгоритмов. Для иллюстрации алгоритмов приводятся следующие лабораторные работы:

- **Линейный алгоритм**
  - «Вычисление значения выражения» (1 лаб. работа)
- **Разветвлённый алгоритм**
  - «Вычисление значения величины, заданной ветвлением» (2)
- **Циклический алгоритм**
  - «Табулирование функции одной переменной» (3)
  - «Табулирование функции двух переменных» (2)
  - «Одномерный массив» (6)
  - «Двумерный массив – матрица» (5)

Для каждой лабораторной работы приводятся листинги программ на языках программирования высокого уровня – С, Fortran, Basic, используемых в учебном процессе. Приведённые программы с большим количеством комментариев, поясняющих программный код, послужат студентам справкой при выполнении заданий.

Кроме того, по каждой теме приведены контрольные задания для проверки знаний, а также задания для самостоятельной работы.

Тема «Сегментация программ» в пособии не рассматривается (см. стр.140, п.5).

Все программы тестировались в средах Microsoft Visual Studio С, QBasic 1.0 (1.1), GNU g95 Fortran compiler. Программы могут быть легко адаптированы для других версий языков.

Цель пособия – научить студентов пользоваться инструментарием языков программирования при решении задач.

# 1. Рекомендации к лабораторной работе

**Для выполнения лабораторной работы необходимо:**

1. В соответствии с заданием и типом алгоритма ввести обозначения переменных. Например, сумму, произведение, количество, среднее арифметическое, наибольшее, наименьшее значение с учётом смысла удобнее обозначить как: S, P, K, SR, MAX, MIN.
2. Имена всех переменных, используемых в задаче, свести в таблицу с пояснением их смысла.
3. Выполнить ручной расчёт результатов. Задания подобраны так, что ручной расчёт не является трудоёмким.
4. Разработать схему алгоритма решения задачи.
5. Составить макет печати исходных данных и результатов, предусмотрев при этом поясняющие текстовые заголовки.
6. В соответствии со схемой и макетом печати написать программу. Для повышения её читаемости рекомендуется грамотное использование:
  - отступов при записи программного кода (правило рельефа);
  - осмысленных и кратких имён переменных;
  - комментариев, поясняющих программный код.
7. Отладить и выполнить программу.
8. Проанализировать полученные результаты. Ручной и машинный расчёты должны совпадать.
9. Представить отчёт-презентацию лабораторной работы.

**Презентация состоит из следующих разделов:**

1. Постановка задачи.
2. Таблица имён переменных с пояснением их смысла.
3. Ручной расчёт результатов (контрольных точек).
4. Схема алгоритма и программа решения задачи.
5. Распечатка (макет печати) исходных данных и результатов.

В качестве примера оформления приведён отчёт для работы 1 (стр.137).

## 2. Примеры выполнения лабораторных работ

### Лабораторная работа 1

### Вычисление значения выражения

### Тема. Линейный алгоритм

Индивидуальные задания – табл. 3.1 (стр.92)

Математические функции – табл. 5.1 (стр.136)

#### 1. Задание

Составить программу решения следующей задачи.

Вычислить значение величины  $Z$ :

$$Z = (\ln |b| + \cos a)^2 - \frac{\sqrt{\ln 3 + \cos b}}{a (\ln a + \cos 2^\circ)}, \text{ где } a = 1,5 \quad b = -2.$$

#### 2. Решение

Для удобства отладки программы введем обозначения:

$$R = \ln |B| + \cos A; \quad W = \ln 3 + \cos B; \quad Q = \ln A + \cos 2^\circ.$$

Тогда вычисление  $Z$  можно представить в виде:

$$Z = R^2 - \frac{\sqrt{W}}{A \cdot Q}$$

Обозначения переменных			
Имя переменной	Тип переменной		Пояснение
	Fortran	C	
A, B	REAL	FLOAT	Исходные данные
R, W, Q, Z	REAL	FLOAT	Рабочие переменные и результат

**Примечание.** У тригонометрических функций аргумент указывается в радианах. Градусы переводятся в радианы по формуле  $\frac{x \cdot \pi}{180}$ . Например,  $\cos 2^\circ$  следует записать как  $\text{COS}(2*3.1415/180)$ .

#### 3. Алгоритм решения задачи

Алгоритм имеет линейную структуру. Представим словесное описание и схему алгоритма (рис. 2.1):

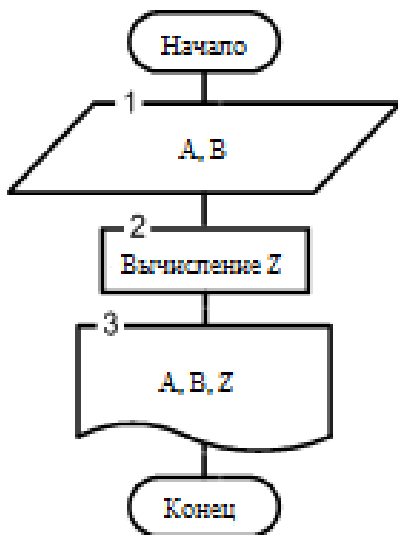


Рис. 2.1. Схема алгоритма к лабораторной работе 1 «Вычисление значения выражения»

Алгоритм состоит из следующих действий:

- 1) Ввод данных – значений A, B;
- 2) Вычисление результата Z;
- 3) Вывод значений A, B, Z;
- 4) Конец.

Приведем два варианта C-программы: с функцией и без использования функции. Выделим в функцию с именем **f** повторяющееся арифметическое выражение вида

$$\ln x + \cos y$$

В процессе вычисления значения Z обращение к функции **f** происходит три раза, а именно:

Имя переменной	Выражение	Вызов функции F
R	$\ln  B  + \cos A$	$F( B , A)$
W	$\ln 3 + \cos B$	$F(3, B)$
Q	$\ln A + \cos 2^\circ$	$F(A, 2^\circ)$

#### 4. Листинг C-программы

**Внимание.** При вводе программы следует иметь в виду, что C различает регистр букв, т.е., «x» и «X» – это две разные переменные.

Без функции	С функцией
<b>Программный код</b>	
<pre> #include&lt;stdio.h&gt; #include&lt;conio.h&gt; #include&lt;math.h&gt; void main() { // Объявление типа данных </pre>	<pre> #include&lt;stdio.h&gt; #include&lt;conio.h&gt; #include&lt;math.h&gt; // Прототип функции F float f(float x, float y); void main() </pre>

<pre> double a = 1.5,b = -2; const double pi = 3.1415; double r,w,q,z; // Вычисление R,W,Q,Z r = log(abs(b))+ cos(a); w = log(3.0)+ cos(b); q = log(a)+cos(2*pi/180); z = r*r - sqrt(w)/(a*q); // Вывод A,B,Z printf(" A = %.1lf",a); printf(" B = %.0lf\n",b); printf(" Z = %.4lf",z); getch(); return; } </pre>	<pre> { // Объявление типа данных float a = 1.5,b = -2; const float pi = 3.1415; float r,w,q,z; // Вызов функции F r = f(abs(b),a); w = f(3.0,b); q = f(a,2*pi/180); z = r*r - sqrt(w)/(a*q); // Вывод A,B,Z printf(" A = %.1f ",a); printf(" B = %.0f\n ",b); printf(" Z = %.4f ",z); getch(); return; } // Функция F float f(float x,float y) { return(log(x)+cos(y)); } </pre>
---	---

## 5. Листинг Fortran и Basic-программ

Программный код	
Fortran	Basic
<pre> ! Объявление данных REAL :: A = 1.5 ; B = -2 REAL :: PI = 3.1415 REAL :: R,W,Q,Z ! Вычисление R,W,Q,Z R = LOG(ABS(B))+ COS(A) W = LOG(3.0)+ COS(B) Q = LOG(A)+ COS(2*PI/180) Z = R**2 - SQR(W)/(A*Q) ! Вывод значений A,B,Z PRINT 1, A,B,Z 1 FORMAT(1X,' A = ', &amp; F3.1,' B = ',F3.0/ &amp; 1X,'Z = ',E11.4 ) END </pre>	<pre> CLS ' Исходные данные A = 1.5: B = -2 PI = 3.1415 ' Вычисление R,W,Q,Z R = LOG(ABS(B))+ COS(A) W = LOG(3) + COS(B) Q = LOG(A)+ COS(2*PI/180) Z = R^2 - SQR(W)/(A*Q) ' Вывод значений A,B,Z PRINT " A = ";A; PRINT " B = ";B PRINT " Z = ";Z END </pre>

## 6. Макет печати результатов

### По C-программе

```
A = 1.5 B = -2
Z = 0.1915
```

### По Fortran-программе

```
A = 1.5 B = -2.
Z = 0.1915E+00
```

## Лабораторная работа 2

### Вычисление значения величины, заданной в форме ветвления

#### Тема. Разветвлённый алгоритм

Индивидуальные задания – табл. 3.2 (стр.94)

Математические функции – табл. 5.1 (стр.136)

### 1. Задание

Составить схему алгоритма и программу для следующей задачи. Вычислить значение величины Z, заданной ветвлением вида:

$$Z = \begin{cases} a \cdot \min(2 \cdot b, k) & \text{при } a < b \text{ и } a = 1 \\ a \cdot b - k^3 & \text{при } a < b \text{ и } a \neq 1 \\ a^2 + \frac{b}{k} & \text{при } a \geq b \text{ и } b \geq 0 \\ \sqrt{|b|} + a \cdot k & \text{при } a \geq b \text{ и } b < 0 \end{cases}$$

Расчёт значений Z приведён в следующей таблице:

Исходные данные			Формула для Z	Значение Z
a	b	k		
1	3	4	$a \cdot \min(2b, k)$	4
2	4	1	$a \cdot b - k^3$	7
3	2	1	$a^2 + b/k$	11
1	-4	3	$\sqrt{ b } + a \cdot k$	5

### 2. Решение

Обозначения переменных			
Имя переменной	Тип переменной		Пояснение
	Fortran	C	
A, B, K	REAL	FLOAT	Исходные данные
Z	REAL	FLOAT	Искомая величина, результат



Алгоритм имеет разветвленную структуру и содержит пять ветвей (рис. 2.2). Каждая ветвь соответствует одному из условий задачи и определяет путь («маршрут») следования вычислительного процесса в соответствии с проверяемым условием. «Маршруты» для каждой из ветвей приведены в следующей таблице.

Условие	Формула для Z	<Маршрут> Номера блоков в схеме алгоритма
$A < B$ и $A = 1, 2B < K$	$a \cdot \min(2 \cdot b, k)$	1 – 2 – 3 – 4 – 5 – 11
$A < B$ и $A = 1, 2B \geq K$	$a \cdot \min(2 \cdot b, k)$	1 – 2 – 3 – 4 – 6 – 11
$A < B$ и $A \neq 1$	$a \cdot b - k^3$	1 – 2 – 3 – 7 – 11
$A \geq B$ и $B \geq 0$	$a^2 + b/k$	1 – 2 – 8 – 10 – 11
$A \geq B$ и $B < 0$	$\sqrt{ b } + a \cdot k$	1 – 2 – 8 – 9 – 11

Схема алгоритма представлена на рис. 2.2:

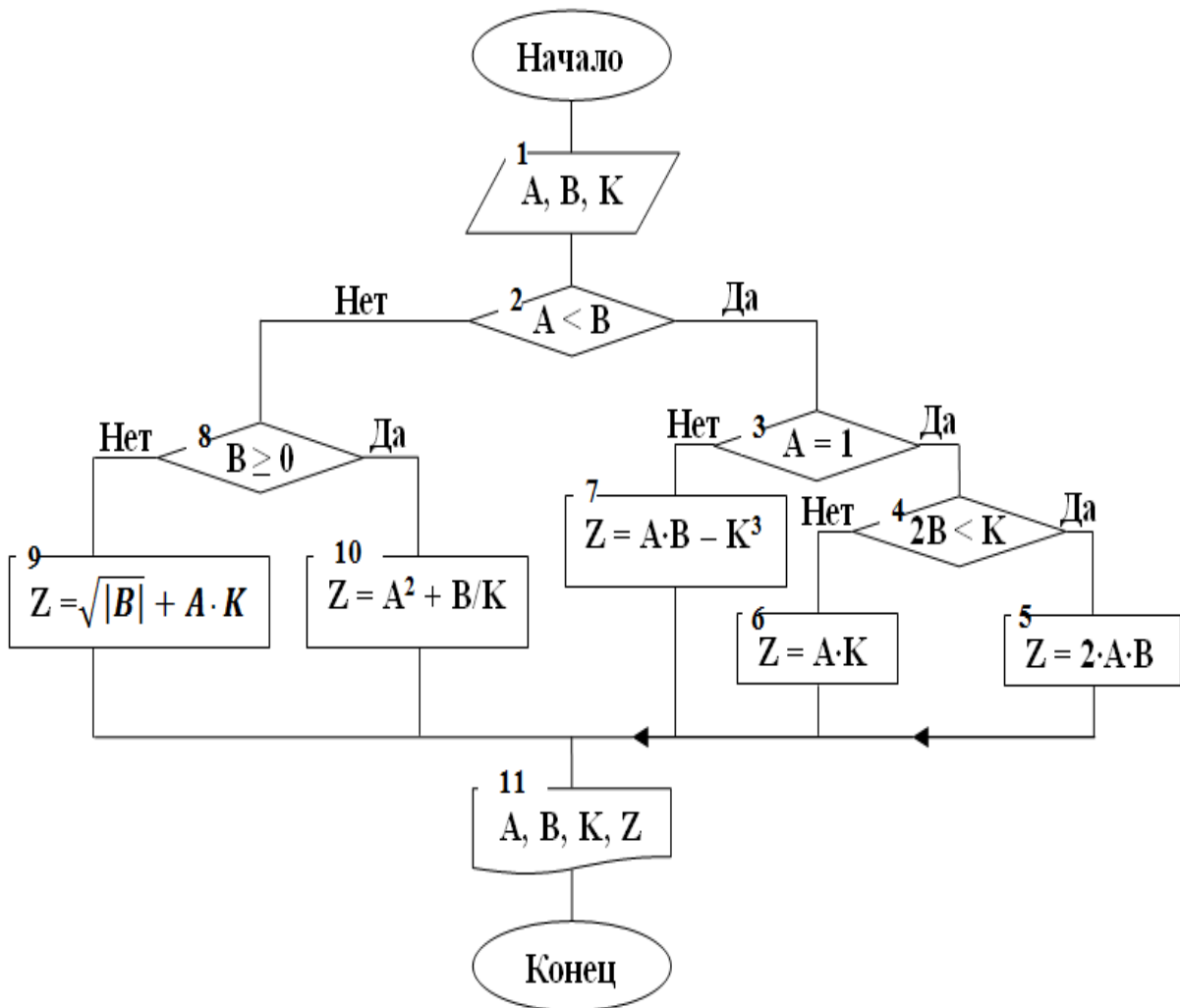


Рис. 2.2 Схема алгоритма к лабораторной работе 2

«Вычисление значения величины, заданной в форме ветвления»

### 3. Листинг С-программы «Разветвлённый алгоритм»

**Внимание.** При вводе программы следует иметь в виду, что С различает регистр букв, т.е., «x» и «X» – это две разные переменные.

```
#include<stdio.h> // Файлы заголовков
#include<conio.h>
#include<math.h>
void main()
{
float a,b,k,z; // Объявление типа данных
// Ввод данных с клавиатуры по запросу с экрана,
// числа при вводе разделяются запятой
printf(" vvedi A,B,K? \n ");
scanf("%f,%f,%f",&a,&b,&k);
printf(" A = %.0f B = %.0f K = %.0f \n",a,b,k);
if(a<b) // Если A < B
    if(a == 1) // Если A = 1
        if(2*b < k) z = 2*b*a; // Вычисление MIN(2·B,K)
        else z = a*k;
    else // Иначе,если A ≠ 1
        z = a*b - pow(k,3);
else // Иначе,если A ≥ B
    if(b ≥ 0) // Если B ≥ 0
        z = a*a + b/k;
    else // Иначе,если B < 0
        z = sqrt(abs(b)) + a*k;
printf(" Result: Z = %.0f ",z); // ВЫВОД Z
getch();
return;
}
```

### 4. Макет печати (по С-программе)

<pre>vvedi A,B,K? 1,3,4 A = 1 B = 3 K = 4 Result: Z = 4</pre>	<pre>vvedi A,B,K? 2,4,1 A = 2 B = 4 K = 1 Result: Z = 7</pre>
<pre>vvedi A,B,K? 3,2,1 A = 3 B = 2 K = 1 Result: Z = 11</pre>	<pre>vvedi A,B,K? 1,-4,3 A = 1 B = -4 K = 3 Result: Z = 5</pre>

Как видно, результаты ручного и машинного расчётов совпадают.

### 5. Листинг Basic-программы «Разветвлённый алгоритм»

```

                                Программный код
INPUT "Введи А, В, К"; А, В, К
IF А < В THEN ' Если А < В
  IF А = 1 THEN ' Если А = 1
    IF 2 * В < К THEN Z = 2 * В * А ELSE Z = К * А
  ELSE ' Иначе, если А ≠ 1
    Z = А * В - К ^ 3
  END IF
ELSE ' Иначе, если А ≥ В
  IF В ≥ 0 THEN ' Если В ≥ 0
    Z = А ^ 2 + В / К
  ELSE ' Иначе, если В < 0
    Z = SQR (ABS (В)) + А * К
  END IF
END IF
PRINT " А = "; А; " В = "; В; " С = "; С
PRINT USING " РЕЗУЛЬТАТ Z = ## "; Z
END
```

Результаты приведены в макете печати в п.4.

### Лабораторная работа 3

## Вычисление значения величины, заданной в форме ветвления

### Тема. Разветвлённый алгоритм

Индивидуальные задания – табл. 3.3 (стр.97)

Математические функции – табл. 5.1 (стр.136)

### 1. Задание

Составить схему алгоритма и программу для следующей задачи.

Вычислить значение величины Z, заданной ветвлением вида:

$$Z = \begin{cases} \min \{a, b \cdot \max(2a, c^2)\} & \text{при } c > b \\ c + \min(a, b) & \text{при } c \leq b \text{ и } a \neq 1 \\ c \cdot \max(2b, \sqrt{|a|}) & \text{при } c \leq b \text{ и } a = 1 \end{cases}$$

Расчёт значений Z приведён в следующей таблице:

Исходные данные			Формула для Z	Значение Z
a	b	c		
1	2	3	$\min\{a, b \cdot \max(2a, c^2)\}$	1
4	3	1	$c + \min(a, b)$	4
1	2	1	$c \cdot \max(2b, \sqrt{ a })$	4

## 2. Решение

Обозначения переменных			
Имя переменной	Тип переменной		Пояснение
	Fortran	C	
A, B, C	REAL	FLOAT	Исходные данные
Z	REAL	FLOAT	Искомая величина, результат

Алгоритм имеет разветвленную структуру и содержит восемь ветвей. В соответствии со схемой (рис. 2.3) необходимо *самостоятельно* представить все его ветви (см. лаб. работу 2, стр.9).

Схема алгоритма решения задачи представлена на рис. 2.3:

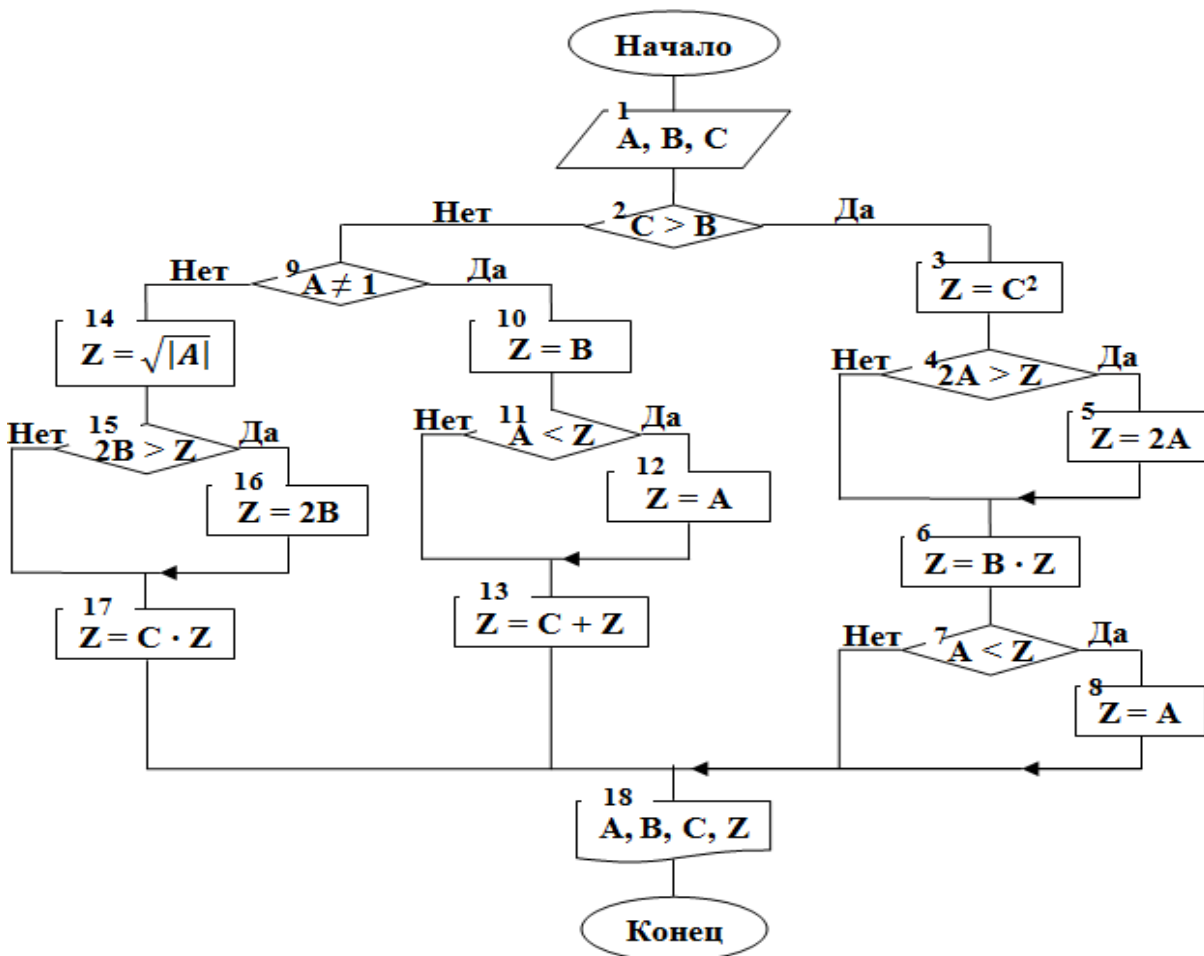


Рис. 2.3 Схема алгоритма к лабораторной работе 3

«Вычисление значения величины, заданной в форме ветвления»  
3. Листинг С-программы «Разветвлённый алгоритм»

**Внимание.** При вводе программы следует иметь в виду, что *C* различает регистр букв, т.е., «*x*» и «*X*» – это две разные переменные.

```
#include<stdio.h> // Файлы заголовков
#include<conio.h>
#include<math.h>
void main()
{
float a,b,c,z; // Объявление типа данных
// Ввод данных с клавиатуры по запросу с экрана,
// числа при вводе разделяются запятой
printf(" Input A,B,C ?\n ");
scanf("%f,%f,%f",&a,&b,&c);
// Вычисление Z = MIN(A,B*MAX(2*A,C*C))
if(c > b) // Если C > B
{
z = c*c;
if(2*a > z) z = 2*a;
z = b*z;
if (a < z) z = a;
}
else // Иначе,если C <= B
// Вычисление Z = C + MIN(A,B)
if(a != 1) // Если A ≠ 1
{
z = b;
if (a < z) z = a;
z = c + z;
}
else // Иначе,если A = 1
// Вычисление Z = C*MAX(2*B,√|A|)
{
z = sqrt(abs(a));
if (2*b > z) z = 2*b;
z = c*z;
}
// Вывод данных A, B, C и результата Z
printf(" A = %.0f B = %.0f C = %.0f \n ",a,b,c);
printf(" Result: Z = %.0f ",z);
getch();
return;
```

```
}
```

#### 4. Макет печати (по С-программе)

```
Input A,B,C ?  
1,2,3  
A = 1 B = 2 C = 3  
Result: Z = 1
```

```
Input A,B,C ?  
4,3,1  
A = 4 B = 3 C = 1  
Result: Z = 4
```

```
Input A,B,C ?  
1,2,1  
A = 1 B = 2 C = 1  
Result: Z = 4
```

Как видно, результаты ручного и машинного расчётов совпадают.

#### 5. Листинг Fortran-программы

##### Программный код

```
! Разветвленный алгоритм  
PROGRAM VETVIZ  
REAL :: A,B,C,Z ! Объявление типа данных  
! Ввод данных по запросу с экрана  
! числа при вводе разделяются пробелом  
PRINT *, ' Введите A,B,C ?'  
READ (*,*)A,B,C  
IF (C > B) THEN ! Если C > B  
! Вычисление Z = MIN(A, B*MAX(2*A, C**2))  
Z = C ** 2  
IF (2*A > Z) Z = 2*A  
Z = B * Z  
IF (A < Z) Z = A  
ELSE ! Иначе если C <= B  
! Вычисление Z = C + MIN(A,B)  
IF (A /= 1) THEN ! Если A ≠ 1  
Z = B  
IF (A < Z) Z = A  
Z = C + Z  
ELSE ! Иначе, если A = 1  
! Вычисление Z = C*MAX(2*B,  $\sqrt{|A|}$ )  
Z = SQRT (ABS (A))  
IF (2*B > Z) Z = 2*B  
Z = C * Z  
ENDIF  
ENDIF  
! Вывод данных A, B, C и результата Z  
WRITE (*, 3) A,B,C,Z  
3 FORMAT (2X, ' A = ',F3.0, ' B = ',F3.0, &  
' C = ',F3.0/ ' РЕЗУЛЬТАТ Z = ',F3.0)  
END
```

Результаты приведены в макете печати в п.4.

## Контрольные задания для проверки знаний

### Тема. Разветвлённый алгоритм

Для п.1-3: Исходные данные и результат представлены в таблице. При несовпадении результата необходимо исправить ошибки и повторно выполнить программу.

Составить схему алгоритма и программу для следующих задач:

#### 1. Вычислить значение сложной функции $Y = F(X)$ .

$$Y = \begin{cases} X + 4 & \text{при } X < -6 \\ 2 \cdot X + 10 & \text{при } -6 \leq X < 0 \\ 10 - X & \text{при } 0 \leq X < 5 \\ 5 & \text{при } X \geq 5 \end{cases}$$

X	Y
-8	-4
-1	8
4	6
7	5

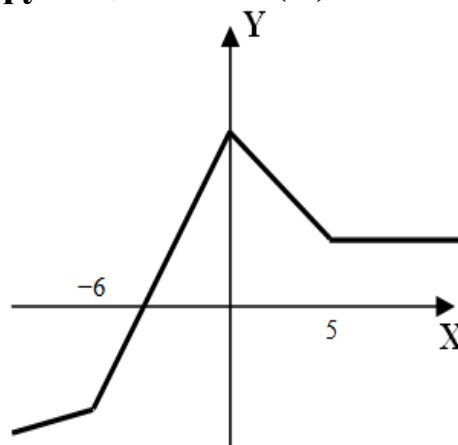


Рис. 2.4 «Вид функции  $Y = F(X)$ »

Для проверки программу необходимо выполнить четыре раза для каждого значения аргумента X.

#### 2. Вычислить значение величины Z, заданной ветвлением.

$$Z = \begin{cases} \max(A; 3 \cdot B) & \text{при } A > B \\ \min(2 \cdot A; B^2) & \text{при } A \leq B \end{cases}$$

Данные		Результат
A	B	Z
4	1	4
1	3	2

Для проверки программу необходимо выполнить дважды.

#### 3. Вычислить значение сложной функции $Y = F(X)$ .

$$Y = \begin{cases} X^2 + 1 & \text{при } X < -1 \\ \sqrt{|X|} & \text{при } -1 \leq X \leq 1 \\ X^3 - 2 & \text{при } X > 1 \end{cases}$$

X	Y
-2	5
-1	1
2	6

Для проверки программу необходимо выполнить три раза, для каждого из заданных значений  $X$ .

**4. Вычислить значение величины  $Z = C \cdot \min\{2 \cdot A, \max(A^2, C), B^3\}$ .**

Исходные данные задать самостоятельно.

**5. Определить попадание точки  $(X, Y)$  в область, ограниченную параболой  $F(X) = -X^2 + 4$  и прямой  $G(X) = 0$ .**

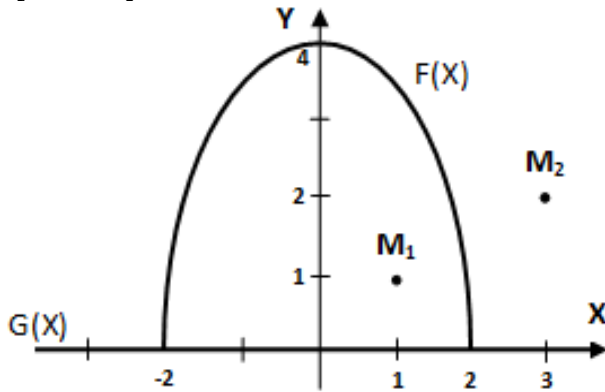


Рис. 2.5 «Вид области»

Необходимо в соответствии с графиком записать условие принадлежности точки  $(X, Y)$  заданной области в виде ветвления.

Для проверки программу следует выполнить для двух точек, например,  $M_1 (1; 1)$  и  $M_2 (3; 2)$ . Согласно рис. 2.5, точка  $M_1 (1; 1)$  попадает в заданную область, а точка  $M_2 (3; 2)$  — нет.

**6. Определить принадлежность точки  $(X, Y)$  области, ограниченной прямыми  $X = -2$ ,  $X = 2$ ,  $Y = -2$ ,  $Y = X$  и  $Y = -X$ .** Область представлена на рис. 2.6.

Необходимо в соответствии с графиком записать условие принадлежности точки  $(X, Y)$  заданной области в виде ветвления.

Для проверки программу следует выполнить для двух точек, например,  $M_1 (1; -1)$  и  $M_2 (0; 2)$ . Согласно рис. 2.6, точка  $M_1 (1; -1)$  лежит внутри заданной области, а точка  $M_2 (0; 2)$  — вне её.



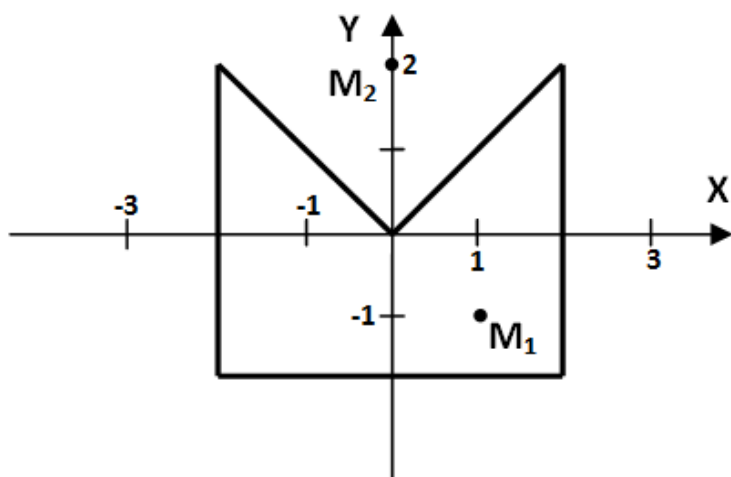


Рис. 2.6 «Вид области»

#### Лабораторная работа 4

### Исследование функции одной переменной

#### Тема. Табулирование функции одной переменной

Индивидуальные задания – табл. 3.4 (стр.100)

Математические функции – табл. 5.1 (стр.136)

#### 1. Задание

Составить программу решения следующей задачи.

Вычислить таблицу значений аргумента  $X$  и функции  $Y = F(X)$  при условии, что аргумент  $X$  изменяется в интервале  $X_{\min} \leq X \leq X_{\max}$  с шагом  $\Delta X$ . Протабулировать функцию и найти наибольшее из её отрицательных значений и соответствующее значение аргумента  $X$  для функции вида

$$F(x) = \sqrt[3]{2x} - e^{-x} \cdot \operatorname{tg}^2(3x)$$

#### Исходные данные

Начальное значение аргумента	Конечное значение аргумента	Шаг изменения аргумента
$X_{\min}$	$X_{\max}$	$\Delta X$
0,6	0,8	0,05

#### 2. Решение

Обозначим:  $Y_{\max}$ ,  $X_m$  – наибольшее из отрицательных значений функции и соответствующее ему значение аргумента  $X$ .

Обозначения переменных				
Имя переменной		Тип переменной		Пояснения
в задаче	в программе	Fortran	C	
$X_{\min}, X_{\max}$	XMIN, XMAX	REAL	DOUBLE	Начальное и конечное значение аргумента X
$\Delta X$	DX	REAL	DOUBLE	Шаг изменения аргумента X
X, Y	X, Y	REAL	DOUBLE	Значение аргумента и функции
$Y_{\max}, X_m$	YMAX, XM	REAL	DOUBLE	Наибольшее из отрицательных значений функции и соответствующее значение аргумента X

### 3. Ручной расчёт результатов

Вычислим значение  $F(X)$  в начальной и конечной точках.

Контрольные точки	Значение X	Значение Y
$X_{\min}$	0,6	-9.02
$X_{\max}$	0,8	0.793

Результаты можно представить графически и в виде таблицы.

#### Графический способ

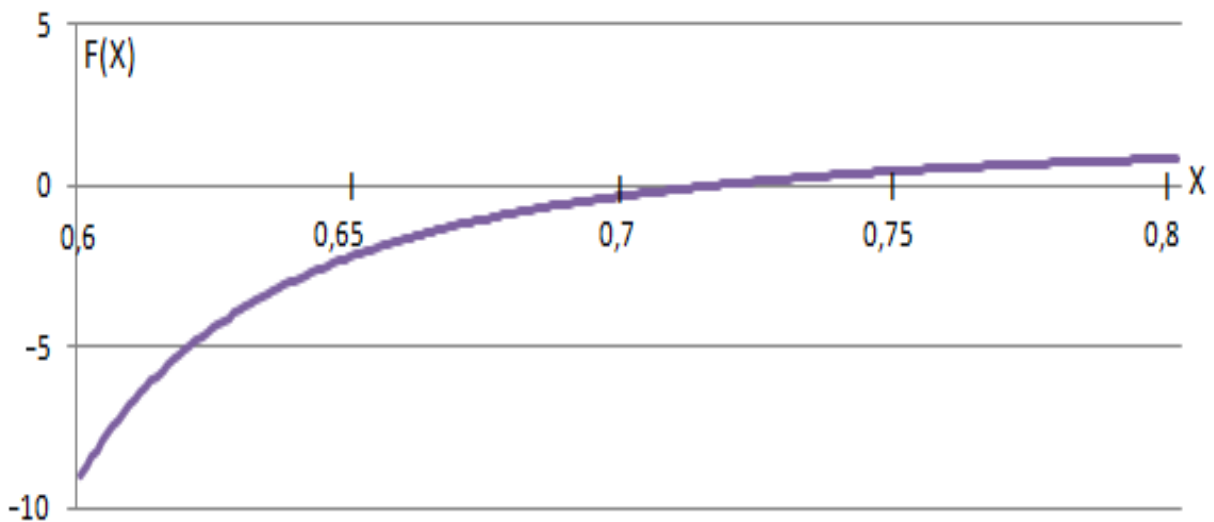


Таблица значений аргумента X и функции  $Y = F(X)$

X	Y
0,6	-9,02

Количество значений (x;y) [строк в таблице]

0,65	-2,196
<b>0,7</b>	<b>-0,333</b>
0,75	0,42
0,8	0,793

определяется как:

$$\frac{X_{\max} - X_{\min}}{\Delta X} + 1 = \frac{0,8 - 0,6}{0,05} + 1 = 5$$

Наибольшее из отрицательных значений функции  $Y_{\max} = -0,333$  достигается при  $X_m = 0,7$ .

Для решения задачи используется циклический алгоритм табулирования функции и вычисления наибольшего значения.

#### 4. Алгоритм

**Алгоритм состоит из следующих действий:**

1. Ввод исходных данных – значений переменных  $X_{\min}$ ,  $X_{\max}$ ,  $\Delta X$ .
2. Задание начального значения  $Y_{\max} = -1.0E38$  (заведомо малого числа  $-10^{38}$ ).
3. *Начало цикла по X.*
4. Вычисление значения функции  $Y = F(X)$ .
5. Вывод текущего значения аргумента  $X$  и функции  $Y$ .
6. Проверка условия  $Y < 0$ . Если условие выполняется – перейти к п.7, иначе – к п.9.
7. Проверка условия  $Y > Y_{\max}$ . Если условие выполняется – перейти к п.8, иначе – к п.9.
8. Присвоить значения переменным:  $Y_{\max} = Y$ ,  $X_m = X$ .
9. *Конец цикла по X.*
10. Вывод результатов – значений  $Y_{\max}$ ,  $X_m$ .
11. Конец.

Схема алгоритма представлена на рис. 2.7 (стр.21).

#### 5. Макет печати результатов

По С-программе

По Basic-программе

---

<pre> Input XMIN XMAX DX ? 0.6 0.8 0.05        X           Y I 0.60 I -9.020 I I 0.65 I -2.196 I I 0.70 I -0.333 I I 0.75 I  0.420 I I 0.80 I  0.793 I YMAX = -0.333 XN = 0.70 </pre>	<pre> ГРАНИЦЫ ИНТЕРВАЛА 0.6 &lt;= X &lt;= 0.8 ШАГ DX = 0.05 ТАБЛИЦА X = 0.60      Y = -9.020E+00 X = 0.65      Y = -2.196E+00 <b>X = 0.70      Y = -3.331E-01</b> X = 0.75      Y =  4.200E-01 X = 0.80      Y =  7.925E-01 YMAX = -3.331E-01 при X = 0.70 </pre>
---	---

Как видно, результаты ручного и машинного расчётов совпадают.

## 6. Листинг С-программы «Табулирование функции»

**Внимание.** При вводе программы следует иметь в виду, что С различает регистр букв, т.е., «x» и «X» – это две разные переменные.

Условия  $Y < 0$  и  $Y > YMAX$  объединены логической связкой **&&**.

```

#include<stdio.h> // Файлы заголовков
#include<conio.h>
#include<math.h>
void main()
{
double xmin, xmax, dx; // Объявление типа данных
double x, y, xm, ymax = -1.0e38;
// Ввод данных с клавиатуры по запросу с экрана,
// числа при вводе разделяются пробелом
printf(" Input XMIN XMAX DX ?\n ");
scanf("%f %f %f", &xmin, &xmax, &dx);
printf("      X           Y \n"); // Вывод заголовка
for(x = xmin; x <= xmax + dx/3; x += dx) // Цикл по X
{
y = pow((2.*x), (1./3.)) - exp(-x) * pow(tan(3*x), 2);
// Вывод значений аргумента X и функции Y
printf(" I %.21f I %6.31f I \n ", x, y);
if(y < 0 && y > ymax) // Если Y < 0 и Y > YMAX

```

```
    {
        ymax = y; xm = x;
    }
}
// Вывод значений YMAX, XM
printf(" YMAX = %.31f  XM = %.21f ", ymax, xm);
getch();
return;
}
```

**Примечание.** При выполнении оператора **for(x = xmin;x<=xmax;x+=dx)** в результате накопления ошибки может произойти «потеря» значения функции при **X = X<sub>max</sub>**. Поэтому значение **xmax** следует увеличить:  
**for(x = xmin;x<=xmax + dx/3;x+=dx).**

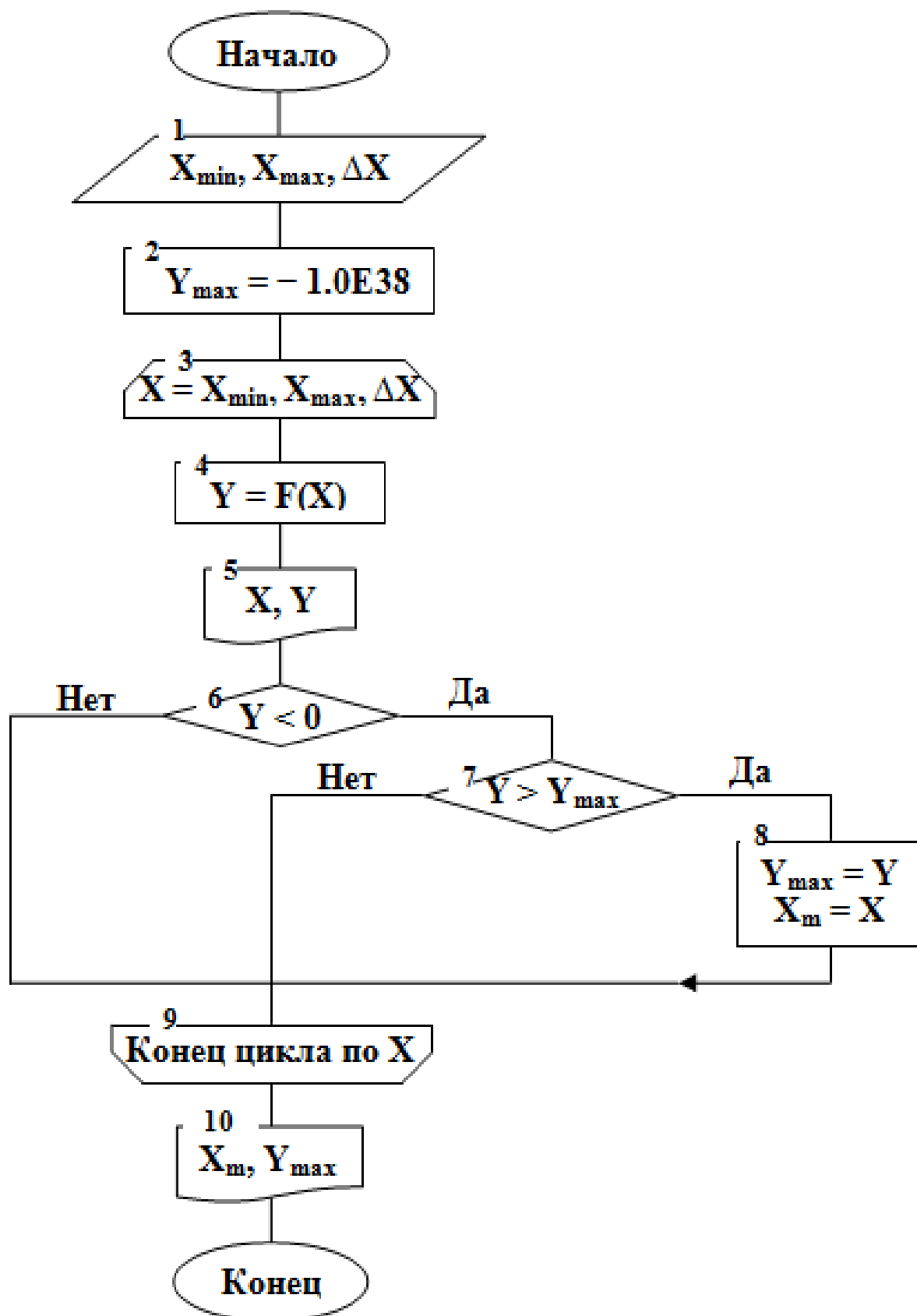


Рис. 2.7 Схема алгоритма к лабораторной работе 4 «Исследование функции одной переменной»

## 7. Листинг Basic-программы «Табулирование функции»

Условия  $Y < 0$  и  $Y > Y_{MAX}$  объединены логической связкой **AND**.

```
CLS
READ XMIN, XMAX, DX ' Ввод исходных данных
DATA 0.6,0.8,0.05
' Вывод исходных данных
PRINT TAB(5); " ГРАНИЦЫ ИНТЕРВАЛА "
PRINT TAB(6); USING " #.# <= X <= #.# "; XMIN; XMAX
PRINT TAB(8); USING " ШАГ DX = #.## "; DX
PRINT TAB(10); " ТАБЛИЦА " ' Вывод заголовка
YMAX = -1E+38
FOR X = XMIN TO XMAX + DX / 3 STEP DX ' Цикл по X
  Y = (2 * X) ^ (1 / 3) - EXP(-X) * TAN(3 * X) ^ 2
  PRINT USING " X = #.## Y = ##.###^^^^ "; X; Y
  IF Y < 0 AND Y > YMAX THEN ' Если Y < 0 и Y > YMAX
    YMAX = Y: XM = X
  END IF
NEXT X ' Конец цикла
' Вывод результатов - значений YMAX, XM
PRINT USING " YMAX = ##.###^^^^ "; YMAX;
PRINT USING " при X = #.## "; XM
END
```

Результаты работы программы представлены в п.5 (стр.19).

### Лабораторная работа 5 Исследование функции одной переменной, зависящей от параметра Тема. Табулирование функции одной переменной

Индивидуальные задания – табл. 3.5 (стр.102)

Математические функции – табл. 5.1 (стр.136)

#### 1. Задание

Составить программу решения следующей задачи.

Вычислить таблицу значений функции  $y = f(X, \alpha)$  и аргумента  $X$  на интервале  $[X_{\min}, X_{\max}]$  с шагом  $\Delta X$  при условии, что параметр  $\alpha$  принимает ряд фиксированных значений  $\alpha_1$  и  $\alpha_2$ . Таким образом, вычисление функции можно представить в виде:

$$y = \begin{cases} f(X, \alpha_1), & \text{если } X_{\min} \leq X \leq X_{\text{cp}} \\ f(X, \alpha_2), & \text{если } X_{\text{cp}} < X \leq X_{\max} \end{cases}$$

где  $X_{\text{cp}}$  – значение аргумента  $X$  из интервала  $X_{\min} < X_{\text{cp}} < X_{\max}$ .

Протабулировать функцию  $f(X, \alpha) = \alpha \cdot X - \cos^2 \alpha X$  и найти количество её отрицательных значений на интервале  $[X_{\min}, X_{\max}]$ .

### Исходные данные

Интервал		Шаг	«Среднее» значение аргумента $X$	Значения параметра $\alpha$	
$X_{\min}$	$X_{\max}$	$\Delta X$	$X_{\text{cp}}$	$\alpha_1$	$\alpha_2$
<b>0</b>	<b>1</b>	<b>0,2</b>	<b>0,4</b>	<b>0,5</b>	<b>1</b>

## 2. Решение

Обозначим:  $K$  – количество отрицательных значений функции.

Обозначения переменных					
Имя переменной		Тип переменной		Пояснения	
в задаче	в программе	Fortran	C		
$X_{\min}, X_{\max}$	XMIN, XMAX	REAL	DOUBLE	Начальное и конечное значения аргумента	
$X_{\text{cp}}$	XCP	REAL	DOUBLE	«Среднее» значение аргумента	
$\Delta X$	DX	REAL	DOUBLE	Шаг изменения аргумента	
$\alpha_1, \alpha_2$	A1,A2	REAL	DOUBLE	Параметры	
$\alpha$	A	REAL	DOUBLE	Текущее значение параметра	
$X, Y$	X, Y	REAL	DOUBLE	Текущие значения аргумента и функции	
$K$	K	INTEGER	INT	Количество отрицательных значений функции	

Количество вычисляемых значений  $(X, Y)$  [строк в таблице] определяется как:

$$\frac{X_{\max} - X_{\min}}{\Delta X} + 1 = 6$$



### 3. Ручной расчёт результатов

Вычислим значение  $y = f(X, \alpha)$  в начальной и конечной точках.

Контрольные точки			
Аргумент X		Функция $y = f(X, \alpha)$	
$X_{\min}$	0	$f(X_{\min}, \alpha_1)$	-1
$X_{\max}$	1	$f(X_{\max}, \alpha_2)$	0,708

#### Таблица значений аргумента X и функции y на интервале

[0;0,4]		[0,6;1]	
X	$f(X, \alpha_1)$	X	$f(X, \alpha_2)$
0	-1	0,6	-0,0812
0,2	-0,89	0,8	0,3146
0,4	-0,7605	1	0,7081

Очевидно, что количество отрицательных значений функции  $K = 4$ .

### 4. Алгоритм

При решении используется алгоритм вычисления количества. В п.4 – п.6 осуществляется выбор параметра  $\alpha$  в зависимости от текущего значения аргумента X (проверка условия  $X \leq X_{cp}$ ).

#### Алгоритм состоит из следующих действий:

1. Ввод данных – значений  $X_{\min}$ ,  $X_{\max}$ ,  $X_{cp}$ ,  $\Delta X$ ,  $\alpha_1$ ,  $\alpha_2$ .
2. Задание начального значения переменным  $X = X_{\min}$ ,  $K = 0$ .
3. Начало цикла по X.
4. Проверка условия  $X \leq X_{cp}$ . Если условие выполнено, перейти к п.5, иначе – к п.6.
5. Параметру  $\alpha$  присваивается значение  $\alpha_1$ :  $\alpha = \alpha_1$ . Перейти к п.7.
6. Параметру  $\alpha$  присваивается значение  $\alpha_2$ :  $\alpha = \alpha_2$ .
7. Вычисление значения функции  $y = f(x, \alpha)$ .
8. Вывод текущих значений аргумента X и функции Y.
9. Проверка условия  $Y < 0$ . Если условие выполнено, перейти к п.10, иначе – к п.11.
10. Увеличение текущего значения счётчика:  $K = K + 1$ .
11. Увеличение текущего значения аргумента:  $X = X + \Delta X$ .
12. Конец цикла по X.
13. Вывод результата – количества K.
14. Конец.

Схема алгоритма представлена на рис. 2.8 (стр.27).

## 5. Листинг С-программы

**Внимание.** При вводе программы следует иметь в виду, что С различает регистр букв, т.е., «*x*» и «*X*» – это две разные переменные.

```
// Табулирование функции, зависящей от параметра
#include<stdio.h> // Файлы заголовков
#include<conio.h>
#include<math.h>
    void main()
{
// Объявление типа и задание данных
int k = 0;
double a, a1 = 0.5, a2 = 1, xcp = 0.4;
double x, y, xmin = 0, xmax = 1, dx = 0.2;
printf("      T a b l e \n"); // Вывод заголовка
x = xmin;
while(x<=xmax + dx/3) // Цикл с предусловием
    {
// Выбор параметра α
    if(x <= xcp) // Если X ≤ Xcp
        a = a1;
    else // Иначе, если X > Xcp
        a = a2;
    y = a*x - pow(cos(a*x), 2);
// Вывод значений X и Y
    printf(" X = %.11f  Y = %7.4lf \n", x, y);
    if(y < 0)k++; // Вычисление значения K
    x += dx;
    }
printf(" Result: K = %d ", k); // Вывод значения K
getch();
return;
}
```

**Примечание.** При выполнении оператора **while** ( $x \leq x_{\max}$ ) в результате накопления ошибки может произойти «потеря» значения функции при  $X = X_{\max}$ . Поэтому значение  $x_{\max}$  следует увеличить: **while** ( $x \leq x_{\max} + dx/3$ ).

## 6. Макет печати (по С-программе)

```
Т а б л и ц а
X = 0.0 Y = -1.0000
X = 0.2 Y = -0.8900
X = 0.4 Y = -0.7605
X = 0.6 Y = -0.0812
X = 0.8 Y = 0.3146
X = 1.0 Y = 0.7081
Result: K = 4
```

Как видно, результаты ручного и машинного расчётов совпадают.

## 7. Листинг Fortran-программы

Программный код
<pre>! Табулирование функции, зависящей от параметра PROGRAM PARAMETR ! Объявление типа и задание данных INTEGER :: K = 0 REAL :: A, A1 = 0.5, A2 = 1, XCP = 0.4 REAL :: X, Y, XMIN = 0, XMAX = 1, DX = 0.2 X = XMIN WRITE(*,*) ' ТАБЛИЦА' ! Вывод заголовка DO WHILE (X &lt;= XMAX + DX/3) ! Начало цикла ! Выбор параметра A IF (X &lt;= XCP) THEN ! Если X ≤ XCP A = A1 ELSE ! Иначе, если X &gt; XCP A = A2 END IF Y = A*X - COS(A*X)**2 ! Вычисление значения Y ! Вывод значений X, Y WRITE(*,2) X, Y 2 FORMAT (1X, ' X = ', F3.1, 2X, ' Y = ', F7.4) IF (Y &lt; 0) K = K + 1 ! Вычисление K X = X + DX END DO ! Конец цикла WRITE(*, "( A, I2) ") 'Результат K = ', K ! Вывод K END</pre>

Результаты представлены в макете печати в п.6.

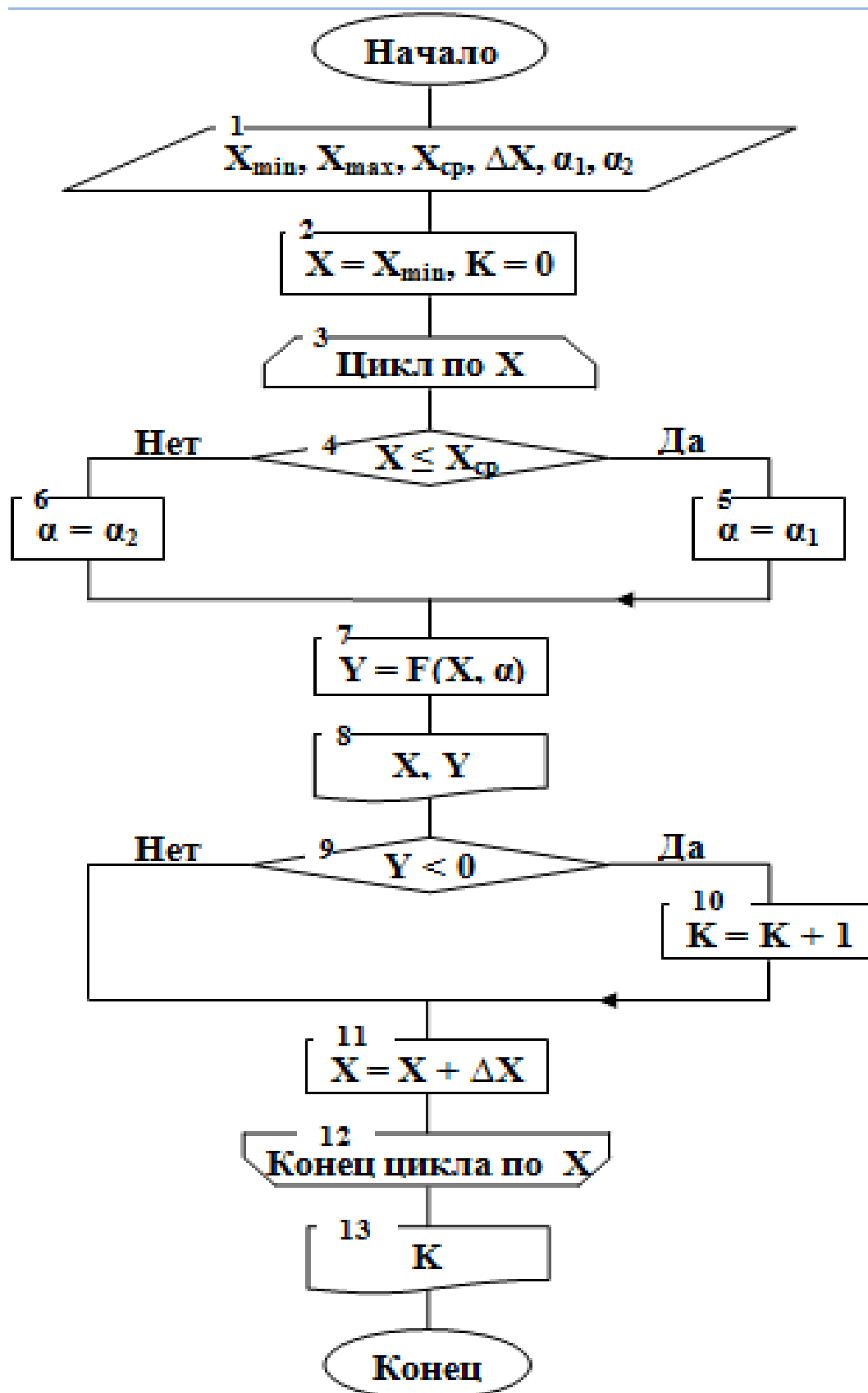


Рис. 2.8 Схема алгоритма к лабораторной работе 5  
«Исследование функции одной переменной, зависящей от параметра»

## Лабораторная работа 6

### Исследование функции одной переменной с различным шагом варьирования аргумента

#### Тема. Табулирование функции одной переменной

Индивидуальные задания – табл. 3.6 (стр.105)

Математические функции – табл. 5.1 (стр.136)

#### 1. Задание

Составить программу решения следующей задачи.

Вычислить таблицу значений функции  $Y = F(X)$  и аргумента  $X$  на интервале  $[X_{\min}, X_{\max}]$  с шагом  $\Delta X$ , причём шаг  $\Delta X$  принимает ряд фиксированных значений  $\Delta X_1$  и  $\Delta X_2$  в зависимости от текущего значения аргумента  $X$ , а именно:

$$\Delta X = \begin{cases} \Delta X_1, & \text{если } X_{\min} \leq X \leq X_{\text{ср}} \\ \Delta X_2, & \text{если } X_{\text{ср}} < X \leq X_{\max} \end{cases}$$

где  $X_{\text{ср}}$  – одно из текущих значений аргумента  $X$  на интервале  $X_{\min} < X_{\text{ср}} < X_{\max}$ .

**Протабулировать функцию  $F(x) = \operatorname{sh}x - \sqrt{x}$  и найти сумму её значений на интервале  $[X_{\min}; X_{\max}]$ . Определить знак суммы и вывести текстовое сообщение.**

#### Исходные данные

Интервал		«Среднее» значение аргумента	Шаг изменения аргумента	
$X_{\min}$	$X_{\max}$	$X_{\text{ср}}$	$\Delta X_1$	$\Delta X_2$
<b>0,5</b>	<b>1,5</b>	<b>1,1</b>	<b>0,2</b>	<b>0,1</b>

## 2. Решение

Обозначения переменных				
Имя переменной		Тип переменной		Пояснение
в задаче	в программе	Fortran	C	
$X_{\min}$ , $X_{\max}$	XMIN, XMAX	REAL	DOUBLE	Начальное и конечное значение X
$X_{\text{cp}}$	XCR	REAL	DOUBLE	Значение аргумента на интервале $X_{\min} < X_{\text{cp}} < X_{\max}$
$\Delta X_1$	DX1	REAL	DOUBLE	Шаг аргумента на интервале [ $X_{\min}$ ; $X_{\text{cp}}$ ]
$\Delta X_2$	DX2	REAL	DOUBLE	Шаг аргумента на интервале [ $X_{\text{cp}} + \Delta X_2$ ; $X_{\max}$ ]
$\Delta X$	DX	REAL	DOUBLE	Текущее значение шага
$X, Y$	X, Y	REAL	DOUBLE	Значения аргумента и функции
S	S	REAL	DOUBLE	Сумма значений функции

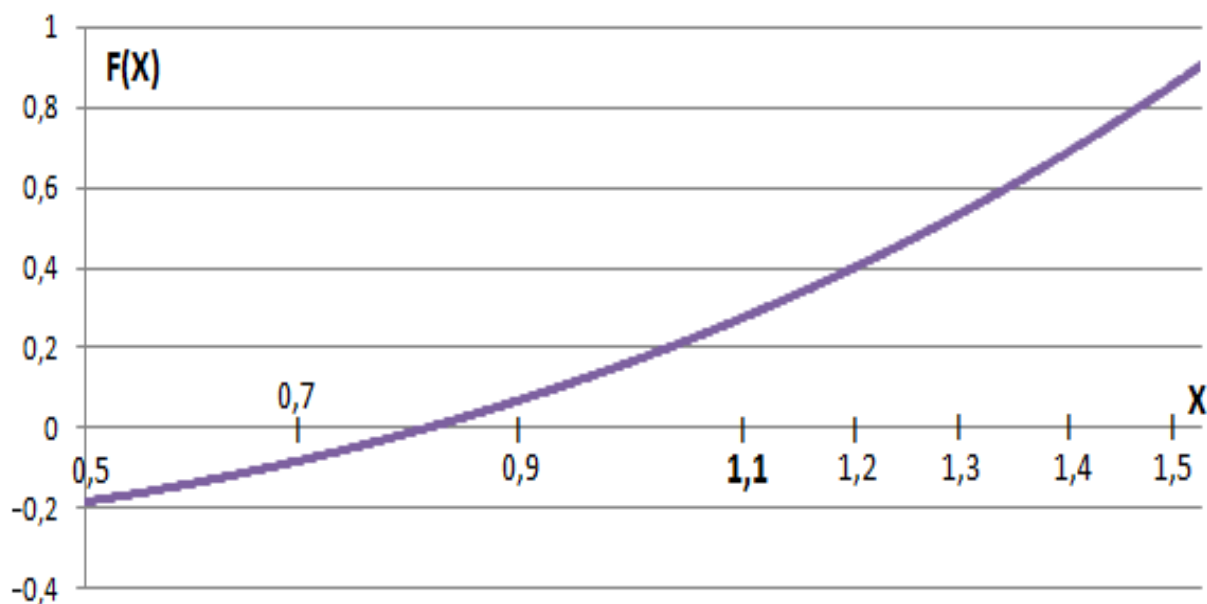
## 3. Ручной расчёт результатов

Вычислим значение  $Y = F(X)$  в начальной и конечной точках.

Контрольные точки	Аргумент	Функция
$X_{\min}$	0,5	-0,186
$X_{\max}$	1,5	0,905

Результаты можно представить в табличном и в графическом виде.

### Графический способ



**Таблица значений функции  $Y = F(X)$  на интервале**

<b>[0,5;1,1] с шагом <math>\Delta X_1 = 0,2</math></b>		<b>[1,2;1,5] с шагом <math>\Delta X_2 = 0,1</math></b>	
X	Y	X	Y
0,5	-0,186	1,2	0,414
0,7	-0,078	1,3	0,558
0,9	0,078	1,4	0,721
<b>1,1</b>	<b>0,287</b>	1,5	0,905

Вычислим сумму значений функции:  $S = 2,698$ ,  $S > 0$ . При решении задачи используется алгоритм вычисления суммы.

#### **4. Алгоритм**

В п.8 – п.9 выполняется смена шага  $\Delta X = \Delta X_2$ , если текущее значение аргумента  $X$  равно  $X_{cp}$  ( $X = X_{cp}$ ). В п.12 – п.14 определяется знак суммы  $S$  и выводится текстовое сообщение.

##### **Алгоритм состоит из следующих действий:**

1. Ввод данных – значений переменных  $X_{min}$ ,  $X_{max}$ ,  $X_{cp}$ ,  $\Delta X_1$ ,  $\Delta X_2$ .
2. Задание начального значения переменным:  $X = X_{min}$ ;  $S = 0$ .
3. Задание начального значения шага  $\Delta X = \Delta X_1$ .
4. Начало цикла по X.
5. Вычисление значения функции  $Y = F(X)$ .
6. Вычисление суммы  $S = S + Y$ .
7. Вывод текущего значения аргумента  $X$  и функции  $Y$ .

8. Проверка условия  $X = X_{\text{ср}}$ . Если условие выполняется, то перейти к п.9, иначе – к п.10.
9. Изменение текущего значения шага  $\Delta X = \Delta X_2$ .
10. Увеличение аргумента  $X$  на шаг  $X = X + \Delta X$ .
11. Конец цикла по  $X$ .
12. Проверка условия  $S > 0$ . Если условие выполняется, то перейти к п.13, иначе – к п.14.
13. Вывод сообщения  $S > 0$  (сумма положительна), перейти к п.15.
14. Вывод сообщения  $S < 0$  (сумма отрицательна).
15. Конец.

Схема алгоритма приведена на рис. 2.9.



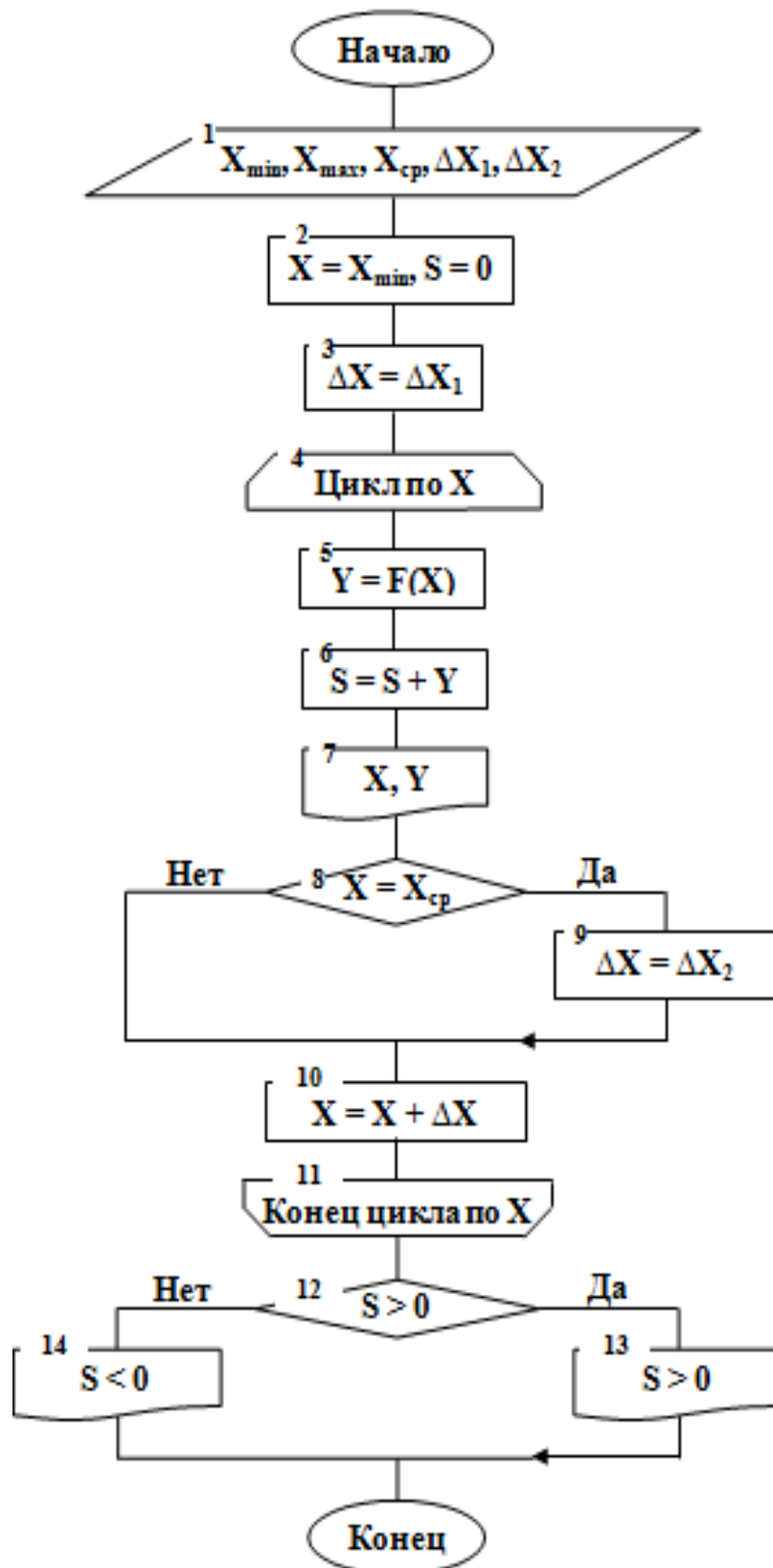


Рис. 2.9 Схема алгоритма к лабораторной работе 6  
«Исследование функции одной переменной  
с различным шагом варьирования аргумента»

## 5. Листинг С-программы

**Внимание.** При вводе программы следует иметь в виду, что С различает регистр букв, т.е., «x» и «X» – это две разные переменные.

```
// Табулирование функции с различным шагом
#include<stdio.h> // Файлы заголовков
#include<conio.h>
#include<math.h>
void main()
{
    double s = 0, xmin, xcr, xmax; // Объявление данных
    double dx, dx1, dx2, x, y;
    // Ввод данных с клавиатуры по запросу с экрана,
    // числа при вводе разделяются запятой
    printf(" Input XMIN, XCR, XMAX, DX1, DX2 ?\n ");
    scanf("%f, %f, %f, %f, %f", &xmin, &xcr, &xmax, &dx1, &dx2);
    x = xmin;
    dx = dx1;
    do // Цикл с постусловием
    {
        y = sinh(x) - sqrt(x);
        s += y; // Вычисление суммы S
        printf(" X = %.11f Y = %6.31f \n", x, y);
        if(abs(x - xcr) <= 0.001) // Если X = Xcr
        {
            printf("----- \n");
            dx = dx2;
        }
        x = x + dx; // Увеличение аргумента на шаг
    }
    while(x <= xmax + dx/3); // Конец цикла
    printf("      R e s u l t \n ");
    // Определение знака суммы S
    if(s > 0) // Если S > 0
        printf(" Sum: S = %.31f S > 0 ", s);
    else // Иначе, если S < 0
        printf(" Sum: S = %.31f S < 0 ", s);
    getch();
    return;
}
```

**Примечание.** При выполнении оператора **while** ( $x \leq x_{\max}$ ) в результате накопления ошибки может произойти «потеря» значения функции при  $X = X_{\max}$ . Поэтому значение  $x_{\max}$  следует увеличить, например: **while** ( $x \leq x_{\max} + dx/3$ ).

## 6. Макет печати (по С-программе)

```
Input XMIN,XCR,XMAX,DX1,DX2 ?
0.5,1.1,1.5,0.2,0.1
T a b l e
X = 0.5 Y = -0.186
X = 0.7 Y = -0.078
X = 0.9 Y = 0.078
X = 1.1 Y = 0.287
-----
X = 1.2 Y = 0.414
X = 1.3 Y = 0.558
X = 1.4 Y = 0.721
X = 1.5 Y = 0.905
R e s u l t
Sum: S = 2.698 S > 0
```

Как видно, результаты ручного и машинного расчётов совпадают.

### Контрольное задание для проверки знаний Расчёт ординат эпюры давления грунта на подпорные стенки

#### Тема. Табулирование функции одной переменной

В инженерной практике процесс табулирования функции, т.е. вычисление значения функции на заданном интервале с заданным шагом, широко используется при создании вычислительных программ.

Составить схему алгоритма и программу для следующей задачи.

Для исследования подпорных стенок на устойчивость и прочность возникает необходимость определения различных нагрузок, которые воспринимают проектируемые сооружения. Одна из основных нагрузок – давление грунта, расположенного за подпорной стенкой.

#### Задание

Рассчитать ординаты эпюры давления грунта на подпорные стенки.

#### Исходные данные

- $\gamma = 19$  – удельный вес грунта для песка,
- $\phi = 10^\circ$  – угол внутреннего трения грунта для песка,
- $h$  – толщина слоя грунта.

Ординаты эпюры определяются по формуле:

$$\sigma(h) = \gamma \cdot h \cdot \operatorname{tg}\left(45^\circ - \frac{\phi}{2}\right)^2$$

### Математическая формулировка задачи:

Вычислить значения аргумента  $h$  и функции  $\sigma(h)$  при условии, что аргумент  $h$  изменяется в интервале  $0 \leq h \leq 4,5$  с шагом  $\Delta h = 0,5$  для функции вида:

$$\sigma(h) = \gamma \cdot h \cdot \operatorname{tg}\left(45^\circ - \frac{h}{2}\right)^2$$

*Таблица значений аргумента  $h$  и функции  $\sigma(h)$*

<b>h</b>	0	0,5	1	1,5	2	2,5	3	3,5	4	4,5
<b><math>\sigma(h)</math></b>	0	6,7	13,4	20,1	26,8	33,4	40,1	46,8	53,5	60,2

Алгоритм циклической структуры, параметр цикла – аргумент  $h$ .

### Лабораторная работа 7

## Исследование функции двух переменных

### Тема. Табулирование функции двух переменных

Индивидуальные задания – табл. 3.7 (стр.107)

Математические функции – табл. 5.1 (стр.136)

#### 1. Задание

Составить программу решения следующей задачи:

Вычислить и напечатать таблицу значений функции  $Z = f(X, Y)$  и аргументов  $X, Y$  в заданной области при условии, что:

- аргумент  $X$  изменяется в интервале  $X_0 \leq X \leq X_k$  с шагом  $\Delta X$ ;
- аргумент  $Y$  изменяется в интервале  $Y_0 \leq Y \leq Y_k$  с шагом  $\Delta Y$ ;

**Протабулировать функцию вида  $Z = X^2 + Y^2$  и найти её среднее арифметическое в заданной области.**

**Для выполнения лабораторной работы необходимо:**

1. Представить заданную область на плоскости  $(X, Y)$ .
2. Пронумеровать точки в области в соответствии с алгоритмом (рис. 2.11).
3. Представить макет печати таблицы значений аргументов.

Порядок расположения координат точек  $(X, Y)$  в таблице должен соответствовать их нумерации в заданной области.

#### Исходные данные

Интервал		Шаг	Интервал		Шаг
$X_0$	$X_k$	$\Delta X$	$Y_0$	$Y_k$	$\Delta Y$
1	2	0,5	1	3	1

## 2. Решение

Обозначим: **S**, **K**, **SR** – сумма, количество значений функции **Z** в области и её среднее арифметическое; **SR = S/K**.

Обозначения переменных				
Имя переменной		Тип переменной		Пояснения
в схеме	в программе	Fortran	C	
<b>X<sub>0</sub>, X<sub>k</sub></b>	X0, XK	<b>REAL</b>	<b>FLOAT</b>	Начальное и конечное значение X
<b>Y<sub>0</sub>, Y<sub>k</sub></b>	Y0, YK	<b>REAL</b>	<b>FLOAT</b>	Начальное и конечное значение Y
<b>ΔX, ΔY</b>	DX, DY	<b>REAL</b>	<b>FLOAT</b>	Шаг изменения аргументов X и Y
<b>X, Y, Z</b>	X, Y, Z	<b>REAL</b>	<b>FLOAT</b>	Значения аргументов и функции
<b>K</b>	K	<b>INTEGER</b>	<b>INT</b>	Количество значений функции
<b>S, SR</b>	S, SR	<b>REAL</b>	<b>FLOAT</b>	Сумма значений функции и её среднее арифметическое в области

График функции  $Z = X^2 + Y^2$  – поверхность (рис. 2.10).

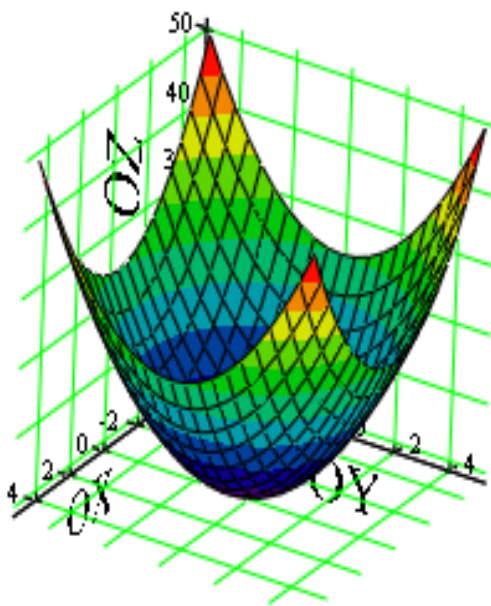


Рис. 2.10 График функции

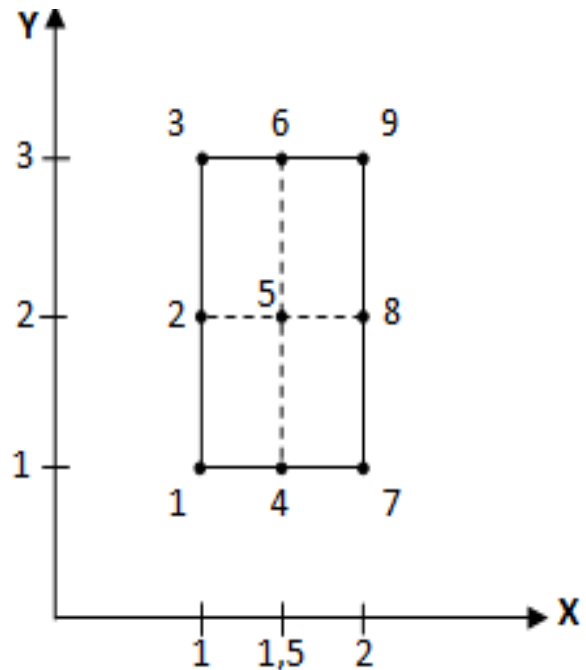


Рис. 2.11 Нумерация точек (значение X фиксировано)

Количество точек  $(X, Y)$ , для которых нужно вычислить значение функции  $Z$ , определяется как:

$$\left(\frac{X_k - X_0}{\Delta X} + 1\right) \cdot \left(\frac{Y_k - Y_0}{\Delta Y} + 1\right) = 9$$

Координаты точек в области приведены в следующей таблице:

Для фиксированного значения аргумента X									
Координаты точек									
Номер точки	1	2	3	4	5	6	7	8	9
Значение X	X = 1			X = 1,5			X = 2		
Координаты	(1;1)	(1;2)	(1;3)	(1,5;1)	(1,5;2)	(1,5;3)	(2;1)	(2;2)	(2;3)

### 3. Алгоритм

Для решения задачи используется алгоритм вычисления суммы и количества.

**Примечание.** Для вычисления среднего арифметического функции во всей заданной области направление «обхода» точек на плоскости значения не имеет.

В алгоритме выбран вариант, когда фиксировано значение аргумента X (рис. 2.11). Приведём словесное описание алгоритма. Алгоритм содержит двойной цикл вложенной структуры. Параметр внешнего цикла – аргумент X, а внутреннего – аргумент Y.

**Алгоритм состоит из следующих действий:**

1. Ввод данных – значений переменных  $X_0, X_k, \Delta X, Y_0, Y_k, \Delta Y$ ;
2. Задание начального значения переменным  $S = 0, K = 0$ .
3. *Начало внешнего цикла по X.*
4. *Начало внутреннего цикла по Y.*
5. Вычисление значения функции  $Z = f(X, Y)$ .
6. Вывод текущих значений аргументов X, Y и функции Z.
7. Вычисление суммы и количества:  $S = S + Z, K = K + 1$ .
8. *Конец внутреннего цикла по Y.*
9. *Конец внешнего цикла по X.*
10. Вывод результата – значения  $SR = S/K$ .
11. Конец.

Схема алгоритма представлена на рис. 2.12 (стр.38).

## 4. Листинг С-программы

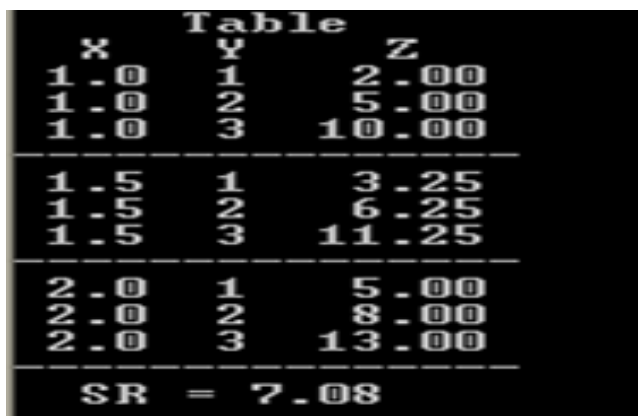
**Внимание.** При вводе программы следует иметь в виду, что С различает регистр букв, т.е., «x» и «X» – это две разные переменные.

```


Программный код


# include<stdio.h> // Файлы заголовков
# include<conio.h>
# include<math.h>
void main()
{
int k = 0; // Объявление типа и задание данных
float x0 = 1, xk = 2, dx = 0.5;
float y0 = 1, yk = 3, dy = 1;
float x, y, z, s = 0;
printf("      Table \n"); // Вывод заголовка
printf("  X    Y    Z \n");
for(x = x0; x<=xk; x += dx) // Внешний цикл по X
  {
  for(y = y0; y<=yk; y += dy) // Внутренний цикл по Y
    {
    z = x*x + pow(y,2); // Вычисление Z, S, K
// Вывод значений X, Y, Z
    printf(" %.1f  %.0f  %5.2f \n", x, y, z);
    s += z; k++;
    }
printf("-----\n"); // Вывод строки "--"
  }
printf("  SR = %.2f ", s/k); // Вывод SR
getch();
return;
}
```

## 5. Макет печати результатов



X	Y	Z
1.0	1	2.00
1.0	2	5.00
1.0	3	10.00
1.5	1	3.25
1.5	2	6.25
1.5	3	11.25
2.0	1	5.00
2.0	2	8.00
2.0	3	13.00
SR =		7.08

Как видно, расположение точек в макете соответствует их нумерации в заданной области (рис. 2.11).

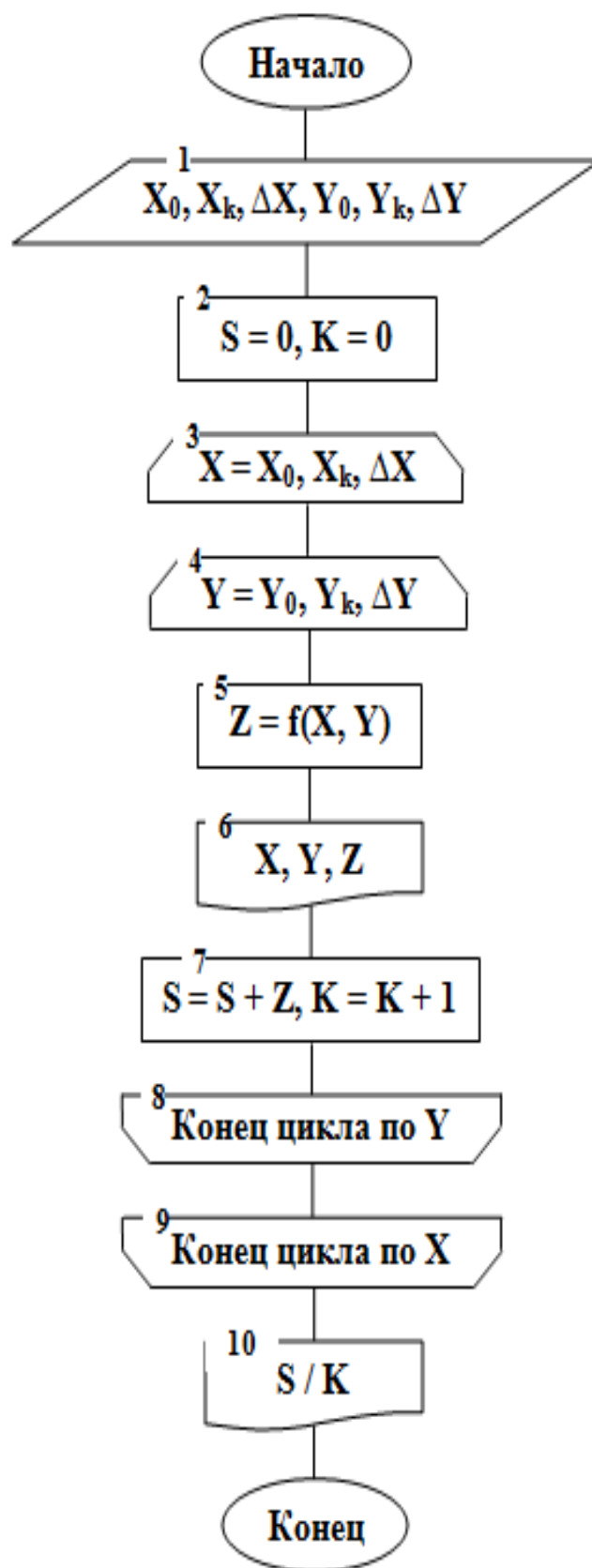


Рис. 2.12 Схема алгоритма к лабораторной работе 7  
«Исследование функции двух переменных»



## Лабораторная работа 8

### Исследование функции двух переменных при фиксированном значении аргумента

#### Тема. Табулирование функции двух переменных

Индивидуальные задания – табл. 3.8 (стр.109)

Математические функции – табл. 5.1 (стр.136)

#### 1. Задание

Составить программу решения следующей задачи:

Вычислить и напечатать таблицу значений функции  $Z = f(X, Y)$

и аргументов  $X, Y$  в заданной области при условии, что:

- аргумент  $X$  изменяется в интервале  $X_0 \leq X \leq X_k$  с шагом  $\Delta X$ ;
- аргумент  $Y$  изменяется в интервале  $Y_0 \leq Y \leq Y_k$  с шагом  $\Delta Y$ ;

Рассмотрим два варианта задания.

**Протабулировать функцию  $Z = X^2 + Y^2$  и найти её среднее арифметическое:**

- для каждого фиксированного значения аргумента  $X$ ;
- для каждого фиксированного значения аргумента  $Y$ .

**Для выполнения лабораторной работы необходимо:**

1. Представить графически заданную область на плоскости.
2. Пронумеровать точки в области в соответствии с алгоритмом (рис. 2.13-2.14).
3. Представить макет печати таблицы значений аргументов. Расположение точек  $(X, Y)$  в таблице должно соответствовать их нумерации в заданной области.

#### Исходные данные

Интервал		Шаг	Интервал		Шаг
$X_0$	$X_k$	$\Delta X$	$Y_0$	$Y_k$	$\Delta Y$
1	2	0,5	1	3	1

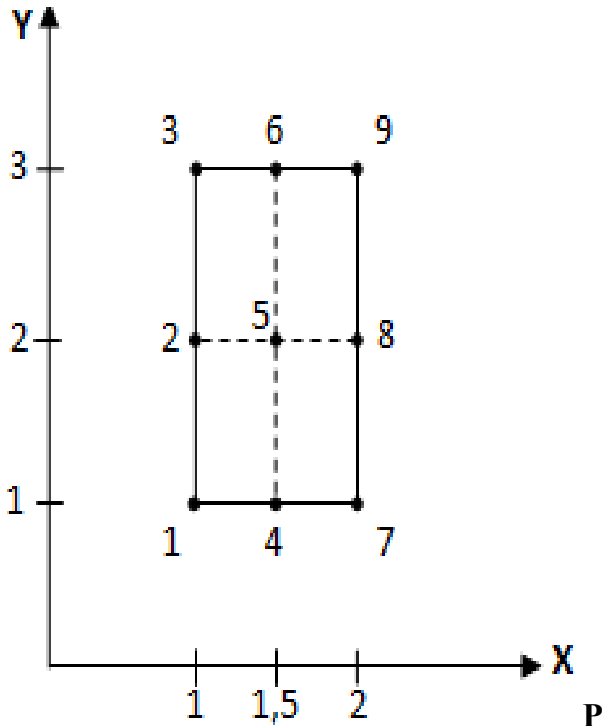
## 2. Решение

Обозначим:  $S$ ,  $K$ ,  $SR$  – сумма, количество значений функции  $Z$  и её среднее арифметическое для фиксированного значения аргумента ( $X$  или  $Y$ );  $SR = S/K$ .

Обозначения переменных				
Имя переменной		Тип переменной		Пояснения
в схеме	в программе	Fortran	C	
$X_0, X_k$	$X_0, X_k$	REAL	FLOAT	Начальное и конечное значение $X$
$Y_0, Y_k$	$Y_0, Y_k$	REAL	FLOAT	Начальное и конечное значение $Y$
$\Delta X, \Delta Y$	$DX, DY$	REAL	FLOAT	Шаг изменения $X$ и $Y$
$X, Y, Z$	$X, Y, Z$	REAL	FLOAT	Значения аргументов и функции
$K$	$K$	INTEGER	INT	Количество значений функции для фиксированного аргумента $X$ или $Y$
$S, SR$	$S, SR$	REAL	FLOAT	Сумма значений функции и её среднее арифметическое для фиксированного аргумента ( $X$ или $Y$ )

График функции  $Z = X^2 + Y^2$  – поверхность (рис. 2.10).

### Графическое представление заданной области



ис. 2.13 Схема нумерации точек для фиксированного значения  $X$

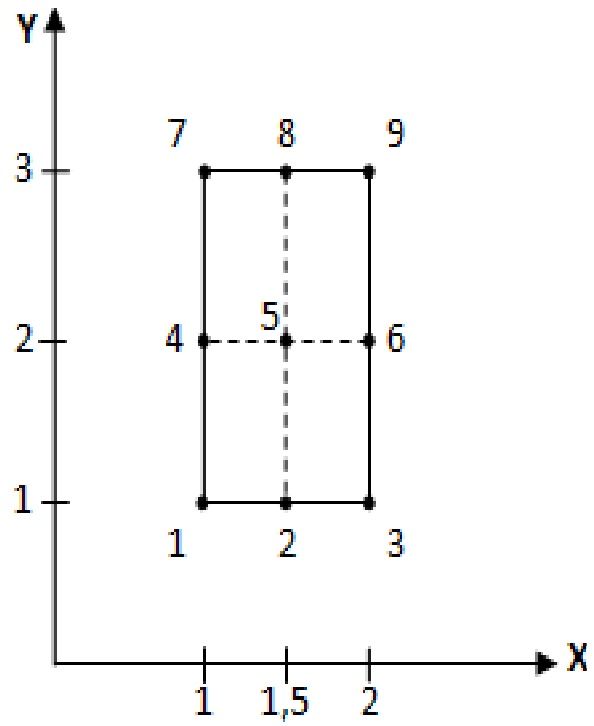


Рис. 2.14 Схема нумерации точек для фиксированного значения  $Y$

Количество точек  $(X, Y)$  в заданной области определяется как:

$$\left(\frac{X_k - X_0}{\Delta X} + 1\right) \cdot \left(\frac{Y_k - Y_0}{\Delta Y} + 1\right) = 9$$

### 3. Алгоритм

Для фиксированного значения аргумента  $X$  параметр внешнего цикла –  $X$ , а внутреннего –  $Y$ . Для фиксированного значения аргумента  $Y$  внешний цикл организуется по параметру  $Y$ , а внутренний – по  $X$ .

Схемы алгоритма представлены на рис. 2.15 - 2.16:

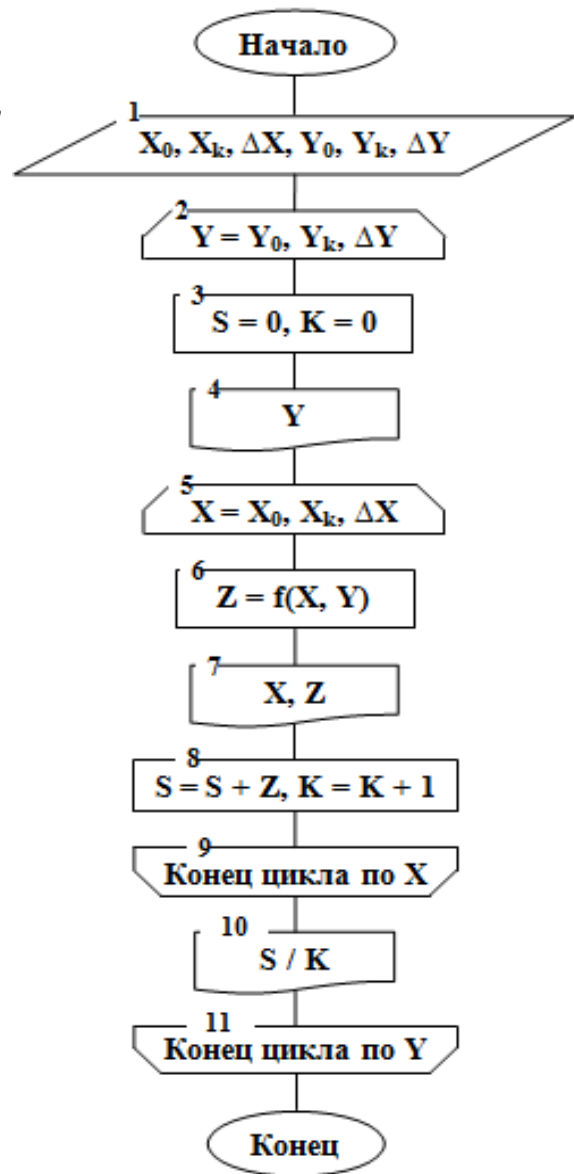
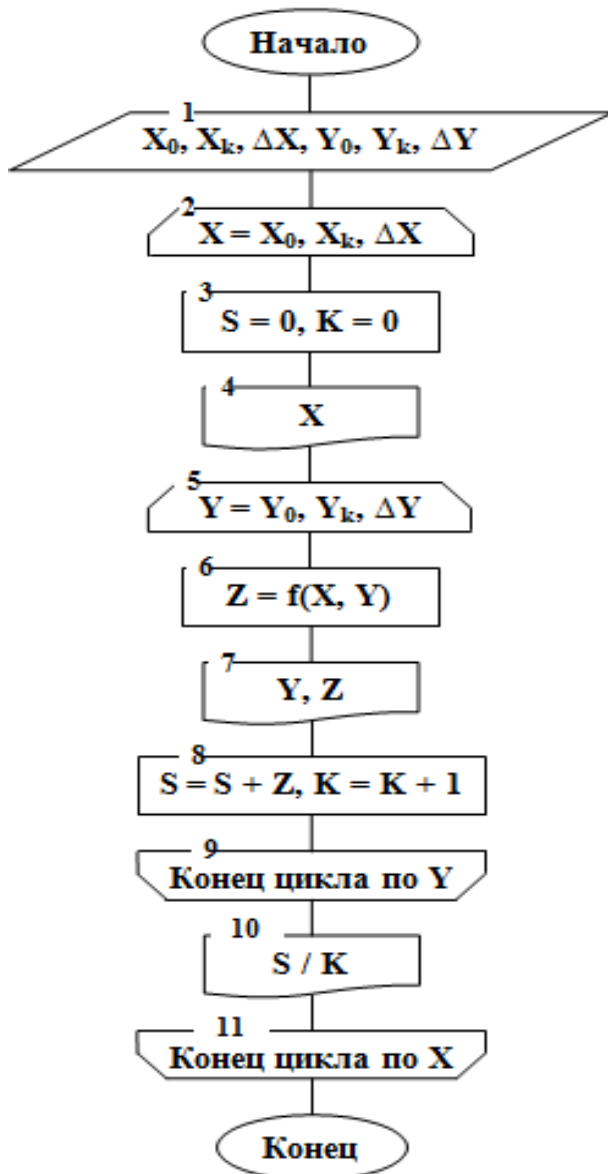


Рис. 2.15 Схема алгоритма к лабораторной работе 8 «Исследование функции двух переменных» при фиксированном  $X$

Рис. 2.16 Схема алгоритма к лабораторной работе 8 «Исследование функции двух переменных» при фиксированном  $Y$

#### 4. Листинг Basic-программы

Для фиксированного значения X	Для фиксированного значения Y
<b>Программный код</b>	
<pre>' Ввод данных <b>READ</b> X0, XK, DX, Y0, YK, DY <b>DATA</b> 1, 2, 0.5, 1, 3, 1 ' Внешний цикл по X <b>FOR</b> X = X0 <b>TO</b> XK <b>STEP</b> DX S = 0: K = 0 <b>PRINT</b> TAB(6); " X = "; X ' Внутренний цикл по Y <b>FOR</b> Y = Y0 <b>TO</b> YK <b>STEP</b> DY   Z = X ^ 2 + Y ^ 2   <b>PRINT</b> "Y ="; Y, " Z ="; Z   S = S + Z: K = K + 1 <b>NEXT</b> Y 'Конец цикла по Y <b>PRINT</b> TAB(6); "SR = "; <b>PRINT</b> USING "##.##"; S/K <b>NEXT</b> X ' Конец цикла по X <b>END</b></pre>	<pre>' Ввод данных <b>READ</b> X0, XK, DX, Y0, YK, DY <b>DATA</b> 1, 2, 0.5, 1, 3, 1 ' Внешний цикл по Y <b>FOR</b> Y = Y0 <b>TO</b> YK <b>STEP</b> DY S = 0: K = 0 <b>PRINT</b> TAB(6); " Y = "; Y ' Внутренний цикл по X <b>FOR</b> X = X0 <b>TO</b> XK <b>STEP</b> DX   Z = X ^ 2 + Y ^ 2   <b>PRINT</b> "X ="; X, " Z ="; Z   S = S + Z: K = K + 1 <b>NEXT</b> X 'Конец цикла по X <b>PRINT</b> TAB(6); "SR = "; <b>PRINT</b> USING "##.##"; S/K <b>NEXT</b> Y ' Конец цикла по Y <b>END</b></pre>

#### 5. Макеты печати (по Basic-программе)

Для фиксированного X	Для фиксированного Y
<b>X = 1</b>	<b>Y = 1</b>
Y = 1    Z = 2	X = 1        Z = 2
Y = 2    Z = 5	X = 1.5      Z = 3.25
Y = 3    Z = 10	X = 2         Z = 5
<b>SR = 5.67</b>	<b>SR = 3.42</b>
<b>X = 1.5</b>	<b>Y = 2</b>
Y = 1    Z = 3.25	X = 1         Z = 5
Y = 2    Z = 6.25	X = 1.5       Z = 6.25
Y = 3    Z = 11.25	X = 2         Z = 8
<b>SR = 6.92</b>	<b>SR = 6.42</b>
<b>X = 2</b>	<b>Y = 3</b>
Y = 1    Z = 5	X = 1         Z = 10
Y = 2    Z = 8	X = 1.5       Z = 11.25
Y = 3    Z = 13	X = 2         Z = 13
<b>SR = 8.67</b>	<b>SR = 11.42</b>

Расположение точек в макете соответствует их нумерации в области (рис. 2.13-2.14, стр.40).

**Лабораторная работа 9**  
**Вычисление координат искомой точки,**  
**наиболее удалённой от начала координат**  
**Тема. Одномерный массив: min, max**

Индивидуальные задания – табл. 3.9 (стр.111)

Математические функции – табл. 5.1 (стр.136)

**1. Задание**

Составить программу решения следующей задачи.

На плоскости  $(x, y)$  заданы точки  $\{M_1(X_1, Y_1), \dots, M_n(X_n, Y_n)\}$  и круг радиуса  $R$  с центром в начале координат,  $n$  – количество точек. Координаты точек задаются в виде массивов абсцисс  $(X_1, \dots, X_n)$  и ординат  $(Y_1, \dots, Y_n)$ , в соответствии с условием задачи и так, чтобы искомая точка была одна.

**Среди точек правой полуплоскости, лежащих внутри круга, определить точку, наиболее удалённую от начала координат.**

**Для выполнения лабораторной работы необходимо:**

- задать точки  $\{M_1(X_1, Y_1), \dots, M_n(X_n, Y_n)\}$  и радиус круга  $R$ .
- представить заданные точки и круг на плоскости  $(X, Y)$ .
- найти и обозначить на графике искомую точку.

**Исходные данные**

Количество точек	Координаты точек $M_i(X_i, Y_i)$				
	Абсциссы				
<b>N = 5</b>	<b>X<sub>1</sub></b>	<b>X<sub>2</sub></b>	<b>X<sub>3</sub></b>	<b>X<sub>4</sub></b>	<b>X<sub>5</sub></b>
	<b>4</b>	<b>1</b>	<b>-2</b>	<b>1</b>	<b>-2</b>
Радиус круга	Ординаты				
<b>R = 3</b>	<b>Y<sub>1</sub></b>	<b>Y<sub>2</sub></b>	<b>Y<sub>3</sub></b>	<b>Y<sub>4</sub></b>	<b>Y<sub>5</sub></b>
	<b>5</b>	<b>2</b>	<b>1</b>	<b>-1</b>	<b>-3</b>

**2. Решение**

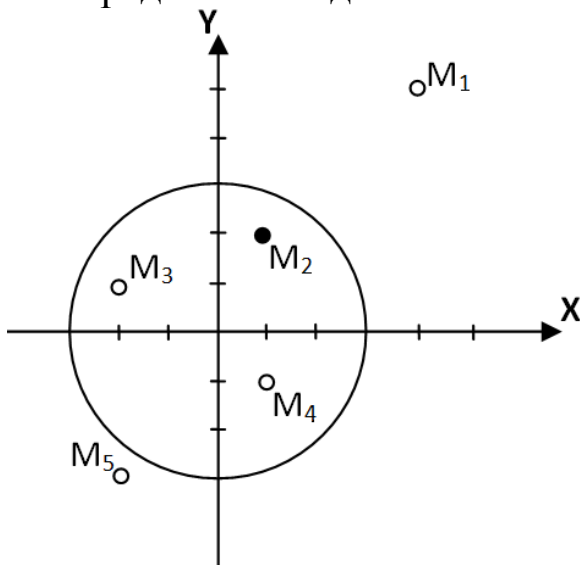
Обозначим:  $RT$  – расстояние от  $M_i(X_i, Y_i)$  до начала координат.

$K, RMAX$  – номер искомой точки, удалённой от начала координат на расстояние  $RMAX$ .

Обозначения переменных			
Имя переменной	Тип переменной		Пояснение
	Fortran	C	
<b>N, i</b>	<b>INTEGER</b>	<b>INT</b>	Количество точек и порядковый номер точки
<b>R</b>	<b>REAL</b>	<b>FLOAT</b>	Радиус круга
<b>X<sub>i</sub>, Y<sub>i</sub></b>	<b>REAL</b>	<b>FLOAT</b>	Координаты точки с порядковым номером i
<b>RT</b>	<b>REAL</b>	<b>FLOAT</b>	Расстояние от M <sub>i</sub> (X <sub>i</sub> , Y <sub>i</sub> ) до начала координат
<b>RMAX</b>	<b>REAL</b>	<b>FLOAT</b>	Наибольшее расстояние
<b>K</b>	<b>INTEGER</b>	<b>INT</b>	Порядковый номер искомой точки
<b>X<sub>k</sub>, Y<sub>k</sub></b>	<b>REAL</b>	<b>FLOAT</b>	Координаты искомой точки

### 3. Ручной расчёт результатов

Представим заданные точки и круг на плоскости (X, Y).



**Координаты точек заданы в соответствии с условием задачи.** Это означает, что среди заданных две точки, M<sub>2</sub>(1; 2) и M<sub>4</sub>(1; -1), расположены в правой полуплоскости и внутри круга.

Расстояние от точки M<sub>i</sub>(X<sub>i</sub>, Y<sub>i</sub>) до начала координат вычисляется по формуле  $RT = \sqrt{X_i^2 + Y_i^2}$ .

Для точек правой полуплоскости выполняется условие  $X_i > 0$ .

Представим точки, из которых осуществляется выбор искомой:

Точки, лежащие в правой полуплоскости и внутри круга			
Номер точки i	Точка M <sub>i</sub> (X <sub>i</sub> , Y <sub>i</sub> )	Расстояние до начала координат	
2	M <sub>2</sub> (1; 2)	$\sqrt{5}$	2,24
4	M <sub>4</sub> (1; -1)	$\sqrt{2}$	1,41

Искомой является точка  $M_2(1; 2)$  с номером  $K = 2$ , удалённая от начала координат на расстояние  $RMAX = 2,24$ .

#### 4. Алгоритм

При решении задачи используется алгоритм вычисления наибольшего значения. Для контроля и наглядности результатов в п.8 выводятся точки, из которых осуществляется выбор искомой.

**Алгоритм состоит из следующих действий:**

1. Ввод координат – значений  $X_i$  и  $Y_i$  ( $i = 1, \dots, n$ ) и радиуса  $R$ .
2. Вывод координат точек  $X_i, Y_i$  ( $i = 1, \dots, n$ ).
3. Задание начального значения переменной  $RMAX = -1.0E38$  (заведомо малого числа).
4. *Начало цикла.*
5. Вычисление значения расстояния  $RT$ .
6. Проверка условия  $X_i > 0$ . Если условие выполнено, перейти к п.7, иначе – к п.11.
7. Проверка условия  $RT < R$ . Если условие выполнено, перейти к п.8, иначе – к п.11.
8. Вывод координат  $X_i, Y_i$  и расстояния  $RT$ .
9. Проверка условия  $RT > RMAX$ . Если условие выполнено, перейти к п.10, иначе – к п.11.
10. Переменным присваиваются значения  $RMAX = RT, K = i$ .
11. *Конец цикла.*
12. Вывод искомой точки  $M_k(X_k, Y_k)$  и расстояния  $RMAX$ .
13. Конец.

Схема алгоритма представлена на рис. 2.17 (стр.47).

#### 5. Макет печати результатов (по С-программе)

```

X      Y
-----
 4      5
 1      2
-2      1
 1     -1
-2     -3

      Point
X(2) = 1  Y(2) = 2  RT = 2.24
X(4) = 1  Y(4) = -1  RT = 1.41

      Result
X(2) = 1  Y(2) = 2
RMAX = 2.24

```

Результаты ручного и машинного расчётов совпадают. В этом легко убедиться: на графике искомая точка обозначена (стр.44).

## 6. Листинг С-программы «Вычисление координат точки»

**Внимание.** При вводе программы следует иметь в виду, что С различает регистр букв, т.е., «x» и «X» – это две разные переменные. Нижняя граница индекса  $i$  (порядкового номера элемента в массиве) задаётся равной нулю.

```
# include<stdio.h> // Заголовочные файлы
# include<conio.h>
# include<math.h>
void main()
{
int i,k,n = 5; // Объявление типа и задание данных
float rt,r = 3,rmax = -1.0e38;
float x[5] = {4,1,-2,1,-2};
float y[5] = {5,2,1,-1,-3};
printf(" X \t Y\n"); // Вывод заголовка
printf("-----\n");
for(i = 0;i<n;i++) // Цикл вывода координат точек
printf(" %3.0f \t %3.0f\n ",x[i],y[i]);
printf(" \n \t Point \n");
for(i = 0;i<n;i++)// Цикл по i (номеру точки)
{
rt = sqrt(pow(x[i],2) + pow(y[i],2));
if(x[i]>0 && rt<r) // Если Xi > 0 и RT < R
{
// Вывод точек правой полуплоскости внутри круга
printf(" X(%d) = %2.0f ", i+1, x[i]);
printf(" Y(%d) = %2.0f ", i+1, y[i]);
printf(" RT = %4.2f \n ", rt);
if(rt>rmax) // Если RT > RMAX
{
rmax = rt; k = i;
}
}
}
// Вывод координат искомой точки (Xk, Yk)
printf(" \t Result \n");
printf(" X(%d) = %3.0f ", k+1, x[k]);
printf(" Y(%d) = %3.0f ", k+1, y[k]);
printf(" \n RMAX = %4.2f ",rmax);
getch();
return;
}
```



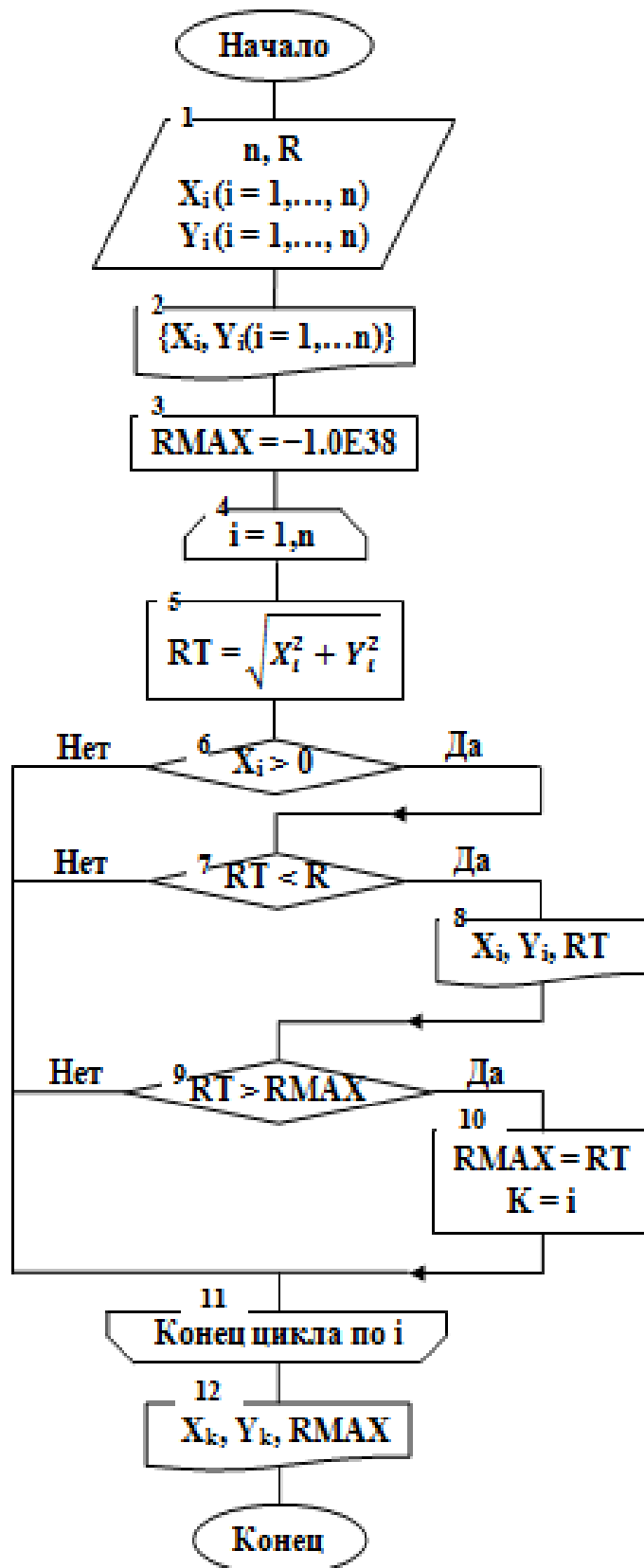


Рис. 2.17 Схема алгоритма к лабораторной работе 9  
«Вычисление координат точки, наиболее удалённой  
от начала координат»

**Лабораторная работа 10**  
**Вычисление координат искомой точки**  
**(ближайшей к заданной)**

**Тема. Одномерный массив: min, max**

Индивидуальные задания – табл. 3.10 (стр.113)

**1. Задание**

Составить программу решения следующей задачи.

На плоскости  $(x, y)$  заданы точки  $\{M_1(X_1, Y_1), \dots, M_n(X_n, Y_n)\}$  и точка  $Q(X_q, Y_q)$ . Координаты точек задаются в виде массивов абсцисс  $(X_1, \dots, X_n)$  и ординат  $(Y_1, \dots, Y_n)$ , в соответствии с условием задачи и так, чтобы искомая точка была одна.

Среди точек первого и третьего квадранта определить точку, ближайшую к точке  $Q(X_q, Y_q)$ , количество точек  $n = 5$ .

Для выполнения лабораторной работы необходимо:

- задать все точки и представить их на плоскости  $(x, y)$ .
- найти и обозначить на графике искомую точку.

**Исходные данные**

Точка Q	Координаты точек $M_i(X_i, Y_i)$				
$X_q$	$X_1$	$X_2$	$X_3$	$X_4$	$X_5$
-2	4	1	-2	5	-2
$Y_q$	$Y_1$	$Y_2$	$Y_3$	$Y_4$	$Y_5$
4	5	2	1	-2	-3

**2. Решение**

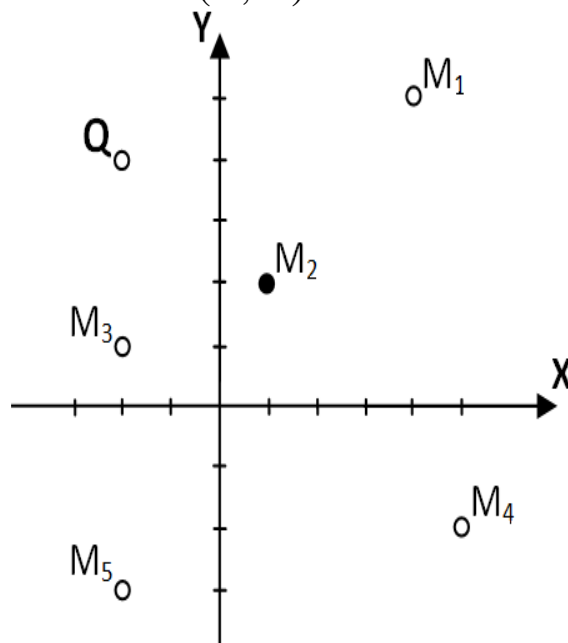
Обозначения переменных			
Имя переменной	Тип переменной		Пояснение
	Fortran	C	
$N, i$	INTEGER	INT	Количество точек и порядковый номер точки
$XQ, YQ$	REAL	FLOAT	Абсцисса и ордината точки $Q(X_q, Y_q)$
$X_i, Y_i$	REAL	FLOAT	Координаты точки с номером $i$
$RQ$	REAL	FLOAT	Расстояние между $M_i(X_i, Y_i)$ и $Q(X_q, Y_q)$
$RMIN$	REAL	FLOAT	Наименьшее расстояние
$K$	INTEGER	INT	Порядковый номер искомой точки
$X_k, Y_k$	REAL	FLOAT	Координаты искомой точки

### 3. Ручной расчёт результатов

Представим заданные точки на плоскости (X, Y).

**Исходные данные заданы в соответствии с условием задачи.**

Это означает, что среди заданных три точки  $[M_1(4; 5), M_2(1; 2)$  и  $M_5(-2; -3)]$  расположены в первом и третьем квадранте. Из них искомой (ближайшей к точке Q) является точка  $M_2(1; 2)$ . Точка  $Q(-2;4)$  выбрана во втором квадранте (может лежать в любом из них).



Расстояние между текущей точкой  $M_i(X_i, Y_i)$  и  $Q(X_q, Y_q)$  вычисляется только для точек первого и третьего квадранта при выполнении условия  $X_i \cdot Y_i > 0$  по формуле

$$RQ = \sqrt{(X_i - X_q)^2 + (Y_i - Y_q)^2}.$$

Точки первого и третьего квадранта			
Номер точки i	Абсцисса	Ордината	Расстояние до точки Q(-2; 4)
1	4	5	$\sqrt{37}$ 6,08
2	1	2	$\sqrt{13}$ 3,61
5	-2	-3	$\sqrt{49}$ 7

Искомой является точка первого квадранта  $M_2(1;2)$  с номером **K = 2**, расположенная от точки  $Q(-2;4)$  на расстоянии **RMIN = 3,61**.

### 4. Алгоритм

При решении задачи используется алгоритм вычисления наименьшего значения. Параметр цикла – номер точки **i**.

Для контроля и наглядности результатов в п.7 выводятся точки первого и третьего квадрантов. Из них осуществляется выбор искомой точки, ближайшей к  $Q(X_q, Y_q)$ .

### Алгоритм состоит из следующих действий:

1. Ввод данных – координат точек  $M_i(X_i, Y_i)$  и  $Q(X_q, Y_q)$ .
2. Вывод координат точек  $M_i(X_i, Y_i)$  и точки  $Q(X_q, Y_q)$ .
3. Задание начального значения переменной  $RMIN = 1.0E38$ .
4. Начало цикла по  $i$ .
5. Проверка условия  $X_i \cdot Y_i > 0$ . Если условие выполнено, перейти к п.6, иначе – к п.10.
6. Вычисление расстояния  $RQ$ .
7. Вывод значений переменных  $X_i, Y_i$  и  $RQ$ .
8. Проверка условия  $RQ < RMIN$ . Если условие выполнено, перейти к п.9, иначе – к п.10.
9. Переменным присваиваются значения:  $RMIN = RQ, K = i$ .
10. Конец цикла по  $i$ .
11. Вывод координат искомой точки  $M_k(X_k, Y_k)$  и  $RMIN$ .
12. Конец.

Схема алгоритма представлена на рис. 2.18 (стр. 51).

### 5. Листинг Basic-программы

Программный код
' Вычисление координат искомой точки <b>CLS</b> <b>DIM</b> X(5), Y(5) <b>DATA</b> 5, -2, 4 <b>READ</b> N, XQ, YQ <b>PRINT</b> "ТОЧКА Q; "; " XQ ="; XQ; " YQ = "; YQ <b>DATA</b> 4, 5, 1, 2, -2, 1, 5, -2, -2, -3 ' Ввод-вывод координат точек <b>PRINT TAB</b> (4); "КООРДИНАТЫ ТОЧЕК:" <b>FOR</b> I = 1 <b>TO</b> N ' Начало цикла <b>READ</b> X(I), Y(I) <b>PRINT</b> "X("; I; ")="; X(I); <b>PRINT</b> " Y("; I; ")="; Y(I) <b>NEXT</b> I ' Конец цикла RMIN = 1E+38 <b>PRINT TAB</b> (6); "ТОЧКИ 1 И 3 КВАДРАНТА" <b>FOR</b> I = 1 <b>TO</b> N ' Начало цикла <b>IF</b> X(I) * Y(I) > 0 <b>THEN</b> RQ = <b>SQR</b> ((X(I) - XQ) ^ 2 + (Y(I) - YQ) ^ 2)

```

' Вывод координат точек 1 и 3 квадранта
  PRINT "X("; I; ") ="; X(I);
  PRINT " Y("; I; ") ="; Y(I); " RQ ="; RQ
  IF RQ < RMIN THEN RMIN = RQ : K = I
END IF
NEXT I      ' Конец цикла
' Вывод координат искомой точки
PRINT TAB(14); "ИСКАМАЯ ТОЧКА:"
PRINT "X("; K; ") = ";X(K); " Y("; K; ") = ";Y(K);
PRINT USING " RMIN = ### "; RMIN
END

```

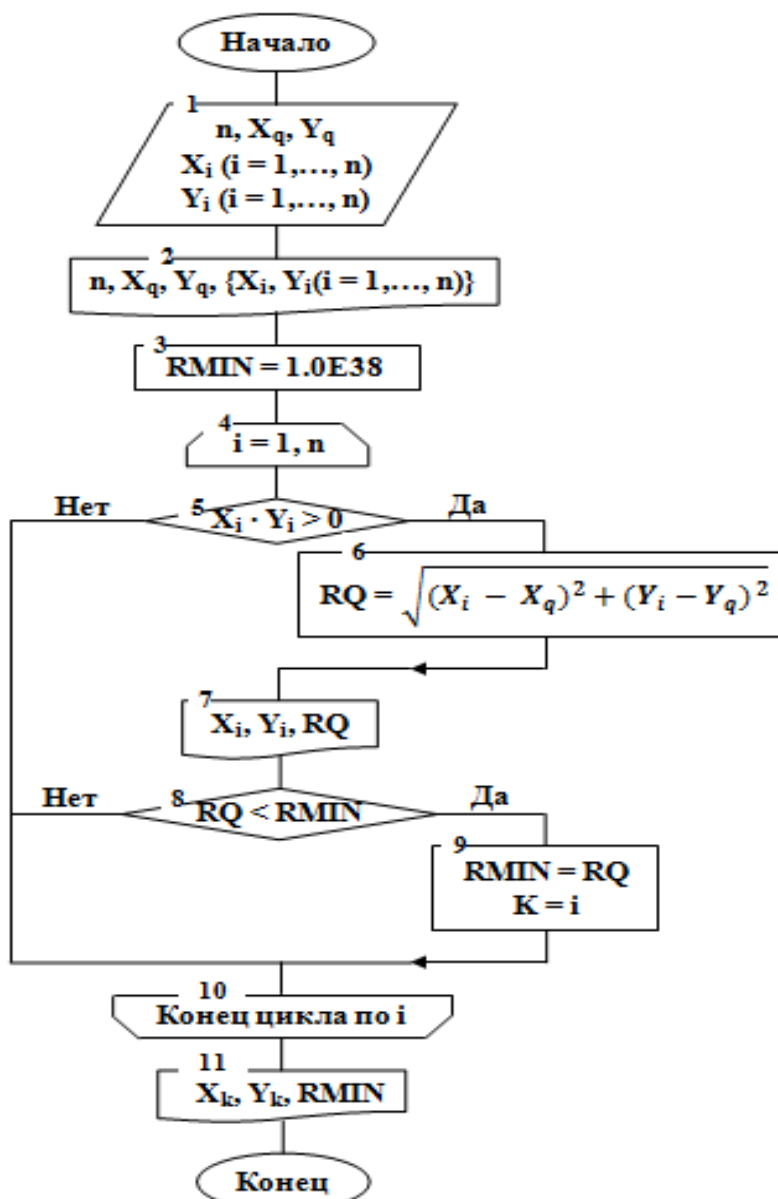


Рис. 2.18 Схема алгоритма к лабораторной работе 10 «Вычисление координат точки, ближайшей к заданной»

## 6. Макет печати результатов (по Basic-программе)

```
ТОЧКА Q;   XQ = -2   YQ = 4
           КООРДИНАТЫ ТОЧЕК:
X( 1 ) = 4           Y( 1 ) = 5
X( 2 ) = 1           Y( 2 ) = 2
X( 3 ) = -2          Y( 3 ) = 1
X( 4 ) = 5           Y( 4 ) = -2
X( 5 ) = -2          Y( 5 ) = -3
           ТОЧКИ 1 И 3 КВАДРАНТА
X( 1 ) = 4   Y( 1 ) = 5   RQ = 6.08
X( 2 ) = 1   Y( 2 ) = 2   RQ = 3.61
X( 5 ) = -2   Y( 5 ) = -3   RQ = 7
           ИСКОМАЯ ТОЧКА:
X( 2 ) = 1   Y( 2 ) = 2   RMIN = 3.61
```

Результаты ручного и машинного расчётов совпадают. В этом легко убедиться: искомая точка обозначена на графике (стр.49).

### Лабораторная работа 11 Вычисление координат искомой точки Тема. Одномерный массив: min, max

Индивидуальные варианты – табл. 3.11 (стр.114)

#### 1. Задание

Составить программу решения следующей задачи.

На плоскости  $(x, y)$  заданы точки  $\{M_1(X_1, Y_1), \dots, M_n(X_n, Y_n)\}$ ,  $n$  – количество точек. Координаты точек задаются в виде массивов абсцисс  $(X_1, \dots, X_n)$  и ординат  $(Y_1, \dots, Y_n)$ , в соответствии с условием задачи и так, чтобы искомая точка была одна.

Среди точек первого квадранта определить точку с наименьшей абсциссой,  $n = 5$ .

Для выполнения лабораторной работы необходимо:

- задать координаты точек  $\{M_1(X_1, Y_1), \dots, M_n(X_n, Y_n)\}$ .
- представить заданные точки на плоскости  $(x, y)$ .
- найти и обозначить на графике искомую точку.

## Исходные данные

Абсциссы	$X_1$	$X_2$	$X_3$	$X_4$	$X_5$
	4	1	-2	5	-2
Ординаты	$Y_1$	$Y_2$	$Y_3$	$Y_4$	$Y_5$
	5	2	1	-2	-3

## 2. Решение

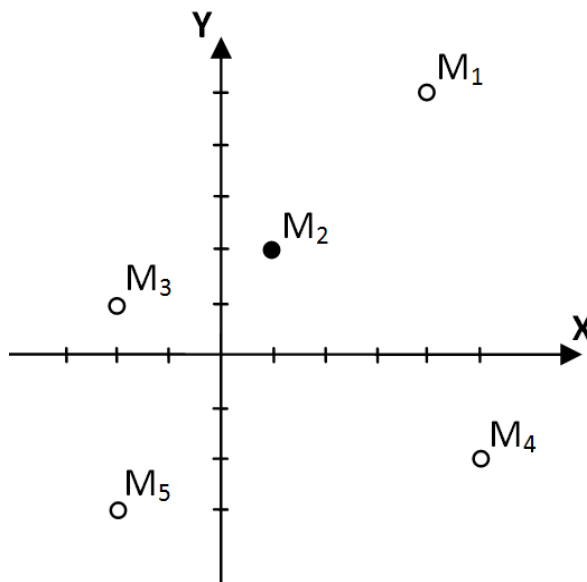
Обозначим:  $K, XMIN$  – номер и абсцисса искомой точки.

Обозначения переменных			
Имя переменной	Тип переменной		Пояснение
	Fortran	C	
$N, i$	<b>INTEGER</b>	<b>INT</b>	Количество точек и номер точки
$X_i, Y_i$	<b>REAL</b>	<b>FLOAT</b>	Координаты точки с номером $i$
$XMIN$	<b>REAL</b>	<b>FLOAT</b>	Абсцисса искомой точки
$K$	<b>INTEGER</b>	<b>INT</b>	Номер искомой точки
$X_k, Y_k$	<b>REAL</b>	<b>FLOAT</b>	Координаты искомой точки

## 3. Ручной расчёт результатов

### Графическое представление точек:

Исходные данные заданы в соответствии с условием задачи. Это означает, что среди заданных две точки  $M_1(4; 5)$  и  $M_2(1; 2)$ , расположены в первом квадранте. Из них искомой (с наименьшей абсциссой) является точка  $M_2(1; 2)$ . Принадлежность точки  $M_i(X_i, Y_i)$  первому квадранту определяется условиями  $X_i > 0$  и  $Y_i > 0$ .



При решении задачи используется алгоритм вычисления наименьшего значения. Параметр цикла – номер точки  $i$ .

Для контроля и наглядности результатов точки первого квадранта выводятся на печать. Из них осуществляется выбор искомой точки с наименьшей абсциссой  $XMIN$ .

Схема алгоритма представлена на рис. 2.19:

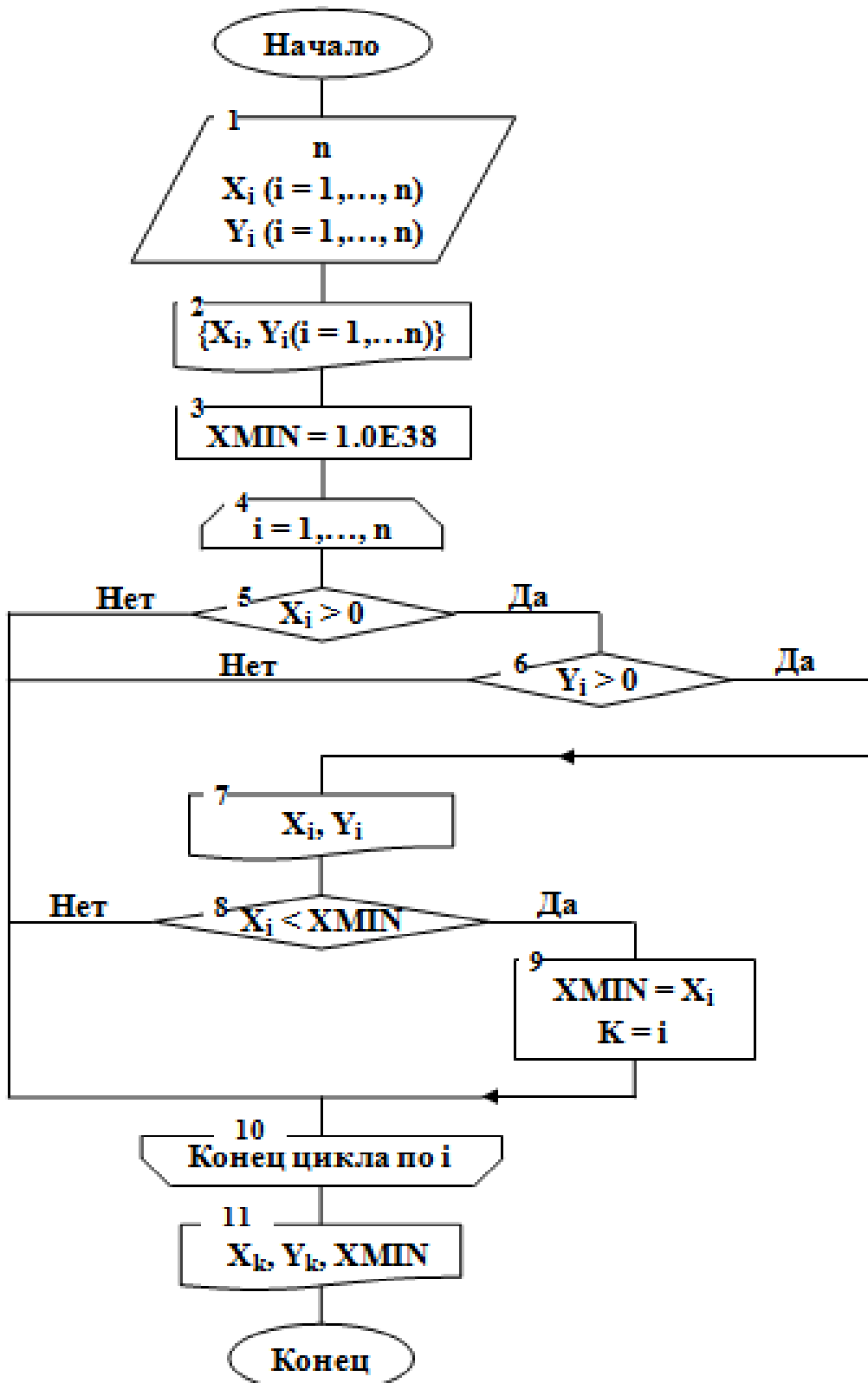


Рис. 2.19 Схема алгоритма к лабораторной работе 11  
«Вычисление координат искомой точки»



## 4. Листинг Fortran-программы

Условия  $X(I) > 0$  и  $Y(I) > 0$  объединены логической связкой **.AND.**

```
! Вычисление координат искомой точки
PROGRAM POINT
! Объявление типа и задание данных
INTEGER :: I,K,N = 5 ! I,K,N - целого типа
REAL :: XMIN = 1.0E38
REAL :: X(1 : 5) = (/4,1,-2,5,-2/)
REAL :: Y(1 : 5) = (/5,2,1,-2,-3/)
! Вывод координат точек
WRITE(*,1) '      КООРДИНАТЫ ТОЧЕК:'
WRITE(*,"(5F5.0/)")X,Y
WRITE(*,*)' ТОЧКИ, ЛЕЖАЩИЕ В 1 КВАДРАНТЕ: '
DO I = 1,N ! Начало цикла
! Вычисление и вывод точек 1 квадранта
IF(X(I) > 0 .AND. Y(I) > 0) THEN ! Если  $X_i > 0$  и  $Y_i > 0$ 
WRITE(*,2)I,X(I),I,Y(I)
2 FORMAT(1X,'X(',I2,') = ',F3.0, &
1X,'Y(',I2,') = ',F3.0)
! Вычисление наименьшей абсциссы
IF(X(I) < XMIN) THEN ! Если  $X_i < X_{min}$ 
XMIN = X(I); K = I
END IF
END IF
END DO ! Конец цикла
! Вывод координат искомой точки  $X_k, Y_k$ 
WRITE(*,*)' ИСКОМАЯ ТОЧКА:'
WRITE(*,12)K,X(K),K,Y(K),XMIN
12 FORMAT(1X,'X(',I2,') = ',F3.0, 1X, &
'Y(',I2,') = ',F3.0,' XMIN = ',F3.0)
END
```

## 5. Макет печати результатов

```
      КООРДИНАТЫ ТОЧЕК:
4.      1.      -2.      5.      -2.
5.      2.      1.      -2.      -3.
ТОЧКИ, ЛЕЖАЩИЕ В 1 КВАДРАНТЕ:
X (1) = 4.   Y (1) = 5.
X (2) = 1.   Y (2) = 2.
      ИСКОМАЯ ТОЧКА:
X (2) = 1.   Y (2) = 2.   XMIN = 1
```

Результаты ручного и машинного расчётов совпадают. На графике искомая точка обозначена (стр.53).

## Лабораторная работа 12

### Вычисление значения выражения

**Тема. Одномерный массив: сумма, произведение, min, max**

Индивидуальные задания – табл. 3.12 (стр.115)

Математические функции – табл. 5.1 (стр.136)

#### 1. Задание

Дан массив  $A_i$ , ( $i = 1, 2, \dots, n$ ),  $n = 4$ . Составить программу вычисления значения величины  $Z$ , заданной выражением:

$$Z = \sum_{i=2}^n i \cdot A_i + \max_i \left( A_i \cdot \prod_{i=1}^{n-1} i^2 \right)$$

При вычислении  $\max$  значение  $i$  изменяется в интервале  $[1, n]$ .

#### Исходные данные

Количество элементов	Элементы массива			
$n$	$A_1$	$A_2$	$A_3$	$A_4$
<b>4</b>	<b>1,5</b>	<b>-2</b>	<b>3</b>	<b>1,2</b>

#### 2. Решение

Введём дополнительные обозначения:

$$S = \sum_{i=2}^n i \cdot A_i, \quad P = \prod_{i=1}^{n-1} i^2, \quad \text{AMAX} = \max_i \left( A_i \cdot \prod_{i=1}^{n-1} i^2 \right), \quad R = A_i \cdot P.$$

Тогда выражение для  $Z$  можно представить:  $Z = S + \text{AMAX}$ ;

Обозначения переменных			
Имя переменной	Тип переменной		Пояснение
	Fortran	C	
<b>i</b>	<b>INTEGER</b>	<b>INT</b>	Порядковый номер элемента в массиве
$A_i$	<b>REAL</b>	<b>FLOAT</b>	Элемент массива с номером $i$
<b>N</b>	<b>INTEGER</b>	<b>INT</b>	Количество элементов в массиве
<b>S, P</b>	<b>REAL</b>	<b>FLOAT</b>	Сумма и произведение
<b>R</b>	<b>REAL</b>	<b>FLOAT</b>	Рабочая переменная
<b>AMAX</b>	<b>REAL</b>	<b>FLOAT</b>	Наибольшее значение
<b>Z</b>	<b>REAL</b>	<b>FLOAT</b>	Искомая величина, результат

### 3. Ручной расчёт результатов

Вычислим значение  $Z$  в следующем порядке:

$$S = \sum_{i=2}^4 i \cdot A_i = 2 \cdot A_2 + 3 \cdot A_3 + 4 \cdot A_4 = \mathbf{9,8} \quad P = \prod_{i=1}^3 i^2 = 1^2 \cdot 2^2 \cdot 3^2 = 36$$

$$\mathbf{AMAX} = \max_i (A_i \cdot P) = \max (A_1 \cdot P, A_2 \cdot P, A_3 \cdot P, A_4 \cdot P) = \mathbf{108}$$

$$\mathbf{Z} = S + \mathbf{AMAX} = 9,8 + 108 = \mathbf{117,8}$$

Алгоритм содержит три цикла с параметром  $i$ . Схема представлена на рис. 2.20 (стр.58).

### 4. Листинг С-программы «Вычисление значения выражения»

**Внимание.** При вводе программы следует иметь в виду, что  $C$  различает регистр букв, т.е., « $x$ » и « $X$ » – это две разные переменные. Нижняя граница индекса  $i$  (порядкового номера элемента в массиве) задаётся равной нулю.

```
# include<stdio.h> // Заголовочные файлы
# include<math.h>
# include<conio.h>
void main()
{
int i,n = 4; // Объявление типа и задание данных
float s = 0, p = 1, amax = -1.0e38, r, z;
float a[4] = {1.5,-2,3,1.2};
printf(" \t Array: \n");
for(i = 0;i<n;i++) // Цикл по i
    printf(" %6.1f ",a[i]); // Вывод A(i)
for(i = 1;i<n;i++) // Цикл вычисления S
    s+=(i+1)*a[i];
for(i = 1;i<=n-1;i++) // Цикл вычисления P
    p = p*i*i;
for(i = 0;i<n;i++) // Цикл вычисления AMAX
    {
        r = a[i] * p;
        if(r > amax) amax = r;
    }
z = s + amax;
printf(" \n \n \t Result \n");
printf("S = %.1f P = %.0f AMAX = %.0f\n",s,p,amax);
printf(" \n \t Z = %.1f ",z);
getch();
return;
}
```

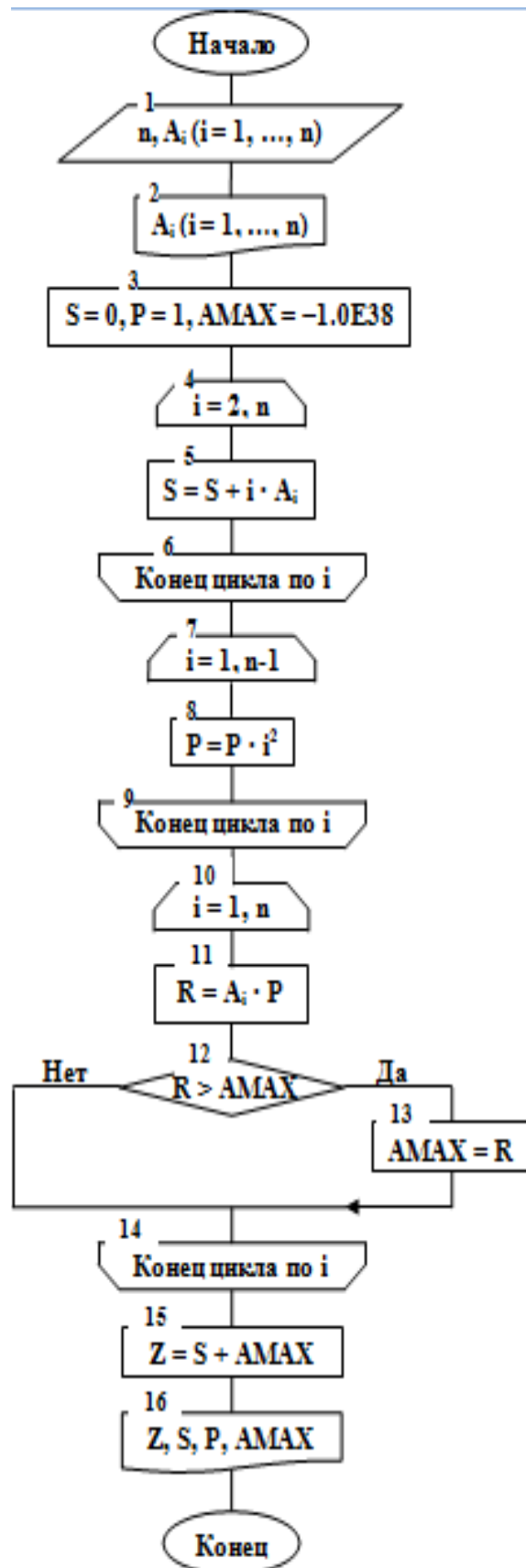


Рис. 2.20 Схема алгоритма к лабораторной работе 12  
«Вычисление значения выражения»

## 5. Макет печати результатов

Как видно, результаты ручного и машинного расчётов совпадают.

```

Array:
  1.5   -2.0   3.0   1.2
Result
S = 9.8   P = 36   AMAX = 108
Z = 117.8
    
```

### Лабораторная работа 13

## Анализ эффективности работы предприятия

Тема. Одномерный массив: сумма, количество, min, max

Индивидуальные варианты – табл. 3.13 (стр.117)

### 1. Задание

Составить программу решения следующей задачи:

Выполнение плана (в процентах) по объемам производства на судостроительном предприятии в течение года и названия месяцев представлены массивами  $A_i$  и  $MC_i$  ( $i = 1, \dots, 12$ ). План – 100%, менее 100% – план не выполнен, более 100% – перевыполнен.

Определить среди месяцев с процентом выполнения плана выше среднего месяц с наименьшим процентом.

### Исходные данные

1-е полугодие						
Месяц	1	2	3	4	5	6
	Январь	Февраль	Март	Апрель	Май	Июнь
%	$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$
	101	98	100	108	97	105
2-е полугодие						
Месяц	7	8	9	10	11	12
	Июль	Август	Сентябрь	Октябрь	Ноябрь	Декабрь
%	$A_7$	$A_8$	$A_9$	$A_{10}$	$A_{11}$	$A_{12}$
	99	107	96	112	100	98

### 2. Решение

Обозначим:  $S$ ,  $SR$  – суммарный и средний процент выполнения

плана.  $S = \sum_{i=1}^{12} A_i$ ,  $SR = S / 12$ .

$K$ ,  $AMIN$  – номер месяца с наименьшим процентом  $AMIN$ .

Обозначения переменных			
Имя переменной	Тип переменной		Пояснение
	Fortran	C	
<b>I</b>	<b>INTEGER</b>	<b>INT</b>	Номер элемента в массиве (месяца)
<b>A<sub>i</sub></b>	<b>REAL</b>	<b>FLOAT</b>	Элемент массива с номером I
<b>MC<sub>i</sub></b>	<b>CHARACTER</b>	<b>CHAR</b>	Элемент массива (месяц) с номером I
<b>S, SR</b>	<b>REAL</b>	<b>FLOAT</b>	Сумма и среднее арифметическое
<b>AMIN</b>	<b>REAL</b>	<b>FLOAT</b>	Наименьший процент выполнения плана
<b>K</b>	<b>INTEGER</b>	<b>INT</b>	Номер месяца с наименьшим процентом
<b>MC<sub>k</sub></b>	<b>CHARACTER</b>	<b>CHAR</b>	Название искомого месяца с номером K

### 3. Ручной расчёт результатов

Вычислим сумму  $S = 1221$  и процент  $SR = S/12 = 101,75$ .

#### Месяцы с процентом выполнения плана выше среднегодового

Месяц	4	6	8	10
		Апрель	Июнь	Август
Процент выполнения плана	108	105	107	112

Очевидно, что наименьший процент  $AMIN = 105$  достигается в июне-месяце с порядковым номером  $K = 6$ .

### 4. Алгоритм

При решении задачи используется алгоритм вычисления суммы и наименьшего значения. Алгоритм содержит два цикла, параметр цикла – номер месяца  $i$ .

В первом цикле осуществляется вычисление суммы  $S$ , а во втором определяются и выводятся названия месяцев с процентом выше среднегодового в соответствии с условием  $A_i > SR$ , а также вычисляется номер месяца  $K$  с наименьшим процентом  $AMIN$ .

#### Алгоритм состоит из следующих действий:

1. Ввод исходных данных – значений  $A_i$  и  $MC_i$  ( $i = 1, \dots, 12$ ).
2. Вывод данных – процентов выполнения плана  $A_i$ .
3. Задание начального значения переменной  $S = 0$ ,  $AMIN = 1.0E38$  (заведомо большого числа).
4. Начало цикла по  $i$ .
5. Вычисление суммы  $S = S + A_i$ .

6. Конец цикла.
7. Вычисление среднегодового процента.  $SR = S / 12$
8. Начало цикла по  $i$ .
9. Проверка условия  $A_i > SR$ .  
Если условие выполняется, перейти к п.10, иначе – к п.13.
10. Вывод  $i$ ,  $MC_i$ ,  $A_i$  – номера, названия месяца и процента.
11. Проверка условия  $A_i < AMIN$ .  
Если условие выполняется, перейти к п.12, иначе – к п.13.
12. Переменным присваиваются значения  $AMIN = A_i$ ,  $K = i$ .
13. Конец цикла.
14. Вывод значений  $SR$ ,  $K$ ,  $MC_k$ ,  $AMIN$ .
15. Конец.

Схема алгоритма представлена на рис. 2.21 (стр.63).

## 5. Листинг С-программы

**Внимание.** При вводе программы следует иметь в виду, что С различает регистр букв, т.е., «x» и «X» – это две разные переменные. Нижняя граница индекса  $i$  (порядкового номера элемента в массиве) задаётся равной нулю.

```
// Анализ итогов работы предприятия
#include<stdio.h> // Заголовочные файлы
#include<string.h>
#include<conio.h>
void main()
{
// Объявление типа и задание данных
int i,k;
float s = 0, amin = 1.0e38, sr;
float a[12] = {101,98,100,108,97,105,
              99,107,96,112,100,98};
char*mc[12] = {"january","february","march","april",
              "may","june","july","august","september",
              "october","november","december"};
printf(" \t\t \t Percent \n");
for(i = 0;i<12;i++) // Цикл по i
{
printf(" %.0f ",a[i]); // Вывод процентов  $A_i$ 
s += a[i]; // Вычисление суммы S
}
}
```

```

sr = s/12;
printf(" \n sr = %.2f \n ",sr); // Вывод SR
printf("  N \t Month \t Percent \n ");
for(i = 0;i<12;i++) // Цикл по i
{
    if(a[i] > sr ) // Если Ai > SR
    {
printf("i = %2d  %s \t %3.0f \n",i+1,mc[i],a[i]);
if(a[i] < amin) // Если Ai < AMIN
    {
        amin = a[i]; k = i;
    }
    }
}
printf(" \t Result \n"); // Вывод значений K,AMIN
printf("k = %d  %s  amin = %.0f ",k+1,mc[k],a[k]);
getch();
return;
}

```

## 6. Макет печати результатов

```

                                     Percent
101  98  100  108  97  105  99  107  96  112  100  98
sr = 101.75
  N      Month      Percent
i = 4    april      108
i = 6    june       105
i = 8    august     107
i = 10   october    112
      Result
k = 6    june      amin = 105

```

Как видно, результаты ручного и машинного расчётов совпадают.



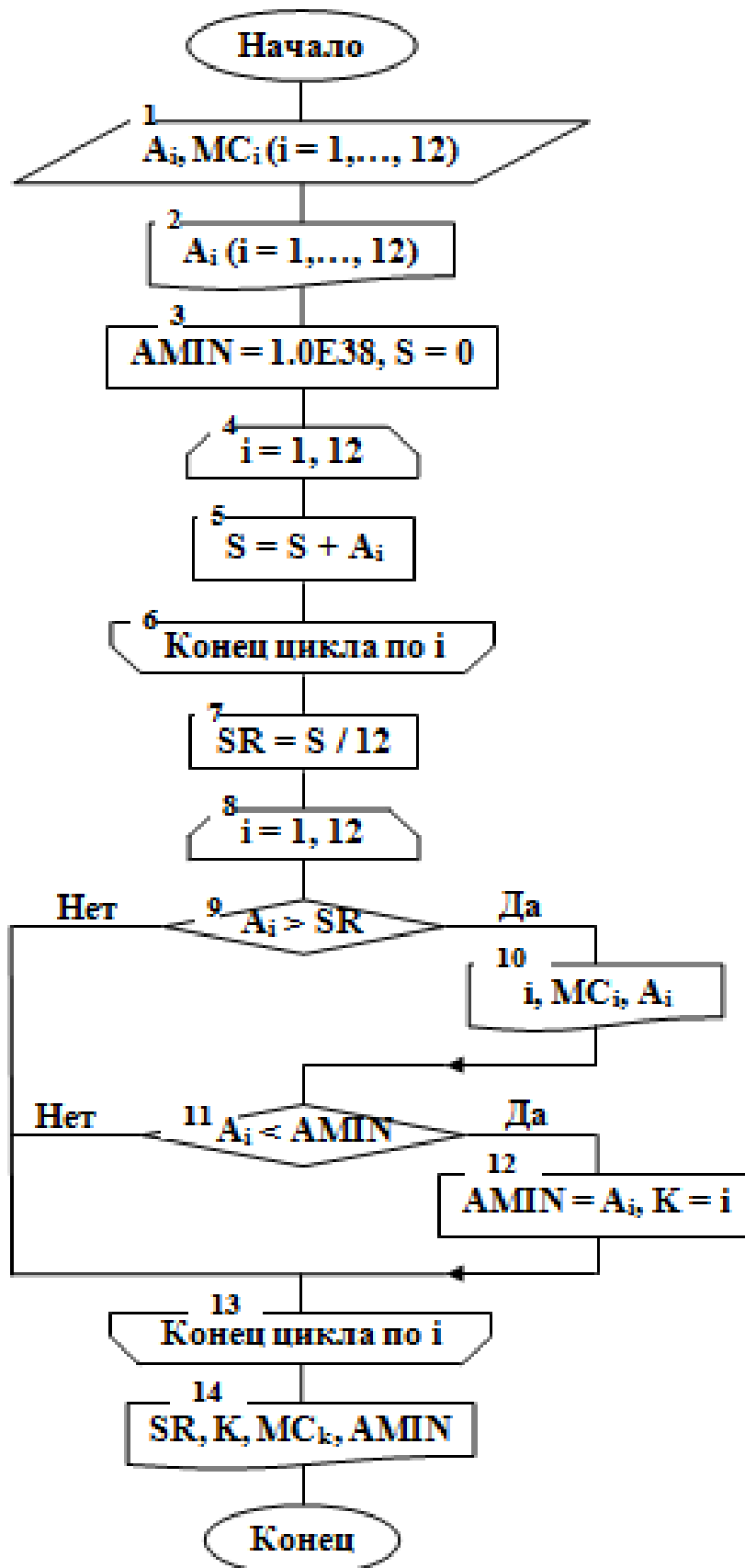


Рис. 2.21 Схема алгоритма к лабораторной работе 13 «Анализ эффективности работы предприятия»

**Лабораторная работа 14**  
**Анализ итогов экзамена в группе**  
**Тема. Одномерный массив: сумма, количество**

Индивидуальные задания – табл. 3.14 (стр.119)

**1. Задание**

Составить программу решения следующей задачи:

Результаты сдачи экзамена и фамилии студентов группы представлены массивами  $A_i$  и  $F_i$  ( $i = 1, \dots, 10$ ).

**Исходные данные**

Студент	$F_1$	$F_2$	$F_3$	$F_4$	$F_5$
Фамилия	Юдин И.В.	Ашин С.В.	Итин К.Л.	Бут С.С.	Бех Л.Л.
Оценка	$A_1$	$A_2$	$A_3$	$A_4$	$A_5$
	5	4	4	3	2
Студент	$F_6$	$F_7$	$F_8$	$F_9$	$F_{10}$
Фамилия	Этин П.Н.	Юров Ю.Ю.	Апин Р.П.	Фет В.И.	Кац Д.Е.
Оценка	$A_6$	$A_7$	$A_8$	$A_9$	$A_{10}$
	5	4	2	4	3

**Примечание.** При вводе фамилий в программе инициалы не учитываются!

**Определить количество студентов-хорошистов, а также количество и фамилии отличников.**

**2. Решение**

Обозначения переменных			
Имя переменной	Тип переменной		Пояснение
	Fortran	C	
$i$	INTEGER	INT	Номер элемента (студента в группе)
$A_i$	REAL	FLOAT	Элемент массива (оценка) с номером $i$
$F_i$	CHARACTER	CHAR	Элемент (фамилия студента) с номером $i$
$K4, K5$	INTEGER	INT	Количество оценок «хорошо» и «отлично»

**3. Ручной расчёт результатов**

Подсчитаем количество оценок «хорошо» и «отлично». Очевидно, что  $K4 = 4$ , а  $K5 = 2$ .

#### 4. Алгоритм

Вычисление количества хороших и отличных оценок в алгоритме осуществляется проверкой условий  $A_i = 4$  и  $A_i = 5$ .

**Алгоритм состоит из следующих действий:**

1. Ввод данных – элементов массивов  $F_i, A_i$  (фамилий и оценок).
2. Вывод данных (оценок студентов)  $A_i$ .
3. Задание начального значения переменным  $K4 = 0$  и  $K5 = 0$ .
4. *Начало цикла.*
5. Проверка условия  $A_i = 5$ .  
Если условие выполнено, перейти к п.6, иначе – к п.8.
6. Увеличить на 1 счётчик  $K5$ :  $K5 = K5 + 1$ .
7. Вывод фамилии студента-отличника  $F_i$ . Перейти к п.10.
8. Проверка условия  $A_i = 4$ .  
Если условие выполнено, перейти к п.9, иначе – к п.10.
9. Увеличить на 1 значение счётчика  $K4$ :  $K4 = K4 + 1$ .
10. *Конец цикла.*
11. Вывод результатов – значений  $K5$  и  $K4$ .
12. Конец.

Схема алгоритма представлена на рис. 2.22 (стр.66).

#### 5. Листинг Fortran-программы «Анализ итогов экзамена»

```
PROGRAM MARKS
! Объявление типа и задание данных
INTEGER :: I, K4 = 0, K5 = 0, N = 10
INTEGER :: A(1:10) = (/5,4,4,3,2,5,4,2,4,3/)
CHARACTER*8 :: F (1:10)
DATA F/' Юдин ',' Ашин ',' Итин ',' Бут ',' Бех ',' &
      ' Этин ',' Юров ',' Апин ',' Фет ',' Кац '/'
WRITE(*,*) ' ОЦЕНКИ '
WRITE(*,' (2X,10I3) ') A ! Вывод оценок
WRITE(*,*) ' ОТЛИЧНИКИ ' ! Вывод заголовка
DO I = 1 TO N ! Начало цикла
  IF(A(I) == 5) THEN ! Если  $A_i = 5$ 
    K5 = K5 + 1
    WRITE(*,' (8X,A) ') F(I) ! Вывод отличников  $F_i$ 
  ELSE ! Иначе, если  $A_i \neq 5$ 
    IF(A(I) == 4) K4 = K4 + 1 ! Если  $A_i = 4$ 
  END IF
END DO ! Конец цикла
```

```

! Вывод значений K4, K5
WRITE (*, 2) K5, K5*100.0/N
2 FORMAT (2X, 'K5 = ', I1, ' % = ', F4.1)
WRITE (*, 3) K4, K4*100.0/N
3 FORMAT (2X, 'K4 = ', I1, ' % = ', F4.1)
END

```

Результаты работы программы представлены в п.7.

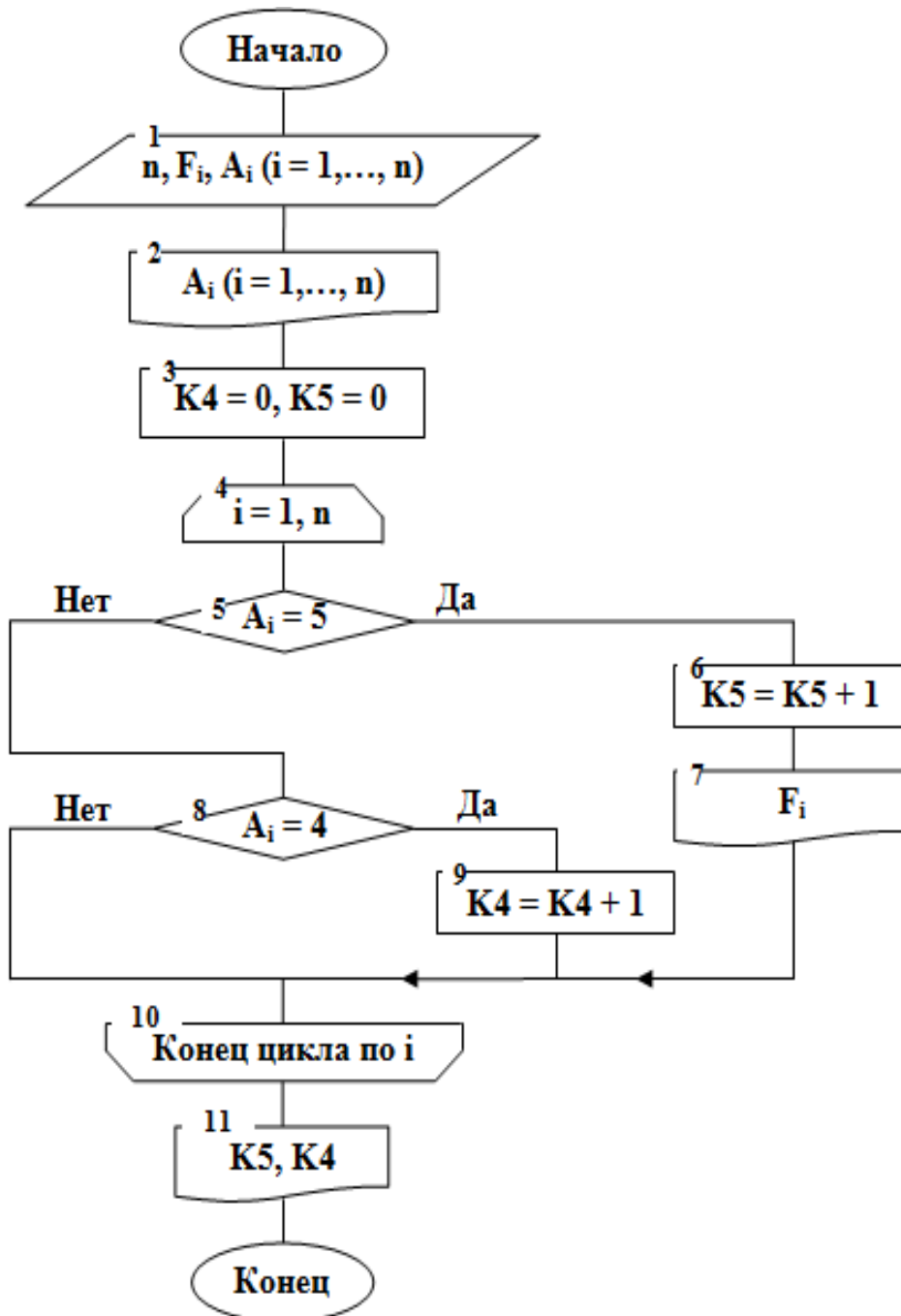


Рис. 2.22 Схема алгоритма к лабораторной работе 14 «Анализ итогов сдачи сессии»

## 6. Листинг Basic-программы

**Примечание.** При вводе фамилий в программе инициалы не учитываются!

```
' Анализ итогов сессии
CLS
DIM A(10), F$(10)
' Задание фамилий и оценок студентов
DATA "ЮДИН", 5, "АШИН", 4, "ИТИН", 4, "БУТ", 3, "БЕХ", 2,
DATA "ЭТИН", 5, "ЮРОВ", 4, "АПИН", 2, "ФЕТ", 4, "КАЦ", 3
PRINT TAB(7); " ОЦЕНКИ "
' Ввод фамилий и оценок
FOR I = 1 TO 10 ' Начало цикла
  READ F$(I), A(I)
  PRINT A(I); ' Вывод оценок Ai
NEXT I ' Конец цикла
PRINT ' Пропуск строки
PRINT STRING$(28, "-") ' Вывод строки символов
K5 = 0: K4 = 0: N = 10
PRINT TAB(7); " ОТЛИЧНИКИ "
FOR I = 1 TO N ' Начало цикла
  IF A(I) = 5 THEN ' Если Ai = 5
    K5 = K5 + 1: PRINT TAB(6); F$(I)
  ELSE ' Иначе, если Ai ≠ 5
    IF A(I) = 4 THEN K4 = K4 + 1
  END IF
NEXT I ' Конец цикла
' Вывод результатов
PRINT USING " K5 = # ## % "; K5, K5*100/N
PRINT USING " K4 = # ## % "; K4, K4*100/N
END
```

Значения переменных K4 и K5 выводятся также и в процентах.

## 7. Макет печати результатов (по Basic-программе)

Как видно, результаты  
ручного и машинного  
расчётов совпадают.

ОЦЕНКИ									
5	4	4	3	2	5	4	2	4	3
ОТЛИЧНИКИ									
ЮДИН									
ЭТИН									
K5 = 2 20%									
K4 = 4 40%									

# Контрольное задание для проверки знаний

## Вычисление площади шпангоута

### Тема. Одномерные массивы: сумма

Составить схему и программу решения для следующей задачи:

Для расчёта мореходных качеств судна необходимо знать элементы теоретического корпуса судна, имеющего сложную геометрическую форму. К таким элементам относят его объём, площади поперечных и продольных сечений, статические моменты объёма и площадей и др. Все выражения для вычисления элементов теоретического корпуса судна имеют интегральный вид. Например, площадь поперечного сечения судна, шпангоута (рис. 2.23), симметричного относительно ДП судна, определяется выражением:

$$\omega = 2 \int_0^T y dz, \text{ где } y - \text{ординаты шпангоута.}$$

Контур половины шпангоута судна представлен на рис. 2.24.

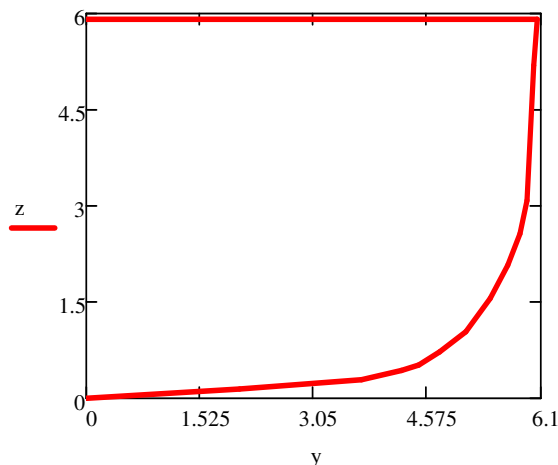
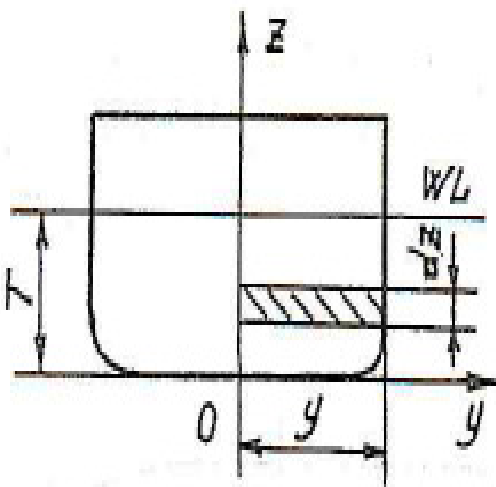


Рис. 2.23 «Контур шпангоута»

Рис. 2.24 «Контур половины шпангоута»

Однако все шпангоуты судна описываются кривыми  $y = f(z)$ , для которых затруднительно подобрать аналитическое выражение. Поэтому расчёт площади  $\omega$  проводят численным методом — по правилу трапеций с переменным шагом по формуле:

$$\omega = \sum_{j=1}^{m-1} f_j,$$

где  $f_j$  – площадь j-той трапеции шпангоута.

$$f_j = \frac{1}{2} (z_{j+1} - z_j) (y_{j+1} + y_j)$$

где  $z_j, y_j$  – координаты j-той точки шпангоута, m – количество точек, j – номер точки. Они снимаются с теоретического чертежа судна в той последовательности, как показано на рис. 2.25.

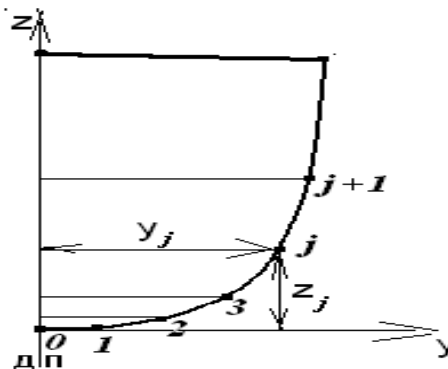


Рис. 2.25 «Точки ветви шпангоута»

### Задание

Вычислить площадь шпангоута методом трапеции по формуле:

$$\omega = \sum_{j=1}^{m-1} \frac{1}{2} (z_{j+1} - z_j) (y_{j+1} + y_j), \text{ где } m = 13.$$

Координаты ветви шпангоута представлены в таблице:

№	z	y
0	0	0
1	0,150	2,050
2	0,300	3,700
3	0,425	4,225
4	0,516	4,475
5	0,725	4,750
6	1,050	5,100
7	1,550	5,425
8	2,075	5,650
9	2,575	5,815
10	3,100	5,911
11	5,180	6,015
12	5,900	6,050
13	5,900	0

**Лабораторная работа 15**  
**Анализ эффективности работы танкерного флота**  
**Тема. Двумерный массив (матрица)**

Индивидуальные задания – табл. 3.15 (стр.120)

**1. Задание**

Составить схему и программу решения для следующей задачи.

Фактические данные по объёмам перевозок грузов пятью судами за четыре месяца навигации представлены в виде матрицы  $A(5 \times 4)$ , где строка – судно, столбец - месяц. Данные приведены в процентном отношении к нормативным объёмам перевозок (плану). За план принимается значение, равное 100%. Менее 100% – план не выполнен, более 100% – перевыполнен.

Матрица задаётся в соответствии с условием задачи. Элемент матрицы  $A_{ij}$  – это процент выполнения плана судном с номером  $i$  за месяц навигации с номером  $j$ .

Рассмотрим три варианта задачи:

**Вариант 1.**

**Определить, каким судном и за какой месяц навигации достигнут наибольший процент выполнения плана.**

**Исходные данные**

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \\ A_{51} & A_{52} & A_{53} & A_{54} \end{pmatrix} = \begin{pmatrix} 102 & 99 & 108 & 103 \\ 98 & 100 & 90 & 93 \\ 101 & 103 & 95 & 105 \\ 100 & 102 & 99 & 104 \\ 97 & 106 & 107 & 98 \end{pmatrix}$$

Названия судов и месяцев заданы массивами  $TN(5)$  и  $MC(4)$ .

Суда					Месяцы навигации			
$TN_1$	$TN_2$	$TN_3$	$TN_4$	$TN_5$	$MC_1$	$MC_2$	$MC_3$	$MC_4$
Танкер-1	Танкер-2	Танкер-3	Танкер-4	Танкер-5	Май	Июнь	Июль	Август

**Примечание.** Возможен более простой и краткий вариант программы без использования названий судов и месяцев. Результаты представлены в макете печати (п.6, стр.74).



## 2. Решение

### Математическая формулировка задачи:

Дана матрица  $A(5 \times 4)$ . Определить номер строки и столбца с наибольшим элементом матрицы.

Переменные, используемые в задаче, приведены в таблице:

Обозначения переменных			
Имя переменной	Тип переменной		Пояснение
	Fortran	C	
<b>N, M</b>	<b>INTEGER</b>	<b>INT</b>	Количество строк и столбцов матрицы
<b>I, J</b>	<b>INTEGER</b>	<b>INT</b>	Номер строки и столбца матрицы
<b>A<sub>ij</sub></b>	<b>REAL</b>	<b>FLOAT</b>	Элемент матрицы
<b>AMAX</b>	<b>REAL</b>	<b>FLOAT</b>	Наибольший элемент матрицы
<b>K, L</b>	<b>INTEGER</b>	<b>INT</b>	Номер строки и столбца с наибольшим элементом
<b>TN<sub>k</sub>, MC<sub>L</sub></b>	<b>CHARACTER</b>	<b>CHAR</b>	Название судна и месяца с наибольшим процентом выполнения плана

## 3. Ручной расчёт результатов

Процент **AMAX** = 108 достигнут судном «Танкер–1» с номером **K** = 1 за месяц навигации «июль» с номером **L** = 3.

## 4. Алгоритм

При решении задачи используется алгоритм вычисления наибольшего значения. Обход матрицы можно осуществлять и по строкам, и по столбцам. В алгоритме выбран вариант «по строкам». Алгоритм – двойной цикл вложенной структуры. Параметр внешнего цикла – номер строки **i** (судна), а внутреннего – номер столбца **j** (месяца).

Схема алгоритма представлена на рис. 2.26:

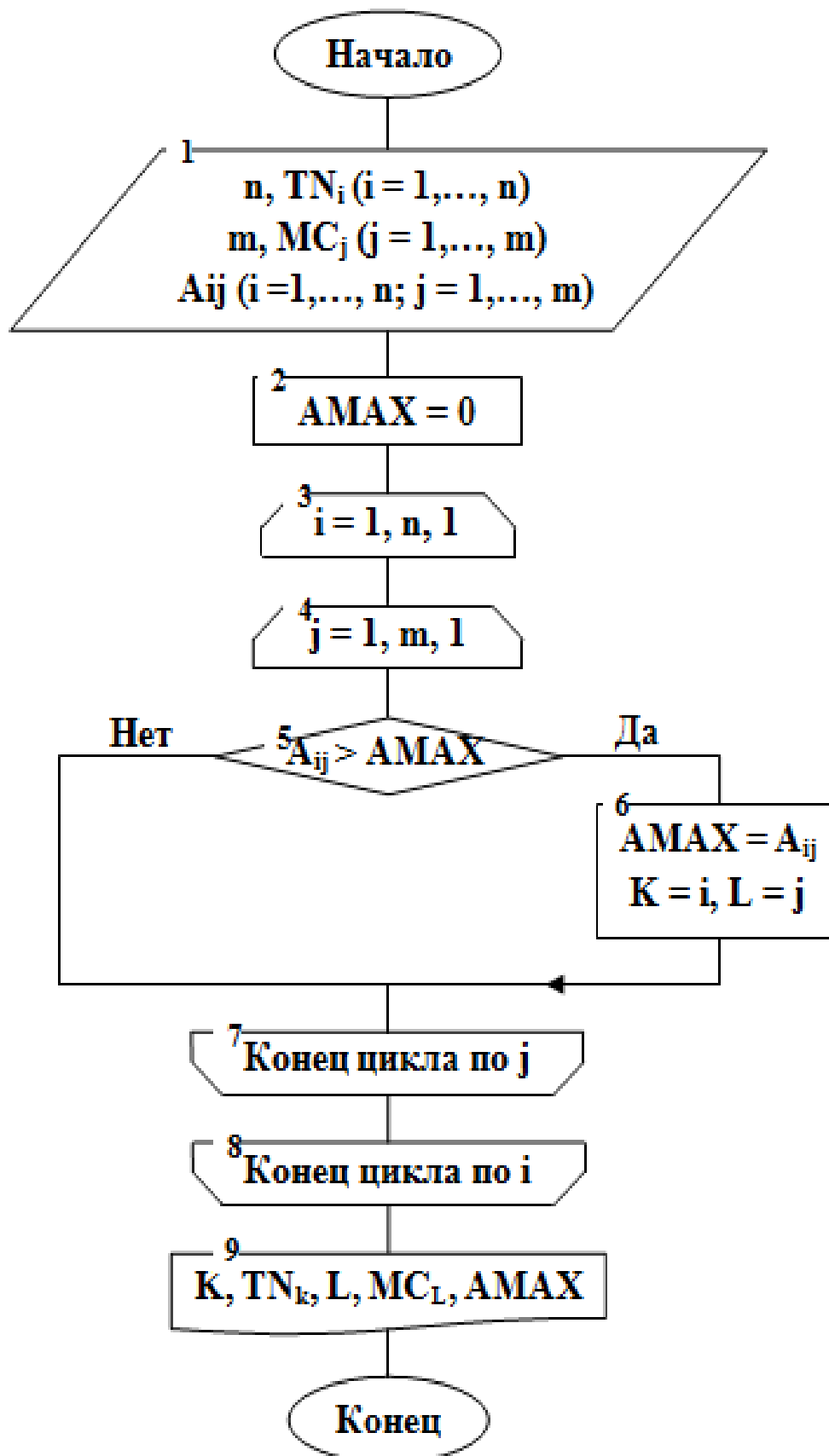


Рис. 2.26 Схема алгоритма к лабораторной работе 15 (вариант 1)  
«Анализ эффективности работы флота»

## 5. Листинг С-программы «Анализ работы флота»

**Внимание.** При вводе программы следует иметь в виду, что С различает регистр букв, т.е., «x» и «X» – это две разные переменные. Нижние границы индексов  $i$  и  $j$  (номера строки и столбца) задаются равными нулю.

```
#include<stdio.h> // Заголовочные файлы
#include<conio.h>
#include<string.h>
void main()
{
// Объявление типа и задание данных
int i, j, k, l, n = 5, m = 4;
char *mc[4] = {"May", "June", "July", "August"};
char *tn[5] = {"Tanкер-1", "Tanкер-2",
              "Tanкер-3", "Tanкер-4", "Tanкер-5"};
float amax, a[5][4] = {102, 99, 108, 103,
                      98, 100, 90, 93,
                      101, 103, 95, 105,
                      100, 102, 99, 104,
                      97, 106, 107, 98};
// Вывод названий месяцев - цикл по i
printf(" Tanкер");
for(i = 0; i < m; i++)
    printf(" %s ", mc[i]);

printf(" \n "); // Пропуск строки
amax = 0;
for(i = 0; i < n; i++) // Внешний цикл по i
{
    printf("%s", tn[i]); // Вывод названия танкера TNi
    for(j = 0; j < m; j++) // Внутренний цикл по j
    {
        printf("%7.0f", a[i][j]); // Вывод процентов Aij
        if(a[i][j] > amax) // Если Aij > AMAX
        {
            amax = a[i][j];
            k = i; l = j;
        }
    }
}
printf(" \n "); // Пропуск строки
}
```

```
// Вывод результатов
printf(" \n \t R e s u l t \n ");
printf(" Tankep  Month Percent \n");
printf(" %s %s %5.0f \n ", tn[k],mc[l],amax);
printf(" K = %d L= %d AMAX =%.0f ",k+1,l+1,amax);
getch();
return;
}
```

## 6. Макет печати (для варианта 1)

Для сравнения приведён макет печати для варианта без названий судов и месяцев. Программу составить самостоятельно.

### По С-программе

Tankep	May	June	July	August		МАТРИЦА			
Tankep-1	102	99	108	103	102	99	<b>108</b>	103	
Tankep-2	98	100	90	93	98	100	90	93	
Tankep-3	101	103	95	105	101	103	95	105	
Tankep-4	100	102	99	104	100	102	99	104	
Tankep-5	97	106	107	98	97	106	107	98	
	R e s u l t					Результат			
Tankep	Month	Percent				K = 1 L = 3			
Tankep-1	July	108				AMAX = 108			
K = 1	L = 3	AMAX=108							

Результаты ручного и машинного расчётов совпадают.

## 7. Листинг Basic-программы

Программный код	Пояснения
' Анализ эффективности работы флота	
<b>CLS</b>	
<b>DIM</b> A(5, 4), MC\$(4), TN\$(5)	
<b>DATA</b> "МАЙ ", " ИЮНЬ ", "ИЮЛЬ", " АВГУСТ "	Задание месяцев и судов
<b>DATA</b> "Tankep-1 ", "Tankep-2", "Tankep-3 "	
<b>DATA</b> "Tankep-4 ", "Tankep-5 "	
N = 5 : M = 4	
<b>DATA</b> 102, 99, 108, 103	Задание элементов матрицы
<b>DATA</b> 98, 100, 90, 93	
<b>DATA</b> 101, 103, 95, 105	
<b>DATA</b> 100, 102, 99, 104	
<b>DATA</b> 97, 106, 107, 98	
<b>PRINT</b> "Tankep"; SPC(7);	

<pre> FOR I = 1 TO M      ' Начало цикла   READ MC\$(I)   PRINT MC\$(I); NEXT I              ' Конец цикла AMAX = 0 FOR I = 1 TO N      ' Внешний цикл по I   READ TN\$(I)   PRINT TN\$(I);   FOR J = 1 TO M      ' Внутренний цикл по J     READ A(I, J)     PRINT USING "#####"; A(I, J);     IF A(I, J) &gt; AMAX THEN       AMAX = A(I, J)       K = I: L = J     END IF   NEXT J            ' Конец внутреннего цикла по J PRINT ' Пропуск строки NEXT I              ' Конец внешнего цикла по I PRINT TAB(10); "Результат" PRINT "Танкер Месяц Процент" PRINT TN\$(K), MC\$(L), AMAX PRINT " K = ";K;" L = ";L;" AMAX = ";AMAX END </pre>	<p>Ввод-вывод месяцев</p> <p>Ввод-вывод танкеров</p> <p>Ввод-вывод строки</p> <p>Вычисление AMAX, K, L</p> <p>Вывод результатов</p>
--	---

Результаты представлены в макете печати (п.6, стр.74).

## Вариант 2.

### 1. Задание

Определить для каждого судна месяц навигации с наибольшим процентом выполнения плана.

#### Исходные данные

$$A = \begin{pmatrix} 102 & 99 & 108 & 103 \\ 98 & 100 & 90 & 93 \\ 101 & 103 & 95 & 105 \\ 100 & 102 & 99 & 104 \\ 97 & 106 & 107 & 98 \end{pmatrix}$$

## 2. Решение

### Математическая формулировка задачи:

Дана матрица  $A(5 \times 4)$ . Определить для каждой строки номер столбца с наибольшим элементом.

Таблица имён переменных приведена в варианте 1 (стр.71).

Здесь:  $AMAX$  – наибольший элемент строки матрицы (процент);  $L, MC_L$  – номер и название месяца с процентом  $AMAX$ .

## 3. Ручной расчёт результатов

Результаты расчёта приведены в следующей таблице:

Судно (строка)		Месяц (столбец)		Наибольший элемент строки (процент $AMAX$ )
Номер $i$	Название $TN_i$	Номер $L$	Название $MC_L$	
1	Танкер–1	3	Июль	108
2	Танкер–2	2	Июнь	100
3	Танкер–3	4	Август	105
4	Танкер–4	4	Август	104
5	Танкер–5	3	Июль	107

## 4. Алгоритм

Алгоритм решения задачи – двойной цикл вложенной структуры. Обход матрицы для данного варианта осуществляется *только по строкам* (судам). Параметр внешнего цикла – номер строки  $i$  (судна), а внутреннего – номер столбца  $j$  (месяца).

Схема алгоритма представлена на рис. 2.27 (стр.80).

## 5. Макет печати (для варианта 2)

Tanкер	May	June	July	August	I	Percent	Month
Tanкер–1	102	99	108	103	I	108	July
Tanкер–2	98	100	90	93	I	100	June
Tanкер–3	101	103	95	105	I	105	August
Tanкер–4	100	102	99	104	I	104	August
Tanкер–5	97	106	107	98	I	107	July

Приведём продолжение С-программы варианта 2. Начало программы: файлы заголовков, объявление типа данных, названия судов, месяцев, матрицы – см. вариант 1 (п.5, стр.73).

## 6. Фрагмент листинга С-программы

**Внимание.** При вводе программы следует иметь в виду, что С различает регистр букв, т.е., «x» и «X» – это две разные переменные. Нижние границы индексов *i* и *j* (номера строки и столбца) задаются равными нулю.

```
printf(" I Percent  Month \n ");
// Внешний цикл по i (по строкам)
for(i = 0; i<n; i++)
{
    amax = 0;
    printf("%s", tn[i]); // Вывод названия танкера
// Внутренний цикл по j (по столбцам)
    for(j = 0; j<m; j++)
    {
        printf("%6.0f", a[i][j]); // Вывод процентов Aij
// Вычисление max элемента в строке AMAX и L
        if(a[i][j]>amax) // Если Aij > AMAX
        {
            amax = a[i][j]; l = j;
        }
    }
// Вывод для каждого танкера max процента и месяца
printf("      I %4.0f      %s \n ", amax, mc[l]);
}
getch();
return;
}
```

Результаты представлены в макете печати (п.5, стр.76).

### Вариант 3.

#### 1. Задание

Определить для каждого месяца навигации судно с наибольшим процентом выполнения плана.

#### Исходные данные

$$A = \begin{pmatrix} \mathbf{102} & 99 & \mathbf{108} & 103 \\ 98 & 100 & 90 & 93 \\ 101 & 103 & 95 & \mathbf{105} \\ 100 & 102 & 99 & 104 \\ 97 & \mathbf{106} & 107 & 98 \end{pmatrix}$$

## 2. Решение

### Математическая формулировка задачи:

Дана матрица  $A(5 \times 4)$ . Определить для каждого столбца матрицы номер строки с наибольшим элементом.

Таблица имён переменных приведена в варианте 1 (стр.71).

Здесь: **АМАХ** – наибольший элемент столбца матрицы (процент); **К**, **TN<sub>к</sub>** – номер и название судна с процентом АМАХ.

## 3. Ручной расчёт результатов

Результаты расчёта приведены в следующей таблице:

Месяц (столбец)		Судно (строка)		Наибольший элемент столбца матрицы (процент АМАХ)
Номер j	Название МС(j)	Номер К	Название TN(k)	
1	Май	1	Танкер – 1	102
2	Июнь	5	Танкер – 5	106
3	Июль	1	Танкер – 1	108
4	Август	3	Танкер – 3	105

## 4. Алгоритм

Алгоритм решения задачи – двойной цикл вложенной структуры. Обход матрицы для данного варианта осуществляется *только по столбцам*. Параметр внешнего цикла – номер столбца  $j$  (месяца), а внутреннего – номер строки  $i$  (судна).

Схема алгоритма представлена на рис. 2.28 (стр.80).

Приведём продолжение С-программы варианта 3. Начало программы: файлы заголовков, объявление типа данных, названия судов, месяцев, задание матрицы – см. вариант 1 (п.5, стр.73).

## 5. Фрагмент листинга С-программы

**Внимание.** При вводе программы следует иметь в виду, что С различает регистр букв, т.е., « $x$ » и « $X$ » – это две разные переменные. Нижние границы индексов  $i$  и  $j$  (номера строки и столбца) задаются равными нулю.

```
printf(" \n "); // Пропуск строки
// Внешний цикл по i (по строкам)
for(i = 0; i < n; i++)
{
    printf("%s", tn[i]); // Вывод названия танкера
```



```

// Внутренний цикл по j (по столбцам)
    for(j = 0;j<m;j++)
        printf("%7.0f",a[i][j]); // Вывод процентов
printf(" \n "); // Пропуск строки
}
// Вычисление для месяцев танкера с max процентом
printf(" \t Result \n Percent   Tankep   Month \n");
// Внешний цикл по j (столбцам)
for(j = 0;j<m;j++)
{
    amax = 0;
// Внутренний цикл по i (строкам)
    for(i = 0;i<n;i++)
    {
        if(a[i][j]>amax) // Если Aij > AMAX
        {
            amax = a[i][j]; k = i;
        }
    }
// Вывод результатов для каждого месяца
    printf(" %5.0f   %s   %s \n ",amax,tn[k],mc[j]);
}
getch();
return;
}

```

### 6. Макет печати (для варианта 3)

Tankep	May	June	July	August
Tankep-1	102	99	108	103
Tankep-2	98	100	90	93
Tankep-3	101	103	95	105
Tankep-4	100	102	99	104
Tankep-5	97	106	107	98
<b>Result</b>				
<b>Percent</b>	<b>Tankep</b>	<b>Month</b>		
102	Tankep-1	May		
106	Tankep-5	June		
108	Tankep-1	July		
105	Tankep-3	August		

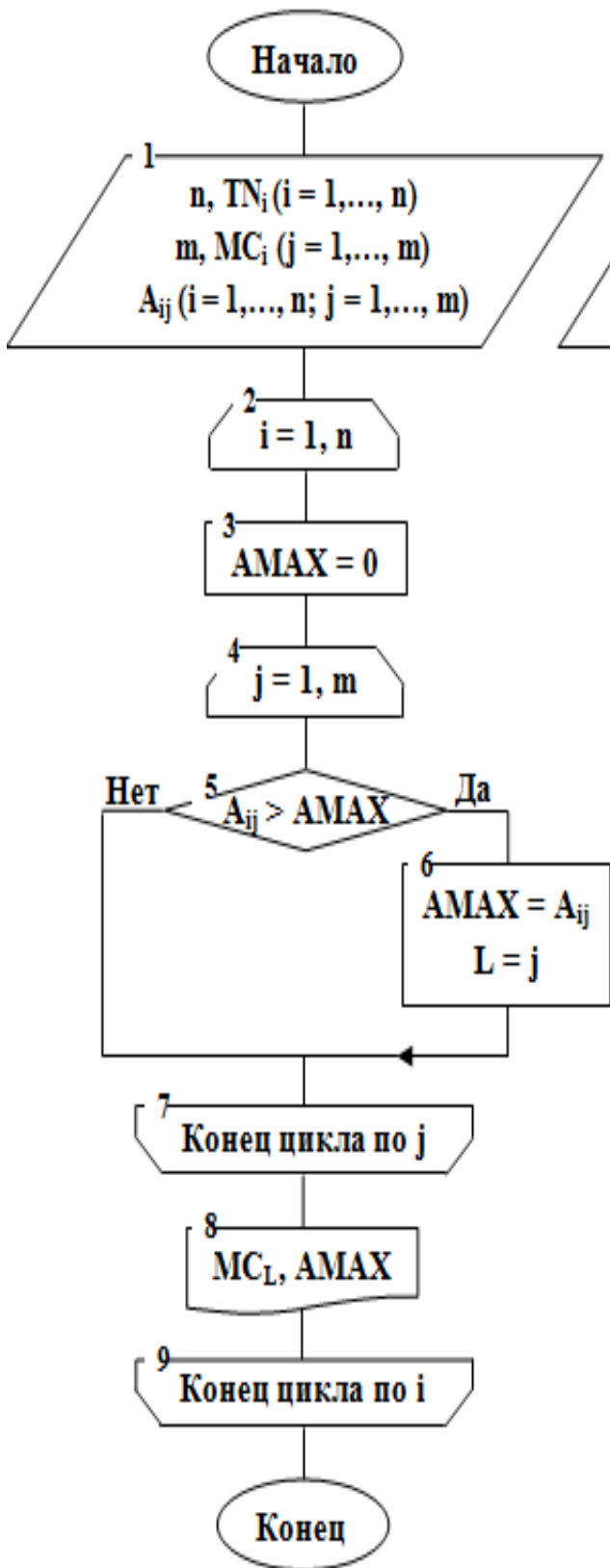


Рис. 2.27 Схема алгоритма к лабораторной работе 15 (вариант 2) «Анализ эффективности работы флота»

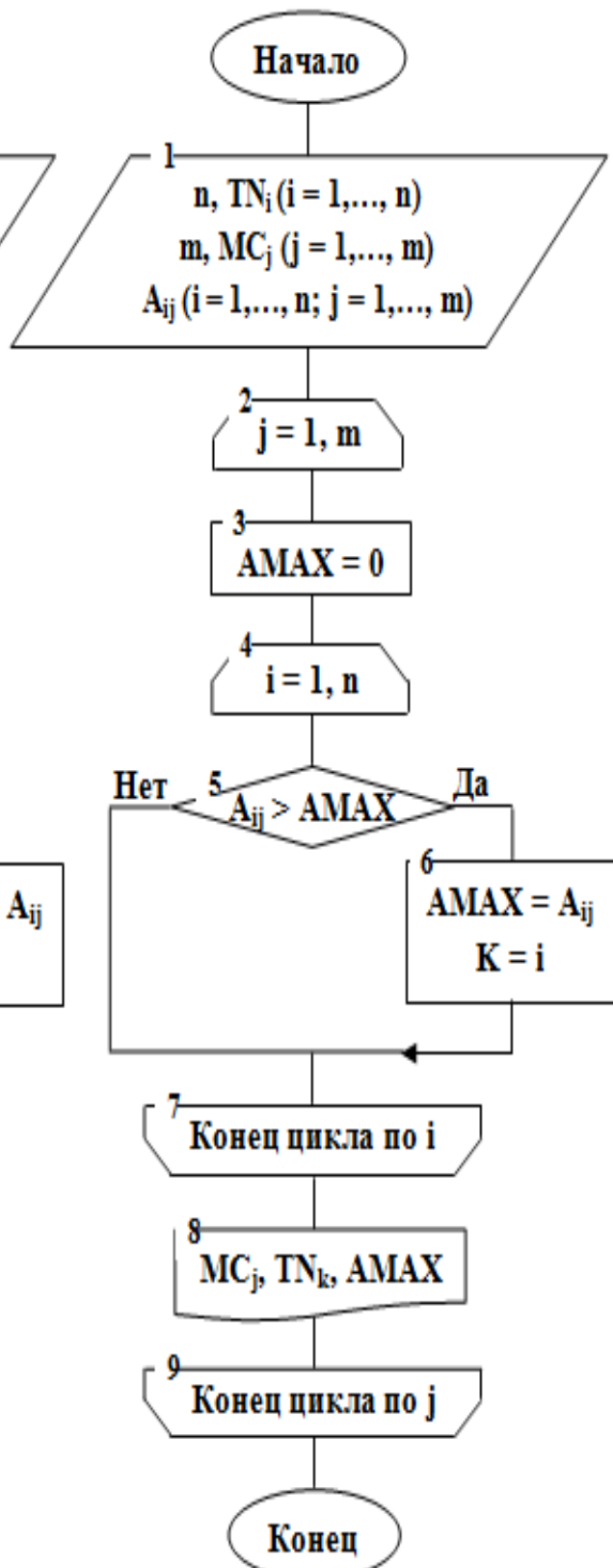


Рис. 2.28 Схема алгоритма к лабораторной работе 15 (вариант 3) «Анализ эффективности работы флота»

**Лабораторная работа 16**  
**Анализ итогов сессии в группе**  
**Тема. Двумерный массив (матрица)**

Индивидуальные задания – табл.3.16 (стр.122)

**1. Задание**

Составить схему и программу решения следующей задачи.

Результаты сессии из четырёх предметов в группе из пяти студентов представлены в виде матрицы  $A(5 \times 4)$ , где строка – студент, столбец – предмет. Элемент матрицы  $A_{ij}$  – оценка, полученная студентом с номером  $i$  по предмету с номером  $j$ . Матрица задаётся в соответствии с условием задачи.

**Определить средний балл каждого студента и студента с наименьшим средним баллом.**

**Исходные данные**

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \\ A_{51} & A_{52} & A_{53} & A_{54} \end{pmatrix} = \begin{pmatrix} 3 & 4 & 4 & 3 \\ 4 & 5 & 4 & 5 \\ 3 & 3 & 3 & 4 \\ 5 & 5 & 4 & 5 \\ 4 & 4 & 3 & 4 \end{pmatrix}$$

**2. Решение**

**Математическая формулировка задачи:**

Дана матрица  $A$  ( $5 \times 4$ ). Найти среднее арифметическое элементов каждой строки и номер строки с наименьшим средним арифметическим.

Переменные, используемые в задаче, приведены в таблице:

Обозначения переменных			
Имя переменной	Тип переменной		Пояснение
	Fortran	C	
<b>I, J</b>	<b>INTEGER</b>	<b>INT</b>	Номер текущей строки и столбца
<b><math>A_{ij}</math></b>	<b>INTEGER</b>	<b>INT</b>	Элемент матрицы (оценка студента)
<b>S, SR</b>	<b>REAL</b>	<b>FLOAT</b>	Суммарный и средний балл студента
<b>SMIN</b>	<b>REAL</b>	<b>FLOAT</b>	Наименьший средний балл
<b>K</b>	<b>INTEGER</b>	<b>INT</b>	Номер студента с баллом SMIN

### 3. Ручной расчёт результатов

Представим результаты расчёта в следующей таблице:

Сведения о студенте						
Номер строки $i$	Оценки				Суммарный балл $S$	Средний балл $SR$
1	3	4	4	3	14	3,5
2	4	5	4	5	18	4,5
3	3	3	3	4	13	3,25
4	5	5	4	5	19	4,75
5	4	4	3	4	15	3,75

Балл  $SMIN = 3,25$  получил студент с номером  $K = 3$ .

### 4. Алгоритм

При решении задачи используется алгоритм вычисления суммы и наименьшего значения. Алгоритм содержит два цикла вложенной структуры. Обход матрицы осуществляется только по строкам. Параметр внешнего цикла – номер строки  $i$  (студента), а внутреннего – номер столбца  $j$  (предмета).

Схема алгоритма приведена на рис. 2.29. Результаты представлены в макете печати (п.5).

### 5. Макет печати результатов

По C-программе					По Fortran-программе				
Session results					SR				
3	4	4	3	I 3.50	ОЦЕНКИ				SR
4	5	4	5	I 4.50	3	4	4	3	3.50
3	3	3	4	I 3.25	4	5	4	5	4.50
5	5	4	5	I 4.75	3	3	3	4	<b>3.25</b>
4	4	3	4	I 3.75	5	5	4	5	4.75
Minimum: SMIN = 3.25 K = 3					4	4	3	4	3.75
					<b>SMIN = 3.25 K = 3</b>				

Результаты ручного и машинного расчётов совпадают.

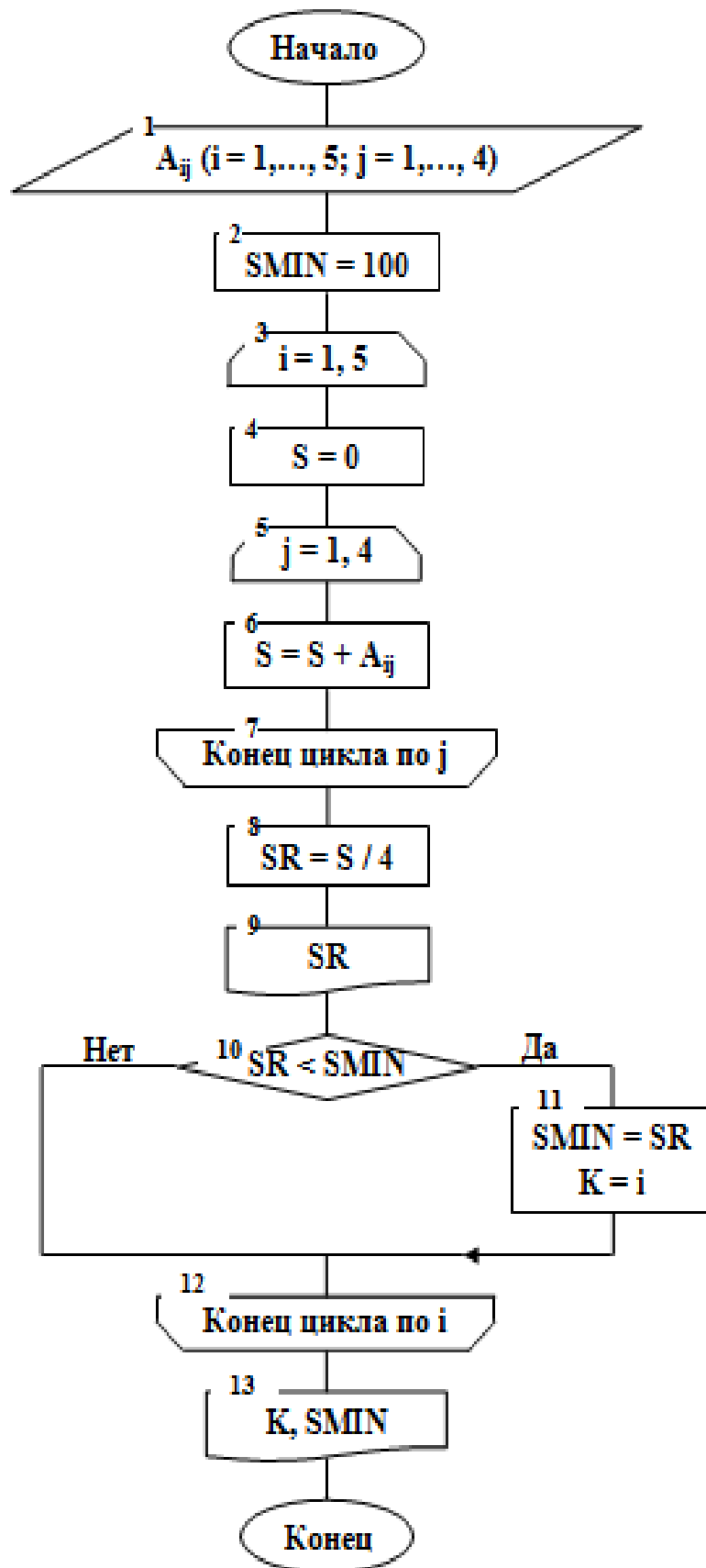


Рис. 2.29 Схема алгоритма к лабораторной работе 16 «Анализ итогов сессии в группе»

## 6. Листинг С-программы «Анализ итогов сессии в группе»

**Внимание.** При вводе программы следует иметь в виду, что С различает регистр букв, т.е., «x» и «X» – это две разные переменные. Нижние границы индексов *i* и *j* (номера строки и столбца) задаются равными нулю.

```
#include<stdio.h> // Заголовочные файлы
#include<conio.h>
void main()
{
    int i,j,k; // Объявление типа данных
    float s, sr, smin = 100;
    // Задание матрицы A(5,4) - оценок (по строкам)
    float a[5][4] = {3,4,4,3,4,5,4,5,3,3,
                    3,4,5,5,4,5,4,4,3,4};
    printf(" Session results      SR \n ");
    // Внешний цикл по строкам (студентам)
    for(i = 0;i<5;i++)
    {
        printf(" \n "); // Пропуск строки
        s = 0;
        // Внутренний цикл по столбцам (предметам)
        for(j = 0;j<4;j++)
        {
            s = s + a[i][j];
            printf(" %2.0f ",a[i][j]); // Вывод Aij
        }
        sr = s/4;
        printf(" I %.2f \n",sr); // Вывод балла SR
        // Вычисление минимального балла SMIN и K
        if(sr<smin) // Если SR < SMIN
        {
            smin = sr;
            k = i;
        }
    }
    // Вывод SMIN и K
    printf(" Minimum: SMIN = %.2f K = %d ",smin,k+1);
    getch();
    return;
}
```

## 7. Листинг Fortran-программы «Анализ итогов сессии»

Примечание. Матрица задаётся в файле с именем FORT.1!

Программный код
<pre>PROGRAM SESSION ! Объявление типа данных INTEGER :: I, J, K, A(1:5, 1:4) REAL :: S, SR, SMIN = 100 ! Ввод матрицы по строкам из файла &lt; FORT.1 &gt; READ(1, *) ((A(I, J), J = 1, 4), I = 1, 5) WRITE(*, *) ' ОЦЕНКИ      SR ' DO I = 1, 5 ! Внешний цикл по I (по строкам)   S = 0.   DO J = 1, 4 ! Внутренний цикл по J (по столбцам)     S = S + A(I, J)   END DO ! Конец цикла по J   SR = S/4   ! Вывод строки матрицы и балла SR   WRITE(*, 3) (A(I, J), J = 1, 4), SR   3 FORMAT(2X, 4I4, F4.2)   ! Вычисление балла SMIN и номера K   IF(SR &lt; SMIN) THEN ! Если SR &lt; SMIN     SMIN = SR; K = I   END IF END DO ! Конец цикла по I ! Вывод значений SMIN и K WRITE(*, "(A, F4.2, A, I1)") ' SMIN = ', SMIN, ' K = ', K END</pre>

Результаты работы представлены в макете печати (п.5, стр.82).

### Лабораторная работа 17 Формирование одномерного массива Тема. Двумерный и одномерный массивы

Индивидуальные задания – табл.3.17 (стр.123)

#### 1. Задание

Составить программу решения следующей задачи.

Дана матрица  $A(3 \times 3)$ . Сформировать одномерный массив  $(B_1, \dots, B_3)$  в соответствии с условием: **каждый элемент массива определяется разностью между элементом главной диагонали и количеством отрицательных элементов строки. Найти произведение положительных элементов массива B.**

### Исходные данные

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} = \begin{pmatrix} 1 & -2 & -3 \\ 2 & 6 & -4 \\ -1 & 3 & 2 \end{pmatrix}$$

### 2. Решение

Переменные, используемые в задаче, приведены в таблице:

Обозначения переменных			
Имя переменной	Тип переменной		Пояснение
	Fortran	C	
<b>I, J</b>	<b>INTEGER</b>	<b>INT</b>	Номер строки и столбца матрицы
<b>A<sub>ij</sub></b>	<b>INTEGER</b>	<b>INT</b>	Элемент матрицы
<b>K</b>	<b>INTEGER</b>	<b>INT</b>	Количество отрицательных элементов строки
<b>V<sub>i</sub></b>	<b>INTEGER</b>	<b>INT</b>	Элемент одномерного массива V с номером i
<b>P</b>	<b>INTEGER</b>	<b>INT</b>	Произведение положительных элементов массива V

### 3. Ручной расчёт результатов

С учётом обозначений элементы массива вычисляются по формуле:  $V_i = A_{ii} - K$ .

Представим результаты расчёта в таблице:

Вычисление элементов массива V <sub>i</sub>			
Номер строки i	Элемент главной диагонали A <sub>ii</sub>	Количество K	Значение V <sub>i</sub>
<b>1</b>	<b>1</b>	<b>2</b>	<b>-1</b>
<b>2</b>	<b>6</b>	<b>1</b>	<b>5</b>
<b>3</b>	<b>2</b>	<b>1</b>	<b>1</b>

Произведение положительных элементов массива **P = 5**.

### 4. Алгоритм

При решении используется алгоритм вычисления количества и произведения. Алгоритм – двойной цикл вложенной структуры. Обход матрицы осуществляется *только по строкам*. Параметр внешнего цикла – номер строки **i**, а внутреннего – номер столбца **j**.



Схема алгоритма представлена на рис. 2.30 (стр.89).

## 5. Листинг С-программы «Формирование массива»

**Внимание.** При вводе программы следует иметь в виду, что С различает регистр букв, т.е., «x» и «X» – это две разные переменные. Нижние границы индексов  $i$  и  $j$  (номера строки и столбца) задаются равными нулю.

```
#include<stdio.h> // Заголовочные файлы
#include<conio.h>
void main()
{
int i,j,k,n = 3; // Объявление типа данных
float b[3],p = 1;
// Задание матрицы А по строкам
float a[3][3] = {1,-2,-3,
                2, 6,-4,
                -1, 3, 2};
printf(" \t ARRAY A(3,3):");
// Внешний цикл по строкам I
for(i = 0;i < n;i++)
{
printf(" \n "); // Пропуск строки
k = 0;
// Внутренний цикл по столбцам J
for(j = 0;j < n;j++)
{
printf(" %2.0f ",a[i][j]); // Вывод Aij
if(a[i][j] < 0) k = k + 1;
}
// Формирование элементов В(I)
b[i] = a[i][i] - k;
if(b[i] > 0) p = p*b[i]; // Вычисление P
}
printf(" \n \t ARRAY B(3): \n ");
for(i = 0;i < n;i++) // Вывод массива В
printf(" %2.0f ",b[i]);
printf(" \n P = %.0f ",p); // Вывод P
getch();
return;
}
```

## 6. Макет печати результатов

### По C-программе

```
Array A(3,3)
 1  -2  -3
 2   6  -4
-1   3   2
Array B(3)
-1   5   1
P = 5
```

### По Fortran-программе

```
Матрица A(3,3)
 1  -2  -3
 2   6  -4
-1   3   2
Массив B(3)
-1   5   1
P = 5
```

Как видно, результаты ручного и машинного расчётов совпадают.

## 7. Листинг Fortran-программы

Программный код
<pre>! Формирование одномерного массива B PROGRAM МАССИВ ! Объявление типа данных - целого типа INTEGER :: A(1:3,1:3),B(3) INTEGER :: I,J,K,N = 3,P = 1 ! Задание матрицы по столбцам DATA A/1,2,-1,-2,6,3,-3,-4,2/ ! Вывод матрицы по строкам WRITE(*,*) '          МАТРИЦА A(3,3) ' WRITE(*,2)((A(I,J),J = 1,3),I = 1,3) 2 FORMAT(2X,3I4) DO I = 1,N ! Внешний цикл по I (по строкам)   K = 0   DO J = 1,N ! Внутренний цикл по J (по столбцам)     IF(A(I,J) &lt; 0) K = K + 1   END DO ! Конец цикла по J ! Вычисление элементов B(I) B(I) = A(I,I) - K IF(B(I) &gt; 0) P = P * B(I) END DO ! Конец цикла по I ! Вывод массива B и произведения P WRITE(*,*) '          МАССИВ B(3) ' WRITE(*,"( 3I4/2X,A,I3)") B,' P = ',P END</pre>

Результаты работы представлены в макете печати (п.6).

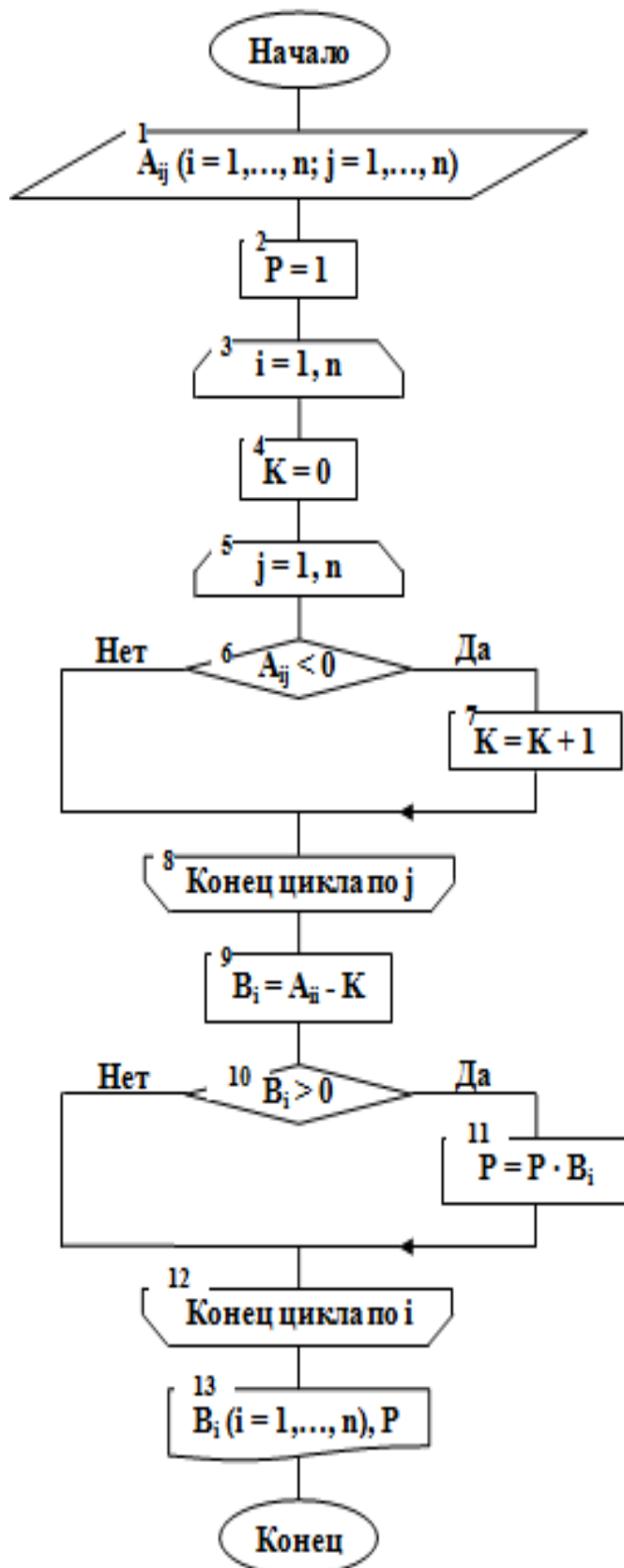


Рис. 2.30 Схема алгоритма к лабораторной работе 17 «Формирование одномерного массива»

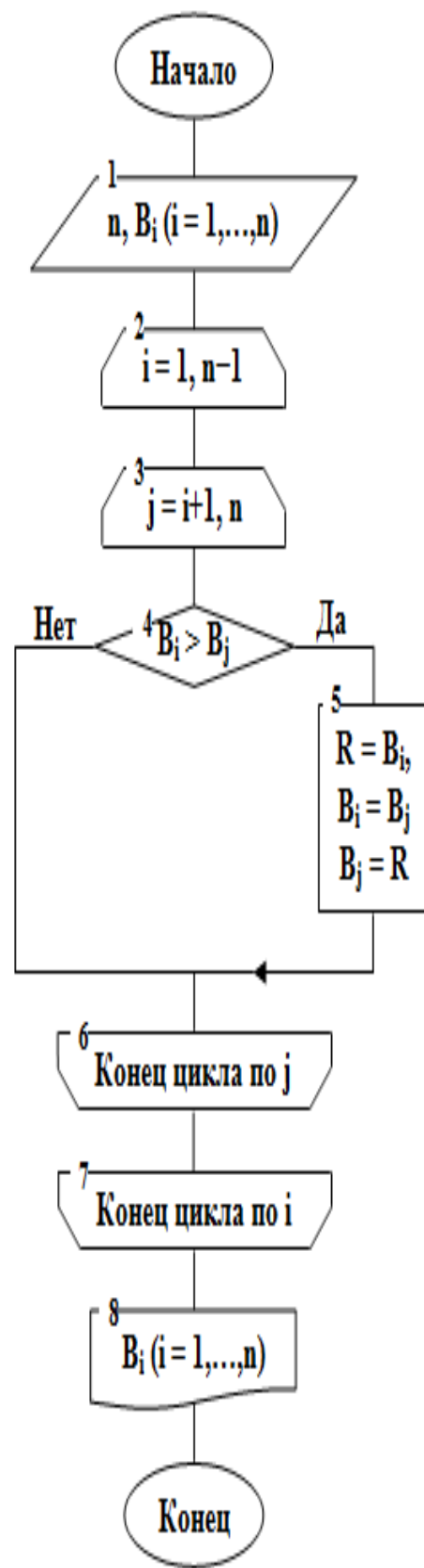


Рис. 2.31 Схема алгоритма к лабораторной работе 17 «Упорядочение массива»

## 8. Листинг Basic-программы «Формирование массива»

```
CLS
DIM A(3, 3), B(3)
' Задание матрицы A по строкам
DATA 1,-2,-3, 2, 6,-4,-1, 3, 2
PRINT TAB(10); " Матрица A(3,3) "
P = 1 : N = 3
FOR I = 1 TO N ' Внешний цикл по I (по строкам)
  K = 0
  FOR J = 1 TO N ' Внутренний цикл по J (по столбцам)
    READ A(I, J)
    PRINT USING " ### "; A(I, J);
    IF A(I, J) < 0 THEN K = K + 1
  NEXT J ' Конец цикла по J
  B(I) = A(I, I) - K
  IF B(I) > 0 THEN P = P * B(I)
  PRINT ' Пропуск строки
NEXT I ' Конец цикла по I
PRINT TAB(10); " Массив B(3) "
' Цикл вывода массива B
FOR I = 1 TO N
  PRINT USING " ## "; B(I);
NEXT I
PRINT ' Пропуск строки
PRINT USING "   P = ## "; P ' Вывод P
END
```

Результаты представлены в макете печати (п.6).

## Контрольное задание для проверки знаний

### Тема. Двумерные массивы (матрицы)

Дана матрица  $A$  ( $n \times n$ ),  $n = 6$ . В матрице можно выделить три совокупности элементов  $A_{ij}$ : главной диагонали (при  $i = j$ ), верхнего и нижнего треугольника (при  $i < j$  и  $i > j$ ). Обход верхнего треугольника удобнее осуществлять по строкам, а нижнего – по столбцам. В этом случае для верхнего треугольника параметр внешнего цикла, номер строки  $i$ , изменяется в интервале  $[1; n - 1]$ , а параметр внутреннего цикла, номер столбца  $j$ , – в интервале  $[i+1;n]$ . Для нижнего треугольника параметр внешнего цикла, номер столбца  $j$ , изменяется в интервале  $[1; n - 1]$ , а параметр внутреннего цикла, номер строки  $i$ , – в интервале  $[j + 1; n]$ . Матрица задаётся самостоятельно в соответствии с условием задачи.

## Задания для лабораторной работы

Составить схему и программу для следующей задачи.

*Определить:*

№	Задача
1	Наименьший из положительных элементов чётных строк и определить его принадлежность верхнему, нижнему треугольнику, либо главной диагонали. Если это элемент верхнего треугольника, то найти сумму его отрицательных элементов.
2	Наибольший из отрицательных элементов нечётных столбцов и определить его принадлежность верхнему, нижнему треугольнику, либо главной диагонали. Если это элемент нижнего треугольника, то найти произведение его положительных элементов.
3	Номер столбца с наименьшим из положительных элементов верхнего треугольника. Если это чётный столбец, то найти сумму его элементов, если нечётный – их произведение.
4	Номер строки с наибольшим из отрицательных элементов нижнего треугольника. Если строка – чётная, то найти количество её положительных элементов, если нечётная – количество отрицательных элементов.
5	Номер строки и столбца наименьшего из элементов верхнего треугольника. Найти сумму положительных элементов строки и произведение отрицательных элементов столбца.
6	Номер строки и столбца наибольшего из элементов нижнего треугольника. Найти наименьший из положительных элементов строки и количество отрицательных элементов столбца.
7	Наименьший элемент матрицы и определить его принадлежность верхнему, нижнему треугольнику, либо главной диагонали. Если это элемент верхнего треугольника, то найти среднее арифметическое его отрицательных элементов.
8	Наибольший элемент матрицы и определить его принадлежность верхнему, нижнему треугольнику, либо главной диагонали. Если это элемент нижнего треугольника, то найти произведение его положительных элементов.
9	Сумму положительных элементов главной диагонали и наименьший из элементов верхнего треугольника, превышающих это значение.
10	Среднее арифметическое элементов нижнего треугольника, не превышающих значение наибольшего элемента главной диагонали.

### 3. Задания на выполнение лабораторных работ

Для таблиц 1-17 приведены примеры выполнения лабораторных работ (раздел 2).

#### Таблица 3.1 Вычисление значения выражения

Математические функции – табл. 5.1 (стр.136)

Пример выполнения лабораторной работы – стр.5-8

#### Задания для лабораторной работы 1

Составить программу вычисления величины Z .

№	Формула для вычисления Z	Исходные данные		Значение Z
		a	b	
1	$Z = \frac{a\sqrt{\ln 2a + \ln b}}{b + \sqrt{\ln ab + \ln 3}} - \frac{a}{b}\sqrt{\ln a + \ln b}$	2	1,5	- 0,5
2	$Z = \sqrt{\sin^2 8^\circ + \sin^2 a} + \frac{\sin^2 \pi + \sin^2 b}{a(\sin^2 ab + \sin^2 2b)}$	- 2	1,25	0,2913
3	$Z = \frac{\ln^2 a + \ln^2 b}{a(\ln^2 3 + \ln^2 3a)} - (\ln^2 \frac{b}{a} + \ln^2 ab)^2$	1,5	2	- 1,5394
4	$Z = \sqrt{ \operatorname{tg} a - \operatorname{tg} b } - \frac{b + (\operatorname{tg} 3^\circ - \operatorname{tg} ab)}{(\operatorname{tg} 2a - \operatorname{tg} 0,2\pi)}$	0,8	3	1,1962
5	$Z = \frac{(\cos 2a + \cos b)^2}{a(\cos 5^\circ + \cos 0,3\pi)} + \sqrt{ \cos a + \cos 2b }$	1,2	1	0,2524
6	$Z = (\cos^2 \frac{\pi}{3} + \ln a)^2 - \frac{\cos^2 3^\circ + \ln  b }{a(\cos^2 b + \ln 2)}$	3	- 2	1,1684
7	$Z = \frac{(\sin 2a + \cos b)^2}{b(\sin 3^\circ + \cos a)} + \sqrt{ \sin b + \cos \frac{a}{b} }$	- 1	1,7	2,4201

8	$Z = \sqrt{ \ln b + \sin 2^\circ } - \frac{(\ln 2 + \sin 0,3\pi)^2}{a(\ln a + \sin b)}$	2	3,1	-0,4556
9	$Z = \frac{(\cos b - \operatorname{tg} a)^2}{\cos 6^\circ - \operatorname{tg} b} + \sqrt{\left \cos a - \operatorname{tg} \frac{a}{b}\right }$	2,4	-2	1,1446
10	$Z = \sqrt{ \cos a + \ln 3 } - \frac{(\cos \pi + \ln b)^2}{\cos 3^\circ + \ln a }$	-3	1,5	0,16103
11	$Z = \frac{(\ln a  - \operatorname{tg} b)^2}{ \ln b - \operatorname{tg} 3^\circ } + \sqrt{\left \ln 2b - \operatorname{tg} \frac{\pi}{3}\right }$	-2	1,85	31,7154
12	$Z = \sqrt{ \cos^2 a - \sin^2 b } - \frac{(\cos^2 b - \sin^2 3^\circ)^2}{a(\cos^2 \pi - \sin^2 a)}$	1,2	2,35	-0,9182
13	$Z = \frac{\sin^2 b + \ln a}{b(\sin^2 3^\circ + \ln 2)} + (\sin^2 a + \ln b )^2$	2	-3	3,3657
14	$Z = \sqrt{ e^{-b} + \operatorname{tg} 3^\circ } - \frac{e^{2a} + \operatorname{tg} b}{b - (e^b + \operatorname{tg} a)^2}$	1,4	0,5	1,1209
15	$Z = \frac{\ln b - e^{-a}}{\ln a  - e^3} + (\ln 2b - e^b)^2$	-2	1,5	11,8053
16	$Z = (\sin \frac{\pi}{3} + e^{2b})^2 - \frac{\sin 5^\circ + e^a}{\sin a  + e^{-a}}$	-2	1	68,1193
17	$Z = \frac{e^3 - \cos 4^\circ}{\sqrt{ e^{-a} - \cos 2b }} + (e^b - \cos \frac{\pi}{3})^2$	1	2,7	243,73
18	$Z = (\ln a  + \cos b)^2 - \frac{\ln 3 + \cos a}{b(\ln \sqrt{b} + \cos 3^\circ)}$	-3	2	0,4254
19	$Z = \frac{\sin a + \ln^2 b}{a(\sin 3^\circ + \ln^2 b)} + (\sin \frac{\pi}{3} + \ln^2 3)^2$	1,5	2	6,1465
20	$Z = (\sqrt{\frac{a}{b}} + \sin 2a)^2 - \frac{\sqrt{a} + \sin 5^\circ}{b(\sqrt{b} + \sin a)}$	1	2,5	2,1975

### Таблица 3.2 Вычисление значения величины, заданной в форме ветвления

Пример выполнения лабораторной работы – стр.8-11

Математические функции – табл. 5.1 (стр.136)

Составить схему и программу вычисления величины Z.

**Для выполнения лабораторной работы необходимо:**

1. При выполнении программы по запросу с экрана ввести с клавиатуры поочерёдно каждое из значений А, В, К. Результат Z сравнить с приведённым в таблице.

2. Действие п.1 повторить для каждой из ветвей алгоритма.

3. При несовпадении результата проанализировать программу, исправить ошибки и повторить действие п.1.

#### Задания для лабораторной работы 2

№	Формула для вычисления Z	Исходные данные			Значение Z
		a	b	k	
1	$Z = \begin{cases} k \cdot \max(a,b) & \text{при } a \geq 1 \text{ и } b > 0 \\ b^2 - a \cdot k & \text{при } a \geq 1 \text{ и } b \leq 0 \\ a + b \cdot k & \text{при } a < 1 \text{ и } b = 1 \\ b + a \cdot k^2 & \text{при } a < 1 \text{ и } b \neq 1 \end{cases}$	3	1	2	6
		3	-1	1	-2
		-1	1	2	1
		-3	2	2	-10
2	$Z = \begin{cases} k \cdot \min(a,b) & \text{при } b > 0 \text{ и } a = 1 \\ a \cdot b + 2 \cdot k & \text{при } b > 0 \text{ и } a \neq 1 \\ a \cdot k^2 + b & \text{при } b \leq 0 \text{ и } a > b \\ a + b \cdot k & \text{при } b \leq 0 \text{ и } a \leq b \end{cases}$	1	5	4	4
		3	2	2	10
		1	-3	2	1
		-2	-1	3	-5
3	$Z = \begin{cases} b \cdot \max(a,k) & \text{при } a < 0 \text{ и } b > 1 \\ b - a \cdot k & \text{при } a < 0 \text{ и } b \leq 1 \\ a + b \cdot k^2 & \text{при } a \geq 0 \text{ и } a > b \\ k \cdot b - a & \text{при } a \geq 0 \text{ и } a \leq b \end{cases}$	-1	2	3	6
		-4	-1	1	3
		2	1	2	6
		1	2	2	3
4	$Z = \begin{cases} a \cdot \min(k,b) & \text{при } a > b \text{ и } b = 1 \\ k - a \cdot b & \text{при } a > b \text{ и } b \neq 1 \\ a \cdot b^2 + k & \text{при } a \leq b \text{ и } a > 0 \\ a \cdot b + k^2 & \text{при } a \leq b \text{ и } a \leq 0 \end{cases}$	3	1	3	3
		3	2	4	-2
		1	2	2	6
		-5	-4	2	24



№	Формула для вычисления Z	Исходные данные			Значение Z
		a	b	k	
5	$Z = \begin{cases} a + \max(b, k) & \text{при } a = b \text{ и } a > 0 \\ a + k \cdot b^2 & \text{при } a = b \text{ и } a \leq 0 \\ a \cdot b + 2k & \text{при } a \neq b \text{ и } b > 1 \\ a^2 + b \cdot k & \text{при } a \neq b \text{ и } b \leq 1 \end{cases}$	2 -1 2 2	2 -1 3 -1	3 2 2 2	5 1 10 2
6	$Z = \begin{cases} b - \min(k, a) & \text{при } a = 1 \text{ и } b \geq 0 \\ b + a \cdot k^3 & \text{при } a = 1 \text{ и } b < 0 \\ a + b \cdot k & \text{при } a \neq 1 \text{ и } a > b \\ a \cdot b - k & \text{при } a \neq 1 \text{ и } a \leq b \end{cases}$	1 1 3 2	4 -1 2 3	3 2 1 2	3 7 5 4
7	$Z = \begin{cases} k + \max(a, b) & \text{при } b \neq 1 \text{ и } a < 0 \\ a^2 \cdot b - 2k & \text{при } b \neq 1 \text{ и } a \geq 0 \\ a \cdot b^2 + k & \text{при } b = 1 \text{ и } a \leq b \\ a \cdot b + k^2 & \text{при } b = 1 \text{ и } a > b \end{cases}$	-2 3 -1 2	2 3 1 1	3 2 2 2	5 5 1 6
8	$Z = \begin{cases} b \cdot \min(2a, k) & \text{при } b > 1 \text{ и } a > 0 \\ a \cdot b + k & \text{при } b > 1 \text{ и } a \leq 0 \\ a \cdot k^2 + b & \text{при } b \leq 1 \text{ и } a \neq 1 \\ b - k \cdot a^2 & \text{при } b \leq 1 \text{ и } a = 1 \end{cases}$	3 -1 -2 1	2 2 -1 -5	4 1 2 2	8 -1 -9 -7
9	$Z = \begin{cases} a \cdot \max(k, 2b) & \text{при } a \leq 0 \text{ и } b = 1 \\ a + b \cdot k & \text{при } a \leq 0 \text{ и } b \neq 1 \\ a \cdot b + 2k^2 & \text{при } a > 0 \text{ и } a > b \\ a \cdot k - b & \text{при } a > 0 \text{ и } a \leq b \end{cases}$	-2 -3 3 1	1 2 1 3	3 1 2 2	-6 -1 11 -1
10	$Z = \begin{cases} k \cdot \min(b, 2a) & \text{при } b \leq 0 \text{ и } a \geq 0 \\ 2k - a \cdot b & \text{при } b \leq 0 \text{ и } a < 0 \\ a + b \cdot k^2 & \text{при } b > 0 \text{ и } a > 1 \\ k - a \cdot b^2 & \text{при } b > 0 \text{ и } a \leq 1 \end{cases}$	1 -1 2 -1	-4 -2 1 2	2 2 2 3	-8 2 6 7
11	$Z = \begin{cases} k \cdot \max(a, 2b) & \text{при } b > 1 \text{ и } a > b \\ 2a + b \cdot k & \text{при } b > 1 \text{ и } a \leq b \\ a \cdot b + k^2 & \text{при } b \leq 1 \text{ и } a < 0 \\ a \cdot b + 2k & \text{при } b \leq 1 \text{ и } a \geq 0 \end{cases}$	3 1 -9 2	2 2 -1 -5	3 1 2 2	12 4 13 -6
12	$Z = \begin{cases} a \cdot \min(b, k^2) & \text{при } a < b \text{ и } a \leq 0 \\ a \cdot b + 2k & \text{при } a < b \text{ и } a > 0 \\ a \cdot k + b^3 & \text{при } a \geq b \text{ и } b = 1 \\ k^2 + a \cdot b & \text{при } a \geq b \text{ и } b \neq 1 \end{cases}$	-1 1 2 2	9 3 1 -1	2 2 3 1	-4 7 7 -1

№	Формула для вычисления Z	Исходные данные			Значение Z
		a	b	k	
13	$Z = \begin{cases} b \cdot \max(a, k^2) & \text{при } a > 0 \text{ и } a \leq b \\ a^2 - b \cdot k & \text{при } a > 0 \text{ и } a > b \\ b - a \cdot k^2 & \text{при } a \leq 0 \text{ и } b < 1 \\ a \cdot b + 3k & \text{при } a \leq 0 \text{ и } b \geq 1 \end{cases}$	2	3	2	12
		4	1	3	13
		-2	-1	2	7
		-4	1	2	2
14	$Z = \begin{cases} k \cdot \min(a, b^2) & \text{при } a < 1 \text{ и } b \leq 0 \\ a \cdot b - 3k & \text{при } a < 1 \text{ и } b > 0 \\ b + a \cdot k^2 & \text{при } a \geq 1 \text{ и } a < b \\ b^2 - a \cdot k & \text{при } a \geq 1 \text{ и } a \geq b \end{cases}$	-1	-4	3	-3
		-1	2	1	-5
		1	2	1	3
		1	-1	2	-1
15	$Z = \begin{cases} k \cdot \max(b, a^2) & \text{при } b = 1 \text{ и } a \geq 0 \\ a \cdot b + 2k & \text{при } b = 1 \text{ и } a < 0 \\ b \cdot k^2 - a & \text{при } b \neq 1 \text{ и } a = b \\ k + a \cdot b & \text{при } b \neq 1 \text{ и } a \neq b \end{cases}$	3	1	2	18
		-2	1	2	2
		2	2	2	6
		3	2	4	10
16	$Z = \begin{cases} k \cdot \min(b, a^2) & \text{при } a \geq 0 \text{ и } b < 1 \\ a \cdot b^2 + 2k & \text{при } a \geq 0 \text{ и } b \geq 1 \\ a \cdot k + b & \text{при } a < 0 \text{ и } b = 1 \\ a + b \cdot k^2 & \text{при } a < 0 \text{ и } b \neq 1 \end{cases}$	5	-4	2	-8
		1	2	2	8
		-1	1	2	-1
		-3	2	2	5
17	$Z = \begin{cases} k \cdot \max(a, b^3) & \text{при } a = b \text{ и } a > 1 \\ a \cdot k^2 + b & \text{при } a = b \text{ и } a \leq 1 \\ a \cdot k - b^2 & \text{при } a \neq b \text{ и } a > 0 \\ b \cdot k - a^3 & \text{при } a \neq b \text{ и } a \leq 0 \end{cases}$	2	2	1	8
		1	1	2	5
		2	3	2	-5
		-1	4	3	13
18	$Z = \begin{cases} b + \min(a, k) & \text{при } b > 0 \text{ и } a = b \\ b + a \cdot k^2 & \text{при } b > 0 \text{ и } a \neq b \\ a^3 - k \cdot b & \text{при } b \leq 0 \text{ и } a \geq 1 \\ k + a \cdot b^2 & \text{при } b \leq 0 \text{ и } a < 1 \end{cases}$	2	2	4	4
		1	2	2	6
		1	-1	2	3
		-1	-2	5	1
19	$Z = \begin{cases} a \cdot \max(b, k) & \text{при } b < 1 \text{ и } a = 1 \\ b^2 - a \cdot k & \text{при } b < 1 \text{ и } a \neq 1 \\ b \cdot k - a^3 & \text{при } b \geq 1 \text{ и } a < b \\ a \cdot b - k^2 & \text{при } b \geq 1 \text{ и } a \geq b \end{cases}$	1	-4	3	3
		-1	-2	2	6
		-2	1	2	10
		2	1	2	-2
20	$Z = \begin{cases} k \cdot \min(a^3, b) & \text{при } a > 1 \text{ и } b \leq 0 \\ 2 \cdot k - a \cdot b & \text{при } a > 1 \text{ и } b > 0 \\ a \cdot k + b^2 & \text{при } a \leq 1 \text{ и } a \neq b \\ b \cdot k - a^3 & \text{при } a \leq 1 \text{ и } a = b \end{cases}$	5	-4	1	-4
		4	2	3	-2
		1	3	2	11
		1	1	2	1

### Таблица 3.3 Вычисление значения величины, заданной в форме ветвления

Пример выполнения лабораторной работы – стр.11-14

Математические функции – табл. 5.1 (стр.136)

Составить схему и программу вычисления величины Z.

**Для выполнения лабораторной работы необходимо:**

1. При выполнении программы по запросу с экрана ввести с клавиатуры поочерёдно каждое из значений А, В, С. Результат Z сравнить с приведённым в таблице.
2. Действие п.1 повторить для каждой из ветвей алгоритма.
3. При несовпадении результата проанализировать программу, исправить ошибки и повторить действие п.1.

#### Задания для лабораторной работы 3

№	Формула для вычисления Z	Исходные данные			Значение Z
		a	b	c	
1	$Z = \begin{cases} \max\{b, a \cdot \min(a^2, 2c)\} & \text{при } c \leq a \\ b^2 + \min(c, 2a) & \text{при } c > a \text{ и } b = 1 \\ \min(2a, \sqrt{ b }) - c & \text{при } c > a \text{ и } b \neq 1 \end{cases}$	2	-1	1	4
		1	1	3	3
		-1	4	3	1
2	$Z = \begin{cases} \max(a, b^2, \sqrt{ c }) & \text{при } a < 1 \\ c + \min(b, a^2) & \text{при } a \geq 1 \text{ и } b > c \\ \min(2b, c^2) - a & \text{при } a \geq 1 \text{ и } b \leq c \end{cases}$	-2	3	9	9
		2	5	3	7
		2	3	4	4
3	$Z = \begin{cases} \min\{a, b \cdot \max(b^2, c)\} & \text{при } a \geq b \\ b + \max(a, 2c) & \text{при } a < b \text{ и } c > 0 \\ \max(b, \sqrt{ c }) - a & \text{при } a < b \text{ и } c \leq 0 \end{cases}$	3	2	5	3
		2	3	4	11
		2	3	-4	1
4	$Z = \begin{cases} \min(b^2, \sqrt{ a }, 2c) & \text{при } b > c \\ a - \max(2b, c) & \text{при } b \leq c \text{ и } a > 1 \\ \max(a^2, c) + b & \text{при } b \leq c \text{ и } a \leq 1 \end{cases}$	4	3	2	2
		6	2	3	2
		-2	2	3	6

№	Формула для вычисления Z	Исходные данные			Значение Z
		a	b	c	
5	$Z = \begin{cases} b + \min(2a, \sqrt{ c }) & \text{при } a = b \\ \min(b^2, c) - a & \text{при } a \neq b \text{ и } c \leq 0 \\ \max(b, a^2, 2c) & \text{при } a \neq b \text{ и } c > 0 \end{cases}$	3	3	16	7
		3	2	-1	-4
		3	2	3	9
6	$Z = \begin{cases} \max(2b, c^3) + a^2 & \text{при } a > c \\ 2a - \min(c^2, b) & \text{при } a \leq c \text{ и } b = 1 \\ \min\{a, b \cdot \min(b^3, c)\} & \text{при } a \leq c \text{ и } b \neq 1 \end{cases}$	3	1	2	13
		2	1	4	3
		2	3	9	2
7	$Z = \begin{cases} b + \max(a^2, c) & \text{при } a \geq b \\ \min(\sqrt{ b }, 2a, c^2) & \text{при } a < b \text{ и } c > a \\ c - \max(2b, a^3) & \text{при } a < b \text{ и } c \leq a \end{cases}$	2	1	5	6
		-1	4	3	-2
		1	4	-2	-10
8	$Z = \begin{cases} \min\{b, a \cdot \max(c, 2a)\} & \text{при } b \geq 0 \\ c^2 - \max(a, b) & \text{при } b < 0 \text{ и } a \neq c \\ \max(2c, b^2) + a^2 & \text{при } b < 0 \text{ и } a = c \end{cases}$	3	2	3	2
		1	-1	2	3
		3	-1	3	15
9	$Z = \begin{cases} \min\{c, b \cdot \min(2a, b^3)\} & \text{при } b > c \\ a - \min(\sqrt{ b }, c^2) & \text{при } b \leq c \text{ и } a > 1 \\ \max(a^3, 3b) + c & \text{при } b \leq c \text{ и } a \leq 1 \end{cases}$	2	3	1	1
		3	4	5	1
		-1	3	6	12
10	$Z = \begin{cases} \min(b^2, 2a) - ac & \text{при } a \leq 1 \\ \max(2b, a^2, c^2) & \text{при } a > 1 \text{ и } b \neq c \\ b^2 + \min(a, 3c) & \text{при } a > 1 \text{ и } b = c \end{cases}$	-2	2	3	2
		2	3	4	16
		2	3	3	11
11	$Z = \begin{cases} \min\{a^2, c \cdot \max(c^3, \sqrt{ b })\} & \text{при } b \geq 0 \\ \max(a, b^3) + 2c & \text{при } b < 0 \text{ и } a \geq c \\ a - \max(b, c^2) & \text{при } b < 0 \text{ и } a < c \end{cases}$	3	4	2	9
		4	-1	3	10
		1	-1	2	-3

№	Формула для вычисления Z	Исходные данные			Значение Z
		a	b	c	
12	$Z = \begin{cases} \min(a^3, \sqrt{ c }, b^2) & \text{при } a > 0 \\ c + \max(b, 2a) & \text{при } a \leq 0 \text{ и } b = c \\ \max(2b, c^2) + a & \text{при } a \leq 0 \text{ и } b \neq c \end{cases}$	2	3	4	2
		-1	3	3	6
		-1	2	3	8
13	$Z = \begin{cases} \max\{b^2, c \cdot \min(a, 2c)\} & \text{при } a > c \\ a + \min(b, c^2) & \text{при } a \leq c \text{ и } a > b \\ \min(2a, \sqrt{ b }) - c^2 & \text{при } a \leq c \text{ и } a \leq b \end{cases}$	3	4	2	16
		2	-1	3	1
		2	4	3	-7
14	$Z = \begin{cases} \max(a, b, c^2) & \text{при } c \geq 1 \\ \min(2c, \sqrt{ a }) - b & \text{при } c < 1 \text{ и } b = a \\ a + \min(c^2, 2b) & \text{при } c < 1 \text{ и } b \neq a \end{cases}$	4	1	3	9
		4	4	-1	-6
		4	3	-1	5
15	$Z = \begin{cases} \min\{a, b \cdot \max(c, b)\} & \text{при } a > c \\ \max(c, b^3) + a & \text{при } a \leq c \text{ и } b > 1 \\ \sqrt{ c } - \max(2a, b^2) + b^2 & \text{при } a \leq c \text{ и } b \leq 1 \end{cases}$	5	2	4	5
		-1	2	1	7
		1	-1	9	1
16	$Z = \begin{cases} \max(c^2, \sqrt{ a }) - b & \text{при } a > b \\  c  + \max(2a, b^3) & \text{при } a \leq b \text{ и } c = 1 \\ \min(b^2, 2c, a^3) & \text{при } a \leq b \text{ и } c \neq 1 \end{cases}$	4	2	3	7
		2	3	1	28
		2	3	4	4
17	$Z = \begin{cases} 2c + \min(a^2, b) & \text{при } a < b \\ \min(2b, c) - 2a & \text{при } a \geq b \text{ и } c \geq 0 \\ \max(2a, b^2, \sqrt{ c }) & \text{при } a \geq b \text{ и } c < 0 \end{cases}$	-1	3	1	3
		3	1	4	-4
		3	2	-4	6
18	$Z = \begin{cases} \max\{b, c \cdot \min(a, b)\} & \text{при } b = c \\ b - \min(a^2, c) & \text{при } b \neq c \text{ и } a \geq 1 \\ \min(a, 2b) + c^2 & \text{при } b \neq c \text{ и } a < 1 \end{cases}$	3	2	2	4
		4	2	3	-1
		-1	4	3	8

№	Формула для вычисления Z	Исходные данные			Значение Z
		a	b	c	
19	$\min(a, 2c, b^2)$ при $a = c$	2	3	2	2
	$b + \max(2c, a^3) - b$ при $a \neq c$ и $b \geq 0$	2	4	3	12
	$\max(b^2, 2a) - c$ при $a \neq c$ и $b < 0$	2	-1	3	1
20	$\min\{\sqrt{ b }, c \cdot \max(c^2, a)\}$ при $b \geq 1$	2	9	1	2
	$\max(a, b^2) + c$ при $b < 1$ и $a \geq c$	3	-2	2	6
	$a - \max(2c, b)$ при $b < 1$ и $a < c$	2	-2	3	-4

**Таблица 3.4 Исследование функции одной переменной**

Пример выполнения лабораторной работы – стр.17-22

Математические функции – табл. 5.1 (стр.136)

Порядок выполнения лабораторной работы – раздел 1 (стр.4)

Составить схему и программу решения для следующей задачи.

Вычислить таблицу значений аргумента  $X$  и функции  $Y = F(X)$  при условии, что аргумент  $X$  изменяется в интервале  $X_{\min} \leq X \leq X_{\max}$  с шагом  $\Delta X$ .

При решении задач используются алгоритмы вычисления суммы, произведения, количества, наибольшего (наименьшего) значения.

#### Задания для лабораторной работы 4

*Протабулировать функцию и найти:*

№	Задача	Исходные данные		
		$X_{\min}$	$X_{\max}$	$\Delta X$
1	Сумму положительных значений функции $y = \cos^3(x) - e^{-x}$	0	1	0,1
2	Произведение отрицательных значений функции $y = \ln^2(x \cdot \sqrt{x}) - e^{-x}$	0,2	2	0,2
3	Количество положительных значений функции $y = \ln^3(1 + \sqrt{ x }) - e^{-x}$	0,4	2,2	0,2
4	Сумму и произведение значений функции. Какая из этих величин больше? $y = \sin^2(2 \cdot x) - e^x$	-1	0,6	0,2

№	Задача	Исходные данные		
		$X_{\min}$	$X_{\max}$	$\Delta X$
5	Значение аргумента $x$ , при котором достигается наименьшее значение функции $y = \sqrt{ x } - \sin^3(2 \cdot x)$	1	2	0,1
6	Наименьшее из положительных значений функции $y = \sin^3 x - e^{-x}$	0,3	1,2	0,1
7	Произведение и количество положительных значений функции $y = tg^3(\sqrt{ x }) - e^{-2x}$	0,1	1	0,1
8	Сумму положительных и произведение отрицательных значений функции $y = tg^3(3 \cdot x) - \sqrt{ x }$	0,1	1	0,1
9	Какому значению аргумента $x$ соответствует наибольшее значение функции $y = \cos^2 x + \sqrt{ x \cdot \ln(x) }$	0,1	1	0,1
10	Произведение положительных и произведение отрицательных значений функции. $y = tg^3(2 \cdot x) - \ln x $	0,2	2	0,2
11	Произведение значений функции. Вывести сообщение о том, положительна эта величина или отрицательна. $y = \sqrt{ x } - tg^2(3 \cdot x)$	1,5	2,3	0,1
12	Категория каких значений функции преобладает на интервале, положительных или отрицательных? $y = \sqrt{ x } + x \cdot tg^3 x$	1	2	0,1
13	Сумму и количество положительных значений функции $y = \sin^2(2 \cdot x) - e^{-x}$	0	1,6	0,2
14	Наименьшее значение функции. Определить знак этой величины. $y = \sin^2(3 \cdot x) + \sqrt{ 1 - x^2 }$	1	2,6	0,2

№	Задача	Исходные данные		
		$X_{\min}$	$X_{\max}$	$\Delta X$
15	Сумму положительных и количество отрицательных значений функции $y = \cos^2(x) - e^{-x}$	-1	1	0,2
16	Сумму и произведение положительных значений функции $y = \sin^2(3 \cdot x) - e^{-x}$	0	1,6	0,2
17	Среднее арифметическое отрицательных значений функции $y = \operatorname{tg}^3(2 \cdot x) - \sqrt{ x }$	0,1	1	0,1
18	Произведение отрицательных и количество положительных значений функции $y = \operatorname{tg}^2(3 \cdot x) - e^{-x}$	0,2	1,8	0,2
19	Сумму положительных и сумму отрицательных значений функции $y = \cos^3(2 \cdot x) - \sqrt{ x }$	0	0,9	0,1
20	Среднее арифметическое значение функции. Вывести сообщение о том, положительна эта величина или отрицательна. $y = \operatorname{tg}^2(3 \cdot x) - e^{-2 \cdot x}$	0	1	0,1

### Таблица 3.5 Исследование функции одной переменной, зависящей от параметра

Пример выполнения лабораторной работы – стр.22-27

Математические функции – табл. 5.1 (стр.136)

Порядок выполнения лабораторной работы – раздел 1 (стр.4)

Составить схему и программу решения для следующей задачи.

Вычислить таблицу значений функции  $y = f(X, \alpha)$  и аргумента  $X$  на интервале  $[X_{\min}, X_{\max}]$  с шагом  $\Delta X$  при условии, что параметр  $\alpha$  принимает ряд фиксированных значений  $\alpha_1$  и  $\alpha_2$ . Таким образом, вычисление функции можно представить в виде:

$$y = \begin{cases} f(X, \alpha_1), & \text{если } X_{\min} \leq X \leq X_{\text{cp}} \\ f(X, \alpha_2), & \text{если } X_{\text{cp}} < X \leq X_{\max}, \end{cases}$$

где  $X_{\text{cp}}$  – значение аргумента  $X$  из интервала  $X_{\min} < X_{\text{cp}} < X_{\max}$ .



При решении используются алгоритмы вычисления суммы, произведения, количества, наибольшего (наименьшего) значения.

### Задания для лабораторной работы 5

*Протабулировать функцию и найти:*

№	Задача	Исходные данные					
		$X_{\min}$	$X_{\text{cp}}$	$X_{\max}$	$\Delta X$	$\alpha_1$	$\alpha_2$
1	Наименьшее значение функции и соответствующее ему значение аргумента $X$ $y = \sqrt{ \alpha \cdot X } - \text{tg}^2 \alpha X + \frac{\alpha X}{1 + \alpha}$	0	0,8	1,5	0,1	0,5	1
2	Наибольшее и наименьшее значение функции $y = \alpha X - \sqrt{\alpha + X^2} - \frac{\cos^2 X}{\alpha X^3}$	1	1,5	2	0,1	1	0,5
3	Наименьшее из положительных значений функции $y = \sqrt{\alpha + X^2} - \text{tg}^2(\pi + \alpha X^2)$	0	1	2	0,2	1	2
4	Наибольшее из отрицательных значений функции $y = \alpha X + \text{tg} \alpha X - \alpha \sqrt{\alpha^2 + X^2}$	2	2,5	3	0,1	0,7	1,2
5	Наименьшее значение функции. Какого знака эта величина? $y = \alpha e^X - \sin^2 X \alpha - \alpha X^3$	0	1	2	0,25	0,5	1
6	Сумму отрицательных значений функции $y = \text{tg} \frac{X}{\alpha} - \cos^2 \alpha X$	0,5	1	2	0,1	1	2
7	Произведение положительных и произведение отрицательных значений функции $y = \sin(X + \alpha) - e^{\alpha X} + \alpha X^2$	-1	0	1	0,2	1	2
8	Сумму положительных и произведение отрицательных значений функции $y = \alpha X^3 - 1,5X + \alpha \text{tg} \alpha X$	-1	0	1	0,2	1	3
9	Сумму и произведение значений функции. Какая из этих величин больше? $y = X^\alpha - \alpha \text{tg} \alpha X$	1,2	1,8	2,5	0,1	1	2

№	Задача	Исходные данные					
		$X_{\min}$	$X_{\text{cp}}$	$X_{\max}$	$\Delta X$	$\alpha_1$	$\alpha_2$
10	Значения аргумента $X$ , при которых достигаются наибольшее и наименьшее значения функции $y = \alpha X - 0,2\sin^2(\alpha + X)$	0	0,5	1	0,1	1	- 1
11	Произведение и количество отрицательных значений функции $y = \cos \alpha X - \alpha \cdot \text{tg}^2 X + \alpha X$	- 1	- 0,5	0	0,1	1	2
12	Количество положительных и количество отрицательных значений функции. Каких на интервале больше? $y = \alpha X^3 + \sqrt{ X\alpha } - \cos^2 \alpha X$	0	0,4	0,7	0,05	2	1
13	Сумму и количество положительных значений функции $y = \sin^2 \alpha X - \alpha \cdot \text{tg} X$	0	1	2	0,2	1	0,5
14	Сумму положительных и сумму всех значений функции $y = X\alpha^2 - \text{tg} \alpha X + \cos^2 X\alpha$	1	2	3	0,2	1	0,6
15	Среднее арифметическое отрицательных значений функции $y = X\alpha^2 - \cos^2 \alpha X$	0	0,5	1	0,1	1	2
16	Сумму и произведение положительных значений функции $y = \text{tg}^2 \alpha X - \sqrt{ \alpha^2 - \ln X }$	1	2	2,5	0,1	1	2
17	Отношение произведения положительных значений функции к их количеству $y = 3X\alpha - 15\ln \alpha X - 3  - 40$	2,8	2,9	3	0,02	1	1,2
18	Сумму и количество отрицательных значений функции $y = \alpha X^3 + \sqrt{ 10\alpha X } - \alpha e^X$	0	0,2	0,3	0,02	1	1,6
19	Сумму положительных и сумму отрицательных значений функции $y = \alpha e^X + \text{tg} \alpha X - (X - \alpha)^2$	1,5	1,7	2	0,05	1	- 1
20	Произведение отрицательных значений функции $y = \sqrt{ \alpha X } + \alpha X^3 + e^{\alpha X} - 60$	3	3,4	3,7	0,05	1	0,9

### Таблица 3.6 Исследование функции одной переменной с различным шагом варьирования аргумента

Пример выполнения лабораторной работы – стр.28-33

Математические функции – табл. 5.1 (стр.136)

Порядок выполнения лабораторной работы – раздел 1 (стр.4)

Составить схему и программу решения для следующей задачи.

Вычислить таблицу значений функции  $Y = F(X)$  и аргумента  $X$  на интервале  $[X_{\min}, X_{\max}]$  с шагом  $\Delta X$ , причём шаг  $\Delta X$  принимает ряд фиксированных значений  $\Delta X_1$  и  $\Delta X_2$  в зависимости от текущего значения аргумента  $X$ , а именно:

$$\Delta X = \begin{cases} \Delta X_1, & \text{если } X_{\min} \leq X \leq X_{\text{ср}} \\ \Delta X_2, & \text{если } X_{\text{ср}} < X \leq X_{\max} \end{cases}$$

где  $X_{\text{ср}}$  – значение аргумента  $X$  на интервале  $X_{\min} < X_{\text{ср}} < X_{\max}$ .

При решении используются типовые алгоритмы вычисления суммы, произведения, наибольшего (наименьшего) значения.

#### Задания для лабораторной работы 6

*Протабулировать функцию и найти:*

№	Задача	Исходные данные				
		$X_{\min}$	$X_{\text{ср}}$	$X_{\max}$	$\Delta X_1$	$\Delta X_2$
1	Наименьшее значение функции. Какого знака эта величина? $y = \sin(1 - \sqrt{ x }) + e^x$	0	1	2,5	0,2	0,25
2	Наибольшее значение функции $y = \sin^2 2x - \sqrt{ x - \text{tg} x }$	1	2	5	0,25	0,5
3	Сумму и количество отрицательных значений функции $y = \text{tg}^2 x - \sqrt{ x - \cos x }$	0	1	3,5	0,25	0,5
4	Количество положительных и количество отрицательных значений функции $y = \sin 2x + \text{tg}^2 \sqrt{ x }$	5,2	5,7	6,1	0,1	0,05

№	Задача	Исходные данные				
		$X_{\min}$	$X_{\text{cp}}$	$X_{\max}$	$\Delta X_1$	$\Delta X_2$
5	Наибольшее из отрицательных значений функции $y = \cos 2x + \operatorname{tg}^2 \ln x $	0,5	1	2,5	0,1	0,25
6	Наименьшее из положительных значений функции $y = \sqrt{ x - \sin^2 x } - \operatorname{tg}(x + \frac{\pi}{3})$	3	4	7	0,2	0,5
7	Сумму и произведение значений функции. Какая из этих величин больше? $y = e^{-x} + \cos x$	3	7	9	0,5	0,25
8	Произведение положительных и количество отрицательных значений функции $y = \sin^2 3x - e^{-x}$	0	1	2	0,25	0,2
9	Среднее арифметическое значение функции. Какого знака эта величина? $y = \operatorname{tg}^2 x - 4 \cdot \operatorname{tg} x - 3$	1	2	5	0,25	0,5
10	Сумму положительных и количество отрицательных значений функции $y = \sin^2 x - \cos^2 x - \cos 2x$	0	1	4	0,2	0,5
11	Количество отрицательных значений функции $y = \operatorname{tg} x - \sin^2 2x$	3,5	4,5	7	0,25	0,5
12	Произведение отрицательных и сумму всех значений функции $y = \sqrt{ x - \operatorname{tg} x } - \cos 2x$	-1	0	1,2	0,25	0,2
13	Среднее арифметическое положительных значений функции $y = e^x - \operatorname{tg} x $	-1	0,5	2,1	0,25	0,2
14	Произведение и количество отрицательных значений функции $y = x^2 + \operatorname{tg} x - \cos^3 x$	0	1,5	2	0,25	0,1
15	Сумму положительных и произведение отрицательных значений функции $y = \sqrt{ x - \sin x } - \frac{1}{\operatorname{tg}^2 x}$	1	1,5	3	0,1	0,25

№	Задача	Исходные данные				
		$X_{\min}$	$X_{\text{cp}}$	$X_{\max}$	$\Delta X_1$	$\Delta X_2$
16	Сумму положительных и сумму всех значений функции $y = x^3 - \text{tg}^2 x - \sin x$	1	4	5	0,5	0,2
17	Отношение произведения положительных значений функции к их количеству $y = x^3 - \sin^2 x + \text{tg}(x + \frac{\pi}{3})$	0	1	1,5	0,2	0,1
18	Произведение положительных и произведение отрицательных значений функции $y = \text{tg}^2 x - 3 \cos x$	5	6	8,5	0,2	0,5
19	Сумму и произведение положительных значений функции $y = x + \text{tg} x  - \cos^2 x$	1,5	1,7	2,3	0,05	0,1
20	Произведение положительных и произведение всех значений функции $y = x - \text{tg}2x + \sin^2 x$	0,25	1,5	2,4	0,25	0,1

**Таблица 3.7 Исследование функции двух переменных**

Пример выполнения лабораторной работы – стр.34-38

Математические функции – табл. 5.1 (стр.136)

Порядок выполнения лабораторной работы – раздел 1 (стр.4)

Составить схему и программу решения для следующей задачи.

Вычислить и напечатать таблицу значений функции  $Z = f(X, Y)$  и аргументов  $X, Y$  в заданной области при условии, что:

- аргумент  $X$  изменяется в интервале  $X_0 \leq X \leq X_k$  с шагом  $\Delta X$ ;
- аргумент  $Y$  изменяется в интервале  $Y_0 \leq Y \leq Y_k$  с шагом  $\Delta Y$ ;

При решении используются алгоритмы вычисления суммы, произведения, количества, наибольшего (наименьшего) значения.

## Задания для лабораторной работы 7

**Протабулировать функцию и найти:**

№	Задача	Исходные данные					
		X <sub>0</sub>	X <sub>k</sub>	ΔX	Y <sub>0</sub>	Y <sub>k</sub>	ΔY
1	Среднее арифметическое положительных значений функции $Z = x \cdot \ln^2( y  + x^2) - e^{x \cdot y}$	1	1,2	0,1	-2	1	1
2	Наименьшее из положительных значений функции $Z = x \cdot \cos^2 y - y \sqrt{ x^2 - \sin y }$	0,8	1,2	0,2	0,5	2	0,5
3	Наибольшее из отрицательных значений функции $Z = y \cdot \sin^2 x - x \sqrt{ y^2 - \cos x }$	1	1,2	0,1	0,5	2	0,5
4	Произведение и количество положительных значений функции $Z = \sqrt{ \ln x - \cos y } - y \cdot \operatorname{tg}^2 x$	1	3	1	0	1,5	0,5
5	Сумму и количество отрицательных значений функции $Z = y \cdot \operatorname{tg}^2 x - \sqrt{ \cos x } e^y$	0,1	0,5	0,2	-3	3	2
6	Сумму положительных и сумму всех значений функции $Z = x \cdot \operatorname{tg}^2 y - \sqrt{ \cos y } e^x$	0,5	1,5	0,5	-1	2	1
7	Произведение отрицательных и количество положительных значений функции $Z = \sqrt{ y - \sin^2 x } - y \cdot e^{2x}$	0,4	0,6	0,1	1	2	0,5
8	Сумму положительных значений функции $Z = x \cdot \operatorname{tg}^2 y - y e^{-x}$	0,1	0,3	0,1	0,5	1,5	0,5
9	Количество отрицательных значений функции $Z = \operatorname{tg}^3(x \cdot y) + \sqrt{ \ln x }$	1	2	0,5	0,5	1,5	0,5
10	Сумму положительных и произведение отрицательных значений функции $Z = y \cdot \operatorname{tg}^2 x - x \cdot e^{-y}$	1	1,6	0,3	1	2	0,5
11	Произведение отрицательных значений функции $Z = y \cdot \operatorname{tg}^2 x - x \cdot e^{-y}$	0,5	0,7	0,1	0,5	1,5	0,5
12	Сумму положительных и сумму отрицательных значений функции $Z = x \cdot \cos^3 y - y \sqrt{ x }$	-1	3	2	0,5	1,5	0,5
13	Сумму отрицательных и количество положительных значений функции $Z = \sqrt{ \cos y } - y \cdot \sin^2 x$	-1	2	1	0,5	1,5	0,25

№	Задача	Исходные данные					
		X <sub>0</sub>	X <sub>k</sub>	ΔX	Y <sub>0</sub>	Y <sub>k</sub>	ΔY
14	Произведение положительных и произведение отрицательных значений функции $Z = x \cdot \sin^2 y - y \sqrt{ \cos y }$	0,3	0,7	0,2	0,5	1,5	0,5
15	Среднее арифметическое значение функции. Определить знак этой величины. $Z = y \cdot \sin^3 x - x \sqrt{ y }$	0,1	0,5	0,2	-1	3	2
16	Какие значения функции в области преобладают: положительные или отрицательные? $Z = y \cdot \operatorname{tg}^2 x - x \sqrt{ y }$	0,5	1,5	0,5	-3	3	2
17	Произведение положительных и произведение всех значений функции $Z = x \cdot \ln^2 x - ye^{-x}$	1	2	0,5	1	1,2	0,1
18	Сумму значений функции. Какого знака эта величина? $Z = \sin^2(x \cdot y) - \sqrt{ \ln x - y }$	1	1,4	0,2	1	2	0,5
19	Сумму и произведение значений функции. Какая из этих величин больше? $Z = \cos^2 3x + \sqrt{ \ln x - y }$	1,2	1,6	0,2	1	2	0,5
20	Отношение произведения отрицательных значений функции к их количеству. $Z = e^{-2y} - \sqrt{ x^3 - \sin^2 y }$	0	1	0,5	1,1	1,5	0,2

**Таблица 3.8 Исследование функции двух переменных при фиксированном значении аргумента**

Пример выполнения лабораторной работы – стр.39-42

Математические функции – табл. 5.1 (стр.136)

Порядок выполнения лабораторной работы – раздел 1 (стр.4)

Составить схему и программу решения для следующей задачи:

Вычислить и напечатать таблицу значений функции  $Z = f(X, Y)$

и аргументов  $X, Y$  в заданной области при условии, что:

- аргумент  $X$  изменяется в интервале  $X_0 \leq X \leq X_k$  с шагом  $\Delta X$ ;
- аргумент  $Y$  изменяется в интервале  $Y_0 \leq Y \leq Y_k$  с шагом  $\Delta Y$ ;

## Задания для лабораторной работы 8

При решении используются алгоритмы вычисления суммы, произведения, количества, наибольшего (наименьшего) значения.

**Протабулировать функцию и найти:**

(п.1 – п.10)

*Для каждого фиксированного значения аргумента X: параметр внешнего цикла – переменная X, а внутреннего – переменная Y.*

№	Задача	Исходные данные					
		X <sub>0</sub>	X <sub>k</sub>	ΔX	Y <sub>0</sub>	Y <sub>k</sub>	ΔY
1	Произведение положительных значений функции $Z = x \cdot \sin^2 y - \sqrt{ x^2 - \cos y }$	1	1,2	0,1	0,5	2	0,5
2	Сумму отрицательных значений функции $Z = x \cdot \operatorname{tg}^2 y - y \sqrt{ \ln x }$	0,2	0,4	0,1	1	2	0,5
3	Наименьшее из положительных значений функции $Z = \sin^2 x - \sqrt{ y \cdot \cos x }$	1	1,2	0,1	0,5	2	0,5
4	Среднее арифметическое отрицательных значений функции $Z = y \cdot \operatorname{tg}^2 x - x \sqrt{ \ln y }$	0,1	0,3	0,1	0,5	2	0,5
5	Количество положительных значений функции $Z = \sin^2(x \cdot y) - \sqrt{ x \cdot \ln y }$	1	1,6	0,3	1	2	0,5
6	Наибольшее из отрицательных значений функции $Z = y \cdot \cos^2 x - x \sqrt{ \ln y }$	1	1,2	0,1	1	2	0,5
7	Сумму и произведение положительных значений функции $Z = \operatorname{tg}^2 x - e^{2y}$	0,5	0,7	0,1	-1	1	1
8	Произведение и количество отрицательных значений функции $Z = y \cdot \ln^2  x  - e^{-y}$	2	3	0,5	-1	1	1
9	Сумму положительных и произведение отрицательных значений функции $Z = \operatorname{tg}^2 y - e^{3x}$	0,1	0,3	0,1	0,5	1,5	0,5
10	Сумму и количество положительных значений функции $Z = x \cdot \ln^2 y - e^{-2x}$	0,4	0,6	0,1	2	3	0,5



**Протабулировать функцию и найти:**

(п.11 – п.20)

Для каждого фиксированного значения аргумента  $Y$ : параметр внешнего цикла – переменная  $Y$ , а внутреннего – переменная  $X$ .

№	Задача	Исходные данные					
		$X_0$	$X_k$	$\Delta X$	$Y_0$	$Y_k$	$\Delta Y$
11	Сумму положительных значений функции $Z = \sin^2(x \cdot y) - x \sqrt{ \cos y }$	1	1,2	0,1	1,9	2,1	0,1
12	Произведение отрицательных значений функции $Z = y \cdot \operatorname{tg}^2 x - x \sqrt{ \ln y }$	0,2	0,6	0,2	1,5	2,1	0,3
13	Наименьшее из положительных значений функции $Z = \operatorname{tg}^2 x - e^{2y}$	0,5	1,5	0,5	0,1	0,3	0,1
14	Количество отрицательных значений функции $Z = \ln^2 x - e^{-y}$	1	2	0,5	1	1,2	0,1
15	Среднее арифметическое положительных значений функции $Z = \sin^2 x - \sqrt{ \ln y }$	1	1,4	0,2	1,9	2,1	0,1
16	Наибольшее из отрицательных значений функции $Z = x \cdot \ln^2 y - ye^{-x}$	0,9	1,1	0,1	2,5	2,7	0,1
17	Произведение и сумму положительных значений функции $Z = y \cdot \operatorname{tg}^2 x - e^{x \cdot y}$	1	2	0,5	0,1	0,3	0,1
18	Сумму положительных и сумму отрицательных значений функции $Z = \operatorname{tg}^2(x \cdot y) - y \cdot e^x$	0,5	1,5	0,5	1	1,4	0,2
19	Сумму и количество положительных значений функции $Z = y \cdot \operatorname{tg}^2 x - x \cdot e^{2y}$	0,5	1,5	0,5	0,1	0,3	0,1
20	Сумму отрицательных и количество положительных значений функции $Z = y \cdot \operatorname{tg}^3 x - x \cdot e^y$	-3	-1	1	0,5	0,7	0,1

**Таблица 3.9 Вычисление координат точки, наиболее удалённой от начала координат (ближайшей к нему)**

Пример выполнения лабораторной работы – стр.43-47

Математические функции – табл. 5.1 (стр.136)

Порядок выполнения лабораторной работы – раздел 1 (стр.4)

Составить схему и программу решения для следующей задачи.

На плоскости  $(x, y)$  заданы точки  $\{M_1(X_1, Y_1), \dots, M_n(X_n, Y_n)\}$  и круг радиуса  $R$  с центром в начале координат,  $n = 10$  – количество точек. Координаты точек задаются в виде массивов абсцисс  $(X_1, \dots, X_n)$  и ординат  $(Y_1, \dots, Y_n)$ , в соответствии с условием задачи и так, чтобы искомая точка была одна.

При решении задачи используются типовые циклические алгоритмы вычисления наибольшего (наименьшего) значения.

### Задания для лабораторной работы 9

**Определить координаты:**

(п.1 – п.10)

**Точки, наиболее удалённой от начала координат и лежащей:**

№	Задача
1	В первом квадранте и внутри круга
2	Во втором или четвертом квадранте и вне круга
3	В нижней полуплоскости и внутри круга
4	В третьем квадранте и вне круга
5	Во втором или третьем квадранте и внутри круга
6	Во втором квадранте и вне круга
7	В первом или третьем квадранте и внутри круга
8	В верхней полуплоскости и вне круга
9	В четвертом квадранте и внутри круга
10	В третьем или четвёртом квадранте и вне круга

(п.11 – п.20)

**Точки, ближайшей к началу координат и лежащей:**

11	Во втором квадранте и внутри круга
12	В первом или втором квадранте и вне круга
13	В первом квадранте и внутри круга
14	В левой полуплоскости и вне круга
15	В первом или третьем квадранте и внутри круга
16	Во втором или четвёртом квадранте и вне круга
17	В третьем квадранте и внутри круга
18	В третьем или четвёртом квадранте и вне круга
19	В четвертом квадранте и внутри круга
20	В первом квадранте и вне круга

### Таблица 3.10 Вычисление координат искомой точки, ближайшей к заданной (наиболее удалённой от неё)

Пример выполнения лабораторной работы – стр.48-52

Математические функции – табл. 5.1 (стр.136)

Порядок выполнения лабораторной работы – раздел 1 (стр.4)

Составить схему и программу решения для следующей задачи.

На плоскости  $(x, y)$  заданы точки  $\{M_1(X_1, Y_1), \dots, M_n(X_n, Y_n)\}$  и  $Q(X_q, Y_q)$ . Координаты точек задаются в виде массивов абсцисс  $(X_1, \dots, X_n)$  и ординат  $(Y_1, \dots, Y_n)$ , в соответствии с условием задачи и так, чтобы искомая точка была одна.

При решении задачи используется типовой алгоритм вычисления наибольшего (наименьшего) значения.

#### Задания для лабораторной работы 10

**Определить координаты:**

(п.1 – п.10)

**Точки, ближайшей к точке  $Q(X_q, Y_q)$  и лежащей:**

№	Задача	n
1	В одной полуплоскости с ней (левой или правой)	8
2	В первом квадранте	9
3	Во втором или четвертом квадранте	7
4	В третьем квадранте	9
5	В четвертом квадранте	8
6	В верхней полуплоскости	9
7	В первом или четвертом квадранте	8
8	Во втором квадранте	7
9	В одном квадранте с точкой $Q(X_q, Y_q)$	9
10	Во втором или третьем квадранте	8

(п.11 – п.20)

**Точки, наиболее удалённой от точки  $Q(X_q, Y_q)$  и лежащей:**

11	В одной полуплоскости с ней (верхней или нижней)	8
12	В четвертом квадранте	7
13	В правой полуплоскости	9
14	Во втором квадранте	7
15	В первом или втором квадранте	8

16	В третьем квадранте	8
17	Во втором или четвертом квадранте	7
18	В первом квадранте	9
19	В нижней полуплоскости	7
20	Во втором или третьем квадранте	8

**Таблица 3.11 Вычисление координат искомой точки**

Пример выполнения лабораторной работы – стр.52-55

Математические функции – табл. 5.1 (стр.136)

Порядок выполнения лабораторной работы – раздел 1 (стр.4)

Составить программу решения следующей задачи.

На плоскости  $(x, y)$  заданы точки  $\{M_1(X_1, Y_1), \dots, M_n(X_n, Y_n)\}$ ,  $n$  – количество точек. Координаты точек задаются в виде массивов абсцисс  $(X_1, \dots, X_n)$  и ординат  $(Y_1, \dots, Y_n)$ , в соответствии с условием задачи и так, чтобы искомая точка была одна.

При решении задачи используется типовой алгоритм вычисления наибольшего (наименьшего) значения.

### Задания для лабораторной работы 11

**Определить координаты:**

(п.1 – п.5)

**Точки с наименьшей абсциссой среди точек, лежащих:**

№	Задача	n
1	В первом и третьем квадранте	8
2	Во втором квадранте	9
3	В третьем и четвертом квадранте	7
4	В третьем квадранте	9
5	В верхней полуплоскости	8

(п.6 – п.10)

**Точки с наименьшей ординатой среди точек, лежащих:**

6	Во втором и третьем квадранте	9
7	Во втором квадранте	8
8	В правой полуплоскости	7
9	В первом и третьем квадранте	9
10	В четвертом квадранте	8

(п.11 – п.15)

**Точки с наибольшей абсциссой среди точек, лежащих:**

11	В первом и втором квадранте	8
12	В нижней полуплоскости	7
13	В третьем квадранте	9
14	Во втором и четвертом квадранте	7
15	В четвертом квадранте	8

(п.16 – п.20)

**Точки с наибольшей ординатой среди точек, лежащих:**

16	В третьем квадранте	8
17	В первом и четвертом квадранте	7
18	В левой полуплоскости	9
19	В первом квадранте	7
20	Во втором и четвертом квадранте	8

### Таблица 3.12 Вычисление значения выражения

Пример выполнения лабораторной работы – стр.56-59

Математические функции – табл. 5.1 (стр.136)

Порядок выполнения лабораторной работы – раздел 1 (стр.4)

Составить схему алгоритма и программу решения задачи:

Вычислить значение величины  $Z$  по заданной формуле. Массив  $A_i$ , ( $i = 1, \dots, n$ ) задаётся самостоятельно. При вычислении  $\min$ ,  $\max$  параметр цикла  $i$  изменяется в интервале  $[1; n]$ .

При решении используются типовые алгоритмы вычисления суммы, произведения, наибольшего (наименьшего) значения.

### Задания для лабораторной работы 12

№	Формула для вычисления значения $Z$	n
1	$Z = \sum_{i=2}^{n-1} (i! + A_i) - \min_i \left( A_i \prod_{i=1}^{n-2} \sqrt{i} \right)$	6
2	$Z = \min_i \left( A_i \prod_{i=1}^{n-3} i^2 \right) + \sum_{i=3}^n A_i^2$	6

3	$Z = \prod_{i=1}^{n-2} \left( A_i + \sum_{i=2}^{n-1} i! \right) - \min_i (iA_i)$	6
4	$Z = \max_i (iA_i) + \sum_{i=2}^{n-2} A_i^i - \prod_{i=1}^{n-3} i!$	6
5	$Z = \sum_{i=2}^{n-1} \left( i^2 - \prod_{i=1}^{n-3} A_i \right) + \max_i (A_i^2)$	6
6	$Z = \max_i \left( A_i^2 - \prod_{i=3}^{n-1} A_i \right) + \sum_{i=1}^{n-2} \sqrt{i}$	7
7	$Z = \sum_{i=2}^{n-3} \sqrt{i!} + \min_i  A_i  - \prod_{i=1}^{n-3} A_i$	7
8	$Z = \min_i (A_i \sqrt{i}) + \sum_{i=2}^{n-2} \left( i! - \prod_{i=1}^{n-3} A_i^2 \right)$	7
9	$Z = \prod_{i=2}^{n-3} (i! - \min_i \{iA_i\}) + \sum_{i=1}^{n-2} \sqrt{i} A_i$	7
10	$Z = \max_i \left( A_i \sum_{i=2}^{n-3} i! \right) - \prod_{i=1}^{n-2} A_i^{-i}$	7
11	$Z = \sum_{i=2}^{n-3} \left( \ln i - \prod_{i=1}^{n-2} A_i \right) + \min_i \left( \sqrt{ A_i } \right)$	6
12	$Z = \max_i (iA_i^2) - \sum_{i=1}^{n-2} \left( i! + \prod_{i=2}^{n-3} A_i \right)$	6
13	$Z = \prod_{i=1}^{n-2} \left( \sqrt{i} - \min_i \{A_i^3\} \right) + \sum_{i=2}^{n-1} A_i$	6
14	$Z = \min_i (i!A_i) + \prod_{i=2}^{n-3} \left( A_i - \sum_{i=1}^{n-2} \sqrt{i} \right)$	6
15	$Z = \sum_{i=1}^{n-2} \left( i! + \max_i  A_i  \right) - \prod_{i=2}^{n-1} \sqrt{ A_i }$	6

16	$Z = \max_i (iA_i^2) \sum_{i=1}^{n-1} \left( \sqrt{i!} - \prod_{i=2}^{n-2} A_i \right)$	5
17	$Z = \prod_{i=2}^{n-1} \left( \sqrt{i!} - \max_i  A_i  \right) + \sum_{i=1}^n A_i^2$	5
18	$Z = \min_i \left( A_i + \prod_{i=2}^{n-1} i! \right) + \sum_{i=1}^{n-2} A_i^2$	5
19	$Z = \sum_{i=2}^n \left( \ln i + \max_i \{i!A_i\} \right) - \prod_{i=2}^n A_i$	5
20	$Z = \max_i \left(  A_i  - \prod_{i=2}^{n-1} A_i^2 \right) + \sum_{i=1}^{n-2} iA_i$	5

**Таблица 3.13 Анализ эффективности  
работы предприятия**

Пример выполнения лабораторной работы – стр.59-63

Порядок выполнения лабораторной работы – раздел 1 (стр.4)

Составить схему алгоритма и программу решения задачи:

Выполнение плана (в процентах) по объемам производства на судостроительном предприятии в течение года и названия месяцев представлены массивами  $A_i$  и  $MC_i$  ( $i = 1, \dots, 12$ ). План – 100%, менее 100% – план не выполнен, более 100% – перевыполнен.

### Исходные данные

1-е полугодие						
Месяц	1	2	3	4	5	6
	Январь	Февраль	Март	Апрель	Май	Июнь
%	$A_1$	$A_2$	$A_3$	$A_4$	$A_5$	$A_6$
	<b>101</b>	<b>98</b>	<b>100</b>	<b>108</b>	<b>97</b>	<b>105</b>
2-е полугодие						
Месяц	7	8	9	10	11	12
	Июль	Август	Сентябрь	Октябрь	Ноябрь	Декабрь
%	$A_7$	$A_8$	$A_9$	$A_{10}$	$A_{11}$	$A_{12}$
	<b>99</b>	<b>107</b>	<b>96</b>	<b>112</b>	<b>100</b>	<b>98</b>

При решении задачи используются типовые алгоритмы вычисления суммы, количества, наименьшего (наибольшего) значения.

**Вычисление среднегодового процента выполнения плана предусмотреть в программе.**

### Задания для лабораторной работы 13

**Определить:**

№	Задача
1	Средний процент выполнения плана в течение тех месяцев, когда план не выполнялся
2	Полугодие с наибольшим средним процентом выполнения плана
3	Месяцы с процентом выполнения, превышающим среднегодовой
4	В каком полугодии достигнут наибольший процент?
5	Средний процент выполнения плана в течение тех месяцев, когда план перевыполнялся
6	Месяц с наибольшим процентом перевыполнения плана
7	Превышает ли среднегодовой процент выполнения плана 100%? Вывести соответствующее текстовое сообщение.
8	Количество месяцев с процентом выполнения плана ниже среднегодового
9	Наибольший процент выполнения плана в течение тех месяцев, когда план не выполнялся
10	Количество месяцев, в течение которых план перевыполнялся
11	Наименьший процент выполнения плана в течение тех месяцев, когда план перевыполнялся
12	В течение каких месяцев план не выполнялся
13	Месяц с наибольшим отставанием от плана в течение года
14	Наибольший процент выполнения плана в каждом полугодии. В каком полугодии этот показатель выше?
15	Для каждого полугодия количество месяцев, когда план не выполнялся. В каком полугодии этот показатель ниже?
16	Месяц с наибольшим процентом среди месяцев, в течение которых процент выполнения плана не превышает среднегодовой.
17	В каком полугодии наблюдается самый низкий процент выполнения?
18	В каком полугодии средний процент выполнения плана выше среднегодового?
19	Месяцы с процентом выполнения плана ниже среднего
20	Каких месяцев в году больше: с перевыполнением или невыполнением плана?



### Таблица 3.14 Анализ итогов экзамена в группе

Пример выполнения лабораторной работы – стр.64-67

Порядок выполнения лабораторной работы – раздел 1 (стр.4)

Составить схему алгоритма и программу решения задачи:

Результаты сдачи экзамена и фамилии студентов группы представлены массивами  $A_i$  и  $F_i$  ( $i = 1, \dots, 10$ ).

#### Исходные данные

Студент	$F_1$	$F_2$	$F_3$	$F_4$	$F_5$
Фамилия	Юдин И.В.	Ашин С.В.	Итин К.Л.	Бут С.С.	Бех Л.Л.
Оценка	$A_1$	$A_2$	$A_3$	$A_4$	$A_5$
	5	4	4	3	2
Студент	$F_6$	$F_7$	$F_8$	$F_9$	$F_{10}$
Фамилия	Этин П.Н.	Юров Ю.Ю.	Апин Р.П.	Фет В.И.	Кац Д.Е.
Оценка	$A_6$	$A_7$	$A_8$	$A_9$	$A_{10}$
	5	4	2	4	3

При решении задачи используются типовой алгоритм вычисления суммы и количества.

#### Задания для лабораторной работы 14

**Вычисление среднего балла предусмотреть в программе!**

**Определить:**

№	Задача
1	Какая категория (хороших или отличных) оценок преобладает?
2	Процент хороших и отличных оценок в группе
3	Количество успевающих студентов
4	Процент оценок каждой категории
5	Количество и фамилии студентов с оценкой «удовлетворительно»
6	Процент и фамилии успевающих студентов
7	Средний балл среди хорошистов и отличников
8	Процент неудовлетворительных оценок
9	Каких оценок в группе больше: «хорошо» и «отлично» или «удовлетворительно»?
10	Средний балл среди успевающих студентов
11	Фамилии и количество неуспевающих студентов
12	Процент успевающих студентов и процент отличников

№	Задача
13	Процент и фамилии студентов-хорошистов
14	Количество успевающих и количество неуспевающих студентов. Какая категория преобладает?
15	Превышает ли средний балл группы оценку «хорошо»? Вывести текстовое сообщение.
16	Фамилии успевающих студентов с оценкой ниже среднего балла
17	Процент оценок, превышающих средний балл группы
18	Количество студентов из числа успевающих с оценкой ниже среднего балла группы
19	Фамилии студентов с оценкой выше среднего балла группы
20	Каких студентов из числа успевающих больше: получивших оценку выше или ниже среднего балла группы?

### Таблица 3.15 Анализ эффективности работы флота

Пример выполнения лабораторной работы – стр.70-80

Составить схему алгоритма и программу решения задачи:

Фактические данные по объёмам перевозок грузов пятью судами за четыре месяца навигации представлены в виде матрицы  $A(5 \times 4)$ , где строка – судно, столбец - месяц. Данные приведены в процентном отношении к нормативным объёмам перевозок (плану). За план принимается значение, равное 100%. Менее 100% – план не выполнен, более 100% – перевыполнен.

Матрица задаётся в соответствии с условием задачи. Элемент матрицы  $A_{ij}$  – это процент выполнения плана судном с номером  $i$  за месяц навигации с номером  $j$ .

#### Порядок выполнения лабораторной работы:

1. В соответствии с обходом матрицы [по строкам/столбцам] определить параметры внешнего и внутреннего цикла ( $i$  или  $j$ ).
2. Определиться с типом алгоритма решения задачи – вычисления суммы, количества, наибольшего или наименьшего значения.
3. В соответствии с алгоритмом при необходимости дополнительно к исходным данным ввести обозначения переменных.
4. Представить имена всех переменных в таблице.
5. Выполнить ручной расчёт результатов.
6. Составить макет печати исходных данных и результатов.

7. Составить схему алгоритма решения задачи.
8. В соответствии со схемой и макетом печати написать программу.
9. Проанализировать полученные результаты. Ручной и машинный расчёты должны совпадать.

**Примечание.** Ввод матрицы можно осуществлять как по строкам, так и по столбцам. Вывод же матрицы организуется в общепринятом виде, т.е. по строкам. Матрица, распечатанная по столбцам, является транспонированной, и не будет соответствовать исходной.

При решении задач используются типовые алгоритмы вычисления суммы, количества, наибольшего (наименьшего) значения.

### Задания для лабораторной работы 15

**Определить:**

(п.1 – п.10)

№	Задача
1	Количество судов, ежемесячно выполнявших план
2	Месяц навигации с наименьшим средним процентом
3	Суда, ежемесячно перевыполнявшие план (100%)
4	Количество месяцев, в которых план выполнялся всеми судами
5	Превышает ли средний процент выполнения плана судами флота за навигацию 100%?
6	Судно с наименьшим средним процентом выполнения плана
7	Месяцы навигации, в которых план не выполнялся ни одним судном
8	Наименьший процент за навигацию. Превышает ли он 100%?
9	Судно с наименьшим средним процентом среди судов, ежемесячно выполнявших план
10	Судно с наибольшим процентом перевыполнения плана

(п.11 – п.15)

*Для каждого судна (строки i):*

№	Задача
11	Количество месяцев, в течение которых план перевыполнялся
12	Превышает ли средний процент выполнения плана 100%?
13	Наибольший процент в течение месяцев, когда план не выполнялся
14	Средний процент в течение месяцев, когда план перевыполнялся
15	Наименьший процент в течение месяцев, когда план выполнялся

(п.16 – п.20)

*Для каждого месяца (столбца j):*

№	Задача
16	Наименьший процент среди судов, перевыполнявших план
17	Превышает ли средний процент выполнения плана 100%?
18	Количество судов, не выполнивших план
19	Средний процент среди судов, перевыполнивших план
20	Наибольший процент среди судов, не выполнивших план

### **Таблица 3.16 Анализ итогов сессии в группе**

Пример выполнения лабораторной работы – стр.81-85

Порядок выполнения лабораторной работы – стр.4, стр.120

Составить схему алгоритма и программу решения задачи:

Результаты сдачи сессии из четырёх предметов в группе из пяти студентов представлены в виде матрицы  $A(5 \times 4)$ , где строка – студент, столбец – предмет. Элемент матрицы  $A_{ij}$  – оценка, полученная студентом с номером  $i$  по предмету с номером  $j$ . Матрица задаётся в соответствии с условием задачи.

При решении задач используются типовые алгоритмы вычисления суммы, количества, наибольшего (наименьшего) значения.

### **Задания для лабораторной работы 16**

*Определить для группы:*

(п.1 – п.10)

№	Задача
1	Процент успевающих студентов
2	Превышает ли средний балл группы оценку «хорошо»?
3	Количество отличников в группе
4	Процент хороших и отличных оценок
5	Какая из оценок [«хорошо» или «удовлетворительно»] преобладает?
6	Количество студентов, сдавших на стипендию (критерий – без троек)
7	Предмет с наименьшим средним баллом
8	Количество студентов, получивших оценку «неудовлетворительно»
9	Процент неудовлетворительных оценок в группе
10	Количество студентов, сдавших все экзамены на оценку «удовлетворительно»

(п.11 – п.15)

**Определить для каждого предмета (столбца j):**

№	Задача
11	Средний балл среди успевающих студентов
12	Процент неудовлетворительных оценок. Предусмотреть вывод текстового сообщения, если таких оценок нет
13	Превышает ли средний балл оценку «удовлетворительно»?
14	Количество хороших и отличных оценок
15	Какая категория оценок преобладает: «отлично» и «хорошо» или «удовлетворительно»?

(п.16 – п.20)

**Определить для каждого студента (строки i):**

№	Задача
16	Количество удовлетворительных оценок. Если таких оценок нет, вывести текстовое сообщение
17	Превышает ли средний балл оценку «хорошо»?
18	Каких оценок получено больше: «отлично» и «хорошо» или «удовлетворительно»?
19	Процент хороших и отличных оценок
20	Процент неудовлетворительных оценок. Предусмотреть вывод текстового сообщения, если таких оценок нет

### **Таблица 3.17 Формирование одномерного массива**

Пример выполнения лабораторной работы – стр.85-90

Порядок выполнения лабораторной работы –стр.4, стр.120

Составить схему алгоритма и программу решения задачи.

При решении используются алгоритмы вычисления суммы, произведения, количества, наибольшего (наименьшего) значения.

### **Задания для лабораторной работы 17**

(п.1 – п.10)

Дана квадратная матрица  $A(6 \times 6)$ .

Сформировать массив  $(B_1, \dots, B_6)$  в соответствии с условием задачи. Упорядочить массив по возрастанию, либо по убыванию (схема алгоритма – рис. 2.31, стр.89). Матрица задаётся самостоятельно.

**Каждый элемент массива определяется как разность между:**

№	Задача
1	Наибольшим и наименьшим элементами строки матрицы. Найти среднее арифметическое элементов массива В
2	Наименьшим и диагональным элементами строки матрицы. Найти произведение элементов массива В
3	Диагональным и наибольшим элементами строки матрицы. Найти сумму отрицательных элементов массива В
4	Количеством положительных и количеством отрицательных элементов строки матрицы. Найти наибольший элемент массива В
5	Суммой положительных и произведением отрицательных элементов строки матрицы. Найти наименьший элемент массива В
6	Диагональным и наименьшим элементами столбца матрицы. Найти сумму отрицательных элементов массива В
7	Наибольшим и диагональным элементами столбца матрицы. Найти произведение положительных элементов массива В
8	Количеством положительных и суммой отрицательных элементов столбца матрицы. Найти наименьший элемент массива В
9	Наибольшим из отрицательных и диагональным элементами столбца матрицы. Найти сумму элементов массива В
10	Суммой отрицательных и произведением положительных элементов столбца матрицы. Найти наибольший элемент массива В

(п.11 – п.20)

По матрице сформировать массив  $(V_1, \dots, V_6)$ . Причём, формула для вычисления элемента массива зависит от чётности строки, либо столбца матрицы. Чётность проверяется условием:

$$(-1)^i > 0 \text{ или } (-1)^j > 0$$

Если условие выполняется (да), строка/столбец – чётные, иначе – нечётные.

**Каждый элемент массива определяется как:**

№	Задача
11	Диагональный элемент в чётной и наименьший элемент в нечётной строке матрицы. Найти среднее арифметическое отрицательных элементов массива В
12	Наибольший элемент в чётной и диагональный элемент в нечётной строке. Найти произведение положительных элементов массива В

13	Количество положительных элементов в чётной и сумма отрицательных элементов в нечётной строке матрицы. Найти наименьший элемент массива В
14	Первый элемент в чётной и наибольший из отрицательных элементов в нечётной строке матрицы. Найти сумму элементов массива В
15	Наименьший элемент в чётной и наибольший – в нечётной строке матрицы. Найти количество положительных элементов массива В
16	Диагональный элемент в чётном и количество положительных элементов в нечётном столбце. Найти наименьший элемент массива В
17	Сумма отрицательных элементов в чётном и диагональный элемент в нечётном столбце матрицы. Найти наибольший элемент массива В
18	Наименьший из положительных элементов в чётном и наибольший из отрицательных элементов в нечётном столбце матрицы. Найти сумму элементов массива В
19	Первый элемент в чётном и наибольший элемент в нечётном столбце матрицы. Найти количество положительных элементов массива В
20	Произведение положительных элементов в чётном и наименьший элемент в нечётном столбце. Найти сумму элементов массива В

## 4. Задания для самостоятельной работы

### Лабораторная работа 4.1

### Вычисление значения функции

#### Тема. Линейный алгоритм

Математические функции – табл. 5.1 (стр.136)

Пример выполнения лабораторной работы – стр.5-8

Составить программу вычисления функции  $Y(X)$ . Значения  $X$  и  $Y(X)$  приведены в таблице. При несовпадении результата необходимо исправить ошибки и повторно выполнить программу.

#### Задания для лабораторной работы 4.1

№	Формула для вычисления функции	Аргумент X	Функция Y
1	$y = \cos^2(3 \cdot x) + \sqrt{\ln x } \cdot e^x$	1,4	2.5926
2	$y = \sqrt[3]{ x } + \sin^2(x) \cdot e^{-2x}$	1,5	1.1943
3	$y = \ln^2 x \cdot \cos 8^0  + \sqrt{ x } \cdot e^x$	0,8	2.0448
4	$y = \sqrt[3]{ x \cdot \cos 2^0 } + \operatorname{tg}^2(x \cdot \ln x)$	0,6	0.9434

№	Формула для вычисления функции	Аргумент X	Функция Y
5	$y = \ln^3(x \cdot \sqrt{ \sin 5^0 }) - e^{-x}$	2,5	-0.1101
6	$y = \cos^2(x) \cdot e^{-x} + \sqrt{\ln x }$	2,7	1.0516
7	$y = \arctg^2 x - \sqrt{ x } \cdot e^{-x}$	0,5	-0.2139
8	$y = \sqrt{ x \cdot \sin 3^0 } + \operatorname{tg}^3(x \cdot e^x)$	0,3	0.2041
9	$y = \sin^2(3 \cdot x) - \ln(x \cdot \sqrt{ x })$	1,4	0.2549
10	$y = \ln^2(e^{-x}) + \sqrt[3]{ x \cdot \cos 7^0 }$	0,5	1.0417
11	$y = \sqrt{\ln x } \cdot e^{2x} - \arctg^2 x$	1,2	3.9393
12	$y = \sin^2(x \cdot e^{-x}) + \sqrt{ x \cdot \operatorname{tg} 5^0 }$	0,7	0.3635
13	$y = x \cdot e^{3x} - \arctg^2(\sqrt{ x })$	-2	-0.9175
14	$y = \cos^2(x \cdot \ln x ) + e^{-2x}$	2,3	0.1243
15	$y = \sqrt[3]{ x \cdot \ln x } + ch^2 x$	0,5	1.9739

### Лабораторная работа 4.2

## Вычисление функции, заданной в виде ветвления

### Тема. Разветвлённый алгоритм

Математические функции – табл. 5.1 (стр.136)

Примеры выполнения лабораторных работ – стр.8-14

Составить схему и программу вычисления функции  $y(x)$ .

**Для выполнения лабораторной работы необходимо:**

1. В таблице даны значения аргумента  $x$  и функции  $y(x)$  для каждой ветви. При выполнении программы по запросу с экрана ввести с клавиатуры первое значение аргумента  $x$ . Полученное значение  $y(x)$  сравнить с приведённым в таблице.

2. Действие п.1 выполняется поочередно для каждой ветви.

3. При несовпадении результата повторить действие п.1.



## Задания для лабораторной работы 4.2

№	Функции $y(x)$	Значение	
		$x$	$y(x)$
1	$y = \begin{cases} 3\sqrt{ x-1 } - 3.25 & \text{при } x < -1.25 \\ -x & \text{при } -1.25 \leq x \leq 1.25 \\ 3.25 - 3\sqrt{ x-1 } & \text{при } x > 1.25 \end{cases}$	-2 1 2	1,95 -1 0,25
2	$y = \begin{cases} \sin^2(x+1) & \text{при } x < -1 \\ -(1+\cos \pi x) & \text{при } -1 \leq x < 1 \\ \sin^2(x-1) & \text{при } x \geq 1 \end{cases}$	-2 0 3	0,7 -2 0,8
3	$y = \begin{cases} -\cos 5\pi(x-0.2) & \text{при } x < 0.2 \\ 1 - 2 \cos 5\pi(x-0.2) & \text{при } 0.2 \leq x < 0.4 \\ 3 & \text{при } x \geq 0.4 \end{cases}$	0 0,3 1	1 1 3
4	$y = \begin{cases} 1 + 2x^2 & \text{при } x \leq 0 \\ 1 & \text{при } 0 < x < 0.5 \\ 1 + (x - 0.5)^4 & \text{при } x \geq 0.5 \end{cases}$	-1 0,2 0,5	3 1 1
5	$y = \begin{cases} 6\sqrt[3]{ x } - 5 & \text{при } x < -1 \\ x^2 & \text{при } -1 \leq x \leq 1 \\ 6\sqrt[3]{x} - 5 & \text{при } x > 1 \end{cases}$	-8 -1 8	7 1 7
6	$y = \begin{cases} 7\sqrt{ x-1 } - 11 & \text{при } x < -3 \\ x & \text{при } -3 \leq x < 3 \\ 7\sqrt{ x+1 } - 11 & \text{при } x \geq 3 \end{cases}$	-8 1 3	10 1 3
7	$y = \begin{cases} \pi - (x + \pi)^2 & \text{при } x < -\pi \\ \pi + \cos x + 1 & \text{при } -\pi \leq x < \pi \\ \pi + (x - \pi)^2 & \text{при } x \geq \pi \end{cases}$	$-2\pi$ 0 $2\pi$	-6,73 5,14 13,01
8	$y = \begin{cases} 0 & \text{при } x < 1 \\ \ln x / x & \text{при } 1 \leq x < e \\ x^2 / e^x & \text{при } x \geq e \end{cases}$	-1 2 3	0 0,34 0,45
9	$y = \begin{cases} 1 + \sin^3(x+0.5) & \text{при } x < -0.5 \\ 1 & \text{при } -0.5 \leq x \leq 0.5 \\ 1 + \sin^3(x-0.5) & \text{при } x > 0.5 \end{cases}$	-1 0 1	0,89 1 1,1

№	Функции $y(x)$	Значение	
		$x$	$y(x)$
10	$y = \begin{cases} x + e^{-x} & \text{при } x \leq 0 \\ 1 & \text{при } 0 < x < 1 \\ 1 + (x-1)^2 & \text{при } x \geq 1 \end{cases}$	0 0,5 2	1 1 2
11	$y = \begin{cases} -\cos(x+1) & \text{при } x \leq -1 \\ \sin \frac{\pi}{2} x & \text{при } -1 < x < 1 \\ \cos(x-1) & \text{при } x \geq 1 \end{cases}$	-1 0,5 1	-1 0,7 1
12	$y = \begin{cases} (x+1)^2 & \text{при } x < -1 \\ 1 + \cos \pi x & \text{при } -1 \leq x < 1 \\ -(x-1)^2 & \text{при } x \geq 1 \end{cases}$	-2 0 2	1 2 -1
13	$y = \begin{cases}  x  & \text{при } x \leq 0 \\ -\sin 2\pi x & \text{при } 0 < x \leq 1 \\ 1 - x & \text{при } x > 1 \end{cases}$	-2 0 2	2 0 -1
14	$y = \begin{cases} x(x+2) & \text{при } x \leq -2 \\ \sin \pi x & \text{при } -2 < x \leq 0 \\ x(x+2) & \text{при } x > 0 \end{cases}$	-3 -1 1	3 0 3
15	$y = \begin{cases} \sqrt{ x } & \text{при } x < 0 \\ \sin^2 x + \sqrt{x} & \text{при } 0 \leq x \leq \pi \\ x^2 & \text{при } x > \pi \end{cases}$	-4 1 4	2 1,7 16

### Лабораторная работа 4.3

## Расчёт грузовой шкалы судна

### Тема. Табулирование функции одной переменной

Примеры выполнения лабораторных работ – стр.17-33

Порядок выполнения лабораторной работы – раздел 1 (стр.4)

#### 1. Задание

Составить схему и программу решения следующей задачи:

При ведении погрузочных работ масса груза на судне  $G$  определяется по таблице грузового размера (грузовой шкале). Рассчитать зависимость между осадкой судна  $T$  и количеством груза  $G$ , задаваемую в виде ветвления:

$$G = \begin{cases} \gamma LB \sigma (T - T_0), & \text{если } T_0 \leq T \leq T_{cp}; \\ \gamma LB (\sigma T - \sigma_0 T_0), & \text{если } T_{cp} < T \leq T_{max}. \end{cases}$$

Плотность пресной воды  $\gamma = 1$ , её в расчётах не учитываем.

### Исходные данные

<i>L</i>	<i>B</i>	$\sigma$	$\sigma_0$	$T_0$	$T_{cp}$	$T_{max}$	$\Delta T$	$\gamma$
8,5	13,8	0,85	0,65	0,2	0,6	1,2	0,2	1

## 2. Решение

### Математическая формулировка задачи.

Вычислить и напечатать таблицу значений функции  $G$  и аргумента  $T$  при условии, что аргумент  $T$  изменяется на интервале

$T_0 \leq T \leq T_{max}$  с шагом  $\Delta T$ . Функция  $G$  определяется как:

$$G = \begin{cases} f1(T), & \text{если } T_0 \leq T \leq T_{cp}; \\ f2(T), & \text{если } T_{cp} < T \leq T_{max}; \end{cases}$$

где  $f1(T) = LB \sigma (T - T_0)$ ,  $f2(T) = LB (\sigma T - \sigma_0 T_0)$

Обозначения переменных				
Имя переменной		Тип переменной		Пояснение
в задаче	в программе	Fortran	C	
<b>L, B</b>	<b>L, B</b>	<b>REAL</b>	<b>FLOAT</b>	Длина и ширина судна
<b><math>\sigma, \sigma_0</math></b>	<b>S, S0</b>	<b>REAL</b>	<b>FLOAT</b>	Коэффициент полноты водоизмещения при текущей осадке и порожнем
<b>T, G</b>	<b>T, G</b>	<b>REAL</b>	<b>FLOAT</b>	Текущее значение аргумента и функции
<b><math>T_0, T_{max}</math></b>	<b>T0, TMAX</b>	<b>REAL</b>	<b>FLOAT</b>	Начальное и конечное значение аргумента $T$ (осадка порожнем и полная осадка судна)
<b><math>T_{cp}</math></b>	<b>TCR</b>	<b>REAL</b>	<b>FLOAT</b>	Среднее значение аргумента на интервале $T_0 < T_{cp} < T_{max}$
<b><math>\Delta T</math></b>	<b>DT</b>	<b>REAL</b>	<b>FLOAT</b>	Шаг изменения аргумента $T$

Алгоритм циклической структуры. Параметр цикла – переменная  $T$  (осадка судна). Вывод результатов осуществляется в виде таблицы текущих значений аргумента  $T$  (осадки) и функции  $G$  (груза).

### 3. Макет печати результатов

$T = 0.2$	$G = 0.0$
$T = 0.4$	$G = 199.41$
$T = 0.6$	$G = 398.8201$
$T = 0.8$	$G = 598.23$
$T = 1.0$	$G = 797.6401$
$T = 1.2$	$G = 997.05$

### Задания для лабораторной работы 4.3

Груз перевозится по реке. Коэффициент полноты водоизмещения изменяется при осадке больше  $T_{cp}$  ( $T > T_{cp}$ ). Рассчитать грузовую шкалу для следующих судов.

Исходные данные								
№	L	B	$T_0$	$T_{max}$	$T_{cp}$	$\Delta T$	$\sigma_0$	$\sigma$
1	85	13,8	0,5	2,5	1,5	0,2	0,66	0,85
2	90	13	0,8	2,4	1,6	0,2	0,68	0,74
3	85	12,6	0,7	2,5	1,5	0,2	0,66	0,74
4	88	12,6	0,5	2,5	1,5	0,25	0,68	0,78
5	90	12	1	5	2,6	0,4	0,68	0,8
6	84	12,2	0,8	2,8	1,8	0,25	0,69	0,86
7	120	16	0,6	2,8	1,6	0,2	0,62	0,82
8	130	16,5	0,9	2,5	1,5	0,2	0,7	0,84
9	110	13,3	1,1	3,5	2	0,3	0,68	0,78
10	114	13	1,2	6	3,2	0,4	0,68	0,8
11	112	13	0,9	4,2	2,4	0,3	0,66	0,82
12	90	12,6	0,8	2,6	1,8	0,2	0,68	0,81
13	92	14	0,7	2,5	1,5	0,2	0,66	0,82
14	102	12,4	0,6	2,6	1,4	0,2	0,68	0,88
15	96	13	1,2	3,4	2,2	0,2	0,7	0,82

### Лабораторная работа 4.4

#### Вычисление координат точки

#### Тема. Одномерные массивы: min, max

Математические функции – табл. 5.1 (стр.136)

Примеры выполнения лабораторных работ – стр.43-55

Составить схему и программу решения для следующей задачи:

На плоскости  $(x, y)$  заданы точки  $\{M_1(X_1, Y_1), \dots, M_n(X_n, Y_n)\}$ , точка  $Q(X_q, Y_q)$ , а также круг радиуса  $R$  с центром в начале координат. Определить координаты точки  $M(X_k, Y_k)$  в соответствии с вариантом,  $k$  – порядковый номер точки, количество точек  $n = 10$ .

**Для выполнения задания необходимо:**

- задать координаты точек  $\{M_1(X_1, Y_1), \dots, M_n(X_n, Y_n)\}$  и  $Q(X_q, Y_q)$ , а также радиус круга  $R$  в соответствии с условием задачи.
- представить точки и круг на плоскости  $(X; Y)$ .
- найти и обозначить на графике искомую точку  $M(X_k, Y_k)$ .

#### **Задания для лабораторной работы 4.4**

При решении используется алгоритм вычисления наибольшего (наименьшего) значения. **Определить координаты точки:**

(п.1 – п.3)

**С наименьшей (наибольшей) абсциссой (ординатой) и лежащей:**

№	Задача
1	Вне круга и в одном квадранте с точкой Q
2	В одной полуплоскости (левой или правой) с точкой Q и внутри круга
3	Вне круга и в одной полуплоскости (верхней или нижней) с точкой Q

(п.4 – п.6)

**Наиболее удалённой от начала координат (ближайшей к нему) и лежащей:**

4	Внутри круга и в одном квадранте с точкой Q
5	В одной полуплоскости (верхней или нижней) с точкой Q и вне круга
6	Внутри круга и в одной полуплоскости (левой или правой) с точкой Q

(п.7 – п.11)

**Ближайшей к точке Q  $(X_q, Y_q)$  и лежащей:**

7	Вне круга и в одном квадранте с точкой Q
8	В первом квадранте и внутри круга
9	В верхней полуплоскости и вне круга
10	Во втором или четвёртом квадранте и внутри круга
11	В третьем квадранте и вне круга

(п.12 – п.15)

**Наиболее удалённой от точки Q  $(X_q, Y_q)$  и лежащей:**

12	Во втором квадранте и вне круга
13	Внутри круга и в одном квадранте с точкой Q
14	В четвёртом квадранте и вне круга
15	В правой полуплоскости и внутри круга

**Лабораторная работа 4.5**  
**Вычисление площади живого сечения**  
**однорукавного русла**

**Тема. Одномерные массивы: вычисление суммы**

Порядок выполнения лабораторной работы – раздел 1 (стр.4)

**1. Задание**

Составить схему и программу решения следующей задачи:

Для проектирования выправительных сооружений и дноуглубительных работ при выполнении расчётов возникает необходимость вычисления площади живого сечения русла по экспериментальным данным – результатам измерений. Вычисление площади сечения осуществляется методом трапеции по формуле:

$$S = \sum_{i=1}^{m-1} \frac{(h_i + h_{i+1})}{2} \cdot (x_{i+1} - x_i), \text{ где } m - \text{ число замеров (точек);}$$

Результаты измерений представлены в виде массивов  $x_i$  и  $h_i$  ( $i = 1, \dots, m$ ),  $m = 8$ .

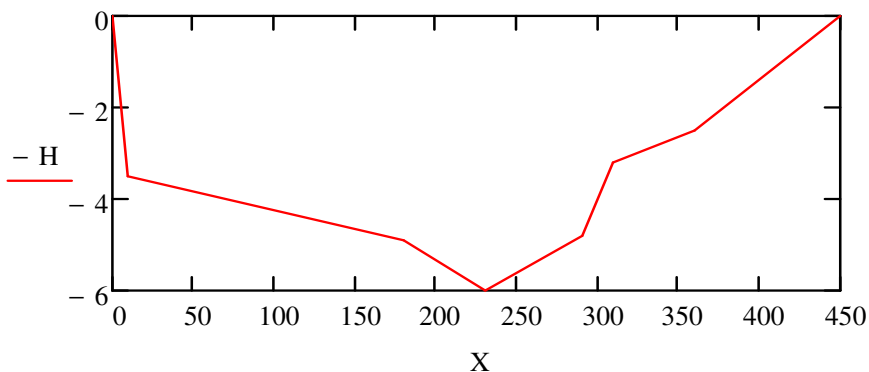
$x_i$  – расстояние от берега до  $i$ -ой точки измерения;

$h_i$  – глубина в  $i$ -ой точке.

**Исходные данные**

<b>Координаты точек</b>							
<b>Расстояние от берега до <math>i</math>-ой точки измерения</b>							
$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	$X_6$	$X_7$	$X_8$
0	10	180	230	290	310	360	450
<b>Глубина в <math>i</math>-ой точке измерения</b>							
$H_1$	$H_2$	$H_3$	$H_4$	$H_5$	$H_6$	$H_7$	$H_8$
0	3,5	4,9	6	4,8	3,2	2,5	0

**График сечения дна русла**



## 2. Решение

Обозначения переменных			
Имя переменной	Тип переменной		Пояснение
	Fortran	C	
$M, i$	<b>INTEGER</b>	<b>INT</b>	Количество точек измерений и номер точки
$X_i$	<b>REAL</b>	<b>FLOAT</b>	Расстояние от берега до $i$ -ой точки
$H_i$	<b>REAL</b>	<b>FLOAT</b>	Глубина в $i$ -ой точке измерения
$S$	<b>REAL</b>	<b>FLOAT</b>	Сумма, площадь сечения русла

При решении используется алгоритм вычисления суммы. Параметр цикла  $i$  (номер точки) изменяется в интервале от 1 до 7 с шагом 1. В цикле осуществляется ввод-вывод данных, а также вычисление суммы  $S$ .

## 3. Макет печати

ИСХОДНЫЕ ДАННЫЕ							
ЧИСЛО ТОЧЕК ИЗМЕРЕНИЙ $M = 8$							
КООРДИНАТЫ ТОЧЕК ИЗМЕРЕНИЙ							
0	10	180	230	290	310	360	450
0	3.5	4.9	6	4.8	3.2	2.5	0
ПЛОЩАДЬ СЕЧЕНИЯ РУСЛА = 1663							

## Задания для лабораторной работы 4.5

По результатам замеров вычислить площадь сечения русла. Число точек измерения  $m = 8$ .

№	Координаты $X_i$	Глубины $H_i$
1	0 100 130 180 240 250 310 390	0,0 1,8 1,1 3,5 2,8 2,0 3,5 0,0
2	0 10 50 100 150 210 270 330	0,0 3,2 1,3 2,9 2,8 2,7 4,0 0,0
3	0 90 130 150 170 180 230 240	0,0 2,9 3,6 2,0 2,8 2,7 3,1 0,0
4	0 10 80 140 160 200 240 260	0,0 1,5 3,6 2,5 2,3 1,4 1,4 0,0
5	0 30 120 220 260 290 330 380	0,0 1,7 3,2 3,9 2,8 3,1 2,3 0,0
6	0 10 90 150 180 210 250 350	0,0 1,3 3,0 2,3 1,4 1,6 2,5 0,0
7	0 20 80 160 230 240 260 350	0,0 1,2 1,8 3,1 3,7 1,4 2,2 0,0
8	0 40 90 130 200 250 260 310	0,0 2,3 3,6 1,8 1,6 1,8 2,9 0,0
9	0 90 180 260 310 360 460 560	0,0 1,3 2,7 1,6 1,4 3,7 1,9 0,0
10	0 90 110 180 230 310 390 400	0,0 1,7 2,7 1,4 2,0 3,6 3,9 0,0
11	0 10 80 80 130 190 190 250	0,0 2,2 2,0 3,0 2,8 1,6 3,6 0,0
12	0 50 110 190 220 250 320 410	0,0 2,8 2,6 3,2 1,5 2,6 2,0 0,0
13	0 80 140 200 260 280 360 460	0,0 2,1 1,9 2,6 1,9 3,6 4,0 0,0
14	0 40 130 150 170 270 320 320	0,0 1,7 2,9 1,7 1,6 3,2 1,7 0,0
15	0 80 170 180 270 350 370 390	0,0 1,9 2,2 3,0 2,2 1,9 3,5 0,0

**Лабораторная работа 4.6**  
**Вычисление границ искомого интервала**  
**Тема. Двумерный массив (матрица)**

Порядок выполнения лабораторной работы – стр.4,120

Примеры выполнения лабораторных работ – стр.70-90

**Для выполнения лабораторной работы необходимо:**

- задать квадратную матрицу  $A(n \times n)$ ,  $n = 8$ . Численные значения элементов  $A_{ij}$ , где  $i, j$  – номер строки и столбца, следует задавать не случайным образом, а подбирать с учётом смысла и условия задачи, что очень важно;
- выполнить ручной расчёт результатов и для контроля вывести их на печать, включая промежуточные.
- составить схему алгоритма и программу решения задачи.

**Внимание!** Выбор границ интервала  $[X_0; X_k]$  осуществляется проверкой условия  $X_0 < X_k$ , что необходимо предусмотреть при составлении схемы и программы решения задачи. При этом значение  $X_0$  должно быть существенно меньше  $X_k$ , а длина интервала  $(X_k - X_0)$  достаточной, так как по условию задачи в матрице должны быть элементы, лежащие внутри (вне) интервала.

**Примечание.** При выборе чётной строки/столбца номер  $i, j$  изменяется в интервале  $[2, n]$ , а нечётной – в интервале  $[1, n - 1]$  с шагом 2.

При решении задач используются алгоритмы вычисления суммы, произведения, количества, наибольшего (наименьшего) значения.

**Задания для лабораторной работы 4.6**

*Определить:*

№	Задача
1	Номера четного столбца с наибольшим количеством положительных и нечётного столбца с наименьшим количеством отрицательных элементов. Эти значения задают интервал $[X_0; X_k]$ . Найти сумму элементов верхнего треугольника, лежащих вне интервала
2	Наибольший из элементов чётных и наименьший из элементов нечётных строк матрицы. Эти значения задают интервал $[X_0; X_k]$ . Найти произведение элементов нижнего треугольника, лежащих внутри интервала



№	Задача
3	Номера строки наибольшего из элементов верхнего и столбца наименьшего из элементов нижнего треугольника. Эти значения задают интервал $[X_0; X_k]$ . Найти сумму элементов главной диагонали, лежащих внутри интервала
4	Наименьший из элементов нечётных и наибольший из элементов чётных столбцов матрицы. Эти значения задают интервал $[X_0; X_k]$ . Найти количество положительных элементов верхнего треугольника, лежащих вне интервала
5	Номера строк с наибольшим и наименьшим из элементов главной диагонали. Эти значения задают интервал $[X_0; X_k]$ . Найти количество элементов нижнего треугольника, лежащих внутри интервала
6	Количество положительных и отрицательных элементов главной диагонали. Эти значения задают интервал $[X_0; X_k]$ . Найти наибольший из элементов верхнего треугольника, лежащих вне интервала
7	Наибольший и наименьший элементы главной диагонали. Эти значения задают интервал $[X_0; X_k]$ . Найти сумму положительных элементов нижнего треугольника, лежащих внутри интервала
8	Наименьший из положительных элементов нижнего и наибольший из отрицательных элементов верхнего треугольника. Эти значения задают интервал $[X_0; X_k]$ . Найти количество элементов главной диагонали, лежащих вне интервала
9	Наибольший элемент верхнего и наименьший элемент нижнего треугольника. Эти значения задают интервал $[X_0; X_k]$ . Найти среднее арифметическое элементов матрицы, лежащих внутри интервала
10	Наименьший из элементов главной диагонали с нечётными номерами и наибольший из них с чётными номерами. Эти значения задают интервал $[X_0; X_k]$ . Найти среднее арифметическое элементов верхнего треугольника, лежащих вне интервала

## 5. Справочные материалы

Существенную помощь в вычислениях с вещественными данными оказывают математические функции. Список некоторых наиболее употребительных функций приведён в таблице 5.1.

**Примечание.** На Fortran'e данные функции и их аргументы имеют тип **REAL**, на C – тип **double**, например для  $\ln x$  – **double log (double x)**.

**Названия стандартных функций в C следует набирать только строчными буквами!**

**Таблица 5.1**  
**Стандартные функции C, Fortran, Basic**

Функция		Запись на		
		C	Fortran	Basic
Натуральный логарифм	$\ln x$	log(x)	LOG(X)	LOG(X)
Десятичный логарифм	$\lg x$	log10(x)	LOG10(X)	
Квадратный корень	$\sqrt{x}$	sqrt(x)	SQRT(X)	SQR(X)
Абсолютная величина (модуль)	$ x $	abs(x)	ABS(X)	ABS(X)
Экспонента	$e^x$	exp(x)	EXP(X)	EXP(X)
<b>Тригонометрические функции (аргумент в радианах)</b>				
Синус	$\sin x$	sin(x)	SIN(X)	SIN(X)
Косинус	$\cos x$	cos(x)	COS(X)	COS(X)
Тангенс	$\operatorname{tg} x$	tan(x)	TAN(X)	TAN(X)
<b>Обратные тригонометрические функции (аргумент <math> x  \leq 1</math>)</b>				
Арксинус	$\arcsin x$	asin(x)	ASIN(X)	
Арккосинус	$\arccos x$	acos(x)	ACOS(X)	
Арктангенс	$\operatorname{arctg}(x)$	atan(x)	ATAN(X)	ATN(X)
<b>Гиперболические функции</b>				
Синус	$\operatorname{sh} x$	sinh(x)	SINH(X)	
Косинус	$\operatorname{ch} x$	cosh(x)	COSH(X)	
Тангенс	$\operatorname{th} x$	tanh(x)	TANH(X)	

**Примечание.** Гиперболические функции  $\operatorname{sh} x$  и  $\operatorname{ch} x$  выражаются через

показательную функцию: 
$$\operatorname{sh} x = \frac{e^x - e^{-x}}{2}, \quad \operatorname{ch} x = \frac{e^x + e^{-x}}{2}.$$

**Пример**

Запись арифметического выражения

$$1.5 \cdot \operatorname{ch}^4 x + \sqrt[3]{|x|} \cdot e^{-x} - \sin \sqrt{\ln 3} + \operatorname{tg} 3^\circ$$

на языках C, Fortran и Basic соответственно:

**C:** набирается строчными буквами

```
1.5*pow(cosh(x), 4) + pow(abs(x), (1./3.)) * exp(-x) -
sin(sqrt(log(3.))) + tan(3*3.1415/180)
```

**Fortran:**

```
1.5*COSH(X)**4 + ABS(X)**(1./3.)*EXP(-X) -
SIN(SQRT(LOG(3.))) + TAN(3*3.1415/180)
```

**Basic:**

```
1.5*(EXP(X) + EXP(-X))^4 + ABS(X)^(1/3)*EXP(-X) -
SIN(SQR(LOG(3))) + TAN(3*3.1415/180)
```

Примеры записи выражений приведены также в работах 1; 4-6.

# Отчёт-презентация к лабораторной работе 1

## Вычисление значения выражения

### 1. Задание

Вычислить значение величины Z:

$$Z = (\ln |b| + \cos a)^2 - \frac{\sqrt{\ln 3 + \cos b}}{a (\ln a + \cos 2^\circ)}, \text{ где } a = 1,5 \quad b = -2.$$

Для удобства написания программы введем обозначения:

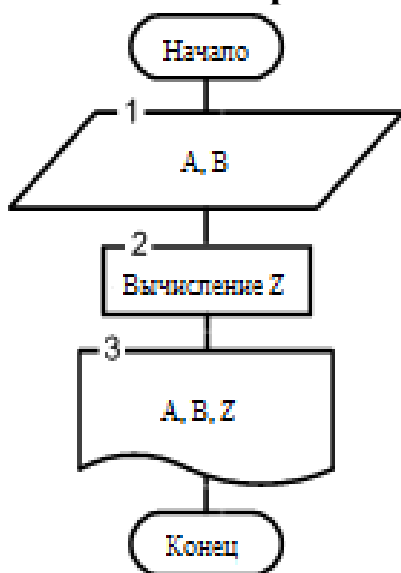
$$R = \ln |B| + \cos A; \quad W = \ln 3 + \cos B; \quad Q = \ln A + \cos 2^\circ.$$

В таком случае вычисление Z можно представить в виде:

$$Z = R^2 - \frac{\sqrt{W}}{A \cdot Q}$$

Обозначения переменных			
Имя переменной	Тип переменной		Пояснение
	Fortran	C	
<b>A, B</b>	<b>REAL</b>	<b>FLOAT</b>	Исходные данные
<b>R, W, Q, Z</b>	<b>REAL</b>	<b>FLOAT</b>	Рабочие переменные и результат

### 2. Схема алгоритма



### 3. Листинг Basic-программы

Программный код
' Исходные данные A = 1.5: B = -2 PI = 3.1415 ' Вычисление Z R = LOG (ABS (B)) + COS (A) W = LOG (3) + COS (B) Q = LOG (A) + COS (2*PI/180) Z = R^2 - SQR (W) / (A*Q) ' Вывод значений A, B, Z PRINT " A = "; A; PRINT " B = "; B PRINT " Z = "; Z END

### 4. Макет печати

```

A = 1.5 B = -2
Z = 0.1915
  
```

## 6. Ввод массива с датчиком случайных чисел

Ввод данных можно поручить датчику случайных чисел.

### Задание

Сформировать матрицу  $A(n \times n)$  и массивы  $X(n)$ ,  $Y(n)$ ,  $n=3$ . Для заполнения массива случайными числами необходимо задать диапазон  $[A, B]$ , в границах которого будут находиться эти числа. На языке C выбор чисел осуществляется по формуле:

$$A + \text{rand}() \% (B - A)$$

Приведём программу ввода-вывода элементов массивов в интервале  $[-10; 10]$  на языках C и Basic.

### Листинг C-программы

**Внимание.** C различает регистр букв, т.е., «x» и «X» – это две разные переменные. Нижняя граница индексов  $i$  и  $j$  задаётся равной нулю.

Ввод-вывод матрицы A	Ввод-вывод массива X, Y
<pre>#include&lt;stdio.h&gt; #include&lt;conio.h&gt; #include&lt;stdlib.h&gt; #include&lt;time.h&gt; void main() { int i, j, n = 3; float a[3][3]; srand(time(0)); // Внешний цикл по i for(i = 0; i&lt;n; i++) { // Внутренний цикл по j for(j = 0; j&lt;n; j++) { a[i][j]=-10+rand()%20; printf("%4.0f ",a[i][j]); } printf(" \n"); } getch(); return; }</pre>	<pre>#include&lt;stdio.h&gt; #include&lt;conio.h&gt; #include&lt;stdlib.h&gt; #include&lt;time.h&gt; void main() { int i, n = 3; float x[3], y[3]; srand(time(0)); // Цикл по i for(i = 0; i&lt;n; i++) { x[i]= -10 + rand()%20; y[i]= -10 + rand()%20; printf("%.0f ",x[i]); printf("%.0f \n",y[i]); } getch(); return; }</pre>

Результаты работы программы приведены в макете печати.

Задание чисел из диапазона [A;B] на языке Basic осуществляется по формуле (для целых чисел):

$$\text{INT}(A + (B - A) \cdot \text{RND})$$

### Листинг Basic-программы

Программный код	
Ввод-вывод матрицы A	Ввод-вывод массива X,Y
<pre>CLS N = 3 RANDOMIZE TIMER DIM A(N,N) PRINT "МАТРИЦА A" FOR I = 1 TO N   FOR J = 1 TO N     A(I,J)=INT(-10 + 20*RND)     PRINT USING "##",A(I,J);   NEXT J PRINT ' Пропуск строки NEXT I END</pre>	<pre>CLS N = 3 RANDOMIZE TIMER DIM X(N),Y(N) PRINT TAB(3); " X Y" FOR I = 1 TO N   X(I)=INT(-10 + 20*RND)   Y(I)=INT(-10 + 20*RND)   PRINT USING "##",X(I);   PRINT USING "##",Y(I) NEXT I END</pre>

### Макет печати результатов (по Basic-программе)

Матрица A		
-4	7	-5
8	6	-6
-7	-8	2

X	Y
8	-3
7	-6
2	8

Функция **srand(time(0))** и оператор **RANDOMIZE TIMER** нужны для того, чтобы каждый раз при выполнении программы создавалась новая последовательность псевдослучайных чисел.

Ручной расчёт результатов возможен только после выполнения программы, в которой необходимо заранее предусмотреть вывод искомого массива.

## Библиографический список

1. **Артемов, И.Л.** Fortran: основы программирования / И.Л. Артемов. – М. : ДИАЛОГ-МИФИ, 2007. – 304 с.
2. **Бартедьев, О.В.** Современный Фортран / О.В. Бартедьев. – 4-е изд, доп. и перераб. – М. : ДИАЛОГ-МИФИ, 2005. – 560 с.
3. **Васильев А.Н.** Самоучитель С++ с примерами и задачами / А.Н. Васильев. – 2-е изд., перераб. – СПб. : Наука и техника, 2012. – 480 с.
4. **ГОСТ 19.701–90 (ИСО 5807–85).** Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения. – Взамен ГОСТ 19.002–80, ГОСТ 19.003–80 ; введ. 01.01.92. – М. : Изд-во стандартов, 1998. – 24 с.
5. **Гурьяшова, Р.Н.** Информатика : Сегментация программ : учеб. пособие для студ. оч. и заоч. обуч. технич. специальностей / Р.Н. Гурьяшова, А.В. Шеянов. – Н. Новгород : Изд-во ФГОУ ВПО «ВГАВТ», 2009. – 108 с.
6. **Гурьяшова, Р.Н.** Информатика. Теоретический курс : учеб. пособие для студ. всех форм обуч. спец-ти 280700.62 «Техносферная безопасность» / Р.Н. Гурьяшова, В.И. Логинов, Е.Ю.Седова. – Н. Новгород : Изд-во ФГОУ ВПО «ВГАВТ», 2013. – 84 с.
7. **Кетков, Ю.Л.** Практика программирования: Бейсик, Си, Паскаль / Ю.Л. Кетков, А.Ю. Кетков. – СПб. : БХВ, 2001. – 480 с.
8. **Культин, Н.Б.** С/С++ в задачах и примерах / Н.Б. Культин. – 2-е изд., перераб. и доп. – СПб. : БХВ-Петербург, 2011. – 368 с.
9. **Ляхович, В.Ф.** Основы информатики : учеб. / В.Ф. Ляхович, С.О. Крамаров. – 6-е изд., доп. и перераб. – Ростов н/Д : Феникс, 2007. – 713 с.
10. **Немнюгин, С.А.** Фортран в задачах и примерах / С.А. Немнюгин, О.Л. Стесик. – СПб. : БХВ-Петербург, 2008. – 320 с.
11. **Подбельский, В.В.** Программирование на языке Си / В.В. Подбельский, С.С. Фомин. – 2-е изд. – М. : Финансы и статистика, 1999. – 600 с.
12. **Рыжиков, Ю.И.** Современный Фортран : учеб. / Ю.И. Рыжиков. – СПб. : КОРОНА принт, 2007. – 288 с.

## Оглавление

<b>ВВЕДЕНИЕ</b> .....	3
<b>1. ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ</b> .....	4
<b>2. ПРИМЕРЫ ВЫПОЛНЕНИЯ ЛАБОРАТОРНЫХ РАБОТ</b> .....	5
ЛАБОРАТОРНАЯ РАБОТА 1. Вычисление значения выражения.....	5
ЛАБОРАТОРНАЯ РАБОТА 2. Вычисление значения величины, заданной в форме ветвления .....	8
ЛАБОРАТОРНАЯ РАБОТА 3. Вычисление значения величины, заданной в форме ветвления .....	11
<b>КОНТРОЛЬНЫЕ ЗАДАНИЯ ДЛЯ ПРОВЕРКИ ЗНАНИЙ</b> .....	15
ЛАБОРАТОРНАЯ РАБОТА 4. Исследование функции одной переменной .....	17
ЛАБОРАТОРНАЯ РАБОТА 5. Исследование функции одной переменной, зависящей от параметра .....	23
ЛАБОРАТОРНАЯ РАБОТА 6. Исследование функции одной переменной с различным шагом варьирования аргумента.....	29
<b>КОНТРОЛЬНОЕ ЗАДАНИЕ ДЛЯ ПРОВЕРКИ ЗНАНИЙ. РАСЧЁТ ОРДИНАТ ЭПЮРЫ ДАВЛЕНИЯ ГРУНТА НА ПОДПОРНЫЕ СТЕНКИ</b> ...	35
ЛАБОРАТОРНАЯ РАБОТА 7. Исследование функции двух переменных .....	36
ЛАБОРАТОРНАЯ РАБОТА 8. Исследование функции двух переменных при фиксированном значении аргумента .....	41
ЛАБОРАТОРНАЯ РАБОТА 9. Вычисление координат искомой точки (наиболее удалённой от начала координат / ближайшей к нему) .....	45
ЛАБОРАТОРНАЯ РАБОТА 10. Вычисление координат искомой точки (ближайшей к заданной / наиболее удалённой от неё) .....	50
ЛАБОРАТОРНАЯ РАБОТА 11. Вычисление координат искомой точки .....	54
ЛАБОРАТОРНАЯ РАБОТА 12. Вычисление значения выражения.....	58
ЛАБОРАТОРНАЯ РАБОТА 13. Анализ эффективности работы предприятия .....	61
ЛАБОРАТОРНАЯ РАБОТА 14. Анализ итогов сдачи сессии.....	66
<b>КОНТРОЛЬНОЕ ЗАДАНИЕ ДЛЯ ПРОВЕРКИ ЗНАНИЙ. ВЫЧИСЛЕНИЕ ПЛОЩАДИ ШПАНГОУТА</b> .....	70
ЛАБОРАТОРНАЯ РАБОТА 15. Анализ эффективности работы танкерного флота .....	72
ЛАБОРАТОРНАЯ РАБОТА 16. Анализ итогов сессии в группе.....	83
ЛАБОРАТОРНАЯ РАБОТА 17. Формирование одномерного массива .....	87
<b>КОНТРОЛЬНОЕ ЗАДАНИЕ ДЛЯ ПРОВЕРКИ ЗНАНИЙ</b> .....	92

<b>3. ЗАДАНИЯ НА ВЫПОЛНЕНИЕ ЛАБОРАТОРНЫХ РАБОТ.....</b>	<b>94</b>
ТАБЛИЦА 3.1 Вычисление значения выражения.....	94
ТАБЛИЦА 3.2	
Вычисление значения величины, заданной в форме ветвления .....	96
ТАБЛИЦА 3.3	
Вычисление значения величины, заданной в форме ветвления .....	99
ТАБЛИЦА 3.4 Исследование функции одной переменной .....	100
ТАБЛИЦА 3.5	
Исследование функции одной переменной, зависящей от параметра .....	104
ТАБЛИЦА 3.6 Исследование функции одной переменной с различным шагом варьирования аргумента .....	107
ТАБЛИЦА 3.7 Исследование функции двух переменных .....	109
ТАБЛИЦА 3.8 Исследование функции двух переменных при фиксированном значении аргумента.....	111
ТАБЛИЦА 3.9 Вычисление координат точки, наиболее удалённой от начала координат (ближайшей к нему).....	113
ТАБЛИЦА 3.10 Вычисление координат искомой точки, ближайшей к заданной (наиболее удалённой от неё) .....	115
ТАБЛИЦА 3.11 Вычисление координат искомой точки .....	116
ТАБЛИЦА 3.12 Вычисление значения выражения .....	117
ТАБЛИЦА 3.13 Анализ эффективности работы предприятия .....	119
ТАБЛИЦА 3.14 Анализ итогов экзамена в группе .....	121
ТАБЛИЦА 3.15 Анализ эффективности работы флота .....	122
ТАБЛИЦА 3.16 Анализ итогов сессии в группе.....	124
ТАБЛИЦА 3.17 Формирование одномерного массива .....	125
<b>4. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ.....</b>	<b>127</b>
ЛАБОРАТОРНАЯ РАБОТА 4.1. Вычисление значения функции.....	127
ЛАБОРАТОРНАЯ РАБОТА 4.2.	
Вычисление функции, заданной в виде ветвления .....	128
ЛАБОРАТОРНАЯ РАБОТА 4.3. Расчёт грузовой шкалы судна .....	130
ЛАБОРАТОРНАЯ РАБОТА 4.4. Вычисление координат точки .....	132
ЛАБОРАТОРНАЯ РАБОТА 4.5.	
Вычисление площади живого сечения однорукавного русла.....	134
ЛАБОРАТОРНАЯ РАБОТА 4.6. Вычисление границ искомого интервала .....	134
<b>5. СПРАВОЧНЫЕ МАТЕРИАЛЫ.....</b>	<b>136</b>
ТАБЛИЦА 5.1. СТАНДАРТНЫЕ ФУНКЦИИ C, FORTRAN, BASIC .....	138
ОТЧЁТ-ПРЕЗЕНТАЦИЯ К ЛАБОРАТОРНОЙ РАБОТЕ 1.....	139
<b>6. ВВОД МАССИВА С ДАТЧИКОМ СЛУЧАЙНЫХ ЧИСЕЛ .....</b>	<b>136</b>
<i>БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....</i>	<i>140</i>



*Гурьяшова Римма Николаевна*

**Лабораторный практикум  
по информатике**

Учебное пособие

*Отпечатано без издательского редактирования и корректирования*

Подписано в печать 16.03.15.  
Формат 60×84 <sup>1</sup>/<sub>16</sub>. Гарнитура «Таймс».  
Ризография. Усл. печ. л. 8,6. Тираж 80 экз. Заказ 190.

---

Издательско-полиграфический комплекс ФГБОУ ВО «ВГУВТ»  
603950, Нижний Новгород, ул. Нестерова, 5

## Требования к содержанию и оформлению отчета по лабораторным работам

Отчет должен содержать:

- 1) Стандартным образом оформленный титульный лист.
- 2) К каждой лабораторной работе:
  - а) формулировку задания;
  - б) блок-схему алгоритма;
  - в) листинг программы;
  - г) скриншот результатов работы программы.

Примерный перечень лабораторных работ:

1. Разветвленные вычислительные процессы (задача 1, задача 2, выполненные на языке Си).
2. Циклические вычислительные процессы. Исследование функции одной переменной (табулирование функции одной переменной).
3. Циклические вычислительные процессы. Обработка данных одномерных массивов (задача 1, задача 2, выполненные на языке Си).
4. Циклические вычислительные процессы. Обработка данные двумерных массивов (задача 1, задача 2, выполненные на языке Си).
5. Программирование пользовательских функций.

Примечание. Все листы отчета должны быть оформлены в простую рамку.

### ***Внимание!***

1. Отчет по лабораторным работам должен быть готов ко дню проведения зачетного занятия!
2. Перед тем как распечатать отчет, продемонстрируйте его преподавателю в электронном виде, а затем устраните замечания!

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**

**Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)**  
ул. Нестерова, 5а, Нижний Новгород,  
603951  
Тел. (831)419 – 79 – 51

Кафедра «Систем информационной  
безопасности, управления и  
телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматики»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №1**

1. Информационно-телекоммуникационные технологии глобальной компьютерной сети Интернет.
2. Ввод-вывод данных в Си. Форматный ввод-вывод.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**

**Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)**  
ул. Нестерова, 5а, Нижний Новгород,  
603951  
Тел. (831)419 – 79 – 51

Кафедра «Систем информационной  
безопасности, управления и  
телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматики»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №2**

1. Структура и принципы функционирования глобальной сети Интернет.
2. Различие между постфиксной и префиксной операцией инкремента(декремента).

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**

**Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)**  
ул. Нестерова, 5а, Нижний Новгород,  
603951  
Тел. (831)419 – 79 – 51

Кафедра «Систем информационной  
безопасности, управления и  
телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматики»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №3**

1. Основы технологии искусственного интеллекта.
2. Операции присваивания в C++.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**

**Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)**  
ул. Нестерова, 5а, Нижний Новгород,  
603951  
Тел. (831)419 – 79 – 51

Кафедра «Систем информационной  
безопасности, управления и телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматики»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №4**

1. Адресация в сетях.
2. Организация подпрограмм-функций. Рекурсия.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**

**Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)**  
ул. Нестерова, 5а, Нижний Новгород,  
603951  
Тел. (831)419 – 79 – 51

Кафедра «Систем информационной  
безопасности, управления и  
телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматики»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №5**

1. Технологии передачи данных. Протоколы.
2. Условный оператор. Примеры.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**

**Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)**  
ул. Нестерова, 5а, Нижний Новгород,  
603951  
Тел. (831)419 – 79 – 51

Кафедра «Систем информационной  
безопасности, управления и  
телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматики»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №6**

1. Информационные технологии программирования на примере языка Си в среде MS Visual Studio 2010.
2. Оператор множественного выбора. Примеры.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**  
Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)  
ул. Нестерова, 5а, Нижний Новгород,  
603951  
Тел. (831)419 – 79 – 51

Кафедра «Систем информационной  
безопасности, управления и  
телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматики»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №7**

1. Алгоритмы. Типы алгоритмов. Способы записи алгоритмов.
2. Оператор цикла for. Примеры.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**  
Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)  
ул. Нестерова, 5а, Нижний Новгород,  
603951  
Тел. (831)419 – 79 – 51

Кафедра «Систем информационной  
безопасности, управления и  
телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматики»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №8**

1. Типовые алгоритмические конструкции. Алгоритмы вычисления суммы, произведения, количества, минимума и максимума.
2. Оператор цикла while. Примеры.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**  
Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)  
ул. Нестерова, 5а, Нижний Новгород,  
603951  
Тел. (831)419 – 79 – 51

Кафедра «Систем информационной  
безопасности, управления и  
телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматизации»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №9**

1. Структура программы на языке Си.
2. Оператор цикла do while. Примеры.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**  
Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)  
ул. Нестерова, 5а, Нижний Новгород,  
603951  
Тел. (831)419 – 79 – 51

Кафедра «Систем информационной  
безопасности, управления и  
телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматизации»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №10**

1. Что такое тип данных? Простые типы данных. Описание данных. Приведение и преобразование типов.
2. Особенности использования операторов цикла. Бесконечные циклы.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**  
Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)  
ул. Нестерова, 5а, Нижний Новгород,  
603951  
Тел. (831)419 – 79 – 51

Кафедра «Систем информационной  
безопасности, управления и  
телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматики»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №11**

1. Что такое переменная? Чем объявление переменной отличается от ее определения? Привести примеры определений и объявлений.
2. Что такое массив? Обработка данных массива.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**  
Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)  
ул. Нестерова, 5а, Нижний Новгород,  
603951  
Тел. (831)419 – 79 – 51

Кафедра «Систем информационной  
безопасности, управления и  
телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматики»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №12**

1. Выражение в языке Си. Составные части выражения. Операнды.
2. Виды массивов и их инициализация. Обработка массивов.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.



**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**  
Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)  
ул. Нестерова, 5а, Нижний Новгород,  
603951  
Тел. (831)419 – 79 – 51

Кафедра «Систем информационной  
безопасности, управления и  
телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматизации»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №13**

1. Что такое отношение? В каком случае оно считается ложным, а в каком – истинным?
2. Объявление и определение функции.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**  
Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)  
ул. Нестерова, 5а, Нижний Новгород,  
603951  
Тел. (831)419 – 79 – 51

Кафедра «Систем информационной  
безопасности, управления и  
телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматизации»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №14**

1. Какие операции называются унарными? Привести примеры.
2. Логические операции. Отличие «одинарных» операций от «двойных».

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**

**Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)**

ул. Нестерова, 5а, Нижний Новгород,  
603951

Тел. (831)419 – 79 – 51

Кафедра «Систем информационной  
безопасности, управления и  
телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматизации»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №15**

1. Какие операции называются бинарными? Привести примеры.
2. Этапы создания исполняемой программы на языке Си.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

## 1.КРАТКАЯ ХАРАКТЕРИСТИКА МИКРОПРОЦЕССОРОВ МП К580

Длина слова МП К 580- 8 битов или байт. При обращении к памяти используются 16-битные адреса, что позволят обращаться к 65536 ячейкам памяти.

МП имеет 8-битный аккумулятор (регистр А ), шесть 8-битных регистров общего назначения (В, С, D, E, H, L ), а также регистр флагов F. Регистры общего назначения могут объединяться в 16-битные регистры двойной длины.

Содержимое аккумулятора и регистра флагов объединяется в «слово состояния процессора» (PSW).

В МП результат характеризуется следующими флагами:

(1- флаг установлен, 0- сброшен):

флаг знака S (SIGN); совпадает со старшим битом результата;

флаг нуля Z (ZERO); устанавливается при нулевом результате;

флаг вспомогательного переноса AC(AUXILIARY CARRY); устанавливается при возникновении переноса из третьего бита в четвертый;

флаг паритета P (PARITY); устанавливается при наличии в результате четного числа единиц;

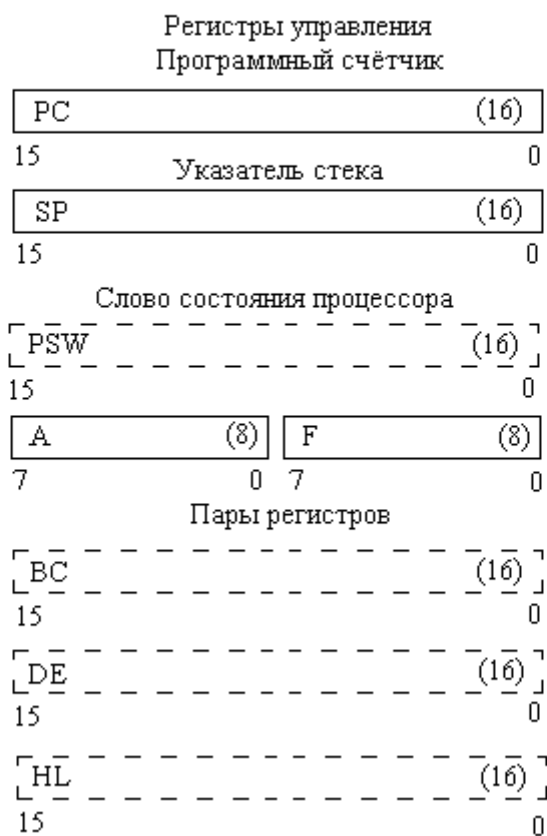
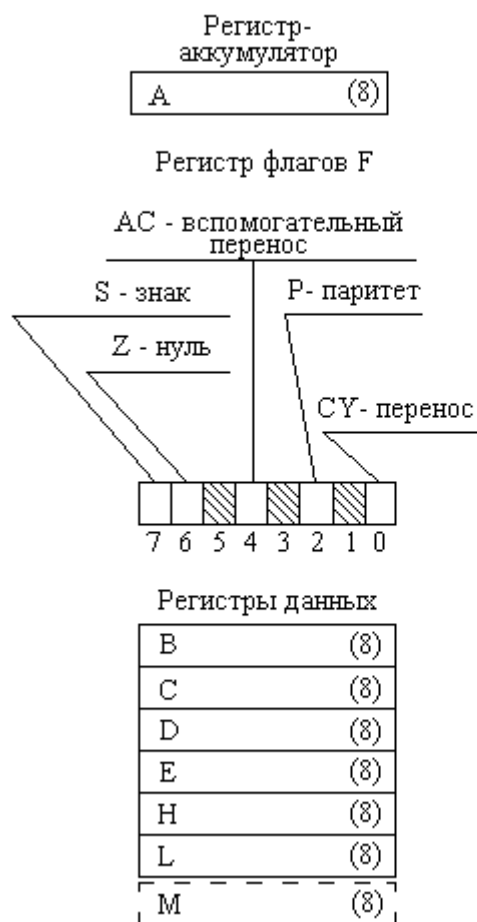
флаг переноса CA (CARRY); устанавливается при переносе из старшего бита результата (при вычитании становится флагом заема);

Команды МП по своим форматам делятся на одно-, двух- и трехбайтные.

При написании программ используются сокращенные английские наименования команд (см. приложение).

Для успешной подготовки программ необходимо представлять совокупность элементов УОУ, их назначение и взаимодействие, т.е. программистскую модель УОУ. Такая модель изображена на рис. 1.

## ВНУТРЕННИЕ РЕГИСТРЫ



## ВНЕШНИЕ РЕГИСТРЫ

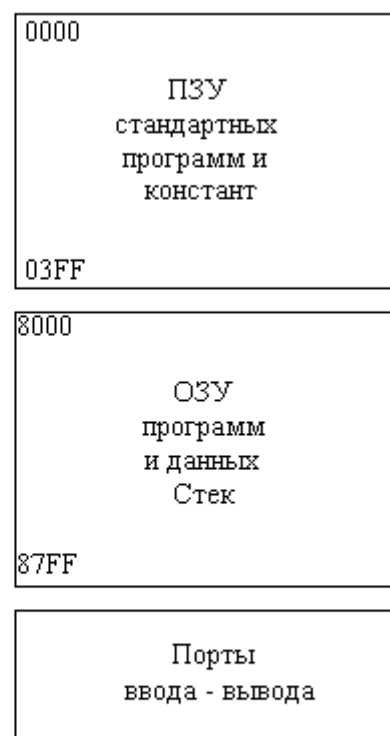


Рис. 1

## 2.СИСТЕМА КОМАНД

### 2.1 Символы и обозначения, используемые в командах МП.

Байт- 8- битное слово данных;

PC (PROGRAM COUNT)- 16- битный программный счетчик

(PC<sub>н</sub> и PC<sub>л</sub> - старший и младший байты счетчика);

SP (STACK POINTER)- 16- битный указатель стека

(SP<sub>н</sub> и SP<sub>л</sub> - старший и младший байты указателя);

SG- 16- битная вершина стека;

PSW (PROCESSOR STATUS WORD)- слово состояния

процессора;

A- регистр- аккумулятор;

r- один из регистров: B, C, D, E, H, L ;

r r- один из двойных регистров: BC, DE, HL, SP,

PSW (r<sub>гн</sub> и r<sub>гп</sub> - старший и младший байты двойного регистра) ;

M (MEMORY)- ячейка памяти, адрес которой хранится в паре регистров HL ;

v- 8-битное значение (байт данных);

vv-16- битное значение;

aa- 16- битный адрес;

S- флаг знака;

Z- флаг нуля;

P- флаг паритета;

AC- флаг вспомогательного переноса;

CY-флаг переноса;

< - операция отсылки;

- - операция обмена;

( )-в скобки заключается содержимое регистра или ячейки памяти.

## 2.1. Команды пересылки

Важно!!! Команды пересылки состояния флагов не изменяют.

Операция	Мнемоника	Кодкоманды	Действие
1	2	3	4
Формат команд	MOV r1, r2	см.	(r1) < (r2)
1 байт	MOV r, M	ниже	(r) < (M)
Пересылка данных	MOV M, r	ОА	(M) < ( )

Косвенная загрузка аккумулятора	LDAX BC	1A	(A) < ((BC))
Косвенная запись аккумулятора	LDAX DE		(A) < ((DE))
		02	((BC)) < (A)
Обмен	STAX BC	12	((DE)) < (A)
Формат команд-	STAX DE		
2 байта		EB	(HL) < (DE)
непосредствен-	XCHG	см.	(r) < v
ная пересылка	MVI, r, v	ниже	(r) < v
Формат команд	MVI M, v		
3 байта	LXI BC, vv	01	(BC) < vv
непосредствен-	LXI DE, vv	11	(DE) < vv
ная загрузка	LXI HL, vv	21	(HL) < vv
Прямая загрузка	LXI SP, vv	31	(SP) < vv
аккумулятора	LDA aa	3A	
Прямая запись аккумулятора	STA aa		(A) < (aa)
Загрузка пары регистров HL	LHLD aa	32	
Запись содержимого HL в память	SHLD aa	2A	(aa) < (A)
		22	(L) < (aa )
			(H) < (a+ 1)
			(aa) < (L)
			(aa+1) < (H)

## Коды команд пересылки

Регистр-приемник	Пересылка данных между регистрами								Непосредственная пересылка MVI r MVI M
	Регистр-источник								
	B	C	D	E	H	L	M	A	
B	40	41	42	43	44	45	46	47	06
C	48	49	4A	4B	4C	4D	4E	4F	0E
D	50	51	52	53	54	55	56	57	16
E	58	59	5A	5B	5C	5D	5E	5F	1E
H	60	61	62	63	64	65	66	67	26
L	68	69	6A	6B	6C	6D	6E	6F	2E
M	70	71	72	73	74	75	-	77	36
A	78	79	7A	7B	7C	7D	7E	7F	3E

## 2.2 Арифметические команды

Операция	Мнемоника	Действие	Флаги
1	2	3	4
<b>Формат команд–2 байта</b>	ADI <i>v</i>	$(A) \leftarrow (A) + v$	Все
Непосредственное сложение	ACI <i>v</i>	$(A) \leftarrow (A) + v + (CY)$	Все
Непосредственное сложение с переносом	SUI <i>v</i>	$(A) \leftarrow (A) - v$	
	SBI <i>v</i>	$(A) \leftarrow (A) - v - (CY)$	
Непосредственное вычитание	ADD <i>r</i>	$(A) \leftarrow (A) + (r)$	
Непосредственное вычитание с заемом	ADD <i>m</i>	$(A) \leftarrow (A) + (m)$	
	ADC <i>r</i>	$(A) \leftarrow (A) + (r) + (CY)$	

<b>Формат команд – 2 байта</b>	ADC <i>m</i>	$(A) < (A) + (m) + (CY)$	
Сложение	SUB <i>r</i>	$(A) < (A) - (r)$	
Сложение с переносом	SUB <i>m</i>	$(A) < (A) - (m)$	
Вычитание	SBB <i>r</i>	$(A) < (A) - (r) - (CY)$	
Вычитание с заемом	SBB <i>m</i>	$(A) < (A) - (m) - (CY)$	
Увеличение	INR <i>r</i>	$(r) < (r) + 1$	Z,S
Уменьшение	INR <i>m</i>	$(m) < (m) + 1$	P,AC
Сложение регистровых пар	INX <i>r r</i>	$(rr) < (rr) + 1$	Нет
Десятичное дополнение аккумулятора	DCR <i>v</i>	$(v) < (v) + 1$	Z,S
	DCR <i>m</i>	$(m) < (m) + 1$	P,AC
	DCX <i>r r</i>	$(r r) < (r r) + 1$	Нет
		$(HL) < (HL) + (rr)$	CY
	DAD <i>r r</i>	(A) приводится к BCD <sup>x)</sup>	Все
	DAR <i>r r</i>		

<sup>x)</sup>BCD(BINARY CODED DECIMAL)–формат двоично-кодированных десятичных чисел. При котором байт содержит две десятичные цифры (тетрады). Команда DAA(ее код–27)в два приема корректирует 8–битное значение аккумулятора:1).Если младшая тетрада больше 9 или установлен флаг вспомогательного переноса AC,то содержимое аккумулятора увеличивается на 6;2).Если после того старшая тетрада больше 9 или установлен флаг CY,то старшая тетрада увеличивается на 6.

### Коды арифметических операций с аккумулятором

Операция	Операция с регистром <i>r</i> или ячейкой <i>M</i>									Операция с байтом данных	
	Мнемоника	Регистр-источник(...)								Мнемоника	Код
		B	C	D	E	H	L	M	A		
Сложение	ADD...	80	81	82	83	84	85	86	87	ADI <i>v</i>	C6



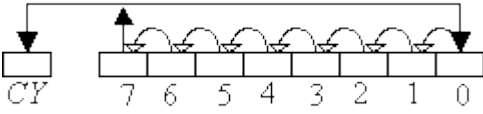
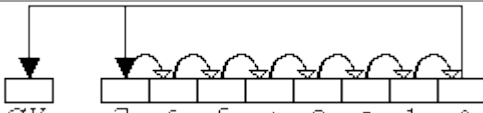
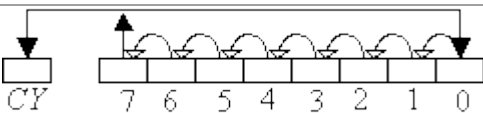
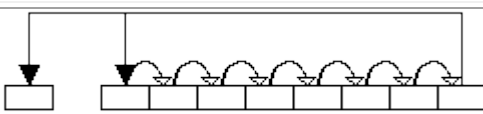
Сложение с переносом	ADC...	88	89	8A	8B	8C	8D	8E	8F	ACI <i>v</i>	CE
Вычитание	SUB...	90	91	92	93	94	95	96	97	SUI <i>v</i>	D6
Вычитание с заемом	SBB...	98	99	9A	9B	9C	9D	9E	9F	SBI <i>v</i>	DE
Увеличение	INR...	04	0C	14	1C	24	2C	34	3C	-	-
Уменьшение	DCR...	05	0D	15	1D	25	2D	35	3D	-	-

### Коды арифметических операций с двойными регистрами

Операции	Мнемоника	Двойной регистр <i>rr(...)</i>			
		BC	DE	HL	SD
Увеличение	INX...	03	13	23	33
Уменьшение	DCX...	0B	1B	2B	3B
Сложение регистровых пар	DAD...	09	19	29	39

### 2.3. Команда логических операций и сдвига

Операция (ее код)	Мнемоника	Действие	Флаги
<u>Формат команда-2 байта</u>			
Логическое "И"	ANI <i>v</i>	$(A) \leftarrow (A) \wedge r$	Z,S,P;AC=A(3);CY=0
Исключающее "ИЛИ"	XRI <i>v</i>	$(A) \leftarrow (A) \vee v$	Z,S,P;AC=CY=0
Логическое "ИЛИ"	ORI <i>v</i>	$(A) \leftarrow (A) \vee v$	
Сравнение	CPI <i>v</i>	$(A) - v$	Все
<u>Формат команда-1 байта</u>			
Логическое "И"	ANA <i>r</i>	$(A) \leftarrow (A) \wedge (r)$	Z,S,P;AC=A(3);CY=0
(поразрядная конъюнкция)	AMA <i>M</i>	$(A) \leftarrow (A) \wedge (M)$	
Исключающее	XRA <i>r</i>	$(A) \leftarrow (A) \vee (r)$	Z,S,P;AC=CY=0

“ИЛИ” (поразрядное сложение по mod 2)	XRA M	$(A) \leftarrow (A) \vee (M)$	
Логическое “ИЛИ” (поразрядная дизъюнкция)	ORA r ORA M	$(A) \leftarrow (A) \vee (r)$ $(A) \leftarrow (A) \vee (M)$	
Сравнение (без записи результата)	CMP r CMP M	$(A) - (r)$ $(A) - (M)$	Всё
Инверсия аккумулятора	CMA	$(A) \leftarrow (\bar{A})$	Нет
Инверсия переноса	CMC	$(CY) \leftarrow (\bar{CY})$	CY
Установка переноса	STC	$(CY) \leftarrow 1$	
Циклический сдвиг влево (07)	RLC		CY=A (7)
Циклический сдвиг вправо (0f)	RRC		CY=A (0)
Арифметический сдвиг влево (17)	RAL		CY=A (7)
Арифметический сдвиг вправо (1F)	RAR		CY=A (0)

### Коды логических операций

Операция	Операция с регистром <i>r</i> или ячейкой <i>M</i>									Операция с байтом данных	
	Мнемоника	Регистр-источник(...)								Мнемоника	Код
		B	C	D	E	H	L	M	A		
Логическое “И”	ANA ...	A0	A1	A2	A3	A4	A5	A6	A7	ANI <i>v</i>	E6
Исключающее	XRA ...	A8	A9	AA	AB	AC	AD	AE	AF	XRI <i>v</i>	EE

“ИЛИ”											
Логическое ”ИЛИ”	ORA ...	B0	B1	B2	B3	B4	B5	B6	B7	ORI v	F6
Сравнение	CMP ...	B8	B9	BA	BB	BC	BD	BE	BF	CPI v	FE

#### 2.4. Команды передачи управления и обращения к подпрограммам

Команды этой группы не изменяют состояние флагов, но могут использовать флаги Z, CY, P, S. Именно так команды перехода, вызова и возврата по условию проверяют по 4 пары дополняющих друг друга условий.

Обозначим (COND)- одно из 8 условий:

NZ-ненулевой результат (Z=0);

Z-нулевой результат (Z=1);

NC-отсутствие переноса из старшего разряда или заёма (CY=0);

C-наличие переноса или заёма (CY=1);

PO-нечётность числа единиц в результате (P=0);

PE -чётность числа единиц в результате (P=1);

P-“плюс” (S=0);

M-“минус” (S=1).

Команды условного и безусловного перехода и вызова содержат адрес aa и поэтому имеют трёхбайтный формат вида:

Код команды
Младший полуадрес
Старший полуадрес

Для перехода к подпрограммам и возврата из подпрограмм в МП К580 используется стек. Стек-группа последовательно пронумерованных ячеек памяти. Указатель стека (SP) постоянно хранит автоматически устанавливаемый адрес вершины стека (SP)- последней 13 занятой ячейки. 8-битное слово при записи его в стек помещается в очередную свободную ячейку стека, а при считывании из стека извлекается слово, поступившее в стек последним. Таким образом, стек автоматически реализует правило “последним пришел - первым ушел” (рис.2).

В МП К580 УОУ стек организован в виде группы ячеек, начинающийся с адреса 83E016 и расширяется по мере заполнения в сторону меньших адресов, т.е. при передаче слова памяти в стек значение SP уменьшается, а при извлечении слова из стека – увеличивается.

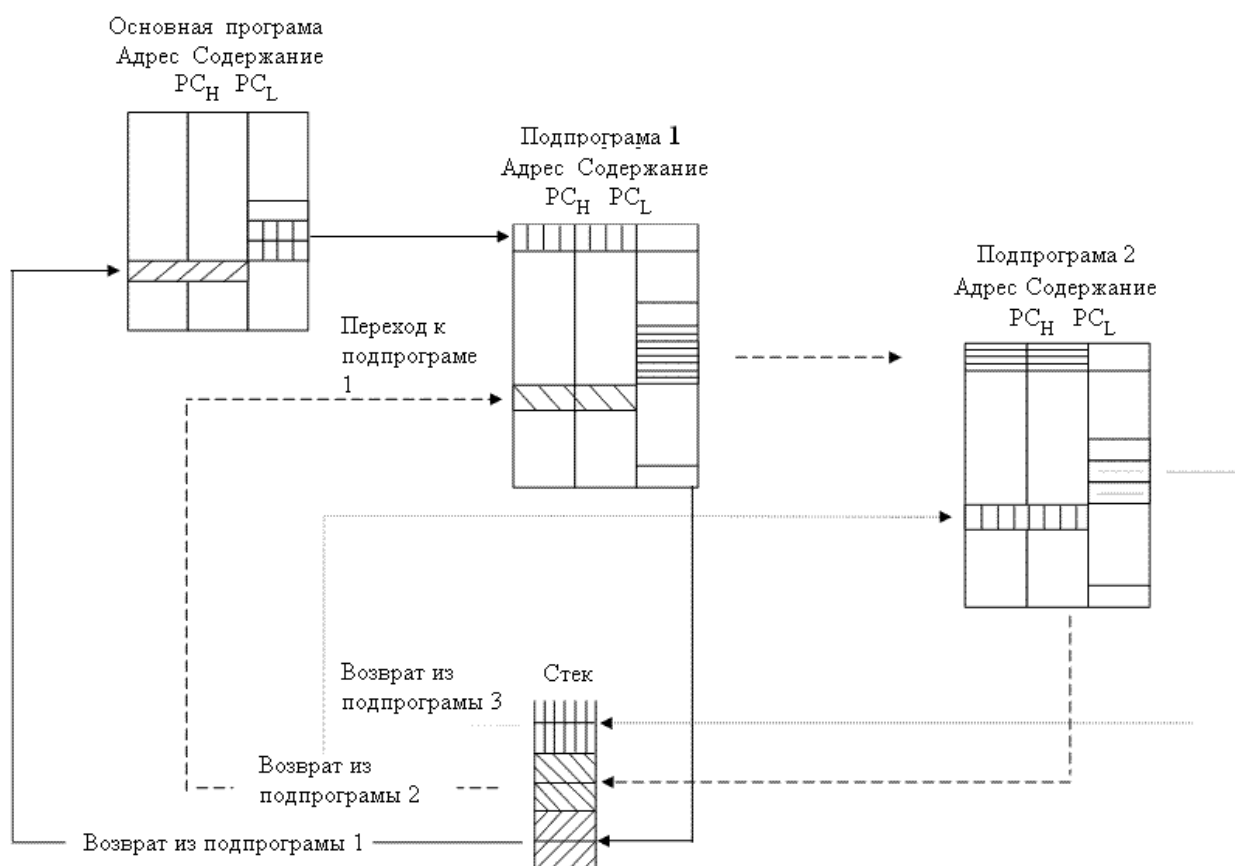
Взаимодействие указателя стека и программного счетчика (PC) при передачах управления показано ниже.

При использовании стека, глубина вложения подпрограмм зависит только от выделенной стеку памяти и практически не ограничена.

Рестарт используется для обработки прерываний и введения контрольных точек при отладке программ.

При рестарте (повторный запуск) текущее значение PC запоминается в стеке, а МП переходит к программе, начинающейся в ячейке с адресом вида  $8N(0 < N < 7)$ : 000016, 000816, 001016, ... 003816.

Особенной командой безусловной передачи управления без возврата является команда PCNL (её код – E9), по которой МП продолжает программу с адреса, хранящегося в паре регистров HL.



2. Упрощенная схема использования стека при обращениях к подпрограмме.

Операции	Мнемоника	Действие
Переход	<i>JVP aa</i>	$(PC) \leftarrow aa$
Переход по условию	<i>J(COND) aa</i>	Если условие верно, то выполняется команда <i>JMP aa</i>
Вызов	<i>CALL aa</i>	$((SP)-1) \leftarrow (PC_H), ((SP)-2) \leftarrow (PC_L),$ $(SP) \leftarrow (SP)-2, (PC) \leftarrow aa$
Вызов по условию	<i>C(COND) aa</i>	Если условие верно, то выполняется команда <i>CALL aa</i>
Возврат	<i>RET</i>	$(PC_L) \leftarrow ((SP)), (PC_H) \leftarrow ((SP)+1),$

		$(SP) \leftarrow (SP)+2$
Возврат по условию	$R(COND)$	Если условие верно, то выполняется команда $RET$
Рестарт ( $0 < N < 7$ )	$RST N$	$((SP-1) \leftarrow (PC_H), ((SP)-2) \leftarrow (PC_L),$ $(SP) \leftarrow (SP)-2, (PC) \leftarrow 8*N$
Передача из $HL$ в $PC$	$PCHL$	$(PC) \leftarrow (HL)$

#### Коды команд управления

Операция	Мнемоника	Безусловный переход	Переход по условию( $COND$ )							
			NZ	Z	NC	C	PO	PE	P	M
Вызов	CALL	CD	C4	CC	D4	DC	E4	EC	F4	FC
Переход	JMP	C3	C2	CA	D2	DA	E2	EA	F2	FA
Возврат	RET	C9	C0	C8	D0	D8	E0	E8	F0	F8

#### Коды рестарта

N	0	1	2	3	4	5	6	7
$RST N$	C7	CF	D7	DF	E7	EF	F7	FF

### 2.5. Команды управления стеком, вводом – выводом и микропроцессором

Команды этой группы состояние флагов не изменяют за исключением команды **POP PSW** восстанавливающей все флаги.

Операция	Мнемоника	Кодкоманды	Действие
Добавление в стек	PUSH rr		$((SP)-1) \leftarrow (r_{rH}),$ $((SP)-2) \leftarrow (r_{rL}),$ $(SP) \leftarrow (SP)-2$
Удаление из стека	POP rr		$(r_{rL}) \leftarrow$ $((SP)), (r_{rH}) \leftarrow$ $((SP)+1), (SP) \leftarrow$ $(SP)+2$
Обмен	XTHL	E3	$(L) \rightarrow ((SP)), (H) \rightarrow ((SP)+1)$

Передача	SPHL	F9	(SP) ← (HL)
Ввод	IN ?	DB	(A) ← (порт ?)
Вывод	OUT ?	D 3	(порт ?) ← (A)
Разрешение	EI	FB	Прерывания разрешены
Запрещение	DI	F3	Прерывания запрещены
Останов	HLT	76	Процесс остановлен
Нет операции	NOP	00	(PC) ← (PC)+1

### 3. КЛАВИВТУРА УОУ. ОСНОВНЫЕ РЕЖИМЫ РАБОТЫ С УОУ

#### 3.1. Назначение клавиш

Управление работой УОУ осуществляется с помощью клавиатуры из 25 клавиш.

Назначение 9 командных клавиш:

RST	формирует команду сброса МП RESET;
MEM	включает режим записи данных в ячейку ОЗУ;
ADDR	включает режим задания адреса ячейки памяти;
CLR	восстанавливает первоначальное значение адреса или данных, если после их ввода не нажимались другие командные клавиши;
NEXT	увеличивает на единицу адрес индицируемой ячейки памяти или регистра
REG	отображает содержимое 8-битного регистра МП;
STEP	запускает в шаговом режиме очередную команду программы;
BRK	задает адрес контрольной точки в программе;
RUN	запускает программу на выполнение. При этом останов

произойдет на введенной контрольной точке или соответствующей команде.

16 клавиш данных служат для ввода 16-ричных цифр от 0 до F. Эти же клавиши задают имена регистров и регистровых пар: A, B, C, D, E, 8/H, 9/L, F обозначают соответственно аккумулятор, регистры общего назначения и регистр флагов.

1/P                    обозначает указатель стека  $SP$ ,

2/T                    обозначает вершину стека  $ST$ .

Ячейки памяти с  $0000_{16}$  до  $03FF_{16}$  отведены под постоянное запоминающее устройство (ПЗУ). В нем хранится программа-монитор, запускаемая в работу сигналами от клавиатуры. Служебная область монитора расположена в оперативном запоминающем устройстве (ОЗУ). Ее адресное поле с  $83E1_{16}$  до  $83FF_{16}$ . Таким образом, в распоряжении пользователя остается не все ОЗУ с адресами  $8000_{16} + 87FF_{16}$ , а два диапазона ячеек ОЗУ:  $8000_{16} + 83E0_{16}$  и  $8400_{16} + 87FF_{16}$ .\*

Клавиатура и монитор позволяют реализовать различные режимы работы:

ввод программы пользователя в ОЗУ;

выполнение программы в шаговом режиме, в режиме с остановом по контрольным точкам и в непрерывном режиме;

запись программы на магнитную ленту магнитофона;

считывание программы с магнитной ленты.

\* Рекомендуемое размещение программы – с адреса 8200.

### 3.2. Чтение и запись содержимого программы и регистров.

Пусть  $v$ ,  $v$  – 16-ричные цифры  $0 + F$ ;

$N$  – любая из 16 клавиш данных.

1. Содержимое ячейки с адресом 8200 выводится на дисплей клавишей  $RST$  в виде

1	2	3	4	5	6	7	8
8	2	0	0			$\alpha$	$\beta$

Содержимое ячейки с адресом  $NNNN$  выводится на дисплей клавишами  $ADDR\ NNNN$  в виде

1	2	3	4	5	6	7	8
N	N	N	N			$\alpha$	$\beta$

Последующее нажатие клавиши  $NEXT$  индицирует адрес и содержимое следующей ячейки памяти.

Повторное нажатие клавиши  $MEM$  выводит на дисплей информацию из предыдущей ячейки.

2. Перед записью программы нужной ячейки ОЗУ устанавливается клавишами  $ADDR\ NNNN\ MEM$  в виде

1	2	3	4	5	6	7	8
N	N	N	N			$\alpha$	$\beta$

(в 6-м разряде дисплея высвечивается точка). После этого данные вводятся нажатием одной или двух клавиш

данных; в 7-м и 8-м разрядах дисплея появляются новые данные.

Переход к следующей ячейке ОЗУ осуществляется клавишей  $NEXT$ .

Повторные нажатия клавиши  $MEM$  уменьшает адрес ячейки на единицу.

3. Пусть  $X$ -любой из регистров  $A, B, C, D, E, F, H, L$ . Содержимое регистра  $X$  выдается на дисплей клавишами

$REG\ X$  в виде

1	2	3	4	5	6	7	8
N	N	N	N	X	-	$\alpha$	$\beta$

(при этом в 1-4 разрядах дисплея сохраняется содержимое программного счетчика  $NNNN$ ).

Нажатие клавиши  $NEXT$  выводит циклически на дисплей содержимое очередного регистра  $A, B, C, D, E, F, H, L, A, B, C, \dots$



Для ввода данных в регистр X следует нажать одну или две клавиши данных.

4.Имя пары регистров отображаются в 5-6 разрядах дисплея, а их содержимое в 1-4 разрядах дисплея с помощью клавиш ADDR R MEM, где

Значения R	B	D	8/H	1/P	2/T
Пара регистров	BC	DE	HL	SP	SF

### 3.3. Ввод контрольных точек.

Ввод контрольной точки по адресу NNNN с числом ее безостановочных проходов NN( $NN < FF_{16}$ ) осуществляется нажатием клавиш ADDR NNNN BRR NN и отображается на дисплее в виде

1	2	3	4	5	6	7	8
N	N	N	N	B	P	N	N

(следует помнить, что нажатие клавиши RST аннулирует все контрольные точки).

### 3.4. Индикация ошибок

Запись данных в ОЗУ разрешена, если в 6-м разряде дисплея (нажатием клавиши MEM) высвечена точка. Код ошибки

1	2	3	4	5	6	7	8
E	r	r					

индицируется при попытках:

- 1) записать байт в ПЗУ или в несуществующую ячейку ОЗУ;
- 2) установить несуществующее имя регистра;
- 3) ввести цифру, если перед этим не была нажата из клавиш ADDR, MEM, REG, BRK;
- 4) запустить программу при адресе, меньшем 4 цифр, и т.д.

Если на дисплее появился код ошибки, то можно восстановить значение:

- 1) счетчика команд и саму команду клавишей CLR или ADDR;
- 2) ячейку памяти и ее адрес клавишей MEM.

### 3.5. Режимы выполнения программы.

Перед выполнением программы в любом режиме следует клавишами ADDR, VNNN задать ее начальный адрес.

Если тумблер выбора режима установлен в позицию ОТЛАДКА, то прерывание и вызов монитора происходит после выполнения каждой команды программы пользователя.

3.5.1. В шаговом режиме останов происходит после выполнения каждой команды. Перед пошаговым выполнением программы тумблер выбора режима следует установить в позицию ОТЛАДКА. Очередная команда программы выполняется по нажатию клавиши STEP. После выполнения этой команды новое значение программного счетчика PC отображается в 1-4 разрядах дисплея, а соответствующее PC содержимое ячейки памяти отображается в 7-8 разрядах дисплея.

После выполнения последней команды следует просмотреть регистры и ячейки ОЗУ с результатами.

В шаговом режиме можно проследить изменения содержимого любого регистра X=A,B,C,D,E,F,H,L.

Для этого перед выполнением первой команды следует нажать клавиши

REG X.

3.5.2. Монитор УОУ позволяет перед пуском программы клавишей RUN создать режим останова по контрольным точкам, т.е. определить адреса, где выполнение программы необходимо прервать для проверки промежуточных результатов. Контрольные точки, каждая из которых занимает 4 байта, записываются в ту же область ОЗУ, где размещается стек.

Напомним, что ввод по адресу NNNN контрольной точки с числом ее безостановочных проходов NN( $NN = \langle FF_{16} \rangle$ )

Если введены контрольные точки, то при исполнении программы в режиме RUN проверяется, соответствует ли содержание программного

счетчика какой-либо контрольной точке. При их совпадении монитор уменьшает на единицу содержимое числа проходов данной контрольной точки и, если оно равно нулю, то происходит останов программы пользователя.

Программа останавливается до выполнения команды, адресуемой контрольной точкой; при этом вызываются входящие в состав монитора подпрограммы клавиатуры и дисплея.

Для выполнения программы с остановами по контрольным точкам необходимо:

- 1) установить тумблер выбора режима ОТЛАДКА;
- 2) ввести контрольные точки;
- 3) клавишами ADDR, NNNN задать начальный адрес программы;
- 4) клавишей RUN пустить программу на выполнение.

В этом режиме программа после выполнения каждой команды автоматически прерывается монитором для контроля точек останова.

Клавиатура и дисплей блокируются до момента, пока текущий адрес программы не сравняется с адресом контрольной точки и число проходов данной контрольной точки не станет равным нулю. После этого будет вызван монитор, а на дисплее отобразится адрес той контрольной точки, на которой произошел останов:

1	2	3	4	5	6	7	8
N	N	N	N				

Нажатие клавиши BRK выводит на дисплей информацию об этой контрольной точке:

1	2	3	4	5	6	7	8
N	N	N	N	B	R	O	O

1	2	3	3	5	6	7
N	N	N	N	B	P	N

При останове в этом режиме, последовательно нажимая клавишу NEXT, можно циклически просмотреть все контрольные точки с переходом от последней к первой. Появившуюся на дисплее информацию о контрольной точке можно изменить, установив новое число проходов NN.

или исключить эту контрольную точку клавишей CLR; при нажатии этой клавиши индицируется адрес следующей контрольной точки. Если контрольных точек больше нет, то дисплей принимает вид

1	2	3	4	5	6	7	8
				В	Р		

Примечания.

1. При вводе или исключении контрольных точек стек сдвигается. Поэтому, если программа пользователя восстанавливает указатель стека, нельзя добавлять или исключать контрольные точки.

2. Необходимо учитывать, что нажатие клавиши RST исключает все контрольные точки.

Пример. В ОЗУ введена следующая программа:

8200	AF		XRA A
1	3C	M:	INR A
2	3C		INR A
3	3C		INR A
4	C3		JMP M
5	01		JMP M
6	82		JMP M

После ввода программы нажаты клавиши:

ADDR	8204	BRK	00	ADDR	8200	RUN
ADDR	8204	BRK	01	ADDR	8200	RUN
ADDR	8204	BRK	02	ADDR	8200	RUN
ADDR	8204	BRK	03	ADDR	8200	RUN

Вид дисплея

1	2	3	4	5	6	7	8
8	2	0	4	A	-	0	3
8	2	0	4	A	-	0	6
8	2	0	4	A	-	0	9
8	2	0	4	A	-	0	C

В данном примере после останова адрес контрольной точки 8204 отображается в 1-4 разрядах дисплея; в 5-м разряде указывается наименование регистра А, а в 7-8 разрядах отображается содержимое этого регистра после последнего выполнения команды 8203.

Кроме описанного, имеется возможность с помощью клавиши BRK прекратить работу программы после выполнения заранее заданного числа команд.

Для этого необходимо после ввода программы в ОЗУ:

1) клавишами ADDR 83E6 BRK NN записать в ячейку с адресом 83E6 уменьшенное на единицу число подлежащих выполнению команд NN

1	2	3	4	5	6	7	8
8	3	E	6	B	P	N	N

2) клавишами ADDR NNNN установить начальный адрес программы;

3) клавишей RUN пустить программу на выполнение.

Пример. В ОЗУ введена следующая программа:

8200	F	A	XRA
1	C	3	INR A
2	C	3	INR A

3	C <sup>3</sup>	INR A
4	C <sup>3</sup>	INR A

Вид дисплея

1	2	3	4	5	6	7	8
8	2	0	1	A	-	0	0
8	2	0	2	A	-	0	1
8	2	0	3	A	-	0	2
8	2	0	4	A	-	0	3

После ввода программы нажать клавиши

ADDR	83E6	BRK	00	ADDR	8200	RUN
ADDR	83E6	BRK	01	ADDR	8200	RUN
ADDR	83E6	BRK	02	ADDR	8200	RUN
ADDR	83E6	BRK	03	ADDR	8200	RUN

В данном примере в 1-4 разрядах дисплея отображается адрес команды, перед выполнением которой произошел останов; в 5-м разряде указывается наименование регистра A, а отображаемое в 7-8 разрядах содержимое этого регистра соответствует значению NN. 25

3.5.3. При выполнении программы в непривычном режиме УОУ работает без подключения монитора, что ускоряет вычисления. Отключение монитора производится установкой тумблера выбора режима в позицию ПРОГОН.

Для того чтобы по окончании программы произошли прерывание и вызов монитора, включающего индикацию, необходимо в качестве команды останова использовать не HLT, а команду RST4(ее код

E7<sub>16</sub>). Иначе, например, при останове по команде HLT(76<sub>16</sub>), монитор не будет вызван и вывод информации на дисплей окажется невозможным.

Для выполнения программы в непрерывном режиме необходимо:

- 1) установить тумблер выбора режима в позицию ПРОГОН;
- 2) клавишами DDDR NNNN задать начальный адрес программы;
- 3) пустить программу клавишей RUN

Сводка режимов выполнения программы

Режим выполнения программы	Положение тумблера выбора режима	Клавиша пуска программы	16-ричной код команды останова
1. Шаговый	ОТЛАДКА	STEP	76 <sub>16</sub>
2.С остановом по контрольным точкам	ОТЛАДКА	RUN	76 <sub>16</sub>
3.Непрерывный	ПРОГОН	RUN	E7 <sub>16</sub>

### 3.6. Рекомендации по использованию прерываний

Каждое прерывание имеет свой вектор, по которому МП определяет начальный адрес программы обработки прерывания.

Как отмечалось, по команде RST N(0=<N=<7) МП переходит на строго фиксированные адреса вида 8\*N, лежащие в диапазоне 0000<sub>16</sub>-0038<sub>16</sub>. Но в УОУ эта область памяти занята монитором, в связи с чем начальные адреса программ обработки прерываний хранятся в специальных ячейках-диспетчерах, куда управление передается из ячеек с адресами вида 8\*N; содержимое диспетчера при поступлении кода прерываний загружается в программный счетчик, а также может быть изменено пользователем.

Монитор использует 3 вектора прерывания:

RST 0 – для начального запуска программ;

RST 4 – для вызова монитора пользователем;

RST 7 – для вызова монитора по прерыванию команды пользователя.

Остальные 5 векторов прерывания пользователь может задействовать без ущерба для работы монитора. Следовательно, для использования прерываний пользователь должен предварительно загрузить соответствующими адресами ячейки – диспетчеры:

(83F4,83F5) – для RST 1;

(83F2,83F3) – для RST 2;

(83F0, 83F1) – для RST 3;

(83EC, 83ED) – для RST 5;

(83EA,83EB) – для RST 6

(в ячейку – диспетчер с меньшим адресом загружается младший байт адреса перехода).

#### 4. ЗАДАЧИ, РЕКОМЕНДУЕМЫЕ ДЛЯ САМОСТОЯТЕЛЬНОГО ВЫПОЛНЕНИЯ И ВЫРАБОТКИ НАВЫКОВ РАБОТЫ С УОУ

##### 4.1. Использование регистров и ячеек памяти

Программа 4.1(выполняется в шаговом режиме)

Адрес	Код	Метка	Мнемокод	Комментарий
8200	3E		MVI A	
8201	19		19	Переслать байт 19 в регистр А
8202	06		MVI B	
8203	25		25	Переслать байт 25 в регистр В
8204	80		ADD B	Прибавить к А регистр В
8205	21		LXI HL	
8206	51		51	Загрузить двойной регистр HL
8207	82		82	Адресом ячейки 8251
8208	70		MOV M,B	Переслать регистр В в ячейку 8251
8209	3D		DCR A	Уменьшить на 1 регистр А
820A	16		MVI D	
820B	47		47	Переслать байт 47 в регистр D



820C	82		ADD D	Прибавить к А регистр D
820D	4F		MOV C,A	Переслать регистр А в регистр С
820E	2C		INR L	Увеличить на 1 регистр L
820F	71		MOV M,C	Переслать регистр С в ячейку 8252
8210	81		ADD C	Прибавить к А регистр С
8211	5F		MOVE,A	Переслать регистр А в регистр Е

Изменение содержимого регистра А после команд:

8200	8204	8209	820C	8210
19 <sub>16</sub>	3E <sub>16</sub>	3D <sub>16</sub>	84 <sub>16</sub>	08 <sub>16</sub>

Содержимое регистров и ячеек после выполнения программ:

A	B	C	D	E	H	L	8251	8252
08 <sub>16</sub>	25 <sub>16</sub>	84 <sub>16</sub>	47 <sub>16</sub>	08 <sub>16</sub>	82 <sub>16</sub>	52 <sub>16</sub>	25 <sub>16</sub>	84 <sub>16</sub>

## 4.2 Сложение двух двухбайтных чисел

Программа 4.2(выполняется в шаговом режиме)

Адрес	Код	Метка	Мnemonic	Комментарий
8200	06		MVI B	Переслать байт 44 в регистр В
8201	44		44	
8202	0E		MVI C	Переслать байт 33 в регистр С
8203	33		33	
8204	16		MVI D	Переслать байт 47 в регистр D
8205	47		47	
8206	1E		MVI E	Переслать байт 53 в регистр Е
8207	53		53	
8208	EB		XCHG	Обменять пару регистров DE с парой HL

8209	09		DAD BC	Прибавить к паре регистров HL содержимое BC
------	----	--	--------	--

Команда сложения регистровых пар DAD проводим суммирование только с двойным регистром HL. Поэтому перед сложением двухбайтных чисел,

Хранящихся в двойных регистрах BC и DE, предварительно осуществляется обмен регистров DE и HL.

Результат сложения помещается в паре регистров HL

4433<sub>16</sub>

+

4753<sub>16</sub>

8B86<sub>16</sub>

Используя непосредственную загрузку пары регистров, можно следующим образом сократить программу:

Адрес	Код	Метка	Мнемокод	Комментарий
8200	01		LXI BC	Загрузить двойной регистр
8201	33		33	BC двухбайтовым числом
8202	44		44	4433
8203	21		LXI HL	Загрузить двойной регистр
8204	52		53	HL двухбайтовым числом
8205	47		47	4753
8206	09		DAD BC	Прибавить к паре HL содержимое BC

#### 4.3 Сложение массива однобайтных чисел

Пусть необходимо найти сумму чисел  $a_1, a_2, \dots, a_{16}$ , хранящихся в ячейках 8301, 8302, ..., 8310. предполагается, что сумма чисел не превосходит 255, а засылка этих чисел предварительно произведена с пульта клавишами:

## ADDR 8301 MEM *a1*NEXT *a2*NEXT.....*a16*NEXT

Блок-схема программы сложения массива однобайтных чисел показана на рис.3

Программа 4.3(выполняется в непрерывном режиме)

8200	21		LXI HL	Загрузить двойной регистр
8201	01		01	HL адресом числа <i>a1</i> ,равным
8202	83		83	8301
8203	06		MVI B	Переслать число слагаемых $10_{16}$ в
8204	10		10	регистр B
8205	AF	SUM:	XRA A	Очистка регистра A
8206	86		ADD M	Прибавить к A содержимое ячейки M
8207	23		INX HL	Увеличить на 1 двойной регистр HL
8208	05		DCR B	Уменьшить на 1 регистр B
8209	C2		JNZ	Если флаг нуля не установлен, то
820A	06		06	Перейти на метку SUM
820B	82		82	
820C	E7		RST 4	Рестарт 4

### 4.4 Сложение массива однобайтных чисел, если их сумма больше 255

Пусть необходимо найти сумму чисел *a1,a2,.....a16*,предварительно записанных в ячейках 8300,8301,830P.Предполагается, что сумма чисел превосходит 255, в связи с чем старший байт суммы накапливается в регистре C ,а младший – в регистре A

Блок-схема программы сложения массива однобайтных чисел ,если их сумма больше 255 ,представлена на рис. 4.

Программа 4.4 (выполняются в непрерывном режиме)

Адресс	Код	Метка	Мнемокод	Комментарий
8200	21		LXI HL	Загрузить двойной регистр
8201	00		00	HL адресом числа <i>al</i> , равным 3300
8202	83		83	
8203	06		MVI B	Переслать число слагаемых $10_{16}$ в регистр В
8204	10		10	
8205	AF		XRA A	Очистка регистра А
8206	4F		MOV C,A	Очистка регистра С
8207	86	SUM	ADD M	Прибавить к А содержимое ячейки М
8208	D2		JNC	Если флаг переноса не установлен, то перейти на метку DLT
8209	0C		0C	
820A	82		82	
820B	0C		INR C	Увеличить на 1 регистр С
820C	23	DLT	INX HL	Увеличить на 1 двойной регистр HL
820D	05		DCR B	Уменьшить на 1 регистр В
820E	C2		JNZ	Если флаг нуля не установлен, то Перейти на метку SUM
820F	07		07	
8210	82		82	
8211	E7		RST 4	Рестарт 4

#### 4.5. Умножение однобайтных чисел.

При вычислении произведения необходимо осуществить многократное суммирование со сдвигом вправо суммы при одновременной проверке содержимого разрядов множителя, начиная со стороны его младшего разряда. При этом, если в очередном разряде множителя записан 1, то множимое прибавляется к сумме и сумма сдвигается вправо на один разряд, а если в разряде записан 0, то производится только сдвиг суммы.

Блок-схема программы умножения однобайтных чисел приведена на рис. 5.

#### Программа 4.5 (выполняется в непрерывном режиме)

Адрес	Код	Метка	Мнемокод	Комментарии
8200	16		MVI D	Загрузить байт множимого в регистр D
8201	*		*	
8202	1E		MVI E	Загрузить байт множителя в регистр E
8203	*		*	
8204	01		LXI BC	Очистить регистры будущего произведения BC
8205	00		00	
8206	00		00	
8207	3E		MVI A	Загрузить регистр A маску 0000 0000 <sub>2</sub>
8208	01		01	
8209	A7	M1:	ANA A	Очистить флаг переноса CY
820A	F5		PUSH PSW	Отослать в стек маску
820B	A3		ANA E	Выделить очередной разряд множителя
*	78		MOV A, B	Вызвать в A старший байт результата
820C				
820D	CA		JZ	Если флаг нуля установлен, то перейти на метку M2
820E	11		11	
820F	82		82	
8210	82	M2:	ADD D	Прибавить можемое к результату

8211	1F		RAR	Сдвиг вправо; младший бит -> CY
8212	47		MOV B, A	Сохранить старший байт в регистре B
8213	79		MOV A, C	Вызвать в A младший байт результата
8214	1F		RAR	Сдвиг вправо; CY -> в старший бит
8215	4F		MOV C, A	Соранить младший байт в регистре C
8216	F1		POP PWS	Вызвать из стека маску
8217	17		RAL	Сдвинуть текущую маску влево
8218	D2		JNC	Если флаг переноса не установлен, то перейти на метку M1
8219	0A		0A	
821A	82		82	
821B	E7		RST 4	Рестарт 4

#### 4.6. «Движущийся» текст

Дисплей из 8 индикаторов при включенном УОУ отражает содержание ячеек с адресами 83F8-83FF. Индикатор (см. рис.6) состоит из 8 светодиодов, каждый из которых светится, если соответствующий его номеру бит ячейки равен 1. Например, для отображения цифры 9 соответствующий байт должен иметь вид:

Номера битов	8 7 6 5	4 3 2 1
Содержание битов	<u>0 1 1 0</u>	<u>1 1 1 1</u>
16-ричная запись байта	6	F

Таким образом, для перемещения на дисплее, например, текста:

«СССР – 1988 г.»

следует, во-первых; предварительно разместить в каких-либо ячейках (например, 8301-830C) следующие:

коды символов: 39 39 39 73 40 06 6F 7F 7F D0 00 00

СИМВОЛЫ: С С С Р - 1 9 8 8 г.

и, во-вторых, организовать в ячейках 83F8-83FF циркуляцию указанных кодов по кольцу справа налево.

Блок-схема программы «движущегося» текста показана на рис.7.

Программа 4.6 (выполняется в непрерывном режиме)

Адрес	Код	Метка	Мnemonic	Комментарий
8200	0E	M1	MVI C	Заслать счетчик длины текста(OB) в регистр C
8201	0B		OB	
8202	11		LXI DE	Загрузить двойной регистр DE адресом 8301
8203	01		01	
8204	83		83	
8205	62		MOV H, D	Переслать адрес 8301 в двойной регистр HL
8206	6B		MOV L, E	
8207	7E	M2:	MOV A, M	Вызвать код символа в регистр A
8208	32		STA	Переслать код символа в ячейку с адресом 83FF
8209	FF		FF	
820A	83		83	
820B	CD		CALL	Обратиться к подпрограмме TIME
820C	16		16	
820D	82		82	
820E	0D		DCR C	Уменьшить на 1 регистр C
820F	C2		JNZ	Если флаг нуля не установлен, то перейти на метку M2
8210	07		07	
8211	82		82	
8212	C3		JMP	Переход на метку M1
8213	00		00	

8214	82		82	
8215	E7		RST 4	Рестарт
8216	06	TIME	MVI B	Заслать счетчик внешнего цикла АО в регистр В
8217	AO		A0	
8218	3E	M3:	MVI A	Заслать счетчик внутреннего цикла FF в регистр А
8219	FF		FF	
821A	3D	M4:	DCR A	Уменьшить на 1 регистр А
821B	C2		JNZ	Если флаг нуля не установлен, то перейти на метку М4
821C	1A		1A	
821D	82		82	
821E	05		DCRB	Уменьшить на 1 регистр В.
821F	C2		JNZ	Если флаг нуля не установлен, то перейти на метку М3
8220	18		18	
8221	82		82	
8222	06		MVI B	Заслать счетчик числа перемещаемых кодов в регистр В
8223	07		07	
8224	21		LXI HL	Загрузить двойной регистр HL адресом 2-го кода символа 83F9
8225	F9		F9	
8226	83		83	
8227	7E	M5:	MOV A, M	Вызвать код символа в регистр А
8228	2B		DCX HL	Уменьшить на 1 двойной регистр HL
8229	77		MOV M,A	Переслать код символа на место предыдущего
822A	23		INX HL	Восстановить адрес вызванного кода символа
822B	23		INX HL	Подготовить адрес следующего кода символа
822C	05		DCR B	Уменьшить на 1 регистр В
822D	C2		JNZ	Если флаг нуля не установлен, то перейти на метку М5
822E	27		27	
822F	82		82	
8230	13		INX DE	Увеличить на 1 двойной регистр DE



8231	62		MOV H, D	Переслать адрес кода символа
8232	6B		MOV L, E	в двойной регистр HL
8233	C9		RET	Выход из подпрограммы

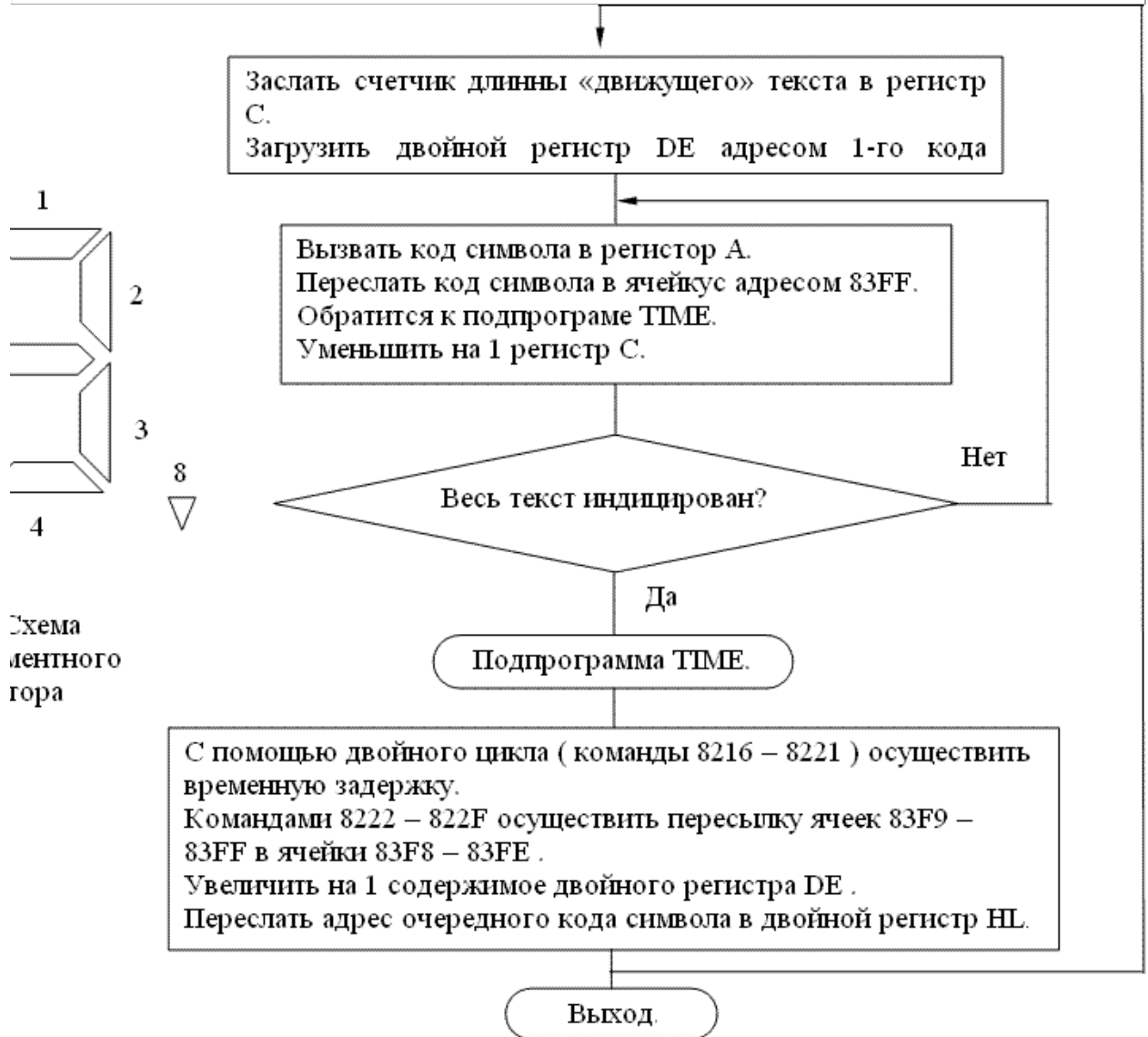


Схема  
ментного  
гора

Рис.7. Блок – схема  
программы  
«движущегося» текста

### Приложение Алфавитный перечень мнемочкодов МП К580.

ACI	v	Прибавить к аккумулятору значение V и флаг переноса (ADDITION CARR IMMEDIATE)
ADC	r	Прибавить к аккумулятору флаг переноса и содержимое регистра r или ячейки памяти M (ADDITION CARRY)

ADD	r	Прибавить к аккумулятору содержимое регистра r или ячейки памяти M (ADDITION)
ADI	v	Прибавить к аккумулятору значение V (ADDITION IMMEDIATE)
ANA	r	Логическое умножение «И» аккумулятора и содержимого регистра r или ячейки памяти M (AND)
ANI	v	Логическое умножение «И» аккумулятора и значения V (AND IMMEDIATE)
CALL	aa	Вызвать подпрограмму
		Вызвать подпрограмму, если:
CZ	aa	Флаг нуля установлен(CALL IF ZERO)
CNZ	aa	Флаг нуля не установлен(CALL IF NOT ZERO)
CP	aa	Аккумулятор положительный(CALL IF PLUS)
CM	aa	Аккумулятор отрицательный(CALL IF MINUS)
CC	aa	Флаг переноса установлен(CALL IF CARRY)
CNC	aa	Флаг переноса не установлен (CALL IF NOCARRY)
CPE	aa	Флаг паритета установлен(CALL IF PARITY EVEN)
CPO	aa	Флаг паритета не установлен(CALL IF PARITY ODD)
CMA		Дополнить аккумулятор(COMPLEMENT ACCUMULATOR)
CMC		Дополнить флаг переноса(COMPLEMENT CARRY)
CMP	r	Сравнить аккумулятор с содержимым регистра r или ячейки памяти M(COMPARE)
CPI	v	Сравнить аккумулятор со значением V(COMPARE IMMEDIATE)
DAA		Десятичная коррекция аккумулятора(DECIMAL ADJUST ACCUMULATOR)
DAD	r r	Прибавить к паре регистров HL содержимое пары регистров rr или указателя стека (DOUBLE ACCEPTATION DATA)
DCR	r	Уменьшить регистр r или ячейку памяти M (DECREMENT REGISTER)
DCX	r r	Уменьшить пару регистров rr или указатель стека (DECREMENT EXTENDED)
DI		Запретить прерывания(DISABLE INTERRUPT)
EI		Разрешить прерывания(ENABLE INTERRUPT)
HLT		Остановить процессор HALT
IN	v	Ввести в аккумулятор байт из порта с адресом V(INPUT)

INR	r	Увеличить регистр r или ячейку памяти M (INCREMENT REGISTER)
INX	rr	Увеличить пару регистров rr или указатель стека (INCREMENT EXTENDED)
JMP	aa	Перейти по адресу aa (JUMP)

Перейти по адресу, если:

JZ	aa	Флаг нуля установлен(JUMP IF ZERO)
JNZ	aa	Флаг нуля не установлен(JUMP IF NOT ZERO)
JP	aa	Аккумулятор положительный(JUMP IF PLUS)
JM	aa	Аккумулятор отрицательный(JUMP IF MINUS)
JC	aa	Флаг переноса установлен(JUMP IF CARRY)
JNC	aa	Флаг переноса не установлен(JUMP IF NO CARRY)
JPE	aa	Флаг паритета установлен (JUMP IF PARITY EVEN)
JPO	aa	Флаг паритета не установлен (JUMP IF PARITY ODD)

LDA	aa	Загрузить в аккумулятор байт, заданный адресом aa (LOAD ACCUMULATOR)
LDAX	rr	Загрузить в аккумулятор байт, заданный адресом, содержащимся в паре регистров BC и DE (LOAD ACCUMULATOR EXTENDED)
LHLD	aa	Загрузить в пару регистров HL байты, заданные адресами aa и aa+1 (LOAD HL DIRECT)
LHI	vv	Загрузить пару регистров 16-битным значением vv (LOAD EXTENDED IMMEDIATE)
MOV	r1,r2	Переслать содержимое (MOVE DATA) регистра
MOV	m,r	R2 в регистр r2 (ячейки M в регистр r)
MOV	r,m	( регистра r в ячейку M)
MVI	r,v	Переслать значение v в регистр r или ячейку памяти M (MOVE IMMEDIATE DATA)
NOP		Нет операции (NO OPERATION)
ORA	r	Логическое сложение «ИЛИ» аккумулятора с содержимым регистра r или ячейки памяти M(OR)
ORI	v	Логическое сложение «ИЛИ» аккумулятора со значением v (OR IMMEDIATE)

OUT	v	Послать содержимое аккумулятора в порт с адресом v (OUTPUT)
PCHL		Переход по адресу, указанному в паре регистров HL (PROGRAM COUNTER)
POP	r r	Прочитать из стека в заданную пару регистров (PULL UA)
PUSH	r r	Записать заданную пару регистров в стек (PUSH DOWN)
RAL		Арифметический сдвиг аккумуляторов влево (ROTATE ARITHMETIC LEFT)
RAR		Арифметический сдвиг аккумуляторов вправо (ROTATE ARITHMETIC RIGHT)
RET		Возвратиться из подпрограммы (RETURN)
		Возвратиться из подпрограммы, если
RZ		Флаг нуля установлен (RETURN IF ZERO)
RNZ		Флаг нуля не установлен (RETURN IF NO ZERO)
RP		Аккумулятор положительный
RM		Аккумулятор отрицательный
RC		Флаг переноса установлен
RNC		Флаг переноса не установлен
RPE		Флаг паритета установлен
RPO		Флаг паритета не установлен
RLC		Циклически сдвинуть аккумулятор влево и во флаг переноса
RRC		Циклически сдвинуть аккумулятор вправо и во флаг переноса
RST	n	Повторный пуск, вызвать подпрограмму по адресу 8N
SBB	r	Вычесть из аккумулятора регистр r или ячейку памяти с учётом переноса (SUBTRACT BORROW)
SBI	v	Вычесть из аккумулятора значение v с учётом переноса (SUBTRACT BORROW IMMEDIATE)
SHLD	aa	Записать содержимое пары регистров HL в ячейки памяти с адресами aa и aa+1 (STORE HL DIRECT)
SPHL		Переслать содержимое пары регистров HL в стек (STACK POINTER)
STA	aa	Загрузить содержимое аккумулятора в ячейку памяти с адресом aa (STORE ACCUMULATOR)
STAX	aa	Загрузить содержимое аккумулятора в ячейку с адресом, указанным в паре регистров BC или DE (STORE

		ACCUMULATOR EXTENDED)
STC		Установить флаг переноса (SET CARRY)
SUB	r	Вычесть из аккумулятора регистр r или ячейку памяти M(SUBTRACT)
SUI	v	Вычесть значение v из аккумулятора(SUBTRACT IMMEDIATE)
XCHG		Обменять пару регистров DE с парой HL (EXCHANGE)
XRA	r	Выполнить логическую операцию «исключающее или» над аккумулятором и содержимым регистра r или ячейки памяти M (EXCLUSIVE OR)
XRI	v	Выполнить логическую операцию «исключающее или» над аккумулятором и значением v (EXCLUSIVE OR IMMEDIATE)
XTHL		Обменять пару регистров HL с содержимым вершины стека (EXCHANGE HL)

### Задания для лабораторной работы

«Адресация (работа с памятью), команды пересылки, сложения, вычитания»

1. Число 2A загрузить в регистр В. Вычитать из него последовательно числа 03, 05, 07, 09, 0B, 0D<sub>16</sub>. Промежуточные результаты сохранять в ячейках с адресами 8300, ... , 8305<sub>16</sub>:

2A-03 -> 8300  
(2A-03)-05->8301...

...

2. Число 19<sub>16</sub> загрузить в регистр С. Прибавлять к нему последовательно числа 21, 23, 25, 27, 29, 2B<sub>16</sub>. Промежуточные результаты сохранять в ячейках 8310, 8313, 8316, ..., 8325<sub>16</sub>:

19+21-> 8310  
(19+21)+23-> 8313...

...

3. Число B5<sub>16</sub> загрузить в ячейку памяти 8300. Прибавлять к нему последовательно числа 05, 07, 09, 0B, 0C<sub>16</sub>. Промежуточные результаты сохранять в ячейках 8301, 82FF, ..., 82F9<sub>16</sub>:

B5+05->8301  
(B5+05)+07->82FF...

...

4. Число 3F<sub>16</sub> загрузить в ячейку памяти 8330. Прибавлять к нему поочередно числа BA, BB, ..., C2<sub>16</sub>. Промежуточные результаты сохранять в ячейках 8328, 8326, 8324, ...:

3F+BA-> 8328  
3F+BB-> 8326...

...

5. Число  $FF_{16}$  загрузить в регистр D. Вычитать из него поочередно числа BF, BE, BD, ...,  $B8_{16}$ . Промежуточные результаты сохранять в ячейках 8340, 8342, ...,  $834E_{16}$ :

$FF-BF \rightarrow 8340$   
 $FF-BE \rightarrow 8342...$

...

6. Число  $A5_{16}$  загрузить в ячейку памяти 8330. Вычитать из него последовательно числа 05, 0A, 0F, 14, 19, 1E,  $23_{16}$ . Промежуточные результаты сохранять в ячейках 8332, 8334, ...:

$A5-05 \rightarrow 8332,$   
 $(A5-05)-0A \rightarrow 8334...$

...

7. Число  $B5_{16}$  загрузить в ячейку памяти 8350. Вычитать из него поочередно числа 02, 05, 08, 0B, 0E,  $11_{16}$ . Промежуточные результаты сохранять в ячейках 834E, 834C, ...,  $8344_{16}$ :

$B5-02 \rightarrow 834E$   
 $B5-05 \rightarrow 834C...$

...

8. Число  $AD_{16}$  загрузить в регистр B. К числу  $AD_{16}$  прибавлять последовательно на 01, 03, ...,  $09_{16}$ . Промежуточные результаты сохранять в ячейках 8300, 8302, ...:

$AD+01 \rightarrow 8300$   
 $(AD+01)+03 \rightarrow 8302...$

...

9. Число  $F5_{16}$  загрузить в ячейку памяти 8300. Вычитать из него последовательно числа 06, 05, 04, 03,  $02_{16}$ . Промежуточные результаты сохранять в ячейках 8327, 8325, ...:

$F5-06 \rightarrow 8327$   
 $(F5-06)-05 \rightarrow 8325, ...$

...

10. Число  $FA_{16}$  загрузить в регистр L. Вычитать из него последовательно числа  $1B_{16}$ , 1A, 19, 18,  $17_{16}$ . Промежуточные результаты сохранять в ячейках 8300, 8302, ...:

$FA - 1B \rightarrow 8300$   
 $(FA - 1B)-1A \rightarrow 8302...$

...

11. Число  $6B_{16}$  загрузить в регистр C. Из числа  $6B_{16}$  вычитать числа 05, 07, ...,  $0D_{16}$ . Промежуточные результаты сохранять в ячейках 8301, 8303, ...:

$6B-05 \rightarrow 8301$   
 $(6B-05)-07 \rightarrow 8303...$

...

12. Число 2A загрузить в регистр B. Прибавлять к нему поочередно числа 03, 05, 07, 09, 0B,  $0D_{16}$ . Промежуточные результаты сохранять в ячейках с адресами 8300, ...,  $8305_{16}$ :

$2A+03 \rightarrow 8300$   
 $2A+05 \rightarrow 8301...$

...  
13. Число  $CC_{16}$  загрузить в регистр С. Вычитать из него последовательно числа 05, 07, ...,  $0D_{16}$ . Промежуточные результаты сохранять в ячейках 8301, 8303, ...:

$CC-05 \rightarrow 8301$   
 $(CC-05)-07 \rightarrow 8303...$

...  
14 Число  $F9_{16}$  загрузить в регистр С. Прибавлять к нему последовательно числа  $V1, V3, V5, V7, V9, V_{16}$ . Промежуточные результаты сохранять в парах ячеек 8310- 8311, 8313-8314, 8316-8317, ..., 8325-8326 $_{16}$ :

$F9+V1 \rightarrow$  ст. байт  $\rightarrow 8310$ , мл. байт  $\rightarrow 8311$   
 $(F9+V1)+V3 \rightarrow$  ст. байт  $\rightarrow 8313$ , мл. байт  $\rightarrow 8314...$

...  
15 Число  $3F_{16}$  загрузить в ячейку памяти 8330. Прибавлять к нему последовательно числа  $01VA, 01VB, \dots, 01C_{216}$ . Промежуточные результаты сохранять в парах ячеек 8328-8329, 8326-8327, 8324-8325, ...:

$3F+01VA \rightarrow$  ст. байт  $\rightarrow 8328$ , мл. байт  $\rightarrow 8329$   
 $(3F+01VA)+01VB \rightarrow$  ст. байт  $\rightarrow 8326$ , мл. байт  $\rightarrow 8327...$

...  
16 Число  $FF_{16}$  загрузить в регистр D. Прибавлять к нему поочередно числа  $VF, VE, VD, \dots, V_{16}$ . Промежуточные результаты сохранять в парах ячеек 8340-8341, 8342-8343, ..., 834E-834F $_{16}$ :

$FF+VF \rightarrow$  ст. байт  $\rightarrow 8340$ , мл. байт  $\rightarrow 8341$   
 $FF+VE \rightarrow$  ст. байт  $\rightarrow 8342$ , мл. байт  $\rightarrow 8343...$

...  
17 Число  $FF15_{16}$  загрузить в ячейки памяти 8330 (ст.байт) и 8331 – мл. байт. Вычитать из него последовательно числа 05, 0A, 0F, 14, 19, 1E, 23 $_{16}$ . Промежуточные результаты сохранять в парах ячеек 8332-8333, 8334-8335, ...:

$FF15-05 \rightarrow$  ст. байт  $\rightarrow 8332$ , мл. байт  $\rightarrow 8333$   
 $(FF15-05)-0A \rightarrow$  ст. байт  $\rightarrow 8334$ , мл. байт  $\rightarrow 8335...$

...  
18 Число  $FFF5_{16}$  загрузить в ячейки памяти 8300 (ст.байт) и 8301 – мл. байт. Вычитать из него последовательно числа 06, 05, 04, 03, 02 $_{16}$ . Промежуточные результаты сохранять в парах ячеек 8327-8326, 8325-8324, ...:

$FFF5-06 \rightarrow$  ст. байт  $\rightarrow 8327$ , мл. байт  $\rightarrow 8326$   
 $(FFF5-06)-05 \rightarrow$  ст. байт  $\rightarrow 8325$ , мл. байт  $\rightarrow 8324...$

...  
19 Число  $F51A_{16}$  загрузить в любые 2 регистра. Вычитать из него числа 1B, 1A, 19, 18, 17 $_{16}$ . Промежуточные результаты сохранять в ячейках 8300-8301, 8302-8303, ...:

$F51A - 1B \rightarrow$  ст. байт  $\rightarrow 8300$ , мл. байт  $\rightarrow 8301$   
 $(F51A - 1B)-1A \rightarrow$  ст. байт  $\rightarrow 8302$ , мл. байт  $\rightarrow 8303...$

...

Министерство транспорта Российской Федерации  
Федеральное государственное образовательное учреждение  
высшего профессионального образования  
Волжская государственная академия водного транспорта

Кафедра информатики, систем управления  
и телекоммуникации

Р.Н. Гурьяшова, А.В. Шеянов

# ИНФОРМАТИКА. ПАКЕТ MATHCAD

Учебное пособие

Издательство ФГОУ ВПО ВГАВТ  
Н. Новгород, 2005



УДК 517.2:681.3

Г 95

Гурьяшова Р.Н., Шеянов А.В.  
Информатика. Пакет Mathcad. Учебное пособие / Р.Н. Гурьяшова,  
А.В. Шеянов – Н. Новгород: Изд-во ФГОУ ВПО ВГАВТ, 2005. – 140 с.

Рассматривается одна из наиболее распространенных математических программ – пакет MathCad фирмы MathSoft, Inc. Пакет в настоящее время широко используется для решения научно-технических и инженерных задач.

При написании пособия использован опыт преподавания авторами дисциплины "Информатика".

Учебное пособие предназначено для студентов высших учебных заведений водного транспорта по специальностям: "Техническая эксплуатация транспортного радиооборудования", "Кораблестроение", "Гидротехническое строительство", "Эксплуатация судового электрооборудования и средств автоматики", "Эксплуатация судовых энергетических установок", "Эксплуатация перегрузочного оборудования портов и транспортных терминалов", "Инженерная защита окружающей среды".

Излагаются основы техники работы с пакетом MathCad. Возможности MathCad иллюстрируются на типовых вычислительных задачах. Приводятся лабораторные работы (рабочие документы MathCad) с подробными комментариями, а также таблицы заданий к ним. Реализация заданий на компьютере позволит студентам приобрести навыки работы с пакетом и эффективно использовать их в учебном процессе.

Авторы выражают благодарность за ценные замечания старшему преподавателю Седовой Е.Ю., проф. д.т.н. Чирковой М.М.

Рекомендовано к изданию кафедрой Информатики, систем управления и телекоммуникации Волжской государственной академии водного транспорта 17.12.04, протокол № 4.

Рецензенты:

- проф. д.т.н. Ю.С. Федосенко;
- кафедра Информатики и систем управления Нижегородского государственного технического университета, зав.кафедрой проф. д.т.н. Ю.С.Бажанов;
- заместитель руководителя Центра подготовки правил Российского речного регистра проф. д.т.н. П.И.Бажан.

© ФГОУ ВПО ВГАВТ, 2005

## Введение

Программа Mathcad позволяет моделировать в электронном документе научно-технические расчеты в форме, близкой к общепринятому в математике виду.

Применение Mathcad в учебном процессе способствует повышению эффективности работы студентов при изучении технических дисциплин, а также при выполнении курсовых и дипломных работ.

Чрезвычайная простота интерфейса Mathcad сделала его одним из самых популярных математических пакетов.

В пособии излагаются средства системы Mathcad и приводится ряд лабораторных работ по следующим темам:

- вычисление сложных математических выражений, записанных в естественной форме;
- табулирование функций одной и двух переменных;
- построение двух- и трехмерных графиков;
- операции с векторами и матрицами;
- символьные преобразования.

В Mathcad широко используются численные методы решения различных математических задач, к которым сводятся многие инженерно-экономические расчеты.

В пособии приведены также элементы численных методов и примеры их реализации в Mathcad, а именно:

- решение уравнений и систем уравнений;
- вычисление интегралов;
- решение дифференциальных уравнений.

Изложение ведется по англоязычной версии Mathcad 2000. Тем не менее, структура пособия такова, что основную часть материала можно использовать с любой версией Mathcad, начиная с версии 6.

Цель пособия – научить студентов быстро и легко решать в среде Mathcad простые математические задачи вычислительного характера. Каждая работа предваряется основными сведениями, необходимыми для ее выполнения, и содержит наборы индивидуальных заданий. Кроме того, к каждой работе прилагается рабочий документ Mathcad, в котором приведены примеры по данной теме. Предполагается, что студенты выполняют их в первую очередь, и только освоив эти примеры и получив те же результаты, приступают к выполнению своих индивидуальных заданий.

# 1. Основы работы

## 1.1. Основные положения

Документ программы Mathcad называется рабочим листом. Он содержит объекты: формулы, графические области и текстовые блоки. В ходе расчетов формулы и графические области обрабатываются последовательно, слева направо и сверху вниз, а текстовые блоки игнорируются. Это означает, что блоки нельзя располагать в документе произвольно. Блоки, готовящие какие-либо операции, должны предшествовать блокам, выполняющим эти операции.

### Пример

Неверно	Верно	
$f(x):=x^2$ $f(3)=$	$f(x):=x^2$ $f(3)=$	$f(x):=x^2$ $f(3)=$
(печатается сообщение об ошибке)	(печатается результат)	

В руководстве Mathcad употребляется термин "живой документ". Это означает следующее: выражения и графики, размещенные на листе, доступны для изменений; изменения, внесенные в каком-либо месте, распространяются по всему документу в направлении сверху вниз и слева направо.


### Важно!


Применительно к сказанному выше о порядке расчета это означает, что перемещение блоков в документе может нарушить ход вычислений.

Ввод информации осуществляется в месте расположения курсора.

Программа Mathcad использует три вида курсоров:

+ Если ни один объект не выбран, используется крестообразный курсор (красный крестик), определяющий место создания следующего объекта.

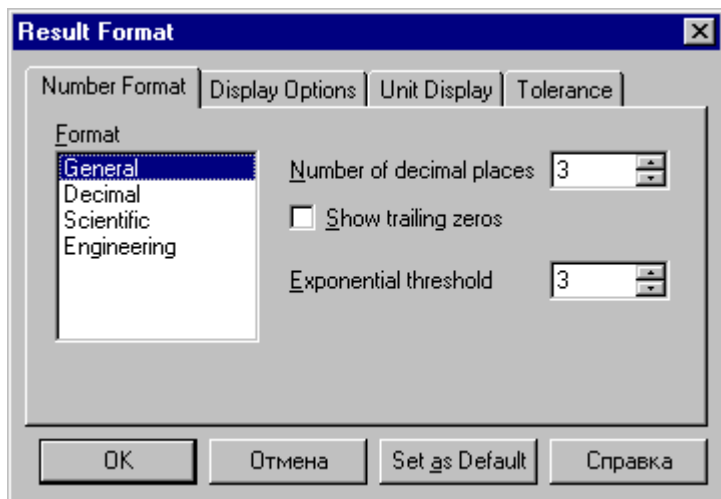
$x:=$   При вводе формул используется уголковый курсор (синий уголок), указывающий направление и место ввода текущего элемента выражения.

 При вводе данных в текстовый блок применяется текстовый курсор в виде вертикальной черты.

### Внимание!

При вводе следует иметь в виду, что система различает регистр букв, т. е. "x" и "X" – это две разные переменные.

Пакет Mathcad ориентирован на получение численного результата; все числа – результаты вычислений в пакете хранятся с точностью 15 значащих цифр. По умолчанию программа показывает только три. Изменить формат отображения результата можно либо через меню Format | Result, параметр Number of decimal places (до 15), или дважды щелкнув мышью на результате.



Поле Exponential threshold определяет, с какой величины число представляется в экспоненциальной записи:  $1.385 \times 10^3$  вместо 1385.456 .

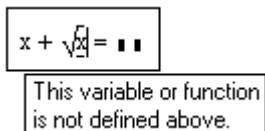
Тем не менее, возможно получение большей, чем 15 цифр, точности. Данная возможность достигается путем применения аппарата символьных вычислений и описывается в соответствующем разделе (п. 2.6.2).

## 1.2. Набор и редактирование формул

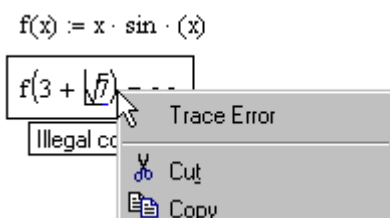
Формулы — основные объекты рабочего листа. По умолчанию новый объект является формулой. Чтобы начать ввод формулы, надо установить крестообразный курсор, расположенный изначально в левом верхнем углу рабочего листа, в нужное место и начать ввод. Перемещение курсора возможно как мышью, так и клавишами управления курсором ("стрелками").

При этом создается область формулы, в которой появляется уголко-вый курсор, охватывающий текущий элемент формулы, например, имя переменной (функции) или число. Если формула заканчивается знаком "=", Mathcad отобразит результат вычисления.

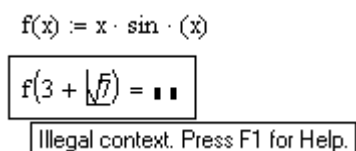
При попытке вычислить формулу система может встретиться с ошибкой (неопределенная переменная, несоответствие числа аргументов и т. п.). При этом проблемный участок формулы выделяется цветом, при наведении указателя мыши на область с ошибкой выводится сообщение:



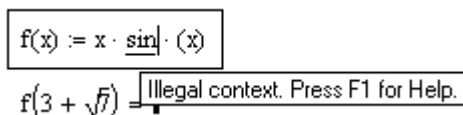
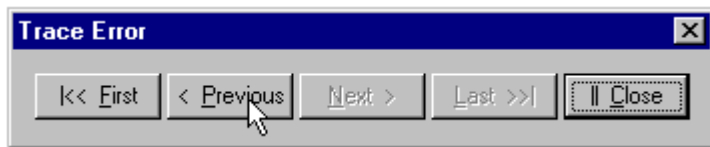
В сложных случаях помочь установить источник ошибки может панель отслеживания ошибок (Trace Error), доступная из контекстного меню при щелчке правой кнопкой мыши на области с ошибкой. Пункт Previous (предыдущая ошибка) позволяет пройти по цепочке вычислений до первого ошибочного звена. Однако, данное средство не всегда работает (пункт Previous неактивен).



Вызов панели отслеживания ошибок...

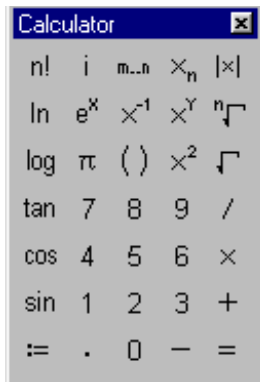


Нажатие кнопки Previous...



И вот оно, ошибочное звено (знак умножения после sin)

"Строительным блоком" для построения выражений является шаблон операции. Так, например, при вводе бинарного оператора по другую сторону знака операции автоматически появляется заполнитель в виде черного прямоугольника  $x + \blacksquare$ . В это место вводят очередной операнд. Шаблоны можно вводить с клавиатуры (что значительно ускоряет работу) или с помощью палитр операций (панелей инструментов, Toolbar). Список горячих клавиш для ввода шаблонов операций с клавиатуры см. в приложении "Горячие клавиши". Пример палитры операций:



Можно сконструировать очень сложные выражения, просто вкладывая шаблоны один в другой, например:

$$\sqrt[3]{x^2 - \ln\left(\left|\frac{\sqrt{x}}{y+1}\right|\right)}$$

При вводе Mathcad форматирует выражение, приводя его к принятому в математике виду.

Так, если мы наберем "3/", Mathcad автоматически преобразует это к записи:



Дальнейший ввод будет происходить в знаменателе.

Мы можем выйти из-под знаменателя, используя клавиши управления курсором ("стрелочки").

Чтобы выделить элементы формулы, которые в рамках операции должны рассматриваться как единое целое, используют клавишу <Пробел>. При каждом нажатии этой клавиши уголкоый курсор "расширяет"

ся", охватывая элементы формулы, примыкающие к данному. После выбора шаблона или ввода знака операции элементы в пределах углового курсора автоматически помещаются в шаблон. Для выхода из шаблонов (например, из-под корня, знаменателя и т. п.) также используется клавиша <Пробел>.

### Пример

Вычислить значение выражения  $\frac{2+3}{4 \cdot \sqrt{7}}$ .

№	Действие	На экране
1.	2+3	$2 + 3$
2.	< Пробел >	$2 + 3$
3.	/	$\frac{2 + 3}{}$
4.	4	$\frac{2 + 3}{4}$
5.	*	$\frac{2 + 3}{4 \cdot }$
6.	\ или знак $\sqrt{\quad}$ с палитры операций	$\frac{2 + 3}{4 \cdot \sqrt{\quad}}$
7.	7	$\frac{2 + 3}{4 \cdot \sqrt{7}}$
8.	< Пробел >	$\frac{2 + 3}{4 \cdot \sqrt{7}}$
9.	< Пробел >	$\frac{2 + 3}{4 \cdot \sqrt{7}}$
10.	=	$\frac{2 + 3}{4 \cdot \sqrt{7}} = 0.472$

Обратите внимание, что знак умножения набирается звездочкой (\*), а отображается точкой (·).



С выделенными уголковым курсором частями выражений можно производить стандартные операции: вырезать / копировать / вставить.

В набранное выражение легко внести изменения. Например, пусть нам нужно вычислить:

$$\frac{2 + 3}{\sqrt{7^{1.5}}}$$

Скопируем предыдущее выражение и модифицируем его:

1.	щелкаем левой кнопкой мыши после цифры 7	$\frac{2 + 3}{4 \cdot \sqrt{7}} = 0.472$ ■
2.	^ или знак x <sup>y</sup> с палитры операций	$\frac{2 + 3}{4 \cdot \sqrt{7}^{\blacksquare}} = \blacksquare \blacksquare$
3.	1.5	$\frac{2 + 3}{4 \cdot \sqrt{7^{1.5}}} = \blacksquare \blacksquare$
4.	щелкаем левой кнопкой мыши после цифры 4	$\frac{2 + 3}{\blacksquare \cdot \sqrt{7^{1.5}}} = \blacksquare \blacksquare$
5.	Два раза нажимаем BackSpace	$\frac{2 + 3}{\sqrt{7^{1.5}}} = \blacksquare \blacksquare$
6.	<Enter> или щелчок мышью вне области формулы	$\frac{2 + 3}{\sqrt{7^{1.5}}} = 1.162$

Еще одна заслуживающая упоминания особенность при вводе – клавиша <Insert> меняет позицию ввода вокруг выделения (справа / слева). При этом курсор меняет вид с  на  и обратно. В частности, таким образом можно поменять знак операции в выражении:



## Пример

$x+y$	$x + y +$	набрано
выделяем $y$	$x + \boxed{y}$	курсор находится справа от $y$
<Insert>	$x + \boxed{y}$	курсор перемещается влево от $y$
<Backspace>	$x \boxed{y}$	стираем знак операции
<->	$x - \boxed{y}$	и ставим на место новый символ
$x-y$	$x - y +$	результат

### 1.3. Запись констант и переменных

Выражения Mathcad могут включать константы, переменные и функции.

#### 1.3.1. Числовые константы

Числа в естественной форме (123.45) записываются как есть, для разделения целой и дробной части используется точка.

Числа в экспоненциальном (научном) формате ( $1.2345 \cdot 10^2$ ) вводятся следующим образом:  $1.2345 * 10^2$ .

Значительным достоинством пакета Mathcad является легкость работы с комплексными числами.

Комплексные числа записываются в форме, подобной  $4 + 5i$ . Мнимую единицу можно выбрать из панели "Calculator"; для ввода числа "мнимая единица" с клавиатуры следует набрать  $1i$ , в противном случае "i" будет воспринято как имя переменной. В качестве символа мнимой единицы допускается использование "j". Кроме мнимой единицы, в системе доступны следующие **встроенные константы**:

Константа	Значение	Способ ввода
$\infty$	бесконечность	Ctrl+Shift+z
e	число e	e
$\pi$	число $\pi$	Ctrl+Shift+p
i	$\sqrt{-1}$	1i

j	$\sqrt{-1}$	Ij
%	0.01	%
deg	множитель для перевода в радианы	deg

Символ "∞" в численных расчетах соответствует "машинной бесконечности" ( $10^{307}$ ). Градусы в радианы переводятся умножением:

косинус 60-ти градусов

$$\cos(60 \cdot \text{deg}) = 0.5$$

### 1.3.2. Переменные

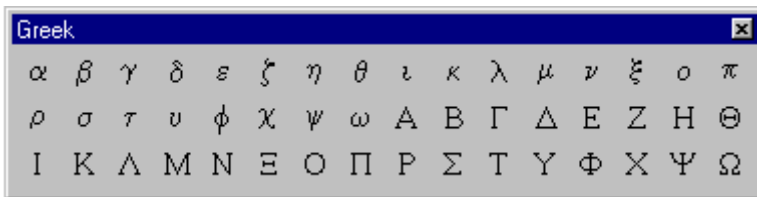
Имена переменных могут состоять из латинских и греческих букв, цифр, символов процента (%), апострофа (') и подчеркивания (\_), должны начинаться с буквы. Переменная может иметь индекс, поясняющий ее смысл. Для набора индекса следует нажать точку (.).

Например, для ввода имени переменной  $x_{\min}$  нужно нажать клавиши `<x><.><min>` (после нажатия точки курсор ввода сместится вниз).

#### Внимание!

Существует опасность визуально спутать индекс в имени переменной с индексом вектора (матрицы). Так,  $x_1$  может означать как "переменная с именем x-первое", так и "первый элемент массива x". Индекс вектора (матрицы) вводится другими средствами и его использование описывается в разделе "Работа с векторами и матрицами" (п. 2.5).

Для ввода греческих букв MathCad предлагает панель "Greek". Другой способ – введя латинский эквивалент греческой буквы (например, "a" для "α"), нажать Ctrl-G.



Напоминаем также, что система различает регистр букв, т. е. "x" и "X" – это две разные переменные.

### 1.3.3. Присваивание

Перед использованием переменной ей необходимо присвоить значение. Использование неинициализированной переменной вызывает сообщение об ошибке (*This variable or function is not defined above* – эта переменная или функция не определена выше). Для присваивания используется обозначение ":= " (вводится нажатием двоеточия (":") или с панели "Calculator").

#### Примеры

В MathCad	действие
$x:=5.25$	Переменной $x$ присваивается значение 5.25
$y:=x$	Переменной $y$ присваивается значение переменной $x$
$z:=x^2+5$	Переменной $z$ присваивается значение выражения
$z =$	"равно" без двоеточия выведет результат

Следует обратить внимание на то, что Mathcad не различает имена функций и переменных. Таким образом, можно ввести переменную с именем  $\sin$  – но функцией  $\sin(x)$  после этого пользоваться уже нельзя.

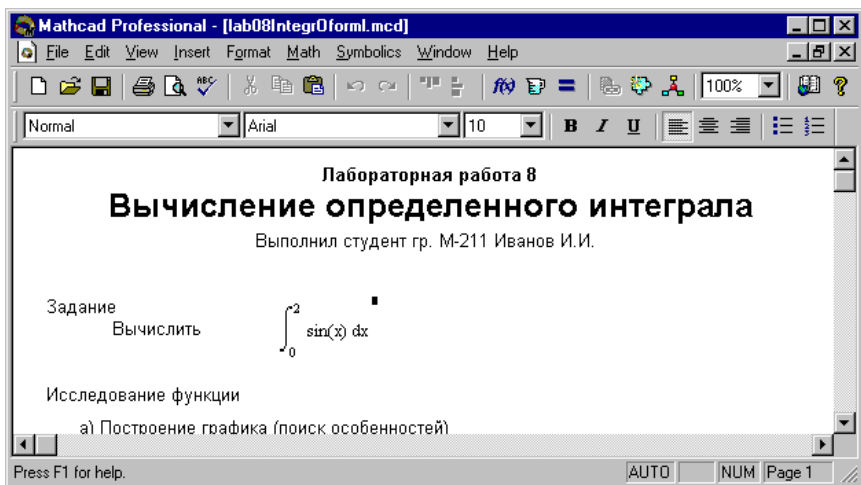
Переменная может иметь размерность. Размерность вводится в операции присваивания после указания значения через меню Insert | Units (горячая клавиша Ctrl-U). Использование размерных величин позволяет избежать некоторых ошибок в решении физических задач.

### 1.3.4. Системные переменные

Для пользователя доступно несколько системных переменных Mathcad, управляющих его работой. Нам пригодятся переменные TOL и STOL, задающие точность численных расчетов (см. глава "Решение уравнений и систем" (п. 2.3)). Переменная ORIGIN задает номер первого элемента массивов (матриц) (см. глава "Работа с векторами и матрицами" (п. 2.5)).

### 1.4. Работа с текстом

Несмотря на то, что работа с текстом не является отличительной особенностью пакета MathCad, эта задача часто возникает в процессе оформления документа.



Текстовый блок воспринимается системой MathCad как комментарий. Он используется только для пояснений и на порядок вычислений не влияет.

Имеется три способа вставки в документ текстового блока:

1. Через меню Insert | Text Region
2. С помощью горячей клавиши (") (двойная кавычка)
3. Система считает, что пользователь вводит текст, если при вводе нажат < Пробел >.

Но если вы нажали клавишу вставки шаблона какой-нибудь операции (например "/" или "["), переход в текстовый режим по нажатию клавиши < Пробел > не произойдет.

По умолчанию текст в текстовом блоке выравнивается влево (на панели инструментов доступны также выравнивания вправо и по центру); при достижении видимой правой границы листа (вертикальная серая линия) текст автоматически переносится на следующую строку. После автоматического переноса строки будут находиться под первой (т. е. блок влево не увеличивается).

В версии MathCad 2000 с панели инструментов "Formatting" доступны следующие возможности форматирования текста:

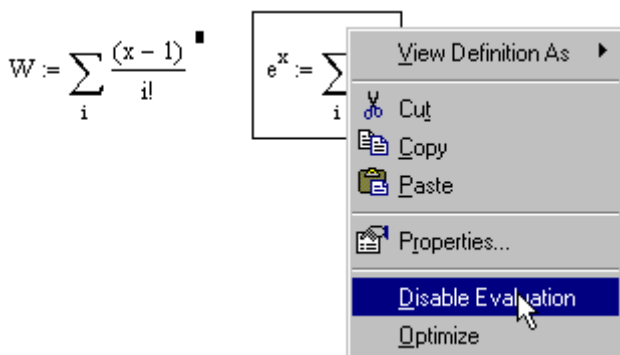
- Выравнивание абзацев – влево, вправо и по центру (по границе текстового блока);
- Специальные абзацы – нумерованные и маркированные списки;
- Форматирование отдельных символов – выделение произвольным шрифтом, начертанием (жирный / курсивный / ...) и размером.

Более сложное форматирование доступно через меню Format.

Интересным случаем является невычисляемая формула. Иногда при оформлении документа нужно привести формулу, вычисления по которой в MathCad невозможны или приводят к ошибке. В таком случае можно запретить вычисления по формуле (Disable Evaluation из контекстного меню). Такие формулы отмечаются черным квадратиком в правом верхнем углу. При необходимости вычисления можно опять разрешить (Enable Evaluation, там же).

### Пример

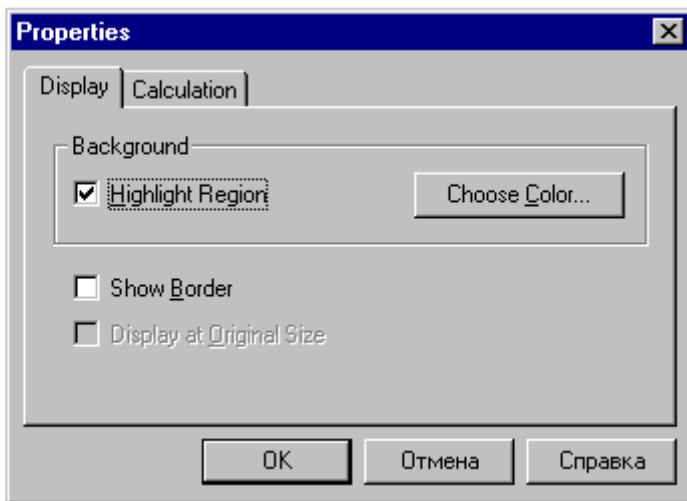
Формула слева отключена (Disabled), формула справа – в процессе отключения.



Для оформления документа большое значение имеет взаимное расположение блоков. При щелчке мышью в блоке становится видна рамка, за которую его можно перемещать; кроме того, на рамке имеются маркеры изменения размера.

Пункт меню "View | Regions" делает блоки видимыми. Отдельно взятый блок можно подсветить произвольным цветом из контекстного меню (пункт Properties, вкладка Display переключатель Highlight Region) или заключить в рамку (там же, переключатель Show Border).

$$X := 1$$



Для выравнивания блоков имеются две команды в меню Format, продублированные на панели инструментов:

- Format | Align Regions | Across (вдоль верхней границы)
- Format | Align Regions | Down (вдоль левой границы)



Данные команды действуют на выделенные блоки (недоступны до выделения, по крайней мере, двух блоков), выравнивание происходит по первому блоку.

Дополнительно, команда

- Format | Separate Regions «раздвигает» перекрывающиеся области (на всем листе – применяйте с осторожностью).

### **Внимание!**

При оформлении (выравнивании, перетаскивании блоков) следует помнить, что вычисления ведутся слева направо / сверху вниз, и этот порядок легко нарушить. В имеющейся у нас версии отмена для операций перемещения блоков не предусмотрена.

С целью предотвратить "разбегание" блоков можно применять возможность вставки математических блоков внутри текстового (команда меню Insert | Math Region). Данные формулы участвуют в вычислениях как обычно, но перемещаются только с объемлющим текстовым блоком.

Из дополнительных средств оформления документа стоит упомянуть разрыв страницы (меню Insert | Page Break) и области (Area, Insert | Area). Область представляет собой две горизонтальные линии через все окно

MathCad с маркерами слева; линии можно перемещать по документу, а двойной щелчок на маркере «схлопывает» область между ними. При этом формулы в области продолжают работать в обычном порядке. Таким образом, области представляют собой удобный способ скрыть детали вычислений, подчеркивая главное в документе. Кроме того, область можно «закрыть» от изменений (меню Format | Area | Lock), возможно – с заданием пароля.

## 1.5. Переменные-диапазоны

Для выполнения повторяющихся вычислений в пакете Mathcad используются переменные-диапазоны (ранжированные переменные, range). Переменная-диапазон содержит последовательность значений (арифметическую прогрессию) и задается указанием первого, второго и граничного значений:

$$x := 1, 3..6 \quad y := 4..8 \quad z := 7..3$$

x =

1
3
5

y =

4
5
6
7
8

z =

7
6
5
4
3

### Внимание!

Шаг явно не задается. Он вычисляется системой автоматически как разность второго и первого значений.

Как следует из примера, при отсутствии второго значения шаг принимается равным единице.

Для ввода символа "две точки" ("..") используется клавиша "точка с запятой" (";") или кнопка на панели инструментов "Calculator".



Если в выражении присутствует переменная-диапазон, MathCad вычисляет выражение столько раз, сколько значений диапазон содержит.

Диапазоны используются:

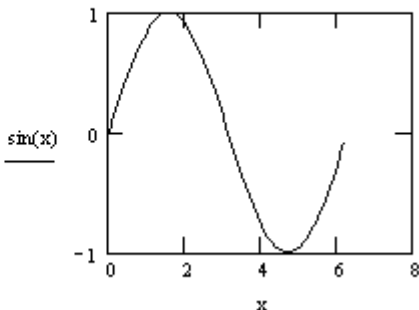
- для организации циклов, как индексы, вместе с векторами и матрицами;
- для построения таблиц значений функции (табулирование);
- для построения графиков.

Примеры последних двух ситуаций приведены ниже.

t := 0,0.2..1

x := 0,0.1..2π

t =	sin(t) =
0	0
0.2	0.199
0.4	0.389
0.6	0.565
0.8	0.717
1	0.841



Необходимо учитывать, что подобная переменная не является обычным числом (скаляром) и ряд действий выполнить с ранжированной переменной нельзя.

### Пример

Задача: вычислить значение выражения  $10 \cdot i$  для  $i$  от 1 до 5.

Последовательность действий, указанных слева, приводит к ошибке, поскольку результатом вычисления будет множество значений (что видно справа).

i := 1..5	10 · i =
f := 10 · i	10
	20
	Illegal context. Press F1 for Help.
	40
	50


## 1.6. Функции в Mathcad

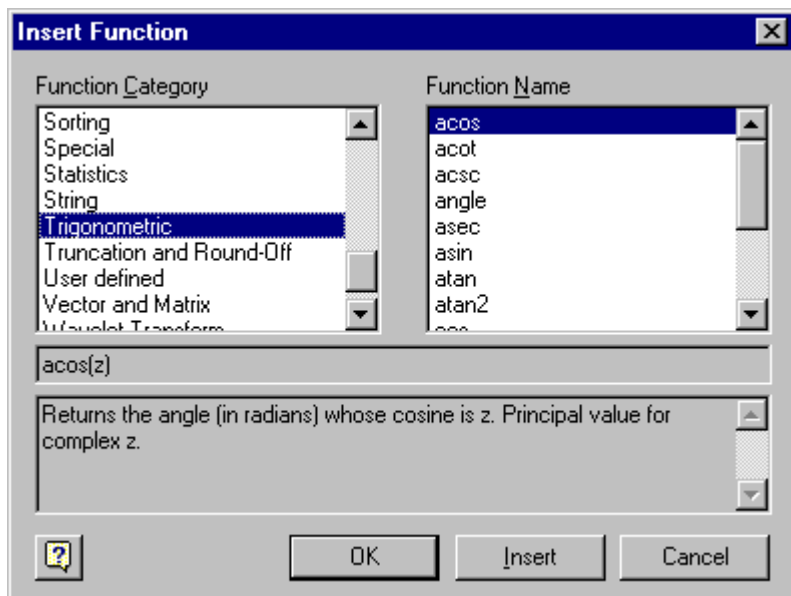
Система Mathcad предоставляет пользователю несколько сотен встроенных функций; кроме того, имеется возможность создания собственных (определенных пользователем) функций.

Имеется три способа ввода стандартных функций Mathcad.

1. Имена наиболее употребительных функций ( $\sin x$ ,  $\sqrt{x}$  и т. д.) можно вводить нажатием кнопок панели "Calculator".



2. Если написание имени функции известно, можно ввести ее с клавиатуры, например,  $\sin(x)$ .
3. Можно ввести функцию (например,  $\arccos(x)$ ), воспользовавшись окном вставки функции (меню Insert | Function, сочетание клавиш Ctrl + E или кнопка  на панели инструментов). Окно вставки функций позволяет осуществить поиск функции по названию, группе (категории) и отображает краткую справку по выделенной функции. Категория All содержит все функции (в алфавитном порядке).



Следует обратить внимание на то, что названия функций в системе Mathcad могут отличаться от привычных нам. Так, тот же  $\arccos(x)$  для Mathcad'a обозначается  $\text{acos}(x)$ .

Аргументы функции вводятся в скобках и отделяются запятыми. Если в момент вставки функции часть выражения была выделена, она становится первым аргументом функции. Аргументы тригонометрических функций задаются в радианах. Для перевода градусов в радианы имеется встроенная константа  $\text{deg}$ , например:

$$\sin(30\text{-deg}) = 0.5$$

даст нам значение синуса 30 градусов.

При вводе функции вручную возможны следующие типичные ошибки:

Неверно	Верно	комментарий
$\sin x$	$\sin(x)$	для вызова функции нужны скобки
$\sin^2(x)$	$\sin(x)^2$	в степень возводится результат вычисления синуса, а не обозначение
$\sin \cdot (x)$	$\sin(x)$	знак умножения здесь не нужен
$x(x+1)$	$x \cdot (x+1)$	знак умножения здесь необходим

Аргументом функции может быть как обычная переменная, так и переменная-диапазон; в последнем случае вычисление функции будет выполнено для каждого значения из диапазона. Ряд функций принимает в качестве аргументов вектора, матрицы и даже имена функций.

Для создания собственной функции необходимо задать ее с помощью оператора присваивания в форме

имя\_ф-ции(аргументы) := выражение

например:

$$f(x, y) := \frac{\sin(x^2 + 1)}{x + y}$$

Все переменные и функции, стоящие в правой части от знака присваивания, должны быть определены ранее или находиться среди списка аргументов, в противном случае будет выдано сообщение "Эта переменная или функция не определены ранее" (*This variable or function is not defined above*).

В дальнейшем определенную пользователем функцию можно использовать так же, как и встроенную.

Следует обратить внимание на то, что имеется возможность (опасность) переопределить уже существующую функцию.

Приведем ряд общеупотребительных функций, названия которых в системе Mathcad отличаются от принятых в русскоязычной литературе.

Принято	В Mathcad	Пояснение
$tg(x)$	$\tan(x)$	
$arccos(x)$	$\text{acos}(x)$	
$arcsin(x)$	$\text{asin}(x)$	
$arctg(x)$	$\text{atan}(x)$	
$\log_b(x)$	$\log(x, b)$	Логарифм $x$ по основанию $b$
$lg(x)$	$\log(x)$	Десятичный логарифм

$sh(x)$	$\sinh(x)$	Гиперболический синус
$ch(x)$	$\cosh(x)$	Гиперболический косинус
$th(x)$	$\tanh(x)$	Гиперболический тангенс

Наряду с функциями в общепринятом смысле MathCad предоставляет большой набор специальных функций, предназначенных для решения тех или иных задач. Приведем те из них, которые будут рассмотрены в данном пособии.

название в MathCad	Назначение
Группа "Решение уравнений и систем"	
<i>root</i>	Нахождение корней функции одной переменной
<i>find</i>	Решение систем уравнений
<i>minerr</i>	Приближенное решение систем уравнений
Группа "Решение дифференциальных уравнений "	
<i>rkfixed</i>	Решение систем обыкн. дифф. ур-й 1-го порядка
<i>odesolve</i>	Решение обыкновенного дифф. ур-я n-го порядка
Группа "Оптимизация"	
<i>minimize</i>	Нахождение минимума функции n переменных
<i>maximize</i>	Нахождение максимума функции n переменных
Группа "Вектора и матрицы"	
<i>cols</i>	Число столбцов матрицы
<i>rows</i>	Число строк матрицы
<i>min</i>	Нахождение минимума в массиве (матрице)
<i>max</i>	Нахождение максимума в массиве (матрице)
<i>mean</i>	Среднее арифметическое массива
<i>sort</i>	Сортировка массива
<i>reverse</i>	Обращение (переворачивание) массива

## 1.7. Графики в Mathcad. Декартовы координаты

### 1.7.1. Общая информация

Известно, что графики представляют собой удобное средство анализа цифровой информации. Система Mathcad предоставляет возможность быстрого и удобного построения графиков.

В системе (версия Mathcad 2000) имеется семь видов графиков:

1. Обычный график в декартовых координатах (сочетание клавиш Shift-2).
2. График в полярных координатах.
3. Поверхность.
4. Линии уровня.
5. Объемная гистограмма.
6. Объемный точечный график.
7. Векторное поле.

Для построения каждого типа графика вставляется соответствующий шаблон, затем поля шаблона заполняются. Шаблоны графиков можно вставить:

1. С помощью меню "Insert | Graph"
2. С панели инструментов "Graph" (ниже)
3. С помощью горячих клавиш (перечислены в меню).



Кроме того, на панели "Graph" имеются две кнопки для исследования графика в декартовых координатах – Увеличить (Zoom) и Трассировка (Trace). Данные кнопки доступны, только когда выделен график функции одной переменной в декартовых или полярных координатах.

В системе Mathcad различаются два способа построения графика – график по умолчанию (быстрый график – QuickPlot), и график по точкам (по массиву данных). Для начала рассмотрим построение графиков вторым способом.

## 1.7.2. Построение графика по точкам

Для построения графика необходим набор точек. Простейший вариант: рассмотрим построение графика функции одной переменной в декартовых координатах.

Иконка на панели инструментов:



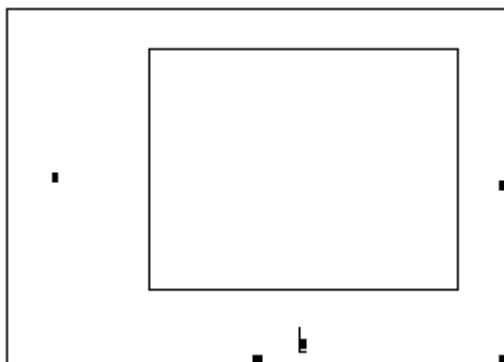
Горячая клавиша Shift-2.

### Задача:

Построить график функции  $\sin(x)$  на интервале  $[1;7]$ .

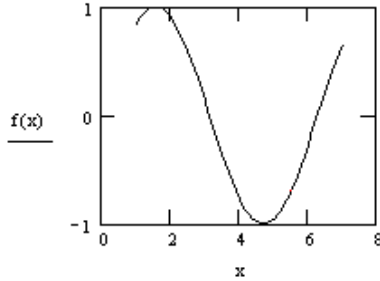
### Действия:

- Определение функции  
 $f(x) := \sin(x)$
- Задание диапазона значений аргумента (ранжированной переменной)  
 $x := 1, 1.1..7$
- Вставка шаблона графика в декартовых координатах. Левый верхний угол шаблона появляется в месте нахождения курсора (красного крестика).



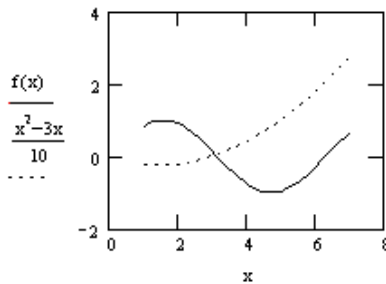
Поля в шаблоне – абсцисса (внизу) и ордината (слева), черные квадратики на внешней рамке – маркеры изменения размера.

- Действия по заполнению шаблона  
В нижнее поле подставляем переменную-диапазон  $x$ , в левое – функцию введенной переменной  $f(x)$ .
- После заполнения шаблона достаточно нажать  $\langle \text{Enter} \rangle$  или щелкнуть мышкой вне области графика для того, чтобы увидеть результат.

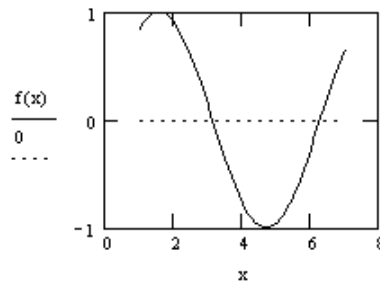


Для того, чтобы добавить на график еще одну функцию от той же переменной, необходимо:

- поместить курсор после функции в шаблоне (не забыть выйти из-под корня, степени, или индекса клавишей <Пробел>);
- нажать <, > (запятая) (курсор переместится на следующую строку);
- ввести имя функции (или формулу).



Если ввести константу, получим горизонтальную линию (например, 0 – имитация оси X):



### Внимание!

Так как график построен по фиксированному количеству точек, имеется опасность пропустить особенности поведения функции (резкие изменения, уход в бесконечность) при большом шаге.

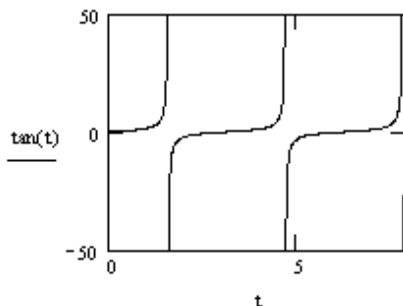
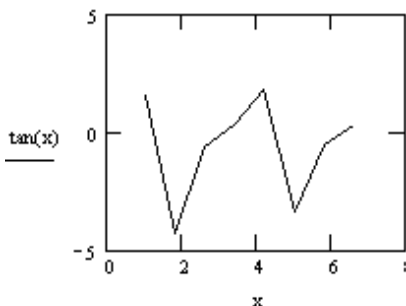
### Пример

Построение графика  $\text{tg}(x)$  на интервале  $[1, 7]$ .

неверно!

на самом деле так

$x := 1, 1.8..7$



В данном случае небольшое число точек приводит к грубому искажению формы графика.

### 1.7.3. Быстрый график

Построение графика по набору точек имеет свои достоинства и недостатки.

#### Достоинства:

- точки, в которых вычисляется функция, задаются пользователем. Разумный выбор количества точек позволяет увеличить скорость работы системы (например, при перерисовке экрана во время прокрутки длинного документа).

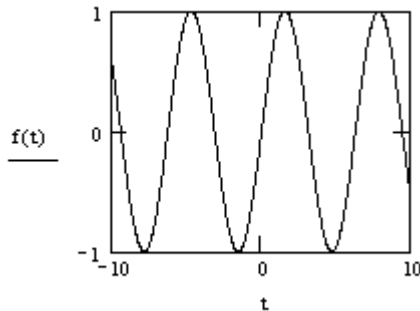
#### Недостатки:

- опасность пропустить особенности поведения функции (проиллюстрирована выше);
- при увеличении масштаба (уменьшение интервала) графика количество оставшихся на графике точек может оказаться недостаточным;
- при увеличении интервала (уменьшении масштаба) новые точки не добавляются, так что добавленная часть графика останется пустой;
- кроме того, задание переменной-диапазона представляет собой лишний (по отношению к задаче построения графика) шаг.

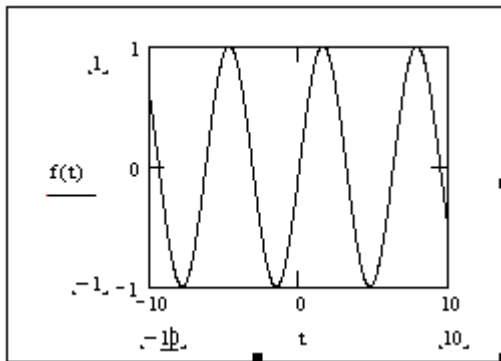
Вариант построения "быстрый график" позволяет получить результат быстрее – и с меньшими ограничениями.

Для этого в шаблон графика вместо переменной-диапазона подставляем переменную, не определенную ранее.

По умолчанию MathCad строит график на интервале  $[-10, 10]$  (или на той его части, где функция определена). При этом число точек выбирается автоматически (достаточно большим).



При необходимости диапазон изменения аргумента может быть исправлен. Щелчок мышью на графике показывает пределы изменения координат на осях. Их можно изменить на нужные нам значения.



Например, если мы хотим получить график в диапазоне  $[1, 7]$  (см. выше), то необходимо:

- изменить левый предел  $(-10)$  на  $1$ ;
- изменить правый предел  $(10)$  на  $7$ .
- нажать  $\langle \text{Enter} \rangle$  или щелкнуть мышкой вне области графика для того, чтобы увидеть результат.



Аналогично можно изменить пределы и по оси ординат. Это может понадобиться, например, если функция на интервале уходит в бесконечность.

В заключение перечислим достоинства и недостатки "быстрого графика".

**Достоинства:**

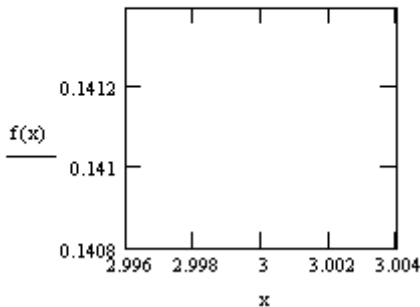
- для построения графика требуется на один шаг меньше, задание (соответственно, изменение) значения переменной-аргумента функции не требуется;
- большее количество точек обеспечивает нам гладкий график, меньше шансов пропустить нерегулярность функции;
- возможность произвольно менять масштаб (при изменении интервала путем замены пределов изменения аргумента на графике новые точки добавляются; при уменьшении интервала с помощью инструмента "Увеличить" – нет).

**Недостатки:**

- большое количество точек вычисления функции замедляет работу системы. Это особенно сказывается, если мы пролистываем длинный документ, или строим график функции, требующей больших вычислений (а именно: производная; функция, заданная неявно; решение дифференциального уравнения). При этом курсор принимает вид "лампочки", показывающий, что система занята расчетами ("думает");
- если переменная, использованная как аргумент функции на графике, уже задана (имеет числовое значение), то график не строится. Распознать такую ситуацию можно по пределам изменения аргумента, проставленным на графике, например:

$$x := 3$$

$$f(x) := \sin(x)$$



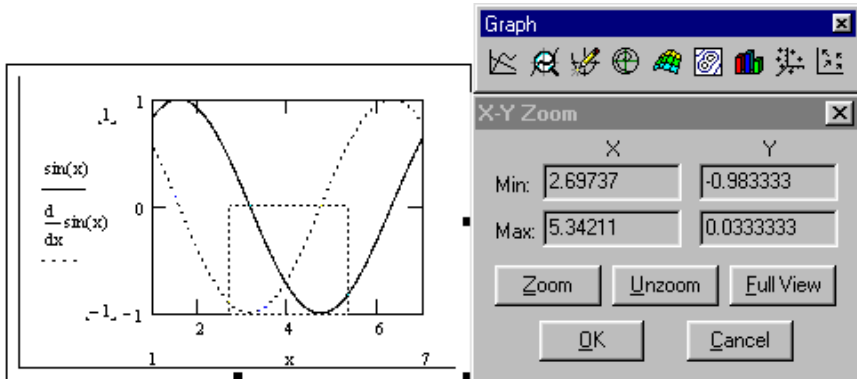
В данном случае построить быстрый график не удастся из-за того, что переменная  $x$  оказывается определена выше. Обратите внимание на диапазон изменения  $x$  на графике.

### 1.7.4. Исследование графика: инструмент Увеличить (Zoom)

Данный инструмент доступен только при выделении графика функции одной переменной в декартовых или полярных координатах.

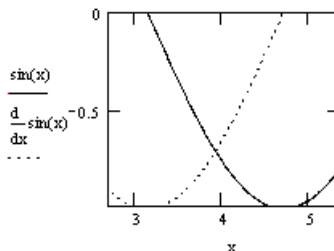


Также может быть вызван из контекстного меню графика (пункт Zoom...).



На рисунке показан вид окна диалога, появляющегося после нажатия кнопки "Увеличить". Окна с минимальным / максимальным значениями заполняются программой при выделении области на графике (перемещением курсора мыши при нажатой левой клавише), вручную задать эти значения нельзя. Кнопка "Zoom" увеличивает выбранный участок на весь график, "Unzoom" отменяет одно увеличение. "Full View" возвращает пределы измерения координат к значениям по умолчанию (для быстрого графика: абсцисса – от  $-10$  до  $10$ , ордината – по значению функции, для графика по точкам: исходный размер). Кнопка "Cancel" отменяет одно увеличение и закрывает приведенное окно, "OK" выполняет увеличение и закрывает окно. Кнопка "Отмена" на основной панели инструментов на действия по увеличению масштаба графика не работает.

После нажатия "Zoom":



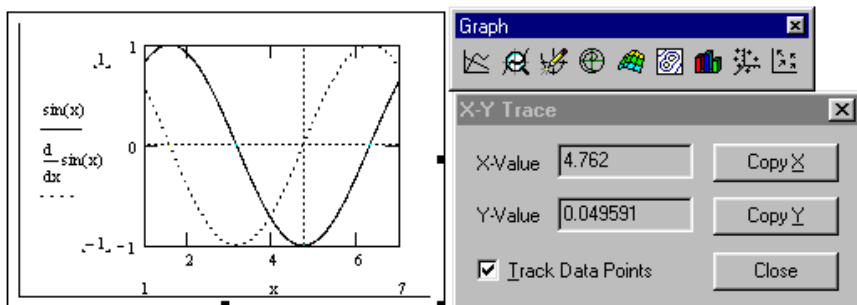
Инструмент может быть полезен для выяснения деталей поведения функции на малых участках. Следует иметь в виду, что использование инструмента "Zoom" не увеличивает количества точек на графике.

### 1.7.5. Исследование графика: инструмент Трассировка (Trace)

Данный инструмент доступен только при выделении графика функции одной переменной в декартовых или полярных координатах.



Также может быть вызван из контекстного меню графика (пункт Trace...).



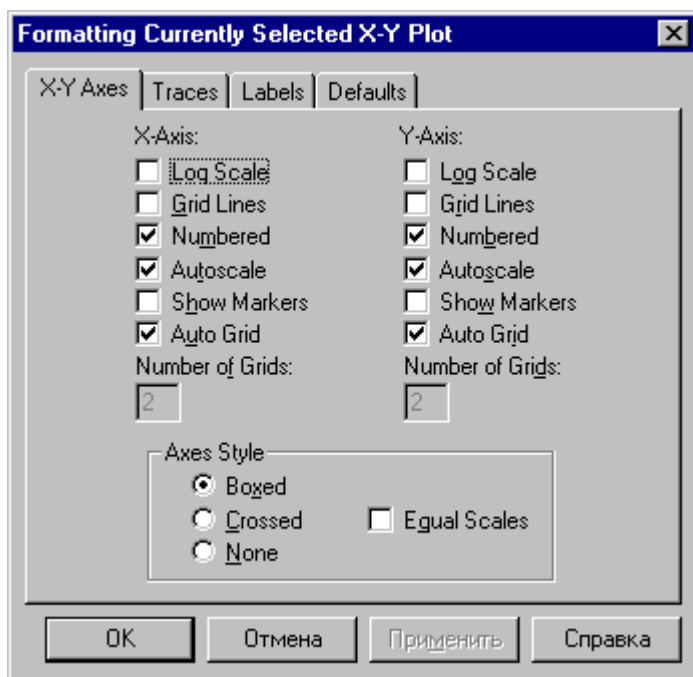
Значения координат отслеживают перемещение курсора мыши при нажатой левой кнопке. Галочка "Track Data Points" заставляет курсор "прилипать" к графику. Кнопки "Copy X", "Copy Y" копируют значение соответствующей координаты в буфер. Оттуда его можно вставить на рабочий лист.

Инструмент может быть полезен для приближенного решения уравнений (решение как точка пересечения кривых).

### 1.7.6. Настройка графика

Окно настройки графика функции одной переменной в декартовых или полярных координатах вызывается двойным щелчком на графике (если вы попадете при этом на ось, получите его часть: только настройки оси). Это же окно доступно через пункт меню Format | Graph | X-Y Plot... и через контекстное меню (Format). Это окно состоит из четырех вкладок; мы рассмотрим наиболее употребляемые возможности первых двух.

Вкладка "X-Y Axes": настройка осей на графике.

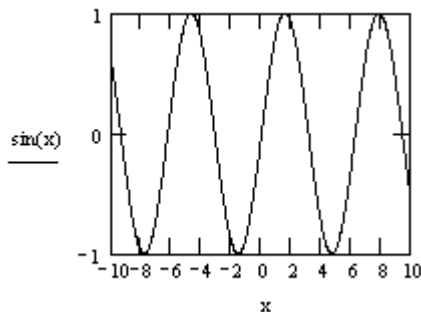


Доступные настройки включают:

- установку логарифмической шкалы по одной или обеим осям (Log Scale)
- включение сетки (Grid lines)
- значения на осях (Numbered)
- автомасштабирование (Autoscale)
- число засечек на осях (Number of Grids при выключенном пункте Auto Grid)
- обеспечение равного масштаба по осям (Equal Scales)

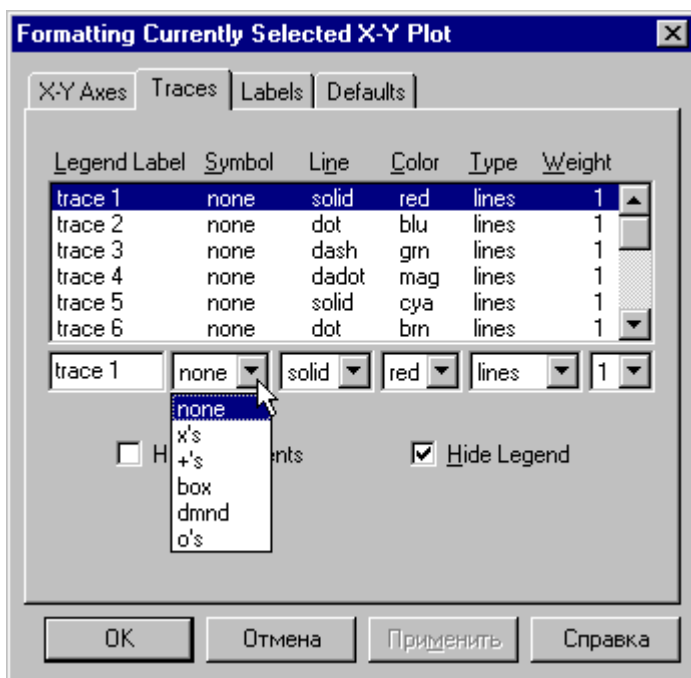
### Пример

Чтобы увеличить число засечек по оси X на быстром графике, выполним следующее:



1. Щелкаем правой кнопкой на графике
2. Выбираем из контекстного меню "Format"
3. Выбираем вкладку "X-Y Axes"
4. Снимаем галочку Auto Grid в первой колонке (X-Axis)
5. Ставим в поле Number of Grids число 10.

Вкладка "Traces": настройка вида графика.

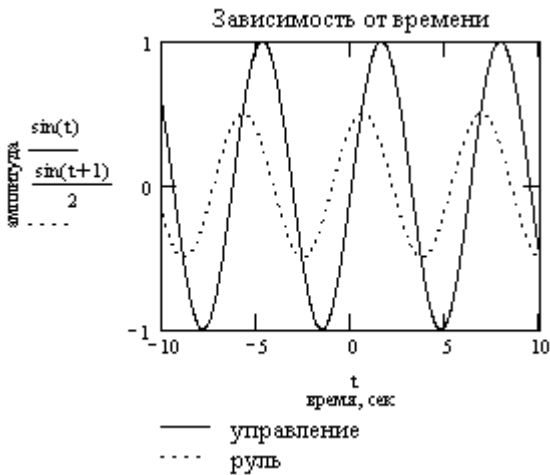


Доступные настройки включают:

- отображение символов в точках графика (крестики, кружочки ...) (Symbol)
- стиль линии (сплошной, пунктирный ...) (Line)
- цвет линии (Color)

- тип графика (линия, точки, ступеньки...) (Type)
  - толщина линии (Weight)
- Установка типа графика "Bar" для быстрого графика (много точек) приводит к сплошному закрашиванию области под графиком.
- Кроме того, имеется возможность спрятать имена переменных и функций, по которым был построен график (Hide Arguments), и показать произвольную подпись для каждого графика (Legend Label; появляются при снятой галочке Hide Legends)

Вкладка Labels позволяет задать подписи на графике и осях, вкладка Defaults – сбросить параметры графика и вернуться к принятым по умолчанию. Пример графика с подписями приведен ниже (возможность использовать русские буквы в подписях зависит от версии MathCad).



## 2. Дополнительные возможности

Далее описываются возможности MathCad, выходящие за границы "основ" работы с системой.

Следующие разделы условно отнесены к "дополнительным" (advanced – "продвинутому"):

- Объемные графики
- Сложные функции
- Решение уравнений и систем

- Решение задач оптимизации
- Работа с векторами и матрицами
- Символьные вычисления
- Решение дифференциальных уравнений
- Интегрирование

## 2.1. Прочие типы графиков. Объемные графики

### 2.1.1. График в полярных координатах

Иконка на панели инструментов:



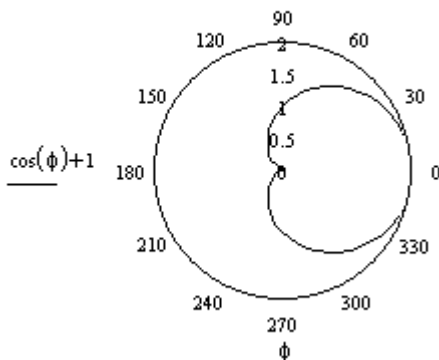
Горячая клавиша Ctrl+7.

График в полярных координатах строится аналогично; поля в шаблоне – полярный угол и радиус.

#### Пример

Построим график кардиоиды  $\rho = \cos(\phi) + 1$ .

$$\phi := 0, .1 .. 2\pi$$



### 2.1.2. Графики "Поверхность", "Линии уровня" и "Объемная гистограмма" по точкам

Иконки на панели инструментов:



Графики "Поверхность", "Линии уровня" и "Объемная гистограмма" отображают функцию двух переменных  $z(x, y)$  или строятся по готовой

матрице данных. Для начала рассмотрим построение графика функции по готовым точкам. Для этого нужно построить матрицу значений функции.

**Пример**

Рассмотрим построение графика поверхности  $z=\sin(x)*\sin(y)$  в области  $x \in [-4, 4], y \in [-1, 5]$ .

Действия:

Задаем переменные-диапазоны  $i, j$ , которые будут служить индексами в матрице. На этом этапе необходимо принять решение о том, сколько будет точек. По оси X возьмем шаг 1, получим 9 значений (8 шагов). По оси Y примем шаг равным 0.6, получим 11 значений (10 шагов).

$i:=0..8$        $j:=0..10$

Запишем формулы, преобразующие индексы  $i, j$  (целые, начинающиеся от 0) в нужные нам координаты  $x_i, y_j$ . В общем случае

$$x_i = x_{min} + i * \text{шаг}, \quad \text{шаг} = \frac{x_{max} - x_{min}}{\text{число шагов}}.$$

Для данного примера

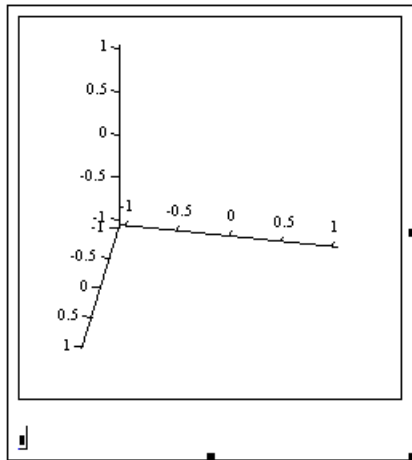
$$X_i := -4 + i \qquad Y_j := -1 + j \cdot \frac{6}{10}$$

Проверим, что получается ( $X_i = \dots$ ).

Зададим матрицу, записав формулу для вычисления ее элементов.

$$M_{i,j} := \sin(X_i) \cdot \sin(Y_j)$$

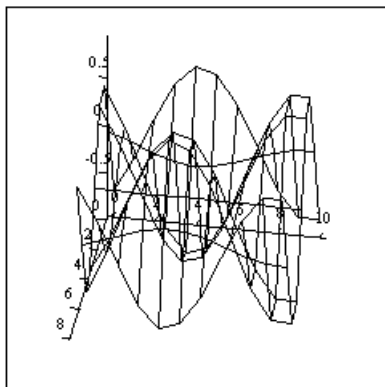
Вставляем шаблон объемного графика (можно сочетанием Ctrl+2).





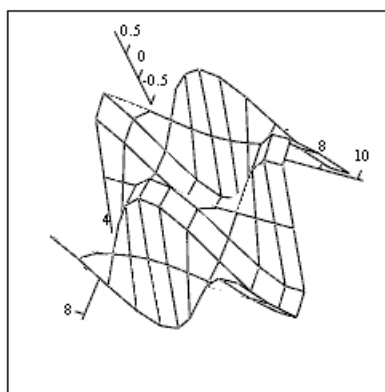
Единственное поле в шаблоне – для матрицы, по которой строится график, черные квадратики на внешней рамке – маркеры изменения размера.

Подставляем в шаблон имя матрицы (в данном случае М). Нажимаем <Enter> или щелкаем мышкой вне области графика для того, чтобы увидеть результат.



М

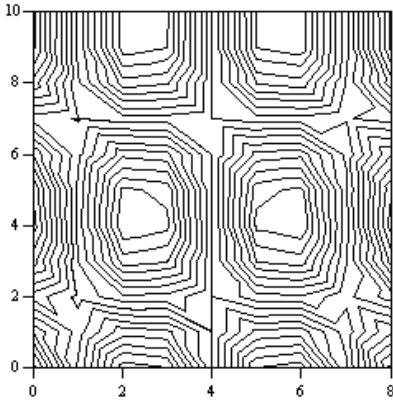
При этом номера строк, столбцов берутся за координаты  $x$ ,  $y$ , а элементы матрицы – за  $z$ . Внешний вид полученного графика можно улучшить (скрыть невидимые линии, закрасить, мышью повернуть в более выгодное положение), но "угловатость", определяемая малым выбранным количеством точек, так и останется.



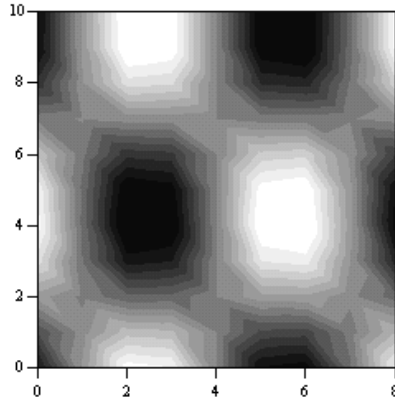
М

График "Объемная гистограмма" отобразит нам похожую картинку столбиками разной высоты. Данный тип больше подходит для исследования набора данных, в котором говорить о значениях "между" имеющимися смысла не имеет.

График "Линии уровня" строится аналогично, но результатом будет серия кривых на плоскости. Данный тип графика хорошо показывает положения экстремумов функции (после настройки закраски).



М



М

Промежуточные результаты для линий уровня строятся интерполированием между имеющимися точками. В данном случае это не очень помогло (контуры ломаные). Число точек необходимо выбирать, соотносясь с задачей (малое количество точек может показать картину, имеющую мало общего с реальностью, большое – "повесить" систему).

Достоинство описанного подхода к построению графиков: меньшее количество точек.

Недостатки: несколько промежуточных этапов (задание диапазонов индексов, преобразование их в координаты, задание матрицы). Кроме того, поскольку график строится по матрице, информация о входных координатах отсутствует (отображаются только номера точек по осям).

### 2.1.3. Объемные кривые по точкам

Иконка на панели инструментов:



Интересной разновидностью объемных графиков являются объемные кривые. Для построения объемной кривой по заданным точкам необходимо задать три вектора равной длины, для координат  $x$ ,  $y$ ,  $z$  соответственно.

#### Пример

Построение объемной спирали.

Задаем переменную-диапазон: индекс  $i$

$i := 0..50$

Задаем вектора

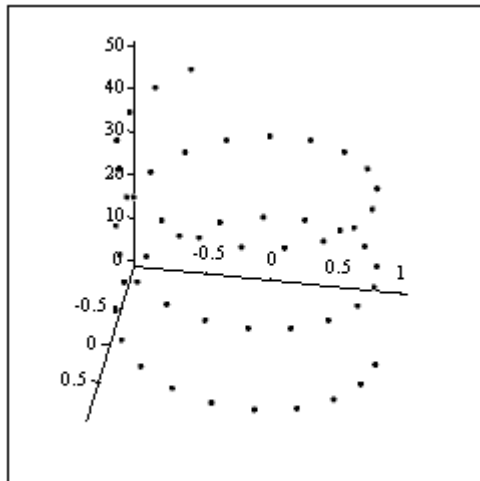
$$x_i := \sin\left(\frac{i}{3}\right)$$

$$y_i := \cos\left(\frac{i}{3}\right)$$

$$z_i := i$$

Вставляем шаблон "Scatter plot"

В поле шаблона вставляем в скобках имена всех трех векторов, через запятую.

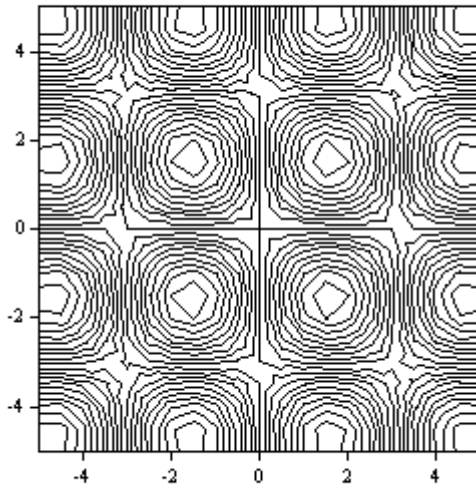


$(x, y, z)$

### 2.1.4. Быстрый объемный график

"Быстрый график" предназначен для построения графика функции без задания матриц. Для графиков "поверхность", "линии уровня" и "гистограмма" входной параметр – функция двух переменных. Вот как выглядит построение "быстрого графика" функции, рассмотренной выше (п. 2.1.2):

$$f(x, y) := \sin(x) \cdot \sin(y)$$



f

Как видно, все, что нужно сделать предварительно – задать функцию двух переменных.

Диапазон изменения переменных задается по умолчанию, но доступен для изменения (двойной щелчок на графике открывает окно настройки, подробнее ниже). Как видно из примера, значения координат доступны на графике. Кроме того, за счет меньшего количества дополнительных шагов уменьшается возможность ошибки.

Для построения объемной кривой нужно задать:

- вектор-функцию одной переменной (сразу три значения);
- или три функции одной переменной.

**Пример**

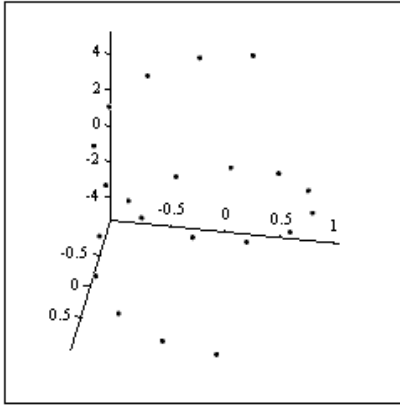
Приведем оба способа построения спирали, подобной построенной выше (п. 2.1.3).

$$f(x) := \begin{pmatrix} \sin(x) \\ \cos(x) \\ x \end{pmatrix}$$

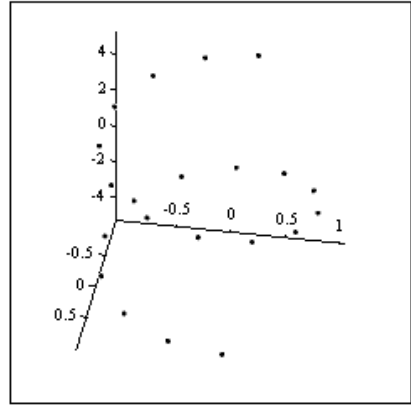
$$x(t) := \sin(t)$$

$$y(t) := \cos(t)$$

$$z(t) := t$$



f



(x, y, z)

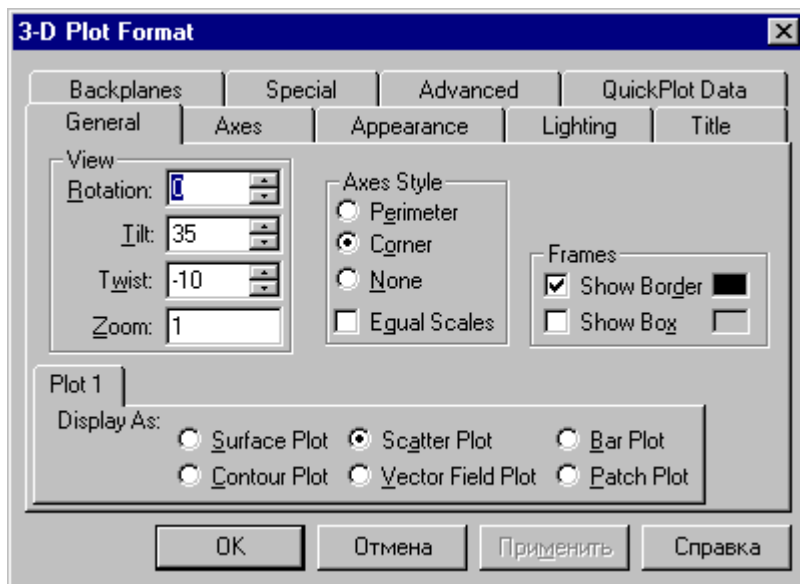
### Внимание!

При построении объемных графиков следует учитывать повышенные требования к ресурсам и опасность "зависания" программы. Сохраните свой документ перед вставкой объемного графика.

## 2.1.5. Модификация объемных графиков

Как было упомянуто, объемный график (кроме линий уровня) можно вращать мышью, рассматривая его со всех сторон. Кроме того, двойной щелчок мышью открывает окно настройки объемного графика. Оно имеет 9 вкладок, рассмотрим те настройки, которые могут быть нам полезны.

Лучшим помощником при изучении данного окна является кнопка "Применить", позволяющая попробовать разные настройки, не закрывая окна.



### 1. General (Общее)

- Переключатель " Display As" внизу определяет вид графика;
- рамка "View" управляет ориентацией (повороты) и масштабом графика;
- рамка " Axes Style" определяет, показываются ли оси, и если показываются, то где;
- рамка " Frames" определяет, будет ли видна рамка вокруг графика.

### 2. Axes (Оси)

- Настройка осей.

### 3. Appearance (Внешний вид)

- Настройка цвета, прозрачности, удаление невидимых линий

### 4. Quick Plot Data

- На этой вкладке можно поменять диапазон аргументов для быстрого графика.

## 2.1.6. График "векторное поле"

Иконка на панели инструментов:



График "векторное поле" строится по комплексной матрице или двум вещественным (в каждой точке необходимы компоненты для двух координат вектора). Практическое применение – моделирование / визуализация полей (физические приложения) и "поле направлений" (при исследовании решений дифференциальных уравнений).

## 2.2. Сложные функции

Как было сказано выше, система Mathcad предоставляет пользователю множество стандартных функций; если их недостаточно, имеется возможность задать свою.

Но при решении ряда задач возникает необходимость в кусочно-гладких или кусочно-непрерывных функциях. Для определения таких функций в Mathcad имеются 4 возможности.

1. Использование встроенных разрывных функций.
2. Использование функции *if*.
3. Использование программной конструкции *if*.
4. Использование логических множителей.

### 2.2.1. Встроенные разрывные функции

Данная группа функций обозначена в Мастере функций как "Piecewise Continuous". Функция Хевисайда (единичная функция) широко применяется для формирования функций сложных сигналов, а также как входная функция при определении переходной характеристики устройства:

$$\Phi(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad \text{функция Хевисайда ("ступенька")}$$

Примеры ее использования для задания функции из двух ветвей:

Функция	Запись со "ступенькой"
$\begin{cases} A(x), & x \geq 0 \\ B(x), & x < 0 \end{cases}$	$A(x) \Phi(x) + B(x) (1-\Phi(x))$
$\begin{cases} A(x), & x > 0 \\ B(x), & x \leq 0 \end{cases}$	$A(x) (1-\Phi(-x)) + B(x) \Phi(-x)$
$\begin{cases} A(x), & x \geq t \\ B(x), & x < t \end{cases}$	$A(x) \Phi(x-t) + B(x) (1-\Phi(x-t))$

Другие функции этой группы включают в себя функцию  $\text{sign}(x)$ , дельта-функцию и функцию *if*, описанную в следующем пункте.

### 2.2.2. Функция if

Данная функция описывается следующим образом:

$$\text{if}(\text{cond}, x, y) = \begin{cases} x, & \text{если условие } \text{cond} \text{ выполнено} \\ y & \text{иначе} \end{cases}$$

Пример использования:

Функция	Запись через "if"
$\begin{cases} A(x), & x \geq t \\ B(x), & x < t \end{cases}$	$\text{if}(x \geq t, A(x), B(x))$

Для записи условий используются знаки отношений. Их можно ввести с палитры "Boolean":





Обратите внимание, что знак равенства (жирное "=") для проверки условия отличается и от операции присваивания (:=), и от оператора вычисления ("=") и вводится нажатием Ctrl= . Знаки ">", "<" можно вводить и с клавиатуры.

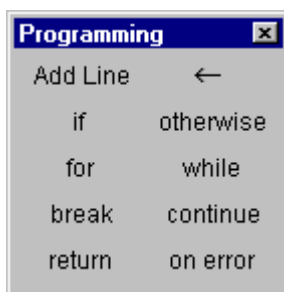
Для записи более сложных функций возможно использование вложенных конструкций.

Пример использования вложенной конструкции *if* (функция, заданная тремя фрагментами):

Функция	Запись через "if"
$\begin{cases} A(x), & x < a \\ B(x), & a \leq x < b \\ C(x), & b \leq x \end{cases}$	$if(x < a, A(x), if(x < b, B(x), C(x)))$

### 2.2.3. Программная конструкция **if**

Данный способ использует возможности программирования, имеющиеся в версиях MathCad с суффиксом "Pro". При этом создается функция, заданная в виде программного блока, который "собирается" из шаблонов панели "Programming".



Приведем последовательность действий для задания функции, упомянутой ранее:

$$\begin{cases} A(x), & x < a \\ B(x), & a \leq x < b \\ C(x), & b \leq x \end{cases}$$

1. На свободном месте листа начнем определять функцию, набрав "f(x):=".
  2. Функция задана тремя строками. Нажимаем два раза на панели "Add Line" (добавить строку), получаем три строки.
  3. Ставим курсор в шаблон ввода на верхней строке и нажимаем кнопку "if". На данном этапе должна получиться следующая картинка:
 

$f(x) :=$	if	
  4. В строке с if правое поле – для условия, левое – для выражения, которое будет значением функции при выполнении этого условия. В данном случае условие первой строки " $x < a$ ".
  5. Повторяем действия пунктов 3 и 4 для второй строки; здесь условие " $x < b$ " (допускается и запись вида " $a \leq x < b$ ").
  6. Ставим курсор в третью строку. Здесь вместо условия нажимаем на кнопку "otherwise" (иначе) и дополним его выражением для функции в случае, когда ни одно из условий не выполнено.
- Результат проделанных операций:

Функция	Запись через "if"
$\begin{cases} A(x), & x < a \\ B(x), & a \leq x < b \\ C(x), & b \leq x \end{cases}$	$f(x) := \begin{cases} A(x) & \text{if } x < a \\ B(x) & \text{if } x < b \\ C(x) & \text{otherwise} \end{cases}$

Как видно, полученная запись похожа на естественную, что является достоинством. Второе достоинство заключается в легкости модификации для функции, заданной на большем количестве участков.

**Внимание!**

Следует иметь в виду, что набирать слова *if* и *otherwise* с клавиатуры нельзя – необходимо использовать кнопки панели "Programming".

### 2.2.4. Использование логических множителей

Данная возможность основана на том, что логические выражения в MathCad вычисляются как числа, причем значением "Истина" является 1, а "Ложь" – 0. Все, что остается сделать – найти выражения, истинные в каждой из областей задания функции, а они как раз нам даны.

Пример:

Функция	Запись с логическими множителями
$\begin{cases} A(x), & x < a \\ B(x), & a \leq x < b \\ C(x), & b \leq x \end{cases}$	$(x < a) * A(x) + (a \leq x < b) * B(x) + (b \leq x) * C(x)$

Нам представляется более удобным использовать функцию if для функций, заданных на двух участках, и программную конструкцию if – для более сложных функций.

### 2.3. Решение уравнений и систем

Решение уравнений и систем – одна из наиболее часто встречающихся в инженерной практике задач.

Изначально система MathCad была ориентирована на получение численного результата, и для численного решения подобных задач предлагаются следующие средства.

#### 2.3.1. Функция root

Функция root предназначена для поиска корня функции одной переменной (точнее, поиска корня по одной переменной). В случае нескольких переменных одна переменная меняется, значения других должны быть фиксированы.

Форма записи:

$$\text{root}(f(x), x) \quad \text{или} \quad \text{root}(f(x), x, x_{\min}, x_{\max}).$$

Здесь:

$f(x)$  – функция, ноль которой мы ищем (может быть выражением),

$x$  – переменная, по которой ищется корень,

$[x_{\min}, x_{\max}]$  – интервал поиска.

В случае, когда интервал не задан, необходимо установить начальное значение для поиска корня ("догадку").

**Обратите внимание:** у функции необходимо указывать список аргументов.

### Случай функции одной переменной

**Задача.** Найти корень функции  $\sin(x)$  на интервале  $[1, 5]$ .

а) поиск от начальной точки

$g(x) := \sin(x)$                       Задаем ф-цию

$x := 3$                                       Задаем начальную точку (догадку)

$\text{root}(g(x), x) = 3.143$                   Получаем результат

б) поиск на интервале

$\text{root}(\sin(x), x, 1, 5) = 3.142$         Получаем результат моментально

### Случай функции нескольких переменных

Функция `root` осуществляет поиск корня по одной переменной. Значения других переменных выполняют роль параметра и должны быть фиксированы.

**Задача.** Найти ноль функции  $f(x,y) = x^2 + y^2 - 1$  при  $y = 0.5$ , интервал поиска по  $x$  задан  $[0, 1]$ .

$f(x,y) := x^2 + y^2 - 1$                       Функция двух переменных

$y := 0.5$                                       Значение второй переменной фиксировано

$\text{root}(f(x,y), x, 0, 1) = 0.866$               Находим значение  $x$ , обращающее ф-цию в 0

$f(0.866, 0.5) = -4.4 \times 10^{-5}$               Проверяем: действительно ноль функции

### Типичные ошибки (возможные неполадки):

Если корня нет вообще, получим комплексное значение корня. В большинстве случаев такой результат для нас смысла не имеет.

$f(x) := \sin(x) + 1.2$

$x := 2$

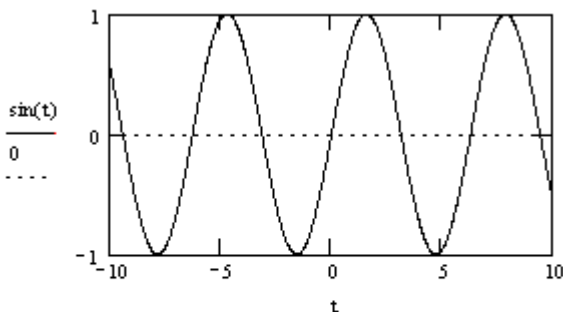
$\text{root}(f(x), x) = 4.712 + 0.622i$

Если начальное значение задано около экстремума функции, можно получить далекое от начальной точки значение или не получить вовсе.

$x := 0$

$\text{root}(\cos(x), x) = 1.459 \times 10^3$

Если интервал задан "неправильно", корень найден не будет.  
"Неправильно" не обязательно означает, что корня на интервале нет.



$\text{root}(\sin(x), x, -5, 2) = \begin{bmatrix} -3.14159 \\ 3.14159 \end{bmatrix}$

The function values on the two bracket end points must be of opposite signs.

В данном примере интервал содержит два корня. Из сообщения об ошибке видно, что для поиска корня функция на концах интервала должна иметь разные знаки.

Все эти ситуации, по существу, предотвращаются очень просто: сначала следует посмотреть на график функции.

### 2.3.2. Конструкция Given / Find

В блоке Given (блок "Дано", блок решения) группируются уравнения и / или неравенства для их совместного решения.

Общая форма записи:

Given

уравнения и/или ограничения

решающая функция

Решающие функции возвращают результат в виде вектора.

Решающие функции:

- Find – точное решение уравнений и систем;

- `minerr` – приближенное решение уравнений и систем;
  - `Maximize / Minimize` – решение задач оптимизации.
- Первые две функции описаны в данной главе, остальные – в разделе "Решение задач оптимизации" (п. 2.4).

Не могут быть включены в блок решения:

- переменные-диапазоны;
- ограничения со знаком  $\neq$  (не равно);
- присваивания (`:=`) (но возможна вставка ограничений-равенств, `Ctrl-=`);
- другие блоки решения.

Для численного решения обязательно задание начальных значений переменным. Общая форма записи для блока `Given / Find`:

начальные значения для переменных

`Given`

уравнения и/или неравенства

`Find` (список переменных через запятую)=

( или вектор:= `Find`(список переменных через запятую) )

Приведем примеры использования блока `Given` для численного решения систем уравнений.

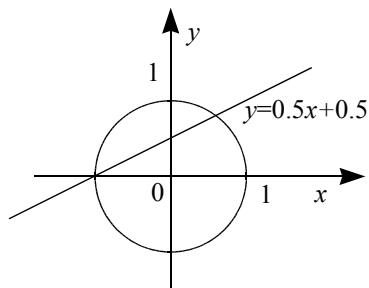
### Решение системы уравнений

Задача. Решить систему

$$\begin{cases} x^2 + y^2 = 1 \\ y = 0.5x + 0.5 \end{cases}$$

### Решение

Данная задача имеет простую геометрическую интерпретацию: пересечение прямой и окружности в двух точках.



### Внимание!

При наборе уравнений в блоке Given знак равенства вводится как Ctrl+= ("жирное равно").

$$x := 1$$

$$y := 1$$

Given

$$x^2 + y^2 = 1$$

$$y = 0.5 \cdot x + 0.5$$

$$\text{Find}(x, y) = \begin{pmatrix} 0.6 \\ 0.8 \end{pmatrix}$$

### Решение системы уравнений с наложением ограничений

А теперь добавим ограничение-неравенство, чтобы найти другую точку:

$$x := 1$$

$$y := 1$$

Given

$$x^2 + y^2 = 1$$

$$y = 0.5 \cdot x + 0.5$$

$$x < 0$$

$$\text{Find}(x, y) = \begin{pmatrix} -1 \\ -6.823 \times 10^{-8} \end{pmatrix}$$

### Использование результата решения в качестве функции

**(функция, заданная неявно)**

Воспользуемся блоком решения, чтобы использовать  $y(x)$  как функцию, не выражая ее явно из уравнения  $g(x, y)=0$ . Для этого:

- задаем начальное значение для  $y$ ;
- записываем условие, связывающее  $x$  и  $y$ ;
- присваиваем возвращаемое функцией Find значение нашей функции  $f(x)$ .

После этого функцией  $f(x)$  можно пользоваться как любой другой – вычислять, строить график... Но поскольку вычисление функции в каждой точке требует решения уравнения, построение графика может занять некоторое время.

**Пример**

$y := 1$

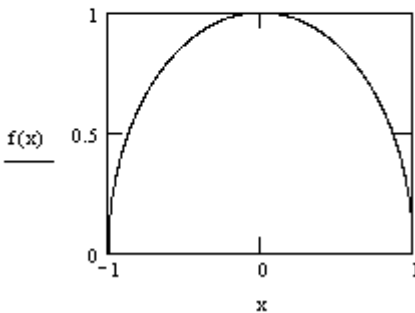
Задали начальное значение

Given

$$x^2 + y^2 = 1$$

Связали  $x$  и  $y$  уравнением; не выражаем  $y$ , даже не приводим уравнение к виду  $g(x,y)=0$

$f(x) := \text{Find}(y)$



Пользуемся результатом как функцией.

**2.3.3. Точность вычислений**

Нахождение корней численными методами производится с некоторой точностью. Система MathCad позволяет эту точность менять.

В пакете имеются встроенные переменные TOL и CTOL, задающие точность численных расчетов. Переменная TOL (Tolerance – допуск) определяет точность вычисления интегралов и производных, а также определяет условия окончания вычислений для алгоритмов нахождения корней (функции root и конструкции Given/Find).

Значение TOL по умолчанию 0.001; уменьшение TOL приводит к увеличению точности (и, соответственно, времени расчета). Кроме того, увеличивается и вероятность не найти решение.

Точность нахождения корня и отображаемая на экране разрядность не связаны. Тем не менее, чтобы увидеть влияние повышения точности расчетов, следует увеличить число отображаемых позиций после запятой (меню Format | Result | Number of decimal places, п.1.1).



### Влияние значения переменной TOL на нахождение корня.

**Задача.** Найти корень функции  $f(x) = \sin^2 x - 0.3$  в окрестности  $x = 3$  с точностью  $10^{-10}$ .

$TOL = 1 \times 10^{-3}$                       Значение по умолчанию

$f(x) := \sin(x)^2 - 0.3$

$x := 3$

$r := \text{root}(f(x), x)$

Отображение значений по умолчанию (с 3-мя цифрами после запятой)

$r = 2.562$                        $f(x) = 2.13 \times 10^{-4}$

Отображаем цифр побольше

$r = 2.56172056052496$

$TOL := 1 \cdot 10^{-10}$                       Уменьшаем допуск (увеличиваем точность)

$r := \text{root}(f(x), x)$

Отображение значений по умолчанию показывает то же  $r$ ,  
а вот значение функции уже гораздо ближе к нулю.

$r = 2.562$                        $f(x) = -5.573 \times 10^{-11}$

Отображаем цифр побольше - вот она разница!

В данном случае, с 4-го знака

$r = 2.5619529132869$

Как мы видим, значение TOL определяет величину допуска  
на значение функции в найденной точке (корне).

Переменная STOL (constraint tolerance – допуск по ограничению) определяет требуемую точность выполнения ограничений в блоке решения (Given/...). Функция, заданная блоком решения (minimize, maximize, find или minerr), возвращает значение, только если все ограничения выполнены; например, ограничение " $x < 2$ " при значении STOL = 0.001 (по умолчанию) считается выполненным, если  $x < 2.001$ .

#### 2.3.4. Конструкция Given / minerr

Форма записи:

начальные значения для переменных  
Given

уравнения и/или неравенства  
minerr(список переменных через запятую)=  
( или вектор:= minerr(список переменных через запятую) )

Функция minerr возвращает значения переменных, при которых уравнения / неравенства в блоке решения выполняются наилучшим образом. Отличие от конструкции Given / Find состоит в том, что при невозможности найти точное (в пределах TOL и CTOL) решение функция Find выдаст ошибку, а функция minerr – наиболее подходящее значение. В справочной системе MathCad рекомендуется использовать эту функцию, если точное решение с помощью Find найти не удалось.

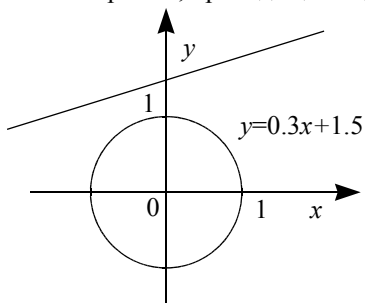
### Пример

Решить систему уравнений

$$\begin{cases} x^2 + y^2 = 1 \\ y = 0.3x + 1.5 \end{cases}$$

### Решение

Нетрудно видеть, что система несовместна: это уравнение единичной окружности и прямой, проходящей над ней.



Тем не менее, функция minerr пытается подобрать наилучшие значения (вектор Z). Подстановка этих значений ( $Z_0, Z_1$ ) в систему (последние две строки) показывает существенную разницу.

$$x := 2 \quad y := 2$$

Given

$$x^2 + y^2 = 1$$

$$y = 0.3 \cdot x + 1.5$$

$$Z := \text{minerr}(x, y)$$

$$Z_0 = -0.121 \quad Z_1 = 1.055$$

$$(Z_0)^2 + (Z_1)^2 = 1.128$$

$$0.3 \cdot Z_0 + 1.5 - Z_1 = 0.409$$

### 2.3.5. Специальные виды уравнений и систем

#### Корни полинома

Система MathCad предлагает функцию для нахождения всех корней полинома. Часть корней может оказаться комплексными.

Функция `polyroots(v)` возвращает вектор, содержащий все корни; значение на входе – вектор коэффициентов полинома, начиная от младших степеней.

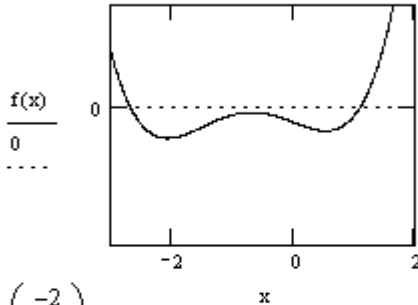
#### Пример

Найти корни полинома  $x^4 + 3x^3 - 3x - 2$ .

#### Решение

Судя по графику, полином имеет два действительных корня. При записи вектора коэффициентов не забываем поставить ноль на месте отсутствующего члена (при  $x^2$ ).

$$f(x) := x^4 + 3 \cdot x^3 - 3 \cdot x - 2$$



$$v := \begin{pmatrix} -2 \\ -3 \\ 0 \\ 3 \\ 1 \end{pmatrix} \quad \text{polyroots}(v) = \begin{pmatrix} -2.688 \\ -0.7 + 0.44i \\ -0.7 - 0.44i \\ 1.088 \end{pmatrix}$$

### Линейные системы

Система MathCad предлагает функцию для решения систем линейных уравнений в матричной форме.

Функция  $\text{lsolve}(M, v)$  возвращает вектор решения линейной системы  $Mx = v$ .  $M$  – квадратная матрица коэффициентов (невырожденная – ее определитель не должен быть равен нулю),  $v$  – вектор правых частей системы.

#### Пример

Решить систему

$$\begin{cases} x_1 + 2x_2 + 3x_3 = 1 \\ -x_1 + \frac{x_2}{2} + x_3 = 2 \\ 2.5x_1 + 2x_2 - x_3 = -0.5 \end{cases}$$

#### Решение

Введем матрицу коэффициентов и проверим, что она невырожденная:

$$M := \begin{pmatrix} 1 & 2 & 3 \\ -1 & \frac{1}{2} & 1 \\ 2.5 & 2 & -1 \end{pmatrix} \quad v := \begin{pmatrix} 1 \\ 2 \\ -0.5 \end{pmatrix}$$

$|M| = -9.25$  не ноль - решение есть

$$\text{lsolve}(M, v) = \begin{pmatrix} -1.432 \\ 1.459 \\ -0.162 \end{pmatrix}$$

### 2.3.6. Средства для символьного решения уравнений и систем

Кроме численных средств (получение решения в виде чисел с какой-то точностью), система располагает возможностями для решения в символьном виде (получение точного решения, в виде формулы). См. пункт Символьные вычисления (п. 2.6).

## 2.4. Решение задач оптимизации

Типовая задача оптимизации состоит в поиске экстремума (наименьшего / наибольшего значения функции). Обычно на практике такие задачи сопровождаются наложением дополнительных условий или ограничений.

Система MathCad предлагает средства для решения следующих задач оптимизации:

- нахождение минимума / максимума функции одной переменной без ограничений;
- нахождение минимума / максимума функции нескольких переменных без ограничений;
- решение задачи оптимизации с ограничениями.

Рассмотрим все три случая (по нарастанию сложности).

### 2.4.1. Оптимизация функции одной переменной без ограничений

Для решения данной задачи используются функции `Maximize` и `Minimize` (для поиска наибольшего и наименьшего значений соответственно).  
Форма записи:

`Maximize(f, x)`

`Minimize(f, x)`

Первый аргумент – функция, экстремум которой ищется; причем указывается только имя функции, без списка аргументов (в отличие от функции `root`). Функция должна быть задана ранее, использование вместо имени функции выражения не допускается.

Второй – переменная, по которой ищется оптимум. Переменной должно быть присвоено начальное значение.

Данные функции возвращают значение аргумента, при котором функция  $f$  принимает оптимальное значение.

Последовательность действий:

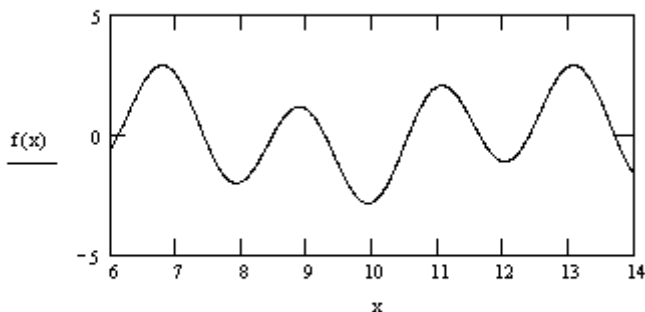
1. Задать функцию.
2. Задать начальное значение для поиска оптимума.
3. Записать оператор поиска экстремума.

#### Пример

Найти максимум функции  $f(x) = 2 \sin 3x + \cos x$ .

Начальное значение возьмем равным 9 (из графика).

$$f(x) := 2 \cdot \sin(3x) + \cos(x)$$



`x := 9`

`Maximize(f, x) = 8.872`

В данном примере величина 8.872 является точкой локального максимума (значение функции в точке  $x = 7$  явно больше). Для того, чтобы узнать значение функции в найденной точке, нужно присвоить результат какой-либо переменной и затем подставить в функцию:

$$x_{\text{opt}} := \text{Maximize}(f, x) \quad x_{\text{opt}} = 8.872 \quad f(x_{\text{opt}}) = 1.141$$

Для поиска глобального максимума (минимума) на заданном участке можно порекомендовать визуально оценить корректность ответа и при необходимости повторить поиск экстремума с другим начальным значением, например:

$$x := 7$$

$$x_{\text{opt}} := \text{Maximize}(f, x) \quad x_{\text{opt}} = 6.78 \quad f(x_{\text{opt}}) = 2.873$$

$$x := 13$$

$$x_{\text{opt}} := \text{Maximize}(f, x) \quad x_{\text{opt}} = 13.063 \quad f(x_{\text{opt}}) = 2.873$$

#### 2.4.2. Оптимизация функции нескольких переменных без ограничений

В случае наличия у функции  $f$  нескольких аргументов, например,  $f(x, y, z)$ , функции `Maximize` и `Minimize` записываются следующим образом:

`Maximize(f, x, y, z)`

`Minimize(f, x, y, z)`

Переменным должны быть присвоены начальные значения.

Возвращаемое значение – вектор.

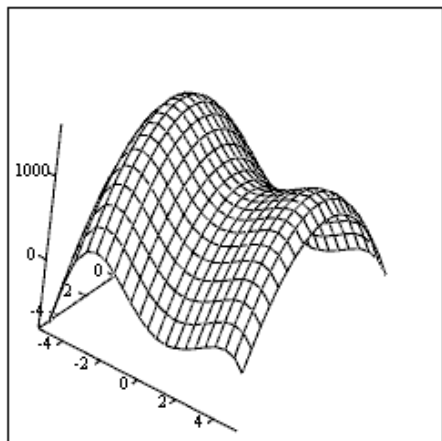
Не стоит забывать, что функция находит локальный экстремум; правдоподобность результата следует оценить. В частности, для случая двух переменных можно построить график поверхности.

##### Пример

Поиск максимума функции двух переменных в области

$x \in [-5, 5], y \in [-5, 5]$ .

$$g(x, y) := 1000 - 3 \cdot x^4 + 8 \cdot x^3 + 66 \cdot x^2 - 144 \cdot x - 50 \cdot y^2$$



g

$$x := 5 \quad y := 5$$

$$\begin{pmatrix} x_{\text{opt}} \\ y_{\text{opt}} \end{pmatrix} := \text{Maximize}(g, x, y)$$

$$x_{\text{opt}} = 4 \quad y_{\text{opt}} = 6.736 \times 10^{-9}$$

$$g(x_{\text{opt}}, y_{\text{opt}}) = 1.224 \times 10^3$$

Другое начальное значение:

$$x := -5 \quad y := -5$$

$$\begin{pmatrix} x_{\text{opt}} \\ y_{\text{opt}} \end{pmatrix} := \text{Maximize}(g, x, y)$$

$$x_{\text{opt}} = -3 \quad y_{\text{opt}} = 6.559 \times 10^{-8}$$

$$g(x_{\text{opt}}, y_{\text{opt}}) = 1.567 \times 10^3$$

### 2.4.3. Задачи оптимизации с ограничениями

Простейший случай ограничений – это ограничение области поиска (типа  $0 < x < 1$ ). На практике встречаются и более сложные ограничения; они возможны как в виде неравенств, так и в виде равенств; в последнем случае знак равенства вводится как Ctrl+= или с панели инструментов "Boolean".

Для учета ограничений в процессе оптимизации используется блок решения (Given). Последовательность действий:

1. Задать функцию.
2. Задать переменным начальные значения.
3. Записать слово Given, задающее начало блока решения.
4. Задать ограничения.
5. Записать функцию Maximize (или Minimize), завершающую блок решения.



### Пример

Найдем точку, лежащую на единичной окружности (ограничение-равенство) в левой полуплоскости (ограничение-неравенство) и наиболее удаленную от точки (2, 1) (расстояние – функция, подлежащая оптимизации).

$$d(x, y) := \sqrt{(x - 2)^2 + (y - 1)^2}$$

$$x := 1 \quad y := 0$$

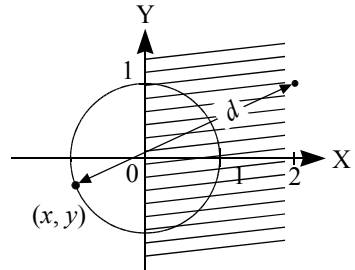
Given

$$x^2 + y^2 = 1$$

$$x < 0$$

$$\begin{pmatrix} x_s \\ y_s \end{pmatrix} := \text{Maximize}(d, x, y)$$

$$x_s = -0.894 \quad y_s = -0.447$$



### Внимание!

Наличие несовместимых ограничений может привести к невозможности решить задачу:

Given

$$x^2 + y^2 = 1$$

$$x < -2$$

$$\begin{pmatrix} x_s \\ y_s \end{pmatrix} := \text{Maximize}(d, x, y)$$

$x_s =$  ■

No solution was found. Try changing the guess value or the value of TOL or CTOL.

## 2.5. Работа с векторами и матрицами

### 2.5.1. Задание векторов и матриц

Массив в MathCad, как и переменная, задается с помощью имени. Массив состоит из отдельных элементов. Порядковый номер элемента определяется одним или несколькими индексами. Нижняя граница индекса задается системной переменной ORIGIN, которой можно присвоить произвольное целое число (по умолчанию 0; для привычной работы с матрицами имеет смысл задать 1). Индексы могут иметь только целочисленные значения.

Значение переменной ORIGIN устанавливается оператором присваивания, например:

ORIGIN:=1

Элементы массива являются индексированными переменными. Это значит, что кроме имени такие переменные имеют один или несколько подстрочных индексов. Индексы вводятся после набора на клавиатуре левой квадратной скобки – < [ > (или с панели инструментов).

#### Внимание!

Имеется опасность визуально спутать индекс вектора (матрицы) с индексом в имени скалярной переменной (вводится через точку). После нажатия клавиши < [ > (для ввода индекса матрицы) появляется шаблончик индекса; после нажатия клавиши < точка > (для ввода описательного индекса) курсор ввода смещается вниз, шаблончик не появляется:



для ввода  $x_j$



для ввода  $x_{min}$

В матрице первый индекс обозначает номер строки, второй – номер столбца, индексы разделяются запятой.

#### Пример

Ввести элемент матрицы – индексированную переменную  $z_{ij}$ , прибавить единицу, напечатать результат  $z_{i,j} + 1$ .

Последовательность нажатий клавиш:

< z > < [ > < i > < , > < j > < Пробел > < Пробел > < + > < 1 > < = >

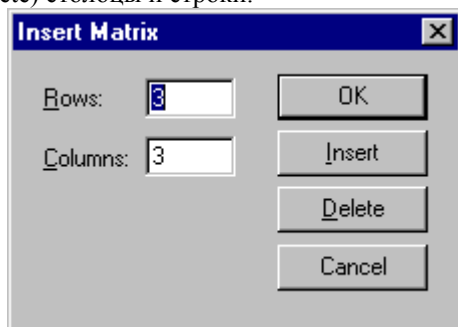
По нажатию клавиши < [ > курсор смещается вниз, по нажатию пробелов – выходит на основной уровень, иначе единица будет добавлена к индексу.

Задать массив можно тремя способами:

1. Через пункт меню Insert | Matrix
2. Через инструментальную панель "Matrix"

### 3. Комбинацией клавиш Ctrl-M

В результате на экране возникает окно, в котором имеется возможность изменить число строк (Rows) и столбцов (Columns). По умолчанию создается массив размером 3×3. Это же окно позволяет вставить (Insert) или удалить (Delete) столбцы и строки.



Нажатие кнопки OK вставляет шаблон ввода значений массива. Отметим, что одномерный массив (вектор) задается как матрица, имеющая один столбец.



Перемещаться между значениями при вводе можно с помощью мыши, клавиши < Tab > и клавиш управления курсором ("стрелочек").

Формировать матрицу можно присвоением значений ее элементам. В этом случае размеры матрицы можно не задавать. Всем неопределенным элементам автоматически присваиваются нулевые значения. Например, при выполнении присваивания

$$M_{5,5}:=1$$

создается матрица размером 6×6, у которой все элементы, кроме  $M_{5,5}$ , равны нулю.

Приведем несколько примеров.

**Пример 1.** Влияние системной переменной ORIGIN.

Задаем массив:

$$a := \begin{pmatrix} 2.5 \\ 3 \\ -15 \end{pmatrix}$$

По умолчанию: (нумерация с нуля)

Выведем значения элементов вектора:

$$a_0 = 2.5 \quad a_1 = 3 \quad a_2 = -15 \quad a_3 = \blacksquare$$

( $a_3$  не определено)

С переменной ORIGIN: (нумерация с единицы)

$$\text{ORIGIN} := 1$$

$$a_0 = \blacksquare \quad a_1 = 2.5 \quad a_2 = 3 \quad a_3 = -15$$

( $a_0$  не определено)

**Пример 2.** Индексация матрицы.

Задаем матрицу:

$$\text{ORIGIN} := 1$$

$$a := \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

Выведем значения элементов матрицы:

$$a_{1,1} = 1 \quad a_{1,2} = 2$$

$$a_{2,1} = 3 \quad a_{2,2} = 4$$

## 2.5.2. Ввод/вывод матриц из внешних источников

Вводя матрицу вручную, нельзя создать матрицы с числом элементов больше 100. Для работы с матрицами больших размеров система MathCAD предоставляет ряд возможностей.

Простейшая из них – чтение массива данных из текстового файла функцией READPRN. Аргументом функции является имя файла в кавычках; файл должен содержать строки матрицы, числа в строке отделяются пробелами или символами табуляции. Функция возвращает массив. Если данные в файле изменились, обновить данные в матрице можно, установив курсор в блок с данной функцией и нажав F9 (или выбрав из меню Math | Calculate).

Парной функцией к READPRN является WRITEPRN. Эта функция используется как левая часть (цель) операции присваивания.

### Пример

Введем матрицу, найдем обратную и запишем результат в файл.

```
m := READPRN("D:\DATA\MATR1.TXT")
```

$$m = \begin{pmatrix} 0.544 & -0.412 & -0.989 \\ -0.412 & -0.989 & -0.657 \\ -0.989 & -0.657 & 0.279 \end{pmatrix}$$

$$n := m^{-1}$$

```
WRITEPRN("D:\DATA\MATR2.TXT") := n
```

Исходный файл (D:\DATA\MATR1.TXT):

```
0.544 -0.412 -0.989
-0.412 -0.989 -0.657
-0.989 -0.657 0.279
```

Файл результата (D:\DATA\MATR2.TXT):

```
2122 -2294 2122
-2294 2479 -2294
2122 -2294 2123
```

Для ввода данных из файлов других форматов (импорт) имеется специальный компонент. Порядок использования:

1. Помещаем курсор ввода на свободный участок документа
2. Выбираем из меню Insert | Component (Вставить / Компонент)
3. Выбираем Input Table, нажимаем Finish. Появляется шаблончик:

$n :=$

	0	1
0	0	
1		

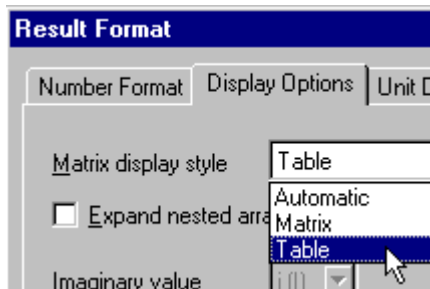
4. В поле вводим имя матрицы, в которую будут помещены данные
5. Правой кнопкой мыши вызываем контекстное меню, выбираем Import
6. Выбираем тип файла (в том числе доступен формат Excel), выбираем файл.

Готово, данные получены:

$n :=$

	0	1
9	0.46	0.56
10	0.13	0.92

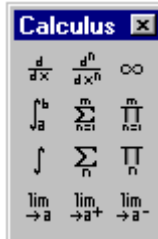
Обратная операция (экспорт) осуществляется выбором пункта Export из контекстного меню таблицы. Если матрица отображается не таблицей (как в приведенных примерах) а матрицей (со скобками), необходимо изменить способ отображения матрицы. Для этого нужно двойным щелчком на матрице вызвать окно форматирования (или через меню Format/Result), выбрать вкладку DisplayOptions и установить пункт Matrix Display Style в Table.



### 2.5.3. Вычисление сумм и произведений

Система MathCAD позволяет очень просто вычислять суммы и произведения некоторого множества величин с помощью расширенных операторов. Основным их достоинством является близость формы записи к математической.

Для вычисления сумм и произведений используют панель инструментов "Calculus".

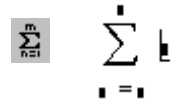


#### Пример

Вычислить значение суммы  $\sum_{i=1}^{10} \frac{1}{i^2}$ .

Последовательность действий:

1. Выбираем на панели соответствующий шаблон. Получаем шаблон для заполнения в документе.



2. Заполняем шаблон конкретными значениями:

$i$ , 1, 10 и  $\frac{1}{i^2}$ .

$$\sum_{i=1}^{10} \frac{1}{i^2}$$

3. Нажимаем клавишу <=>. Получаем результат.

$$\sum_{i=1}^{10} \frac{1}{i^2} = 1.55$$

Следующий вид шаблона суммирования применяется, если индекс задан переменной-диапазоном. В частности, это удобно при работе с векторами и матрицами.



#### Пример

Вычислить сумму значений вектора с использованием оператора суммы. Решение:

$$V := \begin{pmatrix} 1 \\ 10 \\ -3 \end{pmatrix} \quad \text{Задаем вектор}$$

$$\text{ORIGIN} := 1 \quad \text{Задаем нумерацию элементов с 1}$$

$$i := 1..3 \quad \text{Задаем индекс}$$

$$\sum_i V_i = 8 \quad \text{Заполняем шаблон, получаем результат}$$

Часто бывает удобно при вычислении суммы задать ее общий член в виде пользовательской функции от индекса.

### Пример

Вычислить сумму  $\sum_{j=1}^{100} \frac{1}{j^2 + 2j + 10}$

Решение:

$$f(j) := \frac{1}{j^2 + 2 \cdot j + 10} \quad \text{Задаем функцию (общий член суммы)}$$

$$j := 1..100 \quad \text{Задаем индекс}$$

$$\sum_j f(j) = 0.358 \quad \text{Заполняем шаблон, получаем результат}$$

### Вычисление произведений

Процесс вычисления произведений аналогичен вычислению суммы. Используются два вида шаблонов – полный и сокращенный (по переменной-диапазону). Общий член произведения можно задавать и в виде пользовательской функции.



### Пример

Вычислить произведение  $\prod_{i=3}^{100} 1 + \frac{-1^i}{4i-2}$ .

Решение.

$$\prod_{i=3}^{100} \left[ 1 + \frac{(-1)^i}{4 \cdot i - 2} \right] = 0.929$$

Как видно, форма записи произведения практически не отличается от математической.



## 2.6. Символьные вычисления

Несмотря на то, что изначально MathCad была ориентирована на получение численного результата, развиваясь, система приобрела способность к символьным вычислениям. Символьные вычисления – это операции математических преобразований над выражениями, не учитывающие конкретных числовых значений.

В системе MathCad существуют два варианта выполнения символьных преобразований: "живые" символьные вычисления и команды меню Symbolic. Мы рассмотрим только самые необходимые операции.

### 2.6.1. "Живые" символьные вычисления

Самый простой способ объяснить системе, что мы желаем получить ответ в символьном виде – это поставить в конце вместо знака равенства (=) знак символьного вычисления ( $\rightarrow$ , вводится как Ctrl-. (точка)).

Это позволяет:

- выполнять сложение дробей:

$$\frac{1}{2} - \frac{1}{3} = 0.167 \qquad \frac{1}{2} - \frac{1}{3} \rightarrow \frac{1}{6}$$

- решать уравнения и системы в численном виде:

Given

$$x^2 + x + 2 = 0$$

$$\text{Find}(x) \rightarrow \left( \frac{-1}{2} + \frac{1}{2} \cdot i \cdot \sqrt{7} \quad \frac{-1}{2} - \frac{1}{2} \cdot i \cdot \sqrt{7} \right)$$

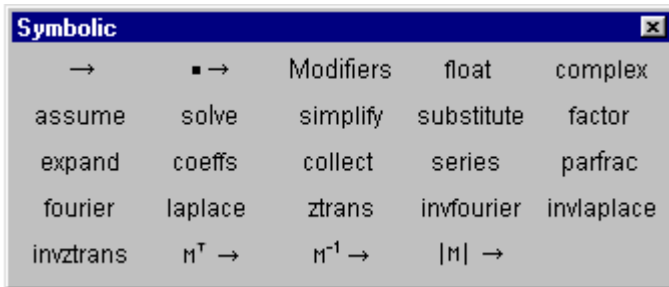
- находить производные:

$$\frac{d}{dx} (\sin(x)^2 - \ln(3 \cdot x)) \rightarrow 2 \cdot \sin(x) \cdot \cos(x) - \frac{1}{x}$$

- вычислять неопределенные интегралы:

$$\int \sin(x)^2 dx \rightarrow \frac{-1}{2} \cdot \sin(x) \cdot \cos(x) + \frac{1}{2} \cdot x$$

Кроме этих простейших случаев, специальная панель инструментов позволяет произвести в этом режиме большое количество операций.



Особенностью режима живых символьных вычислений является то, что система учитывает значения переменных. Если нам этот эффект мешает, можно определить переменную рекурсивно ( $x:=x$ ).

### Пример

Решаем уравнение  $\cos(x) = \sin(y)$  относительно  $y$ .

Первые две строки: заданное ранее значение  $x$  влияет на результат. В третьей строке переменная  $x$  рекурсивно определяется, и последняя строка выдает нам искомую зависимость  $y(x)$ .

$x := 0$

$$\cos(x) = \sin(y) \text{ solve, } y \rightarrow \frac{1}{2} \cdot \pi$$

$x := x$

$$\cos(x) = \sin(y) \text{ solve, } y \rightarrow \frac{1}{2} \cdot \pi - x$$

## 2.6.2. Команды меню Symbolic

Все операции, доступные в режиме живых символьных вычислений, доступны и через меню Symbolic. Отличие состоит в том, что значения переменных при выполнении преобразований в расчет не принимаются.

Рассмотрим наиболее часто используемые команды меню.

- Evaluate | Symbolically (Символьно)  
Вычислить символически. В общем, дает то же, что и вычисление в живом символьном режиме (нажатие Ctrl-точки).
- Evaluate | Floating Point... (С плав. точкой)

Вычислить в заданном числом знаков после запятой. Возможное значение – до 4000 знаков.

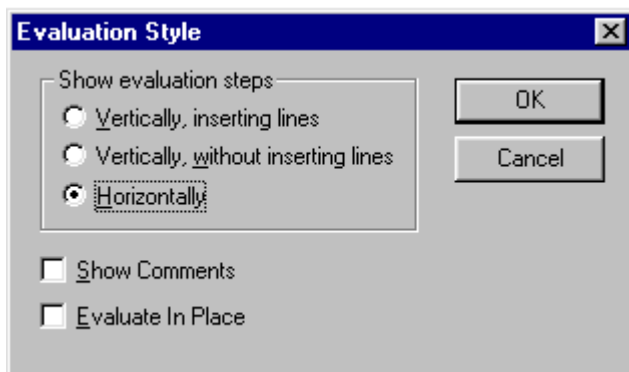
Следующий фрагмент показывает результаты применения двух описанных команд (вторая и третья строки). Число знаков задано равным 30.

$$\text{acos}(0) = 1.571$$

$$\text{acos}(0) \quad \frac{1}{2} \cdot \pi$$

$$\text{acos}(0) \quad 1.57079632679489661923132169164$$

- **Simplify (Упростить)**  
Упростить. Может значительно сократить размер формулы.
- **Variable | Solve (Разрешить)**  
Разрешить уравнения относительно той переменной, на которой стоит курсор
- **Variable | Substitute (Подставить)**  
Подставить вместо переменной, на которой стоит курсор, содержимое буфера обмена. Заменяются все вхождения переменной в формулу.
- **Variable | Differentiate (Продифференцировать)**  
Продифференцировать формулу по переменной, на которой стоит курсор (найти производную)
- **Variable | Integrate (Проинтегрировать)**  
Проинтегрировать формулу по переменной, на которой стоит курсор (найти первообразную)
- **Evaluation Style (Стиль вычислений)**  
Этот пункт меню открывает окно, управляющее расположением результатов численных вычислений через меню Symbolic на экране.



Возможные варианты:

- Vertically, inserting lines – результат помещается под исходной формулой, отделяется чертой
- Vertically, without inserting lines – то же самое без разделителя
- Horisontally – результат помещается справа от исходной формулы
- Show comments – между исходной формулой и результатом записывается, какая операция была выполнена
- Evaluate in place – результат замещает исходную формулу.

## 2.7. Решение дифференциальных уравнений

Дифференциальные уравнения часто являются "сердцем" математической модели, а их решение позволяет исследовать поведение объекта.

MathCad позволяет легко выполнять решение дифференциальных уравнений и их систем численными методами. Полученные результаты также легко (и быстро) отображаются на графиках, что делает процесс исследования математической модели более наглядным.

Пакет MathCad предоставляет ряд средств для решения дифференциальных уравнений и их систем. Мы рассмотрим два, предназначенных для работы с обыкновенными дифференциальными уравнениями.

### 2.7.1. Окружение odesolve

Данное средство предназначено для решения одного обыкновенного дифференциального уравнения, линейного относительно старшей производной. Уравнение и начальные условия должны быть заданы в блоке Given. Уравнение записывается в естественной форме. Начальные условия должны быть представлены в виде  $y(a) = y_1, y'(a) = y_2$ , где  $a$  – начало интервала интегрирования, задается константой. Блок Given должен заканчиваться функцией odesolve, например:

$$y = \text{odesolve}(x, b)$$

Здесь  $x$  – переменная интегрирования,  $b$  – конечная точка интервала интегрирования. Решение дифференциального уравнения возвращается в виде функции  $y$ , определенной на интервале  $[a, b]$ .

#### Пример

Решить дифференциальное уравнение

$$y'' + y' + y = 0$$

на интервале  $[0, 6]$  с начальными условиями  $y(0)=1, y'(0)=0.5$ .

## Решение

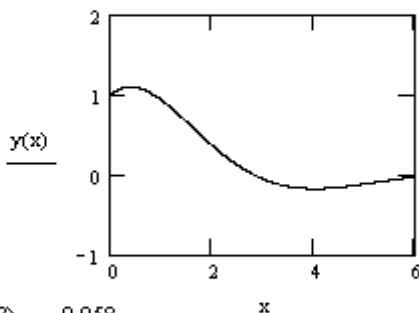
Для окружения `odesolve` функцию (и производные) в уравнении необходимо записывать с указанием аргумента, ( $y(x)$ ,  $y'(x)$  и т. д.). В данном случае исходное уравнение должно быть представлено в виде  $y''(x)+y'(x)+y(x)=0$ . Напоминаем также, что при наборе уравнений знаки равенства вводятся как `Ctrl+=`.

Given

$$y''(x) + y'(x) + y(x) = 0$$

$$y(0) = 1 \quad y'(0) = .5$$

`y := odesolve(x,6)`



$$y(3) = -0.058$$

$$y(6) = -0.028$$

$$y(9) = \blacksquare$$

Argument outside of solution interval.

Мы видим, что в области  $[0, 6]$  можно пользоваться полученной функцией  $y$  как любой другой: можно вычислить значение функции в любой точке интервала, построить ее график и т. п. Попытка же вычислить значение функции за пределами заданного интервала вызывает сообщение "Аргумент вне интервала решения".

### Замечание

При записи дифференциального уравнения производную можно записывать как через апостроф ( $y'$ ,  $y''$  и т. п.), так и с помощью шаблонов с падежами "Calculus" ( $\frac{d}{dx}$ ,  $\frac{d^2}{dx^2}$  и т. п.).

### 2.7.2. Функция `rkfixed`

Функция `rkfixed` предназначена для решения систем обыкновенных дифференциальных уравнения первого порядка, разрешенных относительно производной. Как частный случай, функция `rkfixed` может быть использована и для решения одного уравнения первого порядка. Одно уравнение порядка  $n$  при  $n > 1$  может быть решено после сведения его к системе  $n$  уравнений первого порядка (подробнее см. [9]).

Особенностью данной функции является то, что решение возвращается в виде массива с запрошенным при ее вызове количестве строк (расчитанных точек). Каждая строка содержит значение аргумента и значения рассчитанных в этой точке искомым функций.

Форма записи: `rkfixed(y, x1, x2, noints, D)`

Здесь:

$y$  – вектор начальных условий;

$[x_1, x_2]$  – интервал интегрирования;

`noints` – количество вычисляемых точек (не считая начальной);

$D$  – вектор-функция, вектор правых частей системы уравнений.

Рассмотрим все три варианта использования данной функции.

а). Решение одного уравнения первого порядка, разрешенного относительно производной.

#### Пример

Решить дифференциальное уравнение  $\frac{dy}{dx} = \sin(x) + \frac{1}{y(x)}$

на интервале  $[0, 6]$  с начальными условиями  $y(0)=1$ .

#### Решение

Задаем начальное значение (здесь  $y_0$ ); для одного значения вектор задавать не обязательно. Задаем вектор-функцию  $D$ . Опять-таки, функция у нас только одна. Но поскольку второй аргумент у  $D$  должен быть вектором неизвестных значений функции, писать " $y_0$ " внутри определения функции **ОБЯЗАТЕЛЬНО**. Далее просим посчитать нам на отрезке  $[0, 6]$  100 точек (всего в массиве  $Z$  будет 101 точка: точка номер 0 соответствует начальным условиям). В переменные  $X$  и  $Y$  записываются столбцы мат-

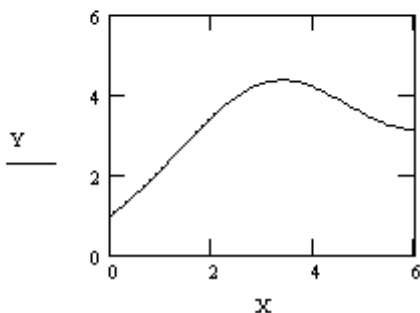
рицы Z, соответствующие значениям аргумента и функции. Оператор выделения столбца <n> берется с панели Matrix (или Ctrl+6).

`y0 := 1`

$$D(x, y) := \sin(x) + \frac{1}{y_0}$$

`Z := rkfixed(y0, 0, 6, 100, D)`

`X := Z <0>      Y := Z <1>`



б). Решение системы уравнений первого порядка, разрешенных относительно производной. Для краткости возьмем два уравнения.

### Пример

Решить систему дифференциальных уравнений первого порядка

$$\begin{cases} y_1' = y_1 + xy_2 \\ y_2' = x - y_1 - y_2 \end{cases}$$

на интервале  $[-3, 6]$

с начальными условиями  $y_1(-3)=1, y_2(-3)=-1$

### Решение

Используем системную переменную ORIGIN. После установки ее в 1 все массивы нумеруются от единицы – в том числе вектор неизвестной функции  $y$ . Таким образом, система для MathCad записывается в практически неизменном виде.

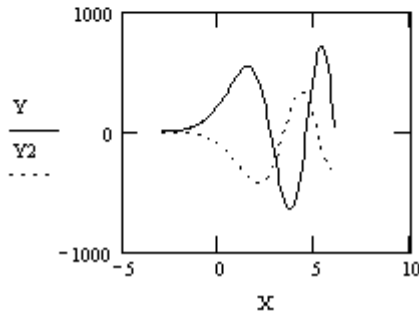
Система записывается в вектор-функцию D. Результат получается в матрице Z. Побочным эффектом изменения ORIGIN является то, что она

тоже нумеруется от 1 (и соответственно, значения  $x$  лежат в столбце  $Z^{(1)}$  и т.п.) В переменные  $Y$ ,  $Y2$  записываются столбцы матрицы  $Z$ , соответствующие значениям функций  $y_1(x)$  и  $y_2(x)$ .

```

ORIGIN := 1
y := ( 1
      -1 )
D(x,y) := ( y1 + x · y2
            x - y1 - y2 )
Z := rkfixed(y,-3,6,100,D)
X := Z (1)   Y := Z (2)   Y2 := Z (3)

```



в). Решение одного уравнения порядка выше первого, разрешенного относительно старшей производной. Для иллюстрации возьмем ту же задачу, что и при рассмотрении `odesolve` (п. 2.7.1).

**Пример**

Решить дифференциальное уравнение

$$y'' + y' + y = 0$$

на интервале  $[0, 6]$  с начальными условиями  $y(0) = 1, y'(0) = 0.5$ .

**Решение**

Сначала необходимо исходное уравнение разрешить относительно старшей производной:

$$y'' + y' + y = 0 \quad \Rightarrow \quad y'' = -y' - y$$

Затем выполнить замену



$$y \Rightarrow y_0, y' \Rightarrow y_1.$$

(По умолчанию все массивы нумеруются от нуля – в том числе вектор искомой функции  $y$ ).

Тогда  $y'' \Rightarrow y_1'$  и уравнение преобразуется в систему

$$\begin{cases} y_0' = y_1 \\ y_1' = -y_1 - y_0 \end{cases}$$

с начальными условиями  $y_0(0)=1, y_1(0)=0.5$ .

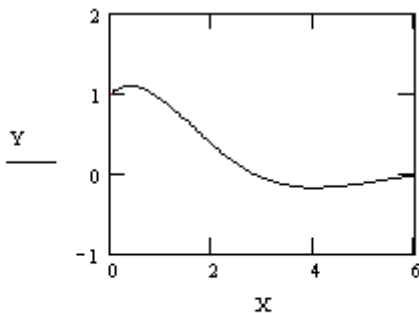
Полученную систему решаем с помощью функции `rkfixed`.

$$y := \begin{pmatrix} 1 \\ .5 \end{pmatrix}$$

$$D(x, y) := \begin{pmatrix} y_1 \\ -y_1 - y_0 \end{pmatrix}$$

$$Z := \text{rkfixed}(y, 0, 6, 100, D)$$

$$X := Z \langle 0 \rangle \quad Y := Z \langle 1 \rangle$$



$$X_{50} = 3 \quad Y_{50} = -0.058$$

$$X_{100} = 6 \quad Y_{100} = -0.028$$

Получили тот же график (и те же значения).

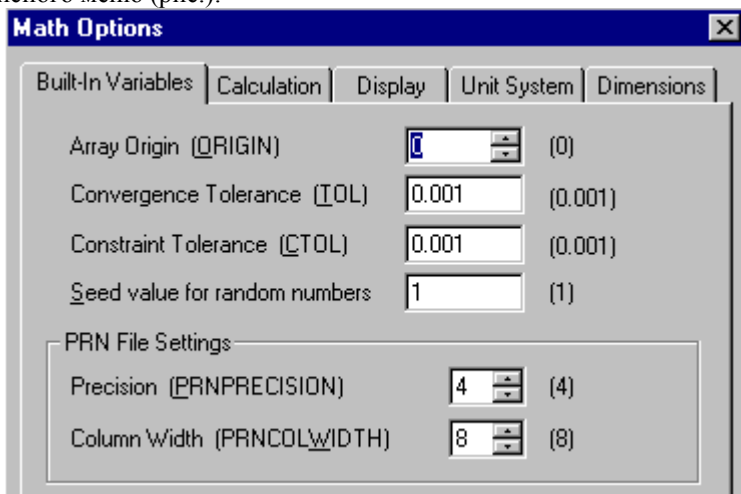
## 2.8. Интегрирование

Интегрирование в MathCAD реализовано в виде вычислительного оператора. Допускается вычислять интегралы от скалярных функций в пределах интегрирования, которые тоже должны быть скалярными.

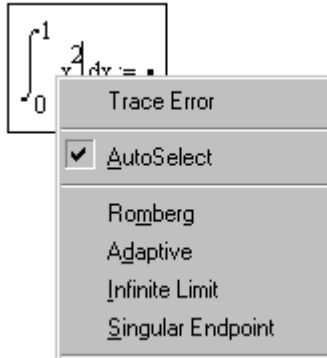
Несмотря на то, что пределы интегрирования должны быть действительными, подынтегральная функция может иметь и комплексные значения, поэтому и значение интеграла может быть комплексным. Если пределы интегрирования имеют размерность, то она должна быть одной и той же для обоих пределов.

### 2.8.1. Алгоритмы вычисления определенного интеграла

Результат численного интегрирования – это не точное, а приближенное значение интеграла, определенное с погрешностью, определяемой встроенной переменной TOL. От величины встроенной переменной зависит не только точность вычислений, но и их длительность. Изменить значение TOL можно присваиванием, например  $TOL:=0.00001$  и с помощью сервисного меню (рис.):



Для вычисления определенного интеграла можно воспользоваться методом интегрирования, установленным в программе MathCAD по умолчанию: *AutoSelect*. При необходимости можно выбрать алгоритм интегрирования самостоятельно, отключив опцию *AutoSelect* (автоматический выбор) через контекстное меню (рис.).



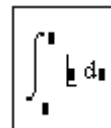
В MathCAD 2000 имеются четыре численных метода интегрирования, описанных ниже.

1. Romberg (метод Ромберга), который применяется для большинства функций, не содержащих особенностей;
2. Adaptive (адаптивный алгоритм) применяемый для быстро меняющихся функций;
3. Infinite Limit (бесконечный предел) для функций с бесконечными пределами;
4. Singular Endpoint – для интегралов с сингулярностью на конце. Это модифицированный алгоритм Ромберга для функций, не определенных на одном или обоих концах.

Для выбора метода надо сделать правый щелчок в левой части подынтегрального выражения (см. рис. выше)

Мы будем пользоваться автоматическим выбором. Тогда для получения решения необходимо:

1. Определить подынтегральную функцию и пределы интегрирования;
2. Выбрать на панели "Calculus" шаблон определенного интеграла (или ввести горячей клавишей – <&>);



3. Вписать соответствующие значения в заполнители (боксы);

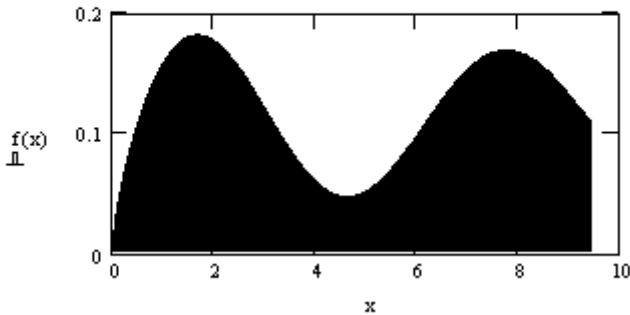
4. Нажать знак равенства на панели или, охватив угловым курсором все выражение, нажать клавишу  $\langle \Rightarrow \rangle$ .

**Пример**

Вычислить площадь криволинейной трапеции, приведенной ниже.

$$f(x) := \frac{\sin(x) + 1 + \sqrt{x}}{x + 10} - 0.1 \quad x := 0, 0.01 \dots 3 \cdot \pi$$

(выбран тип графика "bar")



**Порядок действий:**

- Определить область изменения x.
- Определить вид f(x).
- Заполнить шаблон интеграла.
- Охватить уголковым курсором все выражение.
- Нажать клавишу  $\langle \Rightarrow \rangle$ .

Ответ:

$$\int_0^{3 \cdot \pi} f(x) \, dx = 1.137$$

**Внимание:** как видно из примера, пределами интегрирования могут быть арифметические выражения.

### 3. Приложения

#### Приложение 3.1

#### Контрольные вопросы

Постарайтесь ответить на следующие вопросы. При ответе на каждый вопрос будьте готовы привести примеры.

##### <Введение>

Назовите причины популярности пакета MathCad.

##### <Основы работы>

Как называется документ MathCad?

Из каких блоков может состоять документ MathCad?

В каком порядке обрабатываются блоки на рабочем листе?

Что означает термин "живой документ"?

Что нужно учитывать при форматировании документа MathCad?

Какие виды курсоров имеются в системе MathCad?

Различает ли программа MathCad регистр в названиях переменных?

С какой точностью хранятся результаты вычислений в пакете MathCad?

Как получить результат с большей точностью?

##### <Набор и редактирование формул>

Какие действия необходимо предпринять для набора новой формулы?

Как отобразить результат вычислений?

Как изменить количество отображаемых знаков после запятой?

Что является "строительным блоком" для построения выражений?

Для чего при наборе формул используется клавиша <Пробел>?

Какова последовательность нажатия клавиш для набора  $\frac{x+y}{2}$ ?

Какова последовательность нажатия клавиш для набора  $\sqrt{x+1} - 2$ ?

Для чего при наборе формул используется клавиша <Insert>?

Назовите несколько горячих клавиш ввода шаблонов операций с клавиатуры.

##### <Запись констант и переменных>

Что является разделителем целой и дробной части в системе MathCad?

Как ввести число в экспоненциальной форме?

Как ввести комплексное число?

Назовите несколько встроенных констант системы MathCad.

Правила именования переменных в системе MathCad.

Различает ли система MathCad переменные  $x$  и  $X$ ?

Как ввести поясняющий индекс к имени переменной, например,  $x_{\min}$ .

Как ввести греческие символы.

Присваивание. Способы ввода оператора присваивания.

Можно ли использовать переменную с именем sin?

Встроенные системные переменные.

### <Работа с текстом>

Назначение текстовых блоков в MathCad-документе.

Как влияют текстовые блоки на вычисления?

Способы вставки текстового блока.

Перечислите основные возможности форматирования текста, предоставляемые системой MathCad.

Невычисляемая формула. Назначение. Способ ввода.

Возможности выравнивания блоков.

Средство "область" (Area).

### <Переменные - диапазоны>

Что такое переменная-диапазон.

Как задать переменную-диапазон.

Для чего используются переменные-диапазоны.

### <Функции в Mathcad>

Способы ввода стандартных функций в Mathcad.

Что является разделителем аргументов функции в системе Mathcad?

Какова последовательность нажатий клавиш для вычисления  $\sin^2 x$ ?

Что произойдет, если аргументом функции является переменная - диапазон?

Как создать пользовательскую функцию.

Что означает сообщение "This variable or function is not defined above"?

### <Графики в Mathcad-2d>

Типы графиков в системе Mathcad.

Способы вставки графика функции в декартовых координатах.

Отличие быстрого графика от графика по точкам.

Опишите последовательность действий для построения графика функции  $y = \sin x$  на интервале  $[0; 7]$ .

Размещение нескольких функций на одном графике.

Возможные проблемы при построении графика по точкам.

Оцените число точек при построении быстрого графика.

Каков интервал изменения аргумента для "быстрого графика"?

Что будет, если функция, график которой строится как "быстрый", определена только на неотрицательных значениях?

Как изменить диапазон аргумента на графике?

Как изменить масштаб по оси Y на графике?

Требования к переменной-аргументу при построении "быстрого графика".

Средства исследования графика в декартовых координатах.

Возможности настройки графика в декартовых координатах.

### < Прочие типы графиков >

График в полярных координатах.

Действия для построения графика линий уровня по точкам.

Построение объемной кривой.

Действия для построения быстрого графика - поверхности.

Возможности настройки объемного графика.

### <Сложные функции>

Возможности задания кусочно-непрерывных функций в системе Mathcad.

Пример задания функции с двумя участками непрерывности.

Пример задания функции с тремя участками непрерывности.

Что такое функция Хевисайда?

Пример использования функции if.

### < Решение уравнений и систем >

Функция root – поиск корня функции одной переменной.

Функция root – случай функции двух переменных.

Функция root вернула комплексное значение корня. Ваши действия.

Как найти еще один корень (случай нескольких корней)?

Назначение блока решения.

Элементы, которые нельзя вставлять в блок решения.

Ввод ограничения на равенство.

Пример использования конструкции Given / Find.

Получение с помощью блока решения функции, заданной неявно.

Переменные TOL и CTOL.

Пример использования конструкции Given / Minerr.

Отличие Find от Minerr.

### < Решение задач оптимизации >

Возможности системы MathCad по решению задач оптимизации.

Пример использования функции Maximize.

Пример решения задачи оптимизации с ограничениями.

### < Работа с векторами и матрицами >

Системная переменная ORIGIN.

Ввод индекса при наборе формул.

Вставка массива (матрицы).

Пример вычисления суммы ряда.

### < Символьные вычисления >

Что такое символьные вычисления.

Примеры использования "живых" символьных вычислений.

Как при сложении дробей получить результат в виде дроби.

Как продифференцировать функцию.

Как найти первообразную.

Отличие "живых" символьных вычислений от результатов команд меню Symbolic.

Как получить результат с точностью большей чем 15 знаков после запятой.

Как выполнить подстановку в выражении, скажем  $y+1$  вместо  $x$ .

< **Решение дифференциальных уравнений** >

Пример использования функции `odesolve`.

Назначение функции `rkfixed`.

Пример использования функции `rkfixed` для решения системы уравнений 1го порядка.

< **Интегрирование** >

Пример вычисления определенного интеграла.



### Горячие клавиши

Список горячих клавиш для ввода шаблонов операций с клавиатуры:

Оператор	Сочетание клавиш
квадратный корень	\
возведение в степень	^
корень степени n	Ctrl+\
модуль	
пара скобок (круглых)	' (апостроф)
факториал	!
произведение, общая форма	Ctrl+Shift+3
произведение, по диапазону	#
сумма, общая форма	Ctrl+Shift+4
сумма, по диапазону	\$
замена латинских букв на греческие	Ctrl+G
производная 1-го порядка	?
n-я производная	Ctrl+?
определенный интеграл	&
неопределенный интеграл	Ctrl+I

Список горячих клавиш для ввода логических операций:

Условие	Сочетание клавиш
$x = y$	Ctrl+=
$x \neq y$	Ctrl+3
$x \geq y$	Ctrl+0
$x \leq y$	Ctrl+9

Список горячих клавиш для работы с матрицами / векторами:

Действие	Сочетание клавиш
Вставка матрицы (массива)	Ctrl+M
Выделение столбца матрицы	Ctrl+6
Переход к написанию индекса матрицы / вектора	[ (левая квадратная скобка)
Задание переменной-диапазона	; (точка с запятой)

Горячие клавиши для прочих операций:

Действие	Сочетание клавиш
Ввод символа присваивания Команда «Вычислить» Команда «Вычислить символически» Проверка на равенство Переход к написанию (описательного) индекса в имени переменной Вставка текстового комментария Вставка функции (мастер функций) Вставка шаблона графика в декартовых координатах Вставка шаблона графика в полярных координатах Вставка шаблона графика поверхности	: (двоеточие) = Ctrl+. (Ctrl-точка) Ctrl+= (Ctrl-равно) . (точка) “ (кавычка) Ctrl+E @ Ctrl+7 Ctrl+2

**Исследование поведения математической модели системы  
"судно - авторулевой - рулевая машина" в пакете MathCad.  
(Пример выполнения курсовой работы)**

Данный пример призван проиллюстрировать возможности пакета MathCad. Упор делается на запись мат. модели в форме, близкой к математической, с одновременной реализацией этой модели на ЭВМ. Подробная постановка задачи и процесс построения мат. модели здесь опущены. С ними можно ознакомиться в пособии «Анализ динамических систем на ПЭВМ», Л.С. Грошева, М.И. Фейгин, Н. Новгород, 1996, из которого и была взята данная задача.

1. Математическая модель системы



Обозначим:

$\phi$     угол отклонения судна от курса

$\psi$     угол отклонения пера руля

$I$      момент инерции судна

Уравнение вращения судна вокруг вертикальной оси, проходящей через центр масс:

$$I \cdot \frac{d^2\phi}{dt^2} = M_c + M_p \quad , \quad \text{где}$$

момент сил сопротивления вращению

$$M_c = -H \cdot \frac{d\phi}{dt}$$

момент сил, создаваемых рулем

$$M_p = k \cdot \psi$$

$H$     - коэфф-т вязкого трения,

$k$     - чувствительность к повороту руля

Авторулевой должен формировать сигнал управления и подавать его на рулевую машину.

Известно, что качество управления будет удовлетворительным, если сигнал управления формируется с учетом как величины  $\phi$ ,

так и скорости отклонения судна от курса  $\frac{d}{dt}\phi$ .

В связи с этим сигнал управления рассчитываем по формуле

$$u = \phi + b \cdot \frac{d}{dt}\phi \quad , \text{ где } b - \text{ коэффициент скоростной коррекции,} \\ b \geq 0 \quad (\text{при } b=0 \text{ коррекции нет.})$$

$$u(\phi, d\phi) := \phi + b \cdot d\phi$$

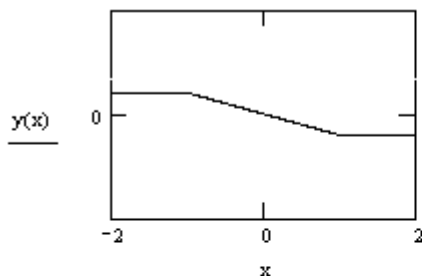
Зададим U в виде функции от угла и скорости, для дальнейшего использования. Параметр b необходимо будет задать глобально.

Реакция рулевой машины на управление описывается нелинейной функцией  $\psi = y(u)$

Константы, зависят от рулевого привода и т.п. :

$$u_0 := 1 \quad y_0 = 35 \quad \text{градусов} \quad y_0 := 35 \cdot \frac{\pi}{180} \quad \text{радиан}$$

$$y(u) := \begin{cases} -\frac{y_0}{u_0} \cdot u & \text{if } |u| < u_0 \\ -y_0 \cdot \text{sign}(u) & \text{otherwise} \end{cases} \quad \text{Задаем } y(u) \text{ в виде функции для} \\ \text{дальнейшего использования.}$$



Вид функции (в качестве аргумента используем неопределенную переменную x)

2. Построение финального уравнения  
(копирование, вставка, ...?)

$$I \cdot \frac{d^2}{dt^2}\phi = M_c + M_p$$

Подставляем формулы моментов

$$I \cdot \frac{d^2 \phi}{dt^2} = -H \cdot \frac{d \phi}{dt} + k \cdot \psi \quad \text{Подставляем формулу для } \psi$$

$$I \cdot \frac{d^2 \phi}{dt^2} = -H \cdot \frac{d \phi}{dt} + k \cdot y(u) \quad \text{Подставляем формулу для } u$$

$$I \cdot \frac{d^2 \phi}{dt^2} = -H \cdot \frac{d \phi}{dt} + k \cdot y\left(\phi + b \cdot \frac{d \phi}{dt}\right) \quad \text{или}$$

$$I \cdot \frac{d^2 \phi}{dt^2} = -H \cdot \frac{d \phi}{dt} + k \cdot y\left(u\left(\phi, \frac{d \phi}{dt}\right)\right)$$

Таким образом, для описания процесса стабилизации курса судна получили одно уравнение второго порядка, линейное относительно старшей производной.

### 3. Решение

Вспользуемся функцией `rkfixed`.

Выражаем старшую производную

$$I \cdot \frac{d^2 \phi}{dt^2} = -H \cdot \frac{d \phi}{dt} + k \cdot y\left(u\left(\phi, \frac{d \phi}{dt}\right)\right)$$

$$\frac{d^2 \phi}{dt^2} = \frac{-H \cdot \frac{d \phi}{dt} + k \cdot y\left(u\left(\phi, \frac{d \phi}{dt}\right)\right)}{I}$$

Далее необходимо переписать уравнение в виде системы. Выполняем замену:

$$\phi \quad \rightarrow \quad s_0$$

$$\frac{d \phi(t)}{dt} \quad \rightarrow \quad s_1$$

Система:

$$\begin{aligned}\frac{d}{dt}s_0 &= s_1 \\ \frac{d}{dt}s_1 &= \frac{-H \cdot s_1 + k \cdot y(u(s_0, s_1))}{I}\end{aligned}$$

Запись системы в форме, пригодной для `rkfixed`:

$$D(t, s) := \begin{pmatrix} s_1 \\ \frac{-H \cdot s_1 + k \cdot y(u(s_0, s_1))}{I} \end{pmatrix}$$

Нач условия:  $s := \begin{pmatrix} 1 \\ 0 \end{pmatrix}$  Первое - угол, второе - угловая скорость

Задаем параметры модели константами:  
(вообще говоря, их нужно подобрать)

$$I \equiv 15 \quad H \equiv .5 \quad k \equiv .4$$

$b \equiv 5$  это значение будем менять, исследуя влияние скоростной коррекции на качество процесса стабилизации

Решаем:

$$Z := \text{rkfixed}(s, 0, 50, 500, D) \quad \text{Интервал решения берем } [0, 50]$$

Выделим столбцы из матрицы решения для построения графиков:

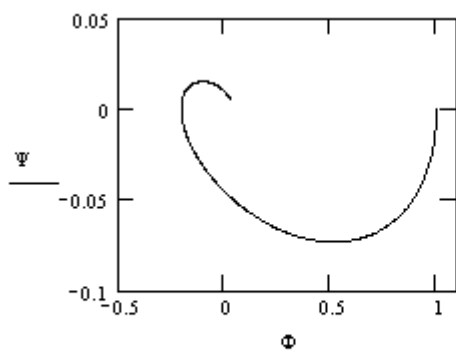
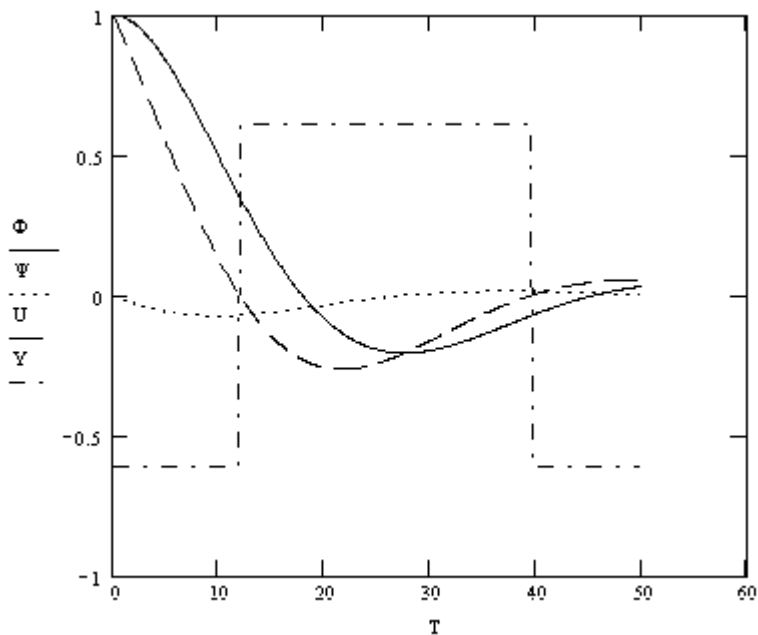
$$T := Z \langle 0 \rangle \quad \text{время}$$

$$\Phi := Z \langle 1 \rangle \quad \text{отклонение от курса}$$

$$\Psi := Z \langle 2 \rangle \quad \text{угловая скорость}$$

$$U := u(\Phi, \Psi) \quad \text{управление}$$

$$Y := y(U) \quad \text{угол поворота руля}$$



Фазовый портрет

Полученные графики автоматически пересчитываются при изменении параметров модели  $I, H, k, b$ .

Примеры выполнения лабораторных работ в пакете mathCad

3.4.1. Лабораторная работа № 1  
Набор формул

(теоретическая часть: см. п.п. 1.2, 1.6. Задание: п.п. 3.5.1)

1. Постановка задачи

Набрать функцию. Вычислить значение на произвольном наборе данных.

$$F(x, y) = e^{-\left(\frac{x+y}{\alpha+\beta}\right)^2} \cdot \frac{\sqrt[4]{x^2 - y^2}}{x^3\sqrt{\alpha \cdot y} + y^3\sqrt{\beta \cdot x}} + \cos(x - 45^\circ)$$

2. Решение

Задаем константы:

$$\alpha := 1 \quad \beta := 5.7$$

Задаем функцию:

$$F(x, y) := e^{-\left(\frac{x+y}{\alpha+\beta}\right)^2} \cdot \frac{\sqrt[4]{x^2 - y^2}}{x \cdot \sqrt[3]{\alpha \cdot y} + y \cdot \sqrt[3]{\beta \cdot x}} + \cos(x - 45 \cdot \text{deg})$$

Вычисляем значение ф-ции в произвольных точках:

$$F(-3, 12) = -0.828 - 0.029i$$

Итак, при  $x=-3$   $y=12$  получаем комплексное число

Пробуем подобрать значения аргументов так, чтобы результат был вещественным:

$$F(1, 1) = 0.977$$



### 3.4.2. Лабораторная работа № 2

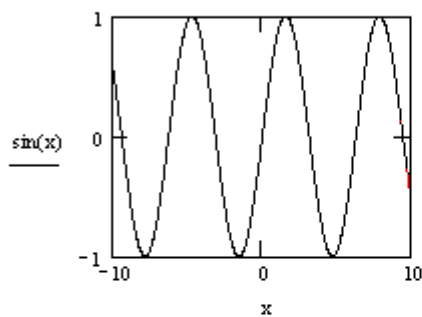
#### Построение графика функции одной переменной в декартовых координатах

(теоретическая часть: см. п.п. 1.5, 1.7. Задание: п.п. 3.5.2)

##### 1. Задание

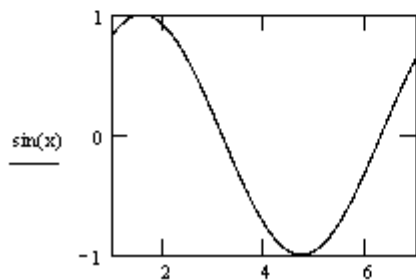
Построить график функции  $y=\sin(x)$  на интервале  $[1,7]$ , исследовать поведение функции

##### 2. Быстрый график



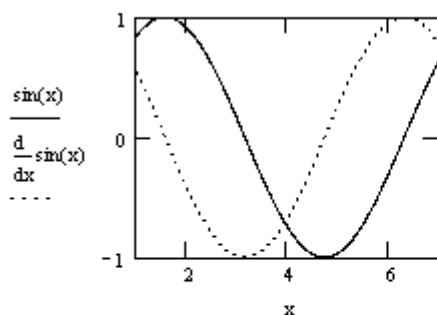
По умолчанию график строится от -10 до 10. Мы записали функцию прямо в шаблоне графика; это допустимо, но задать функцию по имени - ВСЕГДА хорошая идея.

##### 3. Изменяем диапазон графика



Копируем график и вводим нужные нам пределы в шаблоны внизу.

#### 4. Размещение второго графика на той же картинке

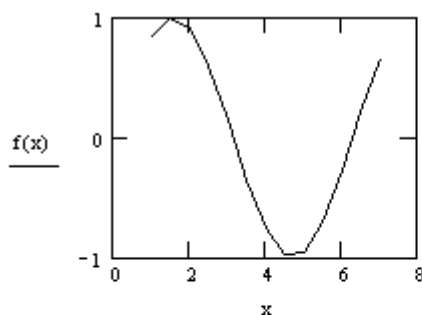


Копируем график, ставим курсор после функции, нажимаем "," (запятая) и вводим новую функцию. Как видим, это может быть даже производная.

#### 5. График, построенный по набору значений.

$f(x) := \sin(x)$  Как было сказано, задать функцию - хорошая идея

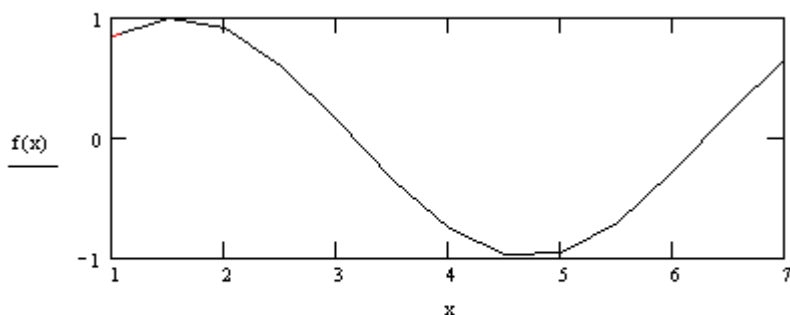
$x := 1, 1.5..7$  Диапазон с небольшим количеством точек (задаются первая, вторая и последняя точки)



Вот эти точки:

x =	f(x) =
1	0.841
1.5	0.997
2	0.909
2.5	0.598
3	0.141
3.5	-0.351
4	-0.757
4.5	-0.978
5	-0.959
5.5	-0.706
6	-0.279
6.5	0.215
7	0.657

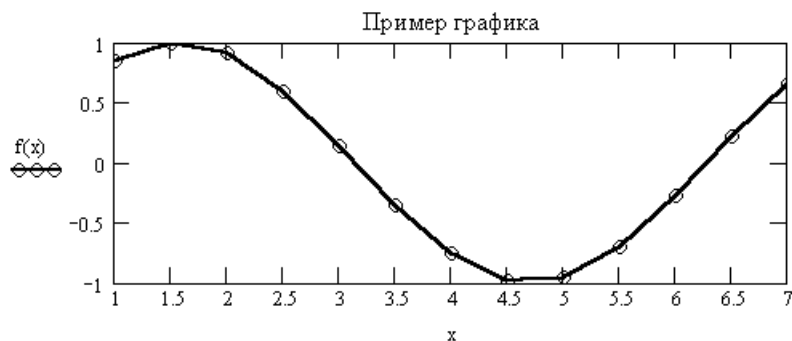
## 6. Меняем размер графика



Изменили размер, потянув за маркеры на рамке (видны, только когда график выделен). Ясно видны изломы графика, построенного по небольшому числу точек.

## 7. Настройка графика

- Меняем деления на шкале
- толщину/цвет линии
- обозначаем точки графика маркерами (их немного)
- добавим подписи к графику



Все эти настройки становятся доступны, если дважды щелкнуть на графике.

**3.4.3. Лабораторная работа № 3**  
**Исследование функции одной переменной**  
**(все, что можно сделать с функцией одной переменной)**

(теоретическая часть: см. п.п. 1.5, 1.7, 2.3, 2.6. Задание: п.п. 3.5.2)

1. Задание

Исследовать функцию  $\sin(x + 1)^2 - e^{\frac{-x^2}{2}} \cdot x + e^{\frac{-x}{5}} - 1$   
на интервале  $[-5; 7]$

- а) задать ф-цию
- б) вычислить значения в крайних точках
- в) протабулировать (не более 21 значения)
- г) построить график, добавить линию нуля
- д) с помощью увеличения графически отделить один корень
- е) найти этот корень
- ж) найти еще корень на участке (если есть)
- з) найти  $\min$ ,  $\max$  ф-ции на участке
- и) Вычислить символически производную; задать ее как функцию
- к) скопировать график и добавить на него производную

2. Решение

а) задаем ф-цию

$$f(x) := \sin(x + 1)^2 - e^{\frac{-x^2}{2}} \cdot x + e^{\frac{-x}{5}} - 1$$

б) задаем константы для крайних точек, вычисляем ф-цию в них

$$a := -5 \quad b := 7$$

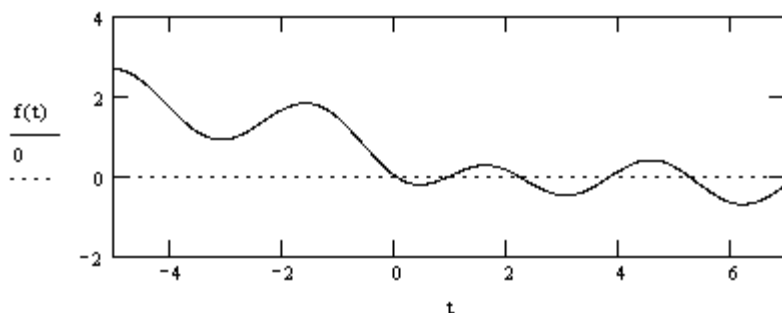
$$f(a) = 2.684 \quad f(b) = -0.222$$

в) задаем переменную-диапазон и строим табличку.

$$x := a, a + 1 .. b \quad \text{Выбираем шаг чтобы получить не очень много значений (не более 21)}$$

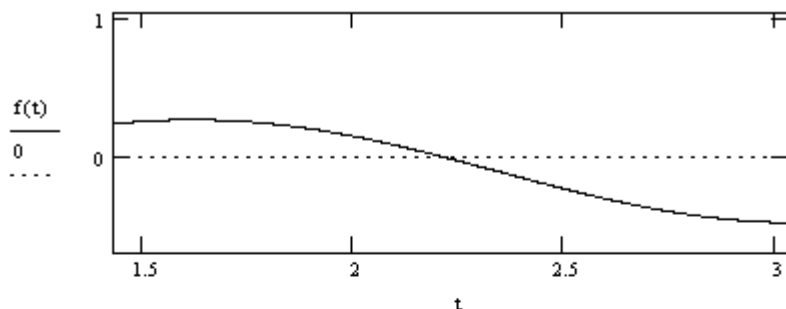
$x =$	$f(x) =$
-5	2.684
-4	1.7
-3	0.913
-2	1.658
-1	1.442
0	$9.967 \cdot 10^{-3}$
1	$6.451 \cdot 10^{-3}$
2	0.145
3	-0.483
4	0.118
5	0.225
6	-0.666
7	-0.222

Строим быстрый график. Аргумент должен быть неопределен, так что  $x$  не годится - берем  $t$



д) отделение корня:

копируем график, отделяем корень  
 Поищем корень в окрестности  $x=2$



е) ищем этот корень функцией root

Используем указание диапазона поиска,  
диапазон берем из последнего графика

```
x := root(f(x), x, 2, 2.5)
```

x = 2.206                  f(x) = 0    Проверка показывает - корень

ж) еще корень.

Используем начальное приближение,  
чтобы найти другой корень.

Поискем в окрестности x=1

```
x := 1
```

```
x := root(f(x), x)
```

x = 0.99                  f(x) =  $4.524 \times 10^{-6}$

з) min, max.

Применяем блок Given чтобы задать интервал  
поиска, если нужно.

Не забываем задать начальные значения.

```
x := 6
```

```
xmin := Minimize(f, x)                  (индекс набирается через точку)
```

x<sub>min</sub> = 6.212                  f(x<sub>min</sub>) = -0.71

```
x := -4
```

$x_{\max} := \text{Maximize}(f, x)$

$x_{\max} = -5.107$        $f(x_{\max}) = 2.693$

Найденное значение вышло за диапазон. Используем блок Given чтобы ограничить  $x$

Given

$$a < x < b$$

$x_{\max} := \text{Maximize}(f, x)$

$x_{\max} = -5$        $f(x_{\max}) = 2.684$       Теперь правильно

и) производная

вычисляем символически  
( → набирается как Ctrl - точка)

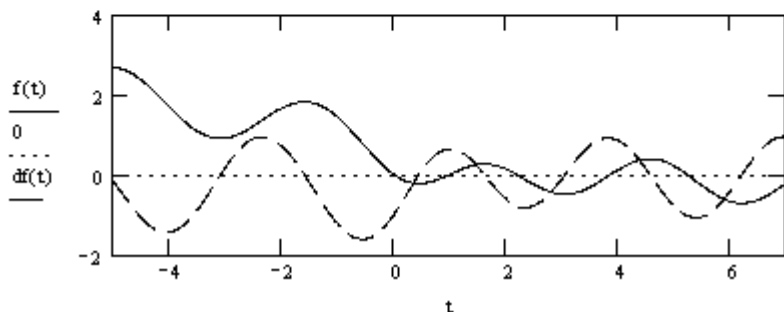
$$\frac{d}{dx} f(x) \rightarrow 2 \cdot \sin(-3.9) \cdot \cos(-3.9) + 15 \cdot \exp(-8) - \frac{1}{5} \cdot \exp\left(\frac{4}{5}\right)$$

задаем как функцию

$$df(x) := \frac{d}{dx} f(x)$$

к) график с производной

Копируем первый график, добавляем производную



### 3.4.4. Лабораторная работа № 4

#### Исследование функции двух переменных

(теоретическая часть: см. п.п. 2.1.2, 2.4. Задание: п.п. 3.5.3)

#### 1. Задание

Исследовать функцию  $f(x, y) := \sin(x) \cdot \cos(y) + 1.5$  в области

$$x \in [-2; 2]$$

$$y \in [-2; 2]$$

- задать функцию
- построить график типа поверхность
- построить график - линии уровня
- найти  $\min$ ,  $\max$
- если они оба одного знака, модифицировать функцию, чтобы проходила через 0
- найти точку в которой функция равна 0

#### 2. Решение

а) задаем функцию

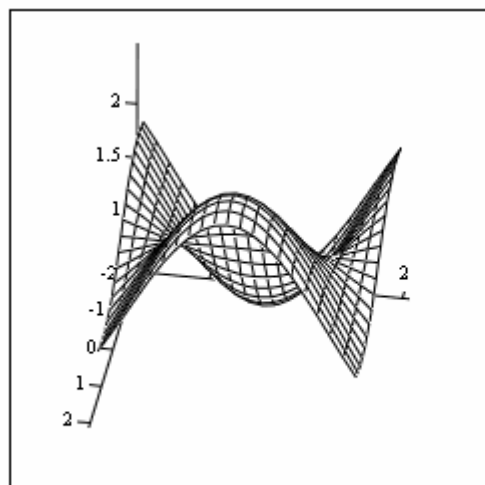
$$f(x, y) := \sin(x) \cdot \cos(y) + 1.5$$

б) график.

Воспользуемся быстрым графиком; функция 2х переменных уже задана, осталось подкорректировать диапазон изменения переменных.

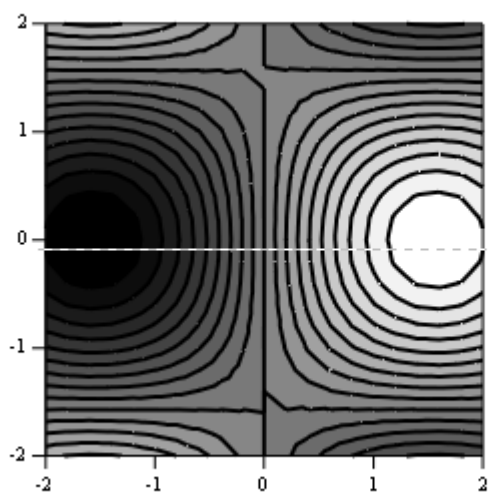
Чтобы видеть  $\min/\max$ , стоит раскрасить график





$f$

в) График - линии уровня строятся аналогично.



$f$

г) Нахождение min, max

min:

т.к. ищем минимум в области, воспользуемся блоком Given для записи ограничений

Не забываем взять начальное приближение (видно на графике "линии уровня")

$x := -1.5$        $y := 0$       начальное приближение

Given

$$-2 < x < 2$$

$$-2 < y < 2$$

$$\text{Minimize}(f, x, y) = \begin{pmatrix} -1.571 \\ 0 \end{pmatrix} \quad \text{Точка минимума}$$

$$f(-1.571, 0) = 0.5 \quad \text{Значение ф-ции в точке минимума}$$

max:

Копируем блок для минимума, вносим коррективы

$x := 1.5$        $y := 0$       начальное приближение

Given

$$-2 < x < 2$$

$$-2 < y < 2$$

$$\text{Maximize}(f, x, y) = \begin{pmatrix} 1.571 \\ 0 \end{pmatrix} \quad \text{Точка максимума}$$

$$f(1.571, 0) = 2.5 \quad \text{Значение ф-ции в точке максимума}$$

д) модификация функции

Т.к. минимальное и максимальное значения ф-ции одного знака, сдвинем поверхность по оси Z, чтобы она проходила через 0.

$$f1(x, y) := f(x, y) - 0.7$$

Теперь

$$f1(-1.571, 0) = -0.2$$

$$f1(1.571, 0) = 1.8 \quad \text{Знаки разные - нули должны быть}$$

е) нахождение одного из нулей функции

Находим одну из точек в которых ф-ция принимает значение 0

$y := 0$       Эту переменную менять не будем

$x := 0$       начальное значение

$$x := \text{root}(f1(x, y), x) \quad x = -0.927$$

Проверяем

$$f1(x, y) = 3.51 \times 10^{-5} \quad \text{Близко к 0 - похоже на корень}$$

### 3.4.5. Лабораторная работа № 5 Вычисление сложной функции

(теоретическая часть: см. п.п. 1.7, 2.2. Задание: п.п. 3.5.4)

#### 1. Задание

Задать сложную функцию.

Построить график.

Убедиться, что значения в точках разрыва соответствуют заданию.

$$f(x) = \begin{cases} -1.5, & x \leq -3 \\ \sin(x), & -3 < x \leq 3 \\ 1.5, & x > 3 \end{cases}$$

#### 2. Решение.

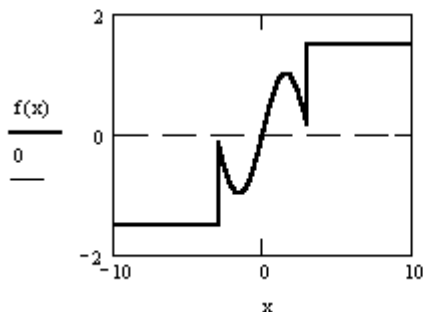
##### 2.1. Задаем функцию.

Т.к. функция имеет 3 участка, описываемых непрерывными функциями, имеет смысл задать ее в виде программной конструкции if (Вариант 1)

$$f(x) := \begin{cases} -1.5 & \text{if } x \leq -3 \\ \sin(x) & \text{if } -3 < x \leq 3 \\ 1.5 & \text{otherwise} \end{cases}$$

##### 2.2. Строим график

Используем быстрый график; если необходимо, вручную меняем пределы изменения аргумента и масштаб по ординате



2.3. Проверяем значение ф-ции в точках разрыва

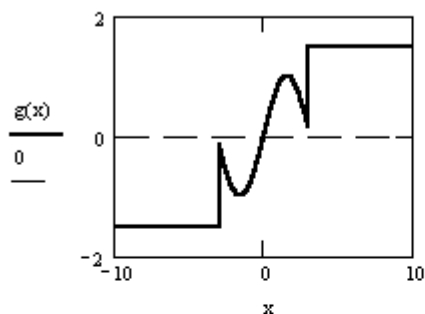
$$f(-3) = -1.5$$

Обращаем внимание где в задании стоит  $\leq$

$$f(3) = 0.141$$

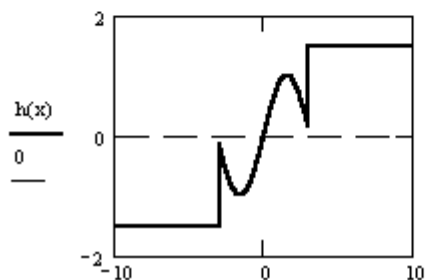
2.4. Вариант 2. Используем функцию If

$$g(x) := \text{if}(x \leq -3, -1.5, \text{if}(x \leq 3, \sin(x), 1.5))$$



2.5. Вариант 3. Используем логические множители

$$h(x) := (x \leq -3) \cdot -1.5 + (-3 < x \leq 3) \cdot \sin(x) + (3 < x) \cdot 1.5$$



### 3.4.6. Лабораторная работа № 6

#### Нахождение корней нелинейного уравнения

(теоретическая часть: см. п.п. 1.5, 1.7, 2.3. Задание: п.п. 3.5.2)

#### 1. Задание

Найти несколько корней уравнения на интервале  $[-10;10]$ :

$$x^3 - 20 \cdot x + e^x = 0$$

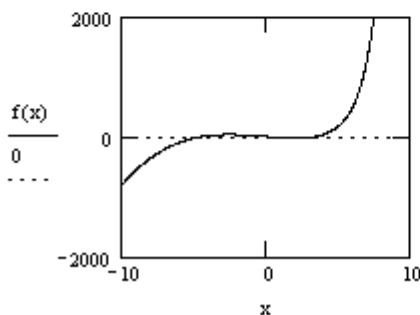
- а) функцией root с заданием интервала
- б) функцией root с заданием нач приближения
- в) функцией find

#### 2. Решение

Задаем ф-цию

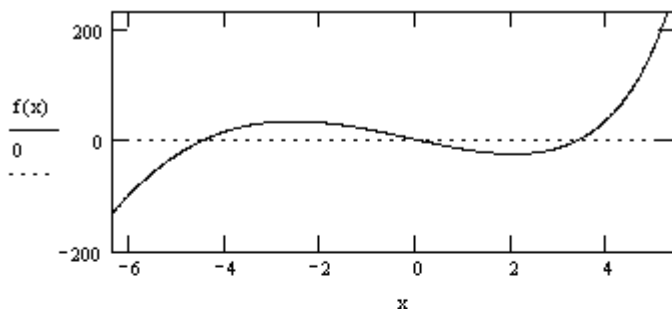
$$f(x) := x^3 - 20x + e^x$$

строим график, визуально отделяем корни.



Если необходимо,  
используем  
увеличение (Zoom).

Применяем инструмент "увеличение" на график;  
видим 3 корня



Протабулируем ф-цию,  
отследим смену знака:

$$X := -5, -2..4$$

$$f(X) =$$

-24.993
32.135
-16.282
38.598

а) ищем корень функцией root с заданием интервала

Первый корень:

$$x_1 := \text{root}(f(x), x, -6, -3) \quad (\text{индекс вводится через точку})$$

$$x_1 = -4.472 \quad f(x_1) = -5.294 \times 10^{-15}$$

б) ищем корень функцией root с заданием нач приближения

Второй корень:

задаем начальное условие - около корня

$$x := 0$$

$$x_2 := \text{root}(f(x), x)$$

$$x_2 = 0.053 \quad f(x_2) = 2.603 \times 10^{-6}$$

Значение корня с большей отображаемой точностью:

$$x_2 = 0.052713577463897$$

Если точность нахождения корня нас не удовлетворяет,  
изменим переменную TOL:

$$\text{было} \quad \text{TOL} = 1 \times 10^{-3} \quad (\text{по умолчанию})$$

$$\text{меняем:} \quad \text{TOL} := 10^{-10}$$

$x_2 := \text{root}(f(x), x)$  Получаем результат с большей точностью.

$$x_2 = 0.053 \quad f(x_2) = 8.005 \times 10^{-12}$$

Значение функции ближе к 0. Для просмотра значения корня увеличим число цифр после запятой (двойной щелчок на результате, увеличиваем разрядность до 15). Теперь видно что значение отличается.

$$x_2 = 0.052713714916075$$

в) ищем корень функцией find

Третий корень:

задаем начальное условие - около корня

$$x := 4$$

Given

$$f(x) = 0 \quad (\text{Условие задается через Ctrl+=})$$

$$x_3 := \text{Find}(x)$$

$$x_3 = 3.371 \quad f(x_3) = -1.511 \times 10^{-5}$$



### 3.4.7. Лабораторная работа № 7 Обработка одномерного массива

(теоретическая часть: см. п.п. 2.5. Задание: п.п. 3.5.5)

#### 1. Задание

Вычислить

$$Z := \sum_{i=1}^{n-2} (a_i)^2 + \min \left( a_i \cdot \prod_{i=2}^{n-1} i! \right)^2 \quad \text{при} \quad n := 4$$

#### 2. Задаем массив

ORIGIN := 1

Определение индекса первого элемента массива (по умолчанию стоит 0)

$$a := \begin{pmatrix} 1 \\ 2 \\ -1 \\ 3 \end{pmatrix}$$

Задаем массив с именем "a" из 4х элементов

n := rows(a)

Количество элементов в массиве, проверим

n = 4

#### 3. Вычисление

а) сумма

$$S := \sum_{i=1}^{n-2} (a_i)^2 \quad S = 5$$

б) произведение

$$P := \prod_{i=2}^{n-1} i! \quad P = 12$$

в) минимум

$$i := 1..n$$

номера элементов (для массива)

$$b_i := a_i \cdot P \quad b = \begin{pmatrix} 12 \\ 24 \\ -12 \\ 36 \end{pmatrix} \quad \text{Задаем вспомогательный массив } b$$

$$M := \min(b) \quad M = -12 \quad \text{Находим в нем минимум}$$

г) искомая величина

$$Z := S + M \quad Z = -7$$

Замечание.

В одну строчку данное выражение записать не удастся из-за необходимости введения вспомогательного массива для минимума.

### 3.4.8. Лабораторная работа № 8 Работа с матрицами

(теоретическая часть: см. п.п. 2.2, 2.5. Задание: п.п. 3.5.6)

#### 1. Задание

По матрице  $A$  ( $n \times k$ ) сформировать одномерный массив  $b$ , каждый элемент которого вычисляется по заданной формуле. Четные элементы массива  $b$  вычисляются по верхней формуле, нечетные - по нижней.

Отсортировать полученный массив и найти среднее арифметическое его элементов.

$$b_i = \begin{cases} \min_j (a_{ij} - j^2) & \text{если } i \text{ четное} \\ \prod_{j=1}^i (j a_{ij}) & \text{если } i \text{ нечетное} \end{cases}$$

для  $\min j$  изменяется в интервале  $[1, k]$

$n=4$   $k=5$

#### 2. Решение

а) Установим системную переменную ORIGIN равной 1. Теперь все массивы будут начинаться с 1

ORIGIN := 1

б) Зададим матрицу.

Создадим две переменные-диапазоны для индексов  
- они нам пригодятся

$n := 4$              $k := 5$

$i := 1..n$          $j := 1..k$

Вводим матрицу вручную

(Варианты включают:

\* Задание матрицы  
формулой от индексов

$$A_{i,j} := i \cdot 10 + j$$

\* Задание матрицы  
случайным образом

$$A_{i,j} := \text{rnd}(10)$$

)

$$A := \begin{pmatrix} 2 & 0 & 1 & 3 & -1 \\ -2 & 3 & -1 & -2 & 0 \\ 2 & 3 & -1 & 1 & 2 \\ 0 & -2 & 2 & 1 & 3 \end{pmatrix}$$

Вот наши исходные данные

в) Формула для вычисления эл-тов массива  $b$  с условием на четность/нечетность:

$$b_i := \text{if} \left[ (-1)^i > 0, \min(A_{i,j} - j^2), \prod_{j=i}^k j \cdot A_{i,j} \right]$$

$$b = \begin{pmatrix} 0 \\ -25 \\ -120 \\ -22 \end{pmatrix}$$

Результат вычислений.

г) Сортировка

$$\text{sort}(b) = \begin{pmatrix} -120 \\ -25 \\ -22 \\ 0 \end{pmatrix}, \text{ по убыванию: } \text{reverse}(\text{sort}(b)) = \begin{pmatrix} 0 \\ -22 \\ -25 \\ -120 \end{pmatrix}$$

е) Среднее арифметическое

$$\text{mean}(b) = -41.75$$

### 3.4.9. Лабораторная работа № 9 Вычисление определенного интеграла

(теоретическая часть: см. п.п. 1.7, 2.6, 2.8. Задание: п.п. 3.5.7)

1. Задание

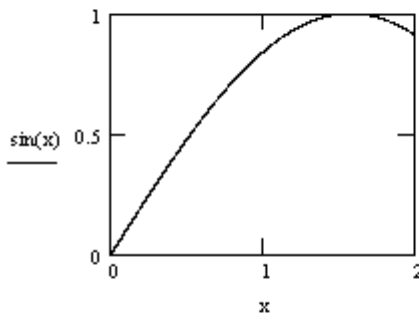
Вычислить

$$\int_0^2 \sin(x) dx$$

2. Исследование функции

а) Построение графика (поиск особенностей)

Используем "быстрый график"



Нет никаких особенностей  
(разрывов). Интеграл  
собственный - можно  
продолжать

б) нахождение первообразной

$\sin(x)$

Записываем исходную функцию

$-\cos(x)$

Вычисляем первообразную  
(Symbolics|Variable|Integrate).

Курсор должен стоять на переменной (x)

**Замечание.** Первообразная может и не существовать  
(или MathCad ее не сможет найти). В таком случае  
будет выведено исходное выражение.

$\sin(x)$                     Дифференцируем обратно  
(Symbolics|Variable|Differentiate)  
- для контроля. Должно совпадать с исходной.

3. Вычисление интеграла по формуле Ньютона-Лейбница  
(если первообразная найдена)

$P(x) := -\cos(x)$     Задаем первообразную как функцию  
(копируем сверху)

$Int := P(2) - P(0)$              $Int = 1.416$

4. Вычисление интеграла средствами MathCad

$Int := \int_0^2 \sin(x) dx$              $Int = 1.416$

### 3.4.10. Лабораторная работа № 10 Вычисление площади живого сечения однорукавного русла и построение рельефа дна

(теоретическая часть: см. п.п. 1.7, 2.5. Задание: п.п. 3.5.8)

#### 1. Постановка задачи

Для проектирования выправительных сооружений и дноуглубительных работ при выполнении расчетов возникает необходимость вычисления площади живого сечения русла по имеющимся замерам. Вычисление площади сечения осуществляется методом трапеции по формуле:

$$W := \sum_{i=1}^{m-1} \frac{(h_i + h_{i+1})}{2} \cdot (x_{i+1} - x_i) \quad , \text{ где}$$

$x_i$  - расстояние от берега до  $i$ -й точки измерения,

$h_i$  - глубина в  $i$ -й точке,

$m$  - число замеров (точек).

По предоставленным данным построить рельеф сечения дна и рассчитать площадь живого сечения русла.

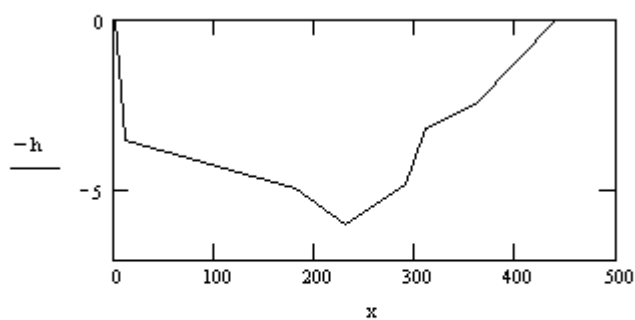
#### 2. Решение

а) Задаем массивы для результатов измерений. Устанавливаем нумерацию массивов от 1 (используется в формуле).

ORIGIN := 1

$$x := \begin{pmatrix} 0 \\ 10 \\ 180 \\ 230 \\ 290 \\ 310 \\ 360 \\ 440 \end{pmatrix} \quad h := \begin{pmatrix} 0 \\ 3.5 \\ 4.9 \\ 6 \\ 4.8 \\ 3.2 \\ 2.5 \\ 0 \end{pmatrix} \quad \begin{array}{l} m := \text{rows}(x) \\ m = 8 \quad \text{точек} \end{array}$$

б) строим сечение дна русла



в) вычисляем площадь сечения методом трапеций

$$W := \sum_{i=1}^{m-1} \frac{(h_i + h_{i+1})}{2} \cdot (x_{i+1} - x_i) \quad W = 1.651 \times 10^3$$



### 3.4.11. Лабораторная работа № 11 Решение ОДУ первого порядка

(теоретическая часть: см. п.п. 2.7. Задание: п.п. 3.5.9)

#### 1. Задание

Решить дифференциальное уравнение  $\frac{d}{dx}y = 1 + x^2 + \frac{2 \cdot xy}{1 + x^2}$

на интервале  $x \in [1, 2]$  с начальным условием  $y(1) = 1$

Построить график.

Проверить значения функции в контрольных точках.

(  $y(1)=4$ ,  $y(2)=15$  )

Построить решение дифференциального уравнения с близкими начальными условиями (семейство интегральных кривых).

Начальные условия подобрать самостоятельно, выделить фоном.

#### 2. Решение

а) Получим решение ДУ в виде функции  $y(x)$ , используя блок `odesolve`:

Given

$$\frac{d}{dx}y(x) = 1 + x^2 + \frac{2 \cdot x \cdot y(x)}{1 + x^2}$$

не забываем указывать аргумент для  $y$  и вводить условия через `Ctrl=`

$$y(0) = 1$$

`y := odesolve(x,2)`

б) Вычислим значения в контрольных точках:

$$y(1) = 4 \qquad y(2) = 15 \qquad (\text{Значения совпадают})$$

в) Получим еще два решения уравнения с близкими начальными условиями. Скопируем и изменим блок `odesolve` для получения функций  $y_1(x)$  и  $y_2(x)$ .

Given

$$\frac{d}{dx} y_1(x) = 1 + x^2 + \frac{2 \cdot x \cdot y_1(x)}{1 + x^2}$$

индекс вводим  
"для красоты".  
Через точку.

$$y_1(0) = 0.8$$

$y_1 := \text{odesolve}(x, 2)$

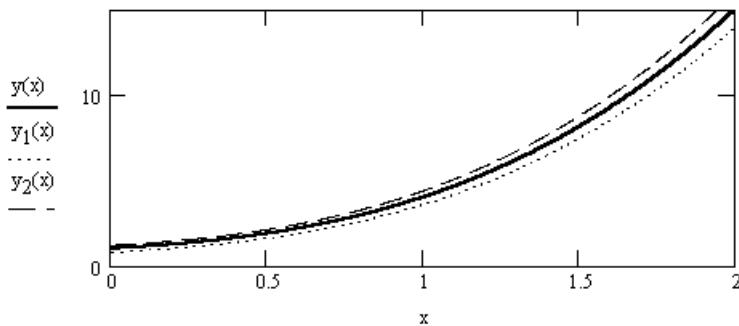
Given

$$\frac{d}{dx} y_2(x) = 1 + x^2 + \frac{2 \cdot x \cdot y_2(x)}{1 + x^2}$$

$$y_2(0) = 1.2$$

$y_2 := \text{odesolve}(x, 2)$

г) Построим график семейства (выделим искомую функцию толщиной линии)



Задания на лабораторные работы

3.5.1. Основы работы: освоение набора формул

(Индивидуальные варианты заданий для лабораторной работы № 1)

**Задание**

Наберите функции; проверьте корректность набора (: вычислите значения функции  $f(x, y)$  на произвольном наборе переменных).

№	Функции для набора $f(x, y)$
1	$\arcsin^2(x - \beta) + \frac{x^3 + x^2y + xy^2 + y^3}{1 - \frac{x}{\alpha} + \frac{y}{\beta} \cos(x + \beta - 15^\circ)}$
2	$\frac{\sin 5^\circ \cdot \sqrt[3]{x^2 + y^2}}{\operatorname{arctg}^3(x^4 + y^4)} \left( \alpha x^2 + \frac{\alpha}{\beta} x^2 y + \frac{\beta}{\alpha} xy^2 + \beta y^2 \right)$
3	$\log_2 \left  \frac{x - \xi}{y - \phi} \right  + e^{\frac{x}{\beta}} \frac{x^3 + \alpha x^2 + \alpha^2 x + \alpha^3}{\sin^2(y - \beta + 25^\circ)^2}$
4	$\sin^2 \left( x - \frac{56^\circ}{\alpha} \right) + \frac{y^2 + xy + x^2}{\alpha^2 + \beta^2} \arccos^2 \left( x - \frac{\pi}{\alpha} \right)$
5	$e^{-(x^2 - y^2)} \frac{\cos \left( \frac{x}{\alpha} \right) - \sin^2 \left( \frac{y}{\beta} - 30^\circ \right)}{\arccos \left( \sqrt{\alpha^2 + xy} \right)}$
6	$\frac{\sqrt[5]{ x^2 - y^2 \cos 48^\circ }}{x\sqrt{\alpha y} + y\sqrt{\beta x}} - \operatorname{arctg}^3 \left( \frac{xy}{\alpha + \beta} \right)$
7	$\frac{\operatorname{tg}^2 \left( \alpha \frac{x}{y} + 10^\circ \right) - \operatorname{ctg}^2 \left( \beta \frac{y}{x} \right)}{\alpha x^2 + \beta y^2} e^{(\sqrt{x} + \sqrt{y})}$
8	$\frac{x^3 + \alpha x^2 + bx + c}{\sqrt{ x - \alpha } + \sqrt{ y - \beta }} \left( \tau \cos^3 \left( 45^\circ + \frac{x}{\alpha} \right) - 3 \arccos \psi y \right)$

№	Функции для набора $f(x, y)$
9	$\frac{\sin(x + 35^\circ)(x - \alpha)^3}{y + (x - \alpha)^2} \operatorname{sh}^2 \left( -\left(\frac{x}{\alpha}\right)^2 - \left(\frac{y}{\beta}\right)^2 \right)$
10	$\frac{x + \frac{y}{\beta}}{y - \frac{x}{\alpha}} \arccos^2 \left( x - \frac{\pi}{4} \right) + \frac{y + \frac{x}{\alpha}}{x - \frac{y}{\beta}} \arcsin^2 3y + 25^\circ$
11	$\frac{\arcsin^2 x}{\cos^2 x - \sin^2(x - 75^\circ)} \sqrt[3]{\alpha x^2 + \alpha \beta xy + \beta y^2}$
12	$\operatorname{sh}(\alpha x^2 + \beta y^2) \frac{\cos^2 \left( \frac{x - \alpha}{\pi} \right) + \sin^2 \left( \frac{y - \beta}{\pi} + 5^\circ \right)}{\sqrt[3]{\left( \frac{x}{\alpha} - \frac{y}{\beta} \right)^2}}$
13	$\frac{\alpha x^2 + (\alpha - \beta)xy + \beta y^2}{\sin \left( \frac{x}{3} + 15^\circ \right) \log_2 \left  \frac{x}{p} - \alpha \right } \left( \operatorname{ch}^2 \frac{\xi}{p} + \operatorname{sh}^2 \frac{\psi}{p} \right)$
14	$\frac{\sqrt[4]{(x - y)^2}}{ x^3 - y^3  - \alpha \beta} \left( \sin^2 \frac{x - 40^\circ}{p} + \arccos^2 \frac{y - \beta}{p} \right)$
15	$\frac{\operatorname{sh}^2 \frac{x}{p} - \operatorname{ch}^2 \frac{y}{p}}{(x - y)^2 + \alpha \beta} \sqrt[3]{\alpha x^3 - \beta y^3} + \operatorname{tg} 17^\circ$
16	$\sin(xy - 21^\circ) \frac{\operatorname{th}^2 3x - \frac{xy^3}{3}}{\alpha(x - y^3)} e^{-x^2 + \beta y}$
17	$\frac{\arcsin^2(x - \psi)}{x + y^2 - \frac{\alpha}{\beta}} + \pi \frac{x^2 + y}{\operatorname{tg} \left( \frac{x}{y} - 30^\circ \right)}$

№	Функции для набора $f(x, y)$
18	$\frac{\frac{x + \alpha}{3} + \operatorname{tg}\left(y + \frac{\pi}{4}\right)}{\operatorname{sh}^2 x + \operatorname{ch}^2 y} \cos \frac{x + y + 2^\circ}{\alpha \beta^2}$
19	$\left(\frac{x^2}{y^2} - \frac{\alpha x}{\beta y} + \alpha \xi + \sqrt{\beta}\right) \arccos^3\left(x^2 + y^2 - \frac{\pi}{6}\right)$
20	$\frac{x + \frac{y}{\delta}}{\delta^2 - 1} \operatorname{sh}^2\left(\cos\left(\frac{x}{p} + 0.5^\circ\right) + \sin\left(\frac{y + x}{p^2 + 1}\right)\right)$

### 3.5.2. Исследование функции одной переменной

(Индивидуальные варианты заданий для лабораторных работ № 2, 3, 6)

#### Задание

Задать функцию. На заданном отрезке  $[x_{\min}, x_{\max}]$ :  
 протабулировать; построить график; найти корни; найти  $\min$ ,  $\max$ ;  
 найти производную и построить ее график.

№	Функция $f(x)$	$x_{\min}$	$x_{\max}$
1	$x^3 - 13x^2 + 56x - \operatorname{tg} x - 83$	3	7
2	$x^4 + 11.4x^3 + 9.53x^2 - 252.84x - 693$	-7	5
3	$\sin^2 x - \cos^2 x - \cos 0.5x$	-5	5
4	$\operatorname{tg}^2 x - 4 \operatorname{tg} x - 5$	2	4.6
5	$\sin 6x - 2 \sin 18x + \cos 5x$	1	2
6	$x^4 - 2.4x^3 - 12.57x^2 + 32.212x - 17.7156$	-4	4
7	$\sin 2x + \cos 2x - \sqrt{\left 3 \sin \frac{x}{2}\right }$	0	10
8	$\operatorname{tg} x - 4 \cos x$	0.5	10
9	$x^4 + 6.4x^3 - 56.23x^2 - 442.414x - 641.273$	-10	10
10	$x^4 + 1.47x^3 + 61.47x^2 + 9.805x - 274.89$	-5	5
11	$x^5 + 4x^3 - 10x^2 + 3x + 50$	-10	10
12	$x^3 - \sqrt[3]{x^2} - 5 + 7 \sin 3x$	-10	10
13	$x - 5.21 \sin(x + 0.5)$	-10	10
14	$\frac{e^x}{2(x-1)^2} - 10$	-10	10
15	$x^4 + 3.7x^3 - 35.91x^2 - 124x + 13$	-10	10
16	$x^4 + 16.7x^3 + 74.46x^2 + 26x - 169$	-10	10
17	$\ln \left 7 \sin x + \sqrt{ x }\right  + 6 \operatorname{tg} x + 5$	-10	10
18	$\frac{1}{x} - 10 \frac{\sin 3x}{x}$	-10	10
19	$2x + 3 \cos x - 7 \sin 2.5x$	-10	10
20	$x^4 - 2.1x^3 - 13.25x^2 - 3x + 3.43$	-10	10

### 3.5.3. Исследование функции двух переменных

(Индивидуальные варианты заданий для лабораторной работы № 4)

#### Задание

Задать функцию.

Построить поверхность в области  $x \in [x_{\min}, x_{\max}]$ ,  $y \in [y_{\min}, y_{\max}]$ ;  
найти  $\min$ ,  $\max$  в области.

№	Функция $f(x, y)$	$x_{\min}$	$x_{\max}$	$y_{\min}$	$y_{\max}$
1	$x \cos^2 y - y \sin^2 x + e^{-xy}$	1	3	0.5	1.5
2	$\sqrt{ x^3 - \ln xy  + y } - (x + 2y)^2$	1	2	2	2.3
3	$2 \sin^2\left(\frac{x}{y} + \frac{\pi}{7}\right) - e^{-\sqrt{ xy }}$	0.1	0.5	1	4
4	$y \operatorname{tg}\left(x + \frac{\pi}{9}\right) - x^3 \ln^2 y + \sqrt{ x  }$	0.5	1.5	1	3
5	$x \sin^2 y - y \sqrt{ x^2 - \sqrt{ \cos y } }$	1	1.2	0.5	2
6	$\cos xy - \frac{\ln x - y^2  + \sqrt{ 2x - y }}{\operatorname{sh}(x + 2y)}$	-1	0	1	2.5
7	$x \ln^2\left(\sqrt{ x  + y^2}\right) - e^{xy}$	1	1.2	-2	1
8	$y \operatorname{tg}^2(x - y) - \sqrt{ \sin x } e^{2y}$	-0.1	0.1	-3	3
9	$\sqrt{ \ln x  - \cos y } - x^2 y^2 \operatorname{tg}^2\left(x + \frac{\pi}{8}\right)$	1	3	0	1.5
10	$\sqrt{ y^2 - \sin x } - x^2(y + e^{2x})$	0.4	0.6	1	2.5
11	$2.1 \operatorname{ch}^2 xy - y^2 e^{-\sqrt{ x }}$	0.1	0.4	0.5	1
12	$\sqrt{ x \cos y } + y \sin x - e^{\frac{x}{y}}$	-1.1	-0.9	0.25	1

№	Функция $f(x, y)$	$x_{\min}$	$x_{\max}$	$y_{\min}$	$y_{\max}$
13	$x^2 e^{-y} + y^2 e^{-x} + \cos^2 x \sin^2 y$	1	1.2	0.5	2
14	$x\sqrt{ y } + y\sqrt{ x } - e^{-2xy}$	1	1.5	2	2.2
15	$\ln^2 x  + 2.3\ln y  + ye^{-x^2}$	2	3	-3	-1
16	$(x^3 - 2y)^2 - \sqrt{ x } e^{-y}$	1	3	0.5	0.7
17	$x\sqrt{ \sin^2 x - \cos y } - y^2 e^{-x}$	0.1	0.3	1	5
18	$\operatorname{sh}^2(x + y) + \sqrt{ x^3 - y\sqrt{ \cos x } }$	0.5	2.5	0.1	0.5
19	$\sin^3(y - 2x) - \sqrt{ \ln^2 x - y  - x^2 \cos y }$	1.7	2.3	-1.2	-1
20	$\cos^2(x + y) - \sqrt{ \ln xy  - x }$	1	1.6	1	2



### 3.5.4. Сложные функции

(Индивидуальные варианты заданий для лабораторной работы № 5)

#### Задание

Сложные функции (конструкция if).

Задать функцию, построить график.

№	Функции для набора
1	$y(x) = \begin{cases} 3\sqrt{ x-1 } - 3.25 & \text{при } x < -1.25 \\ -x & \text{при } -1.25 \leq x \leq 1.25 \\ 3.25 - 3\sqrt{ x-1 } & \text{при } x > 1.25 \end{cases}$
2	$y(x) = \begin{cases} \sin^2(x+1) & \text{при } x < -1 \\ -(1 + \cos \pi x) & \text{при } -1 \leq x < 1 \\ \sin^2(x-1) & \text{при } x \geq 1 \end{cases}$
3	$y(x) = \begin{cases} -\cos 5\pi(x-0.2) & \text{при } x < 0.2 \\ 1 - 2 \cos 5\pi(x-0.2) & \text{при } 0.2 \leq x < 0.4 \\ 3 & \text{при } x \geq 0.4 \end{cases}$
4	$y(x) = \begin{cases} 1 + 2x^2 & \text{при } x \leq 0 \\ 1 & \text{при } 0 < x < 0.5 \\ 1 + (x - 0.5)^4 & \text{при } x \geq 0.5 \end{cases}$
5	$y(x) = \begin{cases} 6\sqrt[3]{ x } - 5 & \text{при } x < -1 \\ x^2 & \text{при } -1 \leq x \leq 1 \\ 6\sqrt[3]{x} - 5 & \text{при } x > 1 \end{cases}$
6	$y(x) = \begin{cases} 7\sqrt{ x-1 } - 11 & \text{при } x < -3 \\ x & \text{при } -3 \leq x < 3 \\ 7\sqrt{ x+1 } - 11 & \text{при } x \geq 3 \end{cases}$
7	$y(x) = \begin{cases} \pi - (x + \pi)^2 & \text{при } x < -\pi \\ \pi + \cos x + 1 & \text{при } -\pi \leq x < \pi \\ \pi + (x - \pi)^2 & \text{при } x \geq \pi \end{cases}$
8	$y(x) = \begin{cases} 0 & \text{при } x < 1 \\ \ln x/x & \text{при } 1 \leq x < e \\ x^2/e^x & \text{при } x \geq e \end{cases}$

№	Функции для набора
9	$y(x) = \begin{cases} 1 + \sin^3(x + 0.5) & \text{при } x < -0.5 \\ 1 & \text{при } -0.5 \leq x \leq 0.5 \\ 1 + \sin^3(x - 0.5) & \text{при } x > 0.5 \end{cases}$
10	$y(x) = \begin{cases} x + e^{-x} & \text{при } x \leq 0 \\ 1 & \text{при } 0 < x < 1 \\ 1 + (x - 1)^2 & \text{при } x \geq 1 \end{cases}$
11	$y(x) = \begin{cases} -\cos(x + 1) & \text{при } x \leq -1 \\ \sin \frac{\pi}{2} x & \text{при } -1 < x < 1 \\ \cos(x - 1) & \text{при } x \geq 1 \end{cases}$
12	$y(x) = \begin{cases} (x + 1)^2 & \text{при } x < -1 \\ 1 + \cos \pi x & \text{при } -1 \leq x < 1 \\ -(x - 1)^2 & \text{при } x \geq 1 \end{cases}$
13	$y(x) = \begin{cases}  x  & \text{при } x \leq 0 \\ -\sin 2\pi x & \text{при } 0 < x \leq 1 \\ 1 - x & \text{при } x > 1 \end{cases}$
14	$y(x) = \begin{cases} x(x + 2) & \text{при } x \leq -2 \\ \sin \pi x & \text{при } -2 < x \leq 0 \\ x(x + 2) & \text{при } x > 0 \end{cases}$
15	$y(x) = \begin{cases} \sqrt{ x } & \text{при } x < 0 \\ \sin^2 x + \sqrt{x} & \text{при } 0 \leq x \leq \pi \\ x^2 & \text{при } x > \pi \end{cases}$
16	$y(x) = \begin{cases} -1 & \text{при } x < -\pi \\ \cos x & \text{при } -\pi \leq x \leq 0 \\ 1 & \text{при } x > 0 \end{cases}$
17	$y(x) = \begin{cases}  (x + 3)^2 - 12  & \text{при } x < -2 \\ \sqrt[3]{x + 3} & \text{при } -2 \leq x \leq 2 \\ 4 - 0.5x & \text{при } x > 2 \end{cases}$

№	Функции для набора
18	$y(x) = \begin{cases} \sin^2 \pi x & \text{при } x < -1 \\ 0 & \text{при } -1 \leq x \leq 2 \\  \cos x  & \text{при } x > 2 \end{cases}$
19	$y(x) = \begin{cases} (x^2 + 1)/x - 1 & \text{при } x < -3 \\ \sqrt{x^2 - 9} & \text{при } -3 \leq x \leq 3 \\ x - 3 & \text{при } x > 3 \end{cases}$
20	$y(x) = \begin{cases} -1 & \text{при } x < -\pi/2 \\ \sin x & \text{при } -\pi/2 \leq x \leq 0 \\ \sqrt{x} & \text{при } x > 0 \end{cases}$

### 3.5.5. Одномерный массив: сумма, произведение, min, max

(Индивидуальные варианты заданий для лабораторной работы № 7)

#### Задание

Введите массив  $a$  из  $n$  элементов; вычислите значение величины  $Z$  по формуле. При вычислении min, max значение  $i$  изменяется в интервале  $[1, n]$ . Исходные данные задать самостоятельно.

№	Формула для вычисления значения $Z$	$n$
1	$Z = \sum_{i=2}^{n-1} (i! + a_i) - \min_i \left( a_i \prod_{i=1}^{n-2} \sqrt{i} \right)$	6
2	$Z = \min_i \left( a_i \prod_{i=1}^{n-3} i^2 \right) + \sum_{i=3}^n a_i^2$	6
3	$Z = \prod_{i=1}^{n-2} \left( a_i + \sum_{i=2}^{n-1} i! \right) - \min_i (ia_i)$	6
4	$Z = \max_i (ia_i) + \sum_{i=2}^{n-2} a_i^i - \prod_{i=1}^{n-3} i!$	6
5	$Z = \sum_{i=2}^{n-1} \left( i^2 - \prod_{i=1}^{n-3} a_i \right) + \max_i (a_i^2)$	6
6	$Z = \max_i \left( a_i^2 - \prod_{i=3}^{n-1} a_i \right) + \sum_{i=1}^{n-2} \sqrt{i}$	7
7	$Z = \sum_{i=2}^{n-3} \sqrt{i!} + \min_i  a_i  - \prod_{i=1}^{n-3} a_i$	7
8	$Z = \min_i (a_i \sqrt{i}) + \sum_{i=2}^{n-2} \left( i! - \prod_{i=1}^{n-3} a_i^2 \right)$	7
9	$Z = \prod_{i=2}^{n-3} \left( i! - \min_i \{ia_i\} \right) + \sum_{i=1}^{n-2} \sqrt{ia_i}$	7
10	$Z = \max_i \left( a_i \sum_{i=2}^{n-3} i! \right) - \prod_{i=1}^{n-2} a_i^{-i}$	7
11	$Z = \sum_{i=2}^{n-3} \left( \ln i - \prod_{i=1}^{n-2} a_i \right) + \min_i \left( \sqrt{ a_i } \right)$	6

№	Формула для вычисления значения $Z$	$n$
12	$Z = \max_i \left( i a_i^2 \right) - \sum_{i=1}^{n-2} \left( i! + \prod_{i=2}^{n-3} a_i \right)$	6
13	$Z = \prod_{i=1}^{n-2} \left( \sqrt{i} - \min_i \{ a_i^3 \} \right) + \sum_{i=2}^{n-1} a_i$	6
14	$Z = \min_i (i! a_i) + \prod_{i=2}^{n-3} \left( a_i - \sum_{i=1}^{n-2} \sqrt{i} \right)$	6
15	$Z = \sum_{i=1}^{n-2} \left( i! + \max_i  a_i  \right) - \prod_{i=2}^{n-1} \sqrt{ a_i }$	6
16	$Z = \max_i \left( i a_i^2 \right) \sum_{i=1}^{n-1} \left( \sqrt{i!} - \prod_{i=2}^{n-2} a_i \right)$	5
17	$Z = \prod_{i=2}^{n-1} \left( \sqrt{i!} - \max_i  a_i  \right) + \sum_{i=1}^n a_i^2$	5
18	$Z = \min_i \left( a_i + \prod_{i=2}^{n-1} i! \right) + \sum_{i=1}^{n-2} a_i^2$	5
19	$Z = \sum_{i=2}^n \left( \ln i + \max_i \{ i! a_i \} \right) - \prod_{i=2}^n a_i$	5
20	$Z = \max_i \left(  a_i  - \prod_{i=2}^{n-1} a_i^2 \right) + \sum_{i=1}^{n-2} i a_i$	5

### 3.5.6. Обработка матриц: сумма, произведение, min, max

(Индивидуальные варианты заданий для лабораторной работы № 8)

#### Задание

Введите матрицу  $a(n, k)$ . По матрице  $a$  сформировать одномерный массив  $b$ , каждый элемент которого вычисляется по заданной формуле. Четные элементы массива  $b$  вычисляются по верхней формуле, нечетные - по нижней. Отсортировать полученный массив и найти среднее арифметическое его элементов. Исходные данные задать самостоятельно.

Функции  $\min$ ,  $\max$  вычисляются по всем значениям  $j$  ( $j=1, k$ ); для произведения и суммы диапазон изменения индекса проставлен в формуле.

№	Формула для вычисления $b_i$	$n$	$k$
1	$b_i = \begin{cases} \max_j (j a_{ij}) \\ \sum_{j=1}^k j a_{ij} \end{cases}$	4	4
2	$b_i = \begin{cases} \prod_{j=1}^i a_{ij}^j \\ \min_j  a_{ij}  \end{cases}$	4	5
3	$b_i = \begin{cases} \sum_{j=i}^k a_{ij}^2 \\ \max_j (j! a_{ij}) \end{cases}$	5	6
4	$b_i = \begin{cases} \min_j (j a_{ij}^2) \\ \prod_{j=1}^k (a_{ij} - j!) \end{cases}$	4	6
5	$b_i = \begin{cases} \max_j (j a_{ij}^i) \\ \sum_{j=1}^i (a_{ij} + j!) \end{cases}$	5	5
6	$b_i = \begin{cases} \min_j (\sqrt{j} a_{ij}) \\ \prod_{j=i}^k j a_{ij} \end{cases}$	4	6

№	Формула для вычисления $b_i$	$n$	$k$
7	$b_i = \begin{cases} \sum_{j=1}^k ja_{ij}^i \\ \max_j ja_{ij} \end{cases}$	4	4
8	$b_i = \begin{cases} \min_j (j! + a_{ij}) \\ \prod_{j=1}^i ja_{ij} \end{cases}$	4	5
9	$b_i = \begin{cases} \sum_{j=i}^k \sqrt{j!} a_{ij} \\ \max_j (ij - a_{ij}) \end{cases}$	5	5
10	$b_i = \begin{cases} \prod_{j=1}^k ja_{ij}^i \\ \min_j \sqrt{ a_{ij} } \end{cases}$	4	4
11	$b_i = \begin{cases} \sum_{j=1}^i i! a_{ij} \\ \max_j (j a_{ij}) \end{cases}$	5	6
12	$b_i = \begin{cases} \prod_{j=1}^k ja_{ij}^2 \\ \min_j (j! a_{ij}) \end{cases}$	4	6
13	$b_i = \begin{cases} \sum_{j=1}^k a_{ij}^i \\ \max_j (j a_{ij}) \end{cases}$	4	4
14	$b_i = \begin{cases} \prod_{j=1}^i a_{ij} \\ \min_j a_{ij}^i \end{cases}$	4	5
15	$b_i = \begin{cases} \max_j  a_{ij}  \\ \sum_{j=i}^k ij! a_{ij} \end{cases}$	4	4

№	Формула для вычисления $b_i$	$n$	$k$
16	$b_i = \begin{cases} \prod_{j=1}^k (j^2 - a_{ij}) \\ \min_j (a_{ij} - ij) \end{cases}$	5	5
17	$b_i = \begin{cases} \sum_{j=1}^i (a_{ij} - j!) \\ \max_j (ij + a_{ij}) \end{cases}$	4	5
18	$b_i = \begin{cases} \prod_{j=i}^k ja_{ij} \\ \min_j (j + a_{ij}) \end{cases}$	4	4
19	$b_i = \begin{cases} \max_j (a_{ij} - j^2) \\ \sum_{j=1}^k ja_{ij} \end{cases}$	4	6
20	$b_i = \begin{cases} \prod_{j=i}^k (a_{ij} - j^2) \\ \min_j (j! a_{ij}) \end{cases}$	5	6



### 3.5.7. Вычисление определенного интеграла

(Индивидуальные варианты заданий для лабораторной работы № 9)

#### Задание

Вычислите определенный интеграл средствами MathCad и по формуле Ньютона-Лейбница. Постарайтесь найти первообразную средствами MathCad.

№	Подынтегральная функция $f(x)$	Интервал $[a;b]$	Первообразная функция $P(x)$	Значение интеграла
1	$x^2 / (1 + x^6)$	[0;3]	$\frac{1}{3} \arctg x^3$	0.511
2	$\sqrt{x} + \sqrt[3]{x^2}$	[0;1]	$\frac{2}{3} x^{3/2} + \frac{3}{5} x^{5/3}$	1.267
3	$\frac{1}{x(1 + \ln^2 x)}$	[1;e]	$\arctg \ln x$	0.785
4	$\frac{\cos \ln x}{x}$	[1;e]	$\sin \ln x$	0.841
5	$\frac{x^2 + 3}{x - 2}$	[3;4]	$\frac{(x+2)^2}{2} + 7 \ln  x - 2 $	10.352
6	$\sqrt[3]{x-1}$	[2;4]	$\frac{3}{4} (x-1)^{4/3}$	2.495
7	$e^x (x+1)^2$	[0;1]	$e^x (x^2 + 1)$	4.437
8	$x\sqrt{x^2 - 1}$	[1;3]	$\frac{1}{3} (x^2 - 1)^{3/2}$	7.542
9	$\frac{1}{x^2(2 + 3x)}$	[0.3;1]	$\frac{1}{2} \left( \frac{3}{2} \ln \left  \frac{2}{x} + 3 \right  - \frac{1}{x} \right)$	0.672
10	$x^3 e^{x^2}$	[0;1]	$\frac{1}{2} (x^2 - 1) e^{x^2}$	0.5
11	$e^x \sqrt{1 + e^x}$	[0;1]	$\frac{2}{3} \sqrt{(1 + e^x)^3}$	2.894
12	$x \ln(x^2 + 1)$	[1;2]	$\frac{1}{2} ((x^2 + 1) \ln(x^2 + 1) - x^2)$	1.83
13	$x^3 / \sqrt{(1 + x^2)^3}$	[1;3]	$\sqrt{1 + x^2} + 1 / \sqrt{1 + x^2}$	1.357
14	$\ln(x + 1)$	[1;2]	$(x + 1) \ln  x + 1  - x$	0.91

№	Подынтегральная функция $f(x)$	Интервал $[a;b]$	Первообразная функция $P(x)$	Значение интеграла
15	$x/(1+x^4)$	$[0;1]$	$\frac{1}{2} \operatorname{arctg} x^2$	0.392
16	$x^2/(2+x^3)$	$[0;2]$	$\frac{1}{3} \ln  2+x^3 $	0.5365
17	$x/(1+2x)$	$[0;2]$	$\frac{1}{2}(x-0,5 \ln  1+2x )$	0.5976
18	$\frac{x^3}{1+2x^2}$	$[1;2]$	$\frac{x^2}{4} - \frac{1}{8} \ln(1+2x^2)$	0.6126
19	$\frac{x}{(1+4x^2)^2}$	$[0;3]$	$\frac{-1}{8(1+4x^2)}$	0.122
20	$e^x/\sqrt{1+e^x}$	$[1;2]$	$2\sqrt{(1+e^x)^3}$	1.9362

### 3.5.8. Вычисление площади живого сечения однорукавного русла и построение рельефа дна

(Индивидуальные варианты заданий для лабораторной работы № 10)

#### Задание

По приведенным результатам замеров построить рельеф сечения дна и вычислить площадь сечения русла. Число точек  $m$  взять равным 8.

№	координаты $x_i$	глубины $h_i$
1.	0 100 130 180 240 250 310 390	0,0 1,8 1,1 3,5 2,8 2,0 3,5 0,0
2.	0 10 50 100 150 210 270 330	0,0 3,2 1,3 2,9 2,8 2,7 4,0 0,0
3.	0 90 130 150 170 180 230 240	0,0 2,9 3,6 2,0 2,8 2,7 3,1 0,0
4.	0 10 80 140 160 200 240 260	0,0 1,5 3,6 2,5 2,3 1,4 1,4 0,0
5.	0 30 120 220 260 290 330 380	0,0 1,7 3,2 3,9 2,8 3,1 2,3 0,0
6.	0 10 90 150 180 210 250 350	0,0 1,3 3,0 2,3 1,4 1,6 2,5 0,0
7.	0 20 80 160 230 240 260 350	0,0 1,2 1,8 3,1 3,7 1,4 2,2 0,0
8.	0 40 90 130 200 250 260 310	0,0 2,3 3,6 1,8 1,6 1,8 2,9 0,0
9.	0 90 180 260 310 360 460 560	0,0 1,3 2,7 1,6 1,4 3,7 1,9 0,0
10.	0 90 110 180 230 310 390 400	0,0 1,7 2,7 1,4 2,0 3,6 3,9 0,0
11.	0 10 80 80 130 190 190 250	0,0 2,2 2,0 3,0 2,8 1,6 3,6 0,0
12.	0 50 110 190 220 250 320 410	0,0 2,8 2,6 3,2 1,5 2,6 2,0 0,0
13.	0 80 140 200 260 280 360 460	0,0 2,1 1,9 2,6 1,9 3,6 4,0 0,0
14.	0 40 130 150 170 270 320 320	0,0 1,7 2,9 1,7 1,6 3,2 1,7 0,0
15.	0 80 170 180 270 350 370 390	0,0 1,9 2,2 3,0 2,2 1,9 3,5 0,0
16.	0 80 110 190 270 320 410 460	0,0 1,7 3,6 2,3 2,7 2,1 1,1 0,0
17.	0 20 110 200 240 280 310 400	0,0 3,7 3,3 1,9 3,0 3,3 2,7 0,0
18.	0 70 80 170 180 190 190 250	0,0 1,1 3,5 3,3 3,5 2,0 2,9 0,0
19.	0 90 110 200 210 280 360 400	0,0 1,7 3,2 3,2 3,6 1,6 2,5 0,0
20.	0 40 110 140 240 250 270 310	0,0 1,6 3,7 2,4 1,4 1,2 2,3 0,0

### 3.5.9. Решение ОДУ первого порядка

(Индивидуальные варианты заданий для лабораторной работы № 11)

#### Задание

Найти решение дифференциального уравнения первого порядка

$\frac{dy}{dx} = f(x, y)$  на интервале  $[x_0, x_k]$  при начальных условиях  $y(x_0) = y_0$ . По-

строить график, проверить значения в контрольных точках.

№	$f(x, y)$	$y_0$	$[x_0, x_k]$	Значения $y$
1	$\frac{(x^2 + y^2)^2}{2x^2 y} - \frac{x}{y}$	1	[1; 1,9]	$y(1,5)=1,9365$ $y(1,9)=5,8643$
2	$\frac{x^2 - y^2}{xy}$	1	[1; 3]	$y(2)=1,4577$ $y(3)=2,1343$
3	$\frac{y^2}{2xy + 3}$	-1	[1; 3]	$y(2) = -0,5$ $y(3) = -1/3$
4	$y(1 - y \cos x)$	1	[0; 2]	$y(1)=1,1451$ $y(2)=3,1822$
5	$\frac{\sqrt{x^2 + y^2} - x}{y}$	1	[0; 2]	$y(1)=1,7321$ $y(2)=2,2361$
6	$x + \frac{y}{x}$	0	[1; 3]	$y(2)=2$ $y(3)=6$
7	$\frac{1 - xy}{1 - x^2}$	1	[0; 0,9]	$y(0,5)=1,366$ $y(0,9)=1,3359$
8	$\frac{2y}{x(x^2 y - 2)}$	1	[1; 3]	$y(2)=0,25$ $y(3)=0,111$
9	$1 + \frac{y(2x - 1)}{x^2}$	2	[1; 3]	$y(2)=6,4261$ $y(3)=13,621$
10	$\cos x(\sin x - y)$	-1	[0; 2]	$y(1) = -0,1585$ $y(2) = -0,0907$
11	$\frac{x + y^2}{2xy}$	1	[1; 3]	$y(2)=1,8402$ $y(3)=2,5091$
12	$\frac{y(1 + xy)}{x}$	1	[1; 1,7]	$y(1,4)=2,69$ $y(1,7)=30,91$

№	$f(x, y)$	$y_0$	$[x_0, x_k]$	Значения $y$
13	$\frac{y^2+1}{x^2+1}$	1	[0; 0,9]	$y(0,5)=3$ $y(0,9)=19$
14	$\frac{y-2}{x(1+2x)}$	1	[1; 3]	$y(2)=0,8$ $y(3)=0,7143$
15	$\frac{y^2-2xy-x^2}{y^2+2xy-x^2}$	-1	[1; 3]	$y(2)=-2$ $y(3)=-3$
16	$\frac{y(xy \ln x - 1)}{x}$	1	[1; 3]	$y(2)=0,6581$ $y(3)=0,8406$
17	$\frac{2y}{6x-y^2}$	2	[2; 4]	$y(3)=2,4495$ $y(4)=2,8284$
18	$1+\frac{y}{x(x+1)}$	0	[1; 3]	$y(2)=1,1287$ $y(3)=2,3239$
19	$x^5+x^2+3x^2y$	1	[0; 1]	$y(0,5)=1,1802$ $y(1)=3,5305$
20	$4x-2y$	0	[0; 2]	$y(1)=1,1353$ $y(2)=3,0183$

## Литература

1. Дьяконов В. MATHCAD 8/2000: специальный справочник – СПб: Питер, 2001. – 592 с.
2. Кирьянов Д.В. Самоучитель MathCad 2001. – СПб.: БХВ-Петербург, 2002. – 544 с.
3. Сдвижков О.А. MathCAD-2000: Введение в компьютерную математику. – М.: Издательско-торговая корпорация "Дашков и К°", 2002. – 204 с.
4. Плис А.И., Сливина Н.А. MathCad. Математический практикум для инженеров и экономистов; Учеб. пособие. – 2-е изд., перераб. и доп. – М.: Финансы и статистика, 2003. – 656 с.
5. Гурский Д.А. Вычисления в MathCad – Мн.: Новое знание, 2003. – 814 с.
6. Порошнев С.В. Компьютерное моделирование физических процессов с использованием пакета MathCad. Учебное пособие – М.; Горячая линия – Телеком, 2002: - 252 с.
7. Очков В.Ф. MathCad 8 Pro для студентов и инженеров. – М.: Компьютер Пресс, 1999. – 523 с.
8. Каганов В.И. Радиотехника + Компьютер + MathCad. – М.: Горячая линия – Телеком, 2001. – 416 с.
9. Гурьяшова Р.Н. Алгоритмы и технологии решения дифференциальных уравнений. Методическое пособие. – Н. Новгород: Изд-во ГОУ ВПО ВГАВТ, 2004. – 56 с.
10. Дьяконов В.П. Энциклопедия MathCad 2001i и MathCad 11. – М.: СОЛОН-Пресс, 2004. – 832с.: ил.– (серия "Библиотека профессионала")
11. Черняк А.А., Черняк Ж.А., Доманова Р.А. Высшая математика на базе MathCad. Общий курс. – СПб.: БХВ-Петербург, 2004. – 608с.: ил.

<b>Содержание</b>	
Введение	3
1. Основы работы	4
1.1. Основные положения	4
1.2. Набор и редактирование формул	5
1.3. Запись констант и переменных	10
1.3.1. Числовые константы	10
1.3.2. Переменные	11
1.3.3. Присваивание	12
1.3.4. Системные переменные	12
1.4. Работа с текстом	12
1.5. Переменные - диапазоны	15
1.6. Функции в Mathcad	17
1.7. Графики в Mathcad. Декартовы координаты	21
1.7.1. Общая информация	21
1.7.2. Построение графика по точкам	22
1.7.3. Быстрый график	24
1.7.4. Исследование графика: инструмент Увеличить (Zoom)	27
1.7.5. Исследование графика: инструмент Трассировка (Trace)	28
1.7.6. Настройка графика	28
2. Дополнительные возможности	31
2.1. Прочие типы графиков. Объемные графики	32
2.1.1. График в полярных координатах	32
2.1.2. Графики "Поверхность", "Линии уровня" и "Объемная гистограмма" по точкам	32
2.1.3. Объемные кривые по точкам	36
2.1.4. Быстрый объемный график	37
2.1.5. Модификация объемных графиков	38
2.1.6. График "векторное поле"	40
2.2. Сложные функции	40
2.2.1. Встроенные разрывные функции	40
2.2.2. Функция if	41
2.2.3. Программная конструкция if	42
2.2.4. Использование логических множителей	43
2.3. Решение уравнений и систем	44
2.3.1. Функция root	44
2.3.2. Конструкция Given / Find	46
2.3.3. Точность вычислений	49
2.3.4. Конструкция Given / Minerr	50

2.3.5. Специальные виды уравнений и систем	52
2.3.6. Средства для символьного решения уравнений и систем	54
2.4. Решение задач оптимизации	54
2.4.1. Оптимизация функции одной переменной без ограничений	55
2.4.2. Оптимизация функции нескольких переменных без ограничений	56
2.4.3. Задачи оптимизации с ограничениями	57
2.5. Работа с векторами и матрицами	59
2.5.1. Задание векторов и матриц	59
2.5.2. Ввод/вывод матриц из внешних источников	62
2.5.3. Вычисление сумм и произведений	64
2.6. Символьные вычисления	66
2.6.1. "Живые" символьные вычисления	66
2.6.2. Команды меню Symbolic	67
2.7. Решение дифференциальных уравнений	69
2.7.1. Окружение odesolve	69
2.7.2. Функция rkfixed	71
2.8. Интегрирование	75
2.8.1. Алгоритмы вычисления определенного интеграла	75
3. Приложения	78
Приложение 3.1. Контрольные вопросы	78
Приложение 3.2. Горячие клавиши	82
Приложение 3.3. Исследование поведения математической модели в пакете MathCad. Авторулевой. (Пример выполнения курсовой работы)	84
Приложение 3.4. Примеры выполнения лабораторных работ в пакете mathCad	89
3.4.1. Лабораторная работа № 1. Набор формул	89
3.4.2. Лабораторная работа № 2. Построение графика функции одной переменной в декартовых координатах	90
3.4.3. Лабораторная работа № 3. Исследование функции одной переменной	93
3.4.4. Лабораторная работа № 4. Исследование функции двух переменных	97
3.4.5. Лабораторная работа № 5. Вычисление сложной функции	101
3.4.6. Лабораторная работа № 6. Нахождение корней нелинейного уравнения	103
3.4.7. Лабораторная работа № 7. Обработка одномерного массива	106



3.4.8. Лабораторная работа № 8. Работа с матрицами	108
3.4.9. Лабораторная работа № 9. Вычисление определенного интеграла	110
3.4.10. Лабораторная работа № 10. Вычисление площади живого сечения однорукавного русла и построение релье- ефа дна	112
3.4.11. Лабораторная работа № 11. Решение ОДУ первого порядка	114
3.5. Задания на лабораторные работы	116
3.5.1. Основы работы: освоение набора формул	116
3.5.2. Исследование функции одной переменной	119
3.5.3. Исследование функции двух переменных	120
3.5.4. Сложные функции	122
3.5.5. Одномерный массив: сумма, произведение, $\min$ , $\max$	125
3.5.6. Обработка матриц: сумма, произведение, $\min$ , $\max$	127
3.5.7. Вычисление определенного интеграла	130
3.5.8. Вычисление площади живого сечения однорукавного русла и построение рельефа дна	132
3.5.9. Решение ОДУ первого порядка	133
Литература	135

*Гурьяшова Римма Николаевна, Шеянов Анатолий Владимирович*

**Информатика. Пакет Mathcad**

Учебное пособие

*Печатается по авторскому оригиналу  
без редактирования и корректуры*

Формат бумаги 60x84 <sup>1</sup>/<sub>16</sub>.

Ризография. Усл. печ. л. 7,79. Уч.-изд. л. 8,25.

Заказ 106. Тираж 810. Цена договорная.

---

Издательско-полиграфический комплекс ФГОУ ВПО «ВГАВТ»

603600, Нижний Новгород, ул. Нестерова, 5

Вопросы для самоподготовки к зачету по дисциплине «Информатика» за 2 семестр  
(специальность 26.05.07 «Эксплуатация судового электрооборудования и средств  
автоматики»,  
профиль «Эксплуатация судового электрооборудования и средств автоматики»)

Цель дисциплины – сформировать у обучающихся следующие компетенции:

ОПК-5: Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности;

ПК-6: Способен осуществлять безопасное техническое использование, техническое обслуживание судовой компьютерной информационной системы в соответствии с международными и национальными требованиями;

1. Безопасность данных на ПК. Средства и приемы безопасной работы. (ПК-6)
2. Понятие о типах вредоносных программ. (ПК-6)
3. Настройки компьютера для повышения безопасности информации. (ПК-6)
4. Брандмауэры (файрволлы). Типы брандмауэров. Их назначение. (ПК-6)
5. Международные стандарты информационного обмена. (ПК-6)
6. Информационная безопасность в глобальных сетях. (ПК-6)
7. Виды возможных нарушений информационной безопасности (ПК-6)
8. Защита информации от несанкционированного доступа. (ПК-6)
9. Основные инструменты текстового редактора MS Word. (ОПК-5)
10. Инструменты табличного процессора MS Excel. (ОПК-5)
11. Адресация данных в MS Excel. (ОПК-5)
12. Назначение микропроцессоров. Характеристика регистров МП. (ОПК-5)
13. Регистры общего назначения. Двойные регистры. Слово состояния процессора. (ОПК-5)
14. Регистр признаков (флагов). Примеры использования (ОПК-5)
15. Регистры: указатель стека, программный счетчик. Назначение. (ОПК-5)
16. Структура машинной команды МП. (ОПК-5)
17. Команды пересылки данных МП. (ОПК-5)
18. Команды МП с косвенной адресацией. Примеры. (ОПК-5)
19. Арифметические команды МП. Примеры. (ОПК-5)
20. Команды условных и безусловных переходов МП. (ОПК-5)
21. Логические команды МП. Примеры использования. (ОПК-5)
22. Принцип маскирования. Примеры использования (ОПК-5)
23. Команды сдвига. Примеры использования. (ОПК-5)
24. Подпрограммы и регистр стека МП. (ОПК-5)
25. Организация и принцип работы стека. (ОПК-5)
26. Алгоритм подпрограммы временной задержки МП. (ОПК-5)
27. Организация индикации с помощью светодиодов. Кодировка символов. (ОПК-5)
28. Алгоритм сдвига символов на индикаторах. (ОПК-5)
29. Алгоритм «движущегося текста» на индикаторах. Общая блок-схема. (ОПК-5)
30. Алгоритм сложения однобайтных чисел, если их сумма больше 255. (ОПК-5)
31. Алгоритм умножения однобайтных чисел, если произведение больше 255. (ОПК-5)
32. Алгоритм умножения однобайтного числа на двухбайтное. (ОПК-5)
33. Использование режима BRK при отладке программ МП. (ОПК-5)
34. Инструменты пакета MathCAD.
35. Основные команды MathCAD.
36. Построение графиков в MathCAD.
37. Исследование функций MathCAD.
38. Способы задания сложных функций MathCAD.
39. Обработка данных одномерных массивов в MathCAD.
40. Обработка данных двумерных массивов в MathCAD.
41. Вычисление определенных интегралов в MathCAD.

Федеральное агентство морского и речного транспорта  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный университет водного транспорта»

Кафедра информатики, систем управления  
и телекоммуникаций

Т.И. Гаврилова

## ОСНОВЫ ЦИФРОВОГО МОДЕЛИРОВАНИЯ

Учебно-методическое пособие  
для студентов очного и заочного обучения  
специальности 26.05.07-65  
«Эксплуатация судового электрооборудования и средств автоматики»

Нижний Новгород  
Издательство ФГБОУ ВО «ВГУВТ»

2016

**УДК 004.942:621.3.011**

**Г**

**Гаврилова, Т.И.**

Основы цифрового моделирования: учеб.-метод. пособие для студентов всех форм обуч. спец-ти 26.05.07-65 «Эксплуатация судового электрооборудования и средств автоматики» / Т.И. Гаврилова.–

Н. Новгород: Изд-во ФГБОУ ВО «ВГУВТ», 2016. – 47 с.

Даётся понятие модели и различных видов моделирования, в том числе математического. Демонстрируются приёмы моделирования на примере электрических цепей. Рассматриваются способы решения дифференциальных уравнений, являющихся широко используемым аппаратом построения математических моделей динамических систем. Приводятся примеры получения решения с помощью языка программирования Си, а также в среде математического пакета MathCAD 14.0 компании Parametric Technology Corporation.

Работа рекомендована к изданию кафедрой информатики, систем управления и телекоммуникаций (протокол №4 от 04.03.2016 г.).

© ФГБОУ ВО «ВГУВТ», 2016



## Введение

Моделирование как вид деятельности, направленной на получение новых представлений об окружающем мире, используется давно. История развития человечества характеризуется постоянным совершенствованием этого метода познания.

Процесс моделирования требует выделения наиболее существенных для проводимого исследования свойств объекта. Так для изучения аэродинамических качеств модели самолета в аэродинамической трубе важно, чтобы модель имела геометрическое подобие оригинала.

Модель — это некий новый объект, который отражает существенные особенности изучаемого объекта, явления или процесса. Для одного объекта может быть построено множество моделей. Примерами моделей могут быть макеты, схемы, чертежи, словесные описания, математические формулы, карты, натурные модели судов, испытываемые в опытовом бассейне и т.д.

Существуют различные классификации моделей.

По способу представления все модели можно разбить на два больших класса: модели предметные (натурные) и модели информационные.

Предметные (натурные) модели воспроизводят геометрические, физические и другие свойства объектов в материальной форме (глобус, анатомические муляжи, модели кристаллических решеток, макеты зданий и сооружений, модели судов и др.).

Информационные модели представляют объекты и процессы в образной (мысленной, разговорной) или знаковой форме, выраженной средствами некоторого формального языка (графики, таблицы, математические формулы, компьютерные программы).

Классификация по фактору времени позволяет выделить модели статические и динамические.

Статическое моделирование служит для описания состояния объекта в фиксированный момент времени, а динамическое — для исследования объекта во времени.

В пособии рассматривается динамическое моделирование, т.е. моделирование объектов, функционирующих в непрерывном времени, непрерывно наблюдаемых и изменяющих свое состояние под воздействием внешних и внутренних причин.



# 1. Компьютерное моделирование

Развитие вычислительной техники привело к появлению нового метода научных исследований – *компьютерного цифрового моделирования*, который основывается на построении математических моделей для описания изучаемых объектов и использовании вычислительных машин для решения возникающих математических задач.

Моделирование на цифровых вычислительных машинах является одним из наиболее мощных средств исследования. Оно применяется при проектировании систем, в задачах прогнозирования, в автоматизированных системах управления, при решении вопросов оптимизации.

Компьютерное моделирование также позволяет изучать объекты, натурные эксперименты с которыми не целесообразны из-за соображений безопасности или дороговизны. Используя этот метод, можно узнать свойства моделируемого объекта, которые редко проявляются при обычных условиях функционирования, но могут проявиться при особых условиях, которые также могут быть смоделированы исследователем.

## *Этапы компьютерного моделирования:*

**1 этап.** Получение математической модели.

Математическая модель – это модель объекта, процесса или явления, представляющая собой математические закономерности, с помощью которых описаны основные характеристики моделируемого объекта, процесса или явления.

Построение математической модели выполняется на основании законов, действующих в исследуемой системе, и завершается записью сформулированных в математических терминах представлений о связях между различными характеристиками системы.

**2 этап.** Выбор математического аппарата для анализа математической модели и программных средств для получения численных результатов решения задачи.

Математическое моделирование объектов различной природы часто приводит к получению одинаковых математических задач.

Такие типичные математические задачи рассматриваются исследователями как самостоятельный объект изучения.

**3 этап.** Проведение серии вычислительных экспериментов с целью выяснить, удовлетворяет ли модель критерию практики.

На этом этапе выясняется, согласуются ли расчетные значения с результатами натуральных испытаний. Если отклонения расчетных значений от тех, что получены в натуральных экспериментах, выходят за пределы точности наблюдений, то говорят, что модель неадекватна, и не может быть принята. В этом случае следует вернуться к первому этапу и уточнить математическую модель. Возможно, учесть факторы и связи, ранее не отраженные в модели.

*Модель признается адекватной*, если степень её соответствия характеру моделируемого явления не выходит за пределы точности наблюдений.

В англоязычной научной литературе существует термин «computer simulation» или «компьютерная имитация», что означает реализованную на ЭВМ математическую модель, оперирующую системой динамических уравнений.

**4 этап.** Проведение серии вычислительных экспериментов на основе модели, признанной адекватной, при различных значениях параметров, характеризующих изучаемый объект и, возможно, разных начальных условиях. На основе данных этих экспериментов исследователи получают новые знания об объекте и его свойствах, находят оптимальные параметры и режимы работы.

**5 этап.** Анализ модели в связи с накоплением новых научных данных об изучаемых объектах и модернизация модели на их основе.

### ***Контрольные вопросы.***

- 1) Что такое модель? Приведите примеры моделей.
- 2) Назовите известные Вам способы классификации моделей.
- 3) На какие два класса можно разбить модели по способу представления? Приведите примеры.
- 4) Что такое математическая модель? Как её можно классифицировать?
- 5) Что подразумевают под компьютерным цифровым моделированием?
- 6) Назовите этапы компьютерного моделирования.
- 7) Что является целью компьютерного моделирования?

## 2. Получение математических моделей электротехнических цепей

Компьютерное моделирование электротехнических систем связано с использованием физических законов, описывающих электрические, а также электромагнитные явления и процессы.

Математическая модель любого объекта характеризуется внутренними, внешними, выходными параметрами и фазовыми переменными. Внутренние параметры модели в нашем случае определяются номиналами элементов принципиальной схемы. Если проектируемый объект содержит  $n$  элементарных компонентов, то и его математическая модель будет определяться параметрами, которые образуют вектор внутренних параметров  $W = |w_1 \dots w_n|^T$ .

Выходные параметры модели могут характеризовать функциональные, эксплуатационные, конструкторско-технологические, экономические и другие характеристики проектируемого объекта. К таким показателям могут относиться коэффициенты передачи, сила тока на участке цепи, надежность, стоимость и т.п. Вектор выходных параметров модели будем обозначать  $Y = |y_1 \dots y_k|^T$ .

Внешние параметры модели — это характеристики внешней по отношению к проектируемому объекту среды, а также рабочие управляющие воздействия. Будем обозначать вектор внешних параметров  $X = |x_1 \dots x_k|^T$ .

В электрических цепях мы будем рассматривать в качестве выходных параметров модели либо напряжения, либо токи, либо заряд конденсатора, в качестве входных параметров — напряжение, подаваемое на вход цепи (напряжение, создаваемое источником э.д.с.).

### 2.1. Математические модели неразветвленных цепей

#### 2.1.1. Математическая модель RC-цепи. Пример 1

Начнем рассмотрение вопроса получения математических моделей с примера цепи, содержащей всего два последовательно

соединенных элемента: резистор номинального сопротивления  $R$  и конденсатор электрической ёмкости  $C$  (рис. 1).

Для построения математической модели примем допущение, что величины  $R$  и  $C$  являются постоянными в течение всего времени наблюдения. Такой приём физической и математической идеализации используется в случаях, когда цель моделирования позволяет пренебречь возможными зависимостями сопротивлений и ёмкостей от температурных режимов эксплуатации и речь не идет об элементах цепи, управляемых током или напряжением (в таких случаях математическая модель превращается в дифференциальное уравнение с переменными коэффициентами).

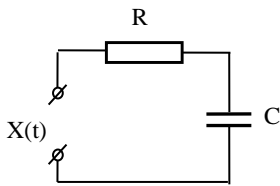


Рис. 1. RC - цепь

В зависимости от того, какой параметр будет выбран в качестве выходного, можно получать различные математические модели.

Рассмотрим в качестве выходного параметра заряд конденсатора  $q(t)$ , который является непрерывной функцией времени. Получим математическую модель

нашей цепи при входном напряжении  $1 \text{ В}$ .

Математическую модель объекта, представляющего объединение компонентов, получают на основе математических моделей этих компонентов. В нашей схеме для резистора справедлив закон Ома (1), выражающий соотношение между током и напряжением:

$$U_R(t) = R \cdot I_R(t). \quad (1)$$

Известно соотношение (2), описывающее связь между зарядом конденсатора и напряжением на нем:

$$q_C(t) = C \cdot U_C(t). \quad (2)$$

Объединение компонентных уравнений в математическую модель объекта осуществляется на основе фундаментальных физических законов, выражающих условия непрерывности и равновесия фазовых переменных, например законов (правил) Кирхгофа (3). Уравнения, описывающие эти законы, называют топологическими; они отражают связи между компонентами в устройстве.

$$U_R(t) + U_C(t) = X(t). \quad (3)$$

Получим из соотношения (2)  $U_C$  и подставим его, а также выражение (1), в выражение (3):

$$R \cdot I_R(t) + \frac{q_C(t)}{C} = X(t). \quad (4)$$

Полученное выражение не может быть признано математической моделью, так как решение этого уравнения получить невозможно: оно содержит две неизвестные функции  $I_R(t)$  и  $q_C(t)$ .

Известно, что токи на различных элементах цепи при их последовательном соединении равны, то есть

$$I_C(t) = I_R(t). \quad (5)$$

Продифференцировав выражение (2), получим:

$$I_C(t) = \frac{dq_C(t)}{dt} = C \cdot \frac{dU_C(t)}{dt}. \quad (6)$$

Используя это выражение и соотношение (4), получим искомую математическую модель:

$$R \cdot \frac{dq_C(t)}{dt} + \frac{q_C(t)}{C} = X(t). \quad (7)$$

Математическая модель содержит внутренние ( $R$  и  $C$ ) и внешние  $X(t)$  параметры цепи, а также одну неизвестную функцию  $q_C(t)$  и её производную.

Таким образом, мы получили математическую модель в виде дифференциального уравнения первого порядка.

### 2.1.2. Математическая модель RC-цепи. Пример 2

Выберем теперь в качестве выходного параметра напряжение на конденсаторе  $U_C(t)$ .

Используя соотношения (2) и (7), получим математическую модель в виде:

$$R \cdot C \cdot \frac{dU_C(t)}{dt} + U_C(t) = X(t).$$

Преобразуем это выражение, приведя его к следующему виду:

$$\frac{dU_C(t)}{dt} = \frac{X(t) - U_C(t)}{R \cdot C}. \quad (8)$$

Если записать это дифференциальное уравнение, применив преобразование Лапласа, широко используемое в электротехнике и теории автоматического управления, получим

$$R \cdot C \cdot p \cdot U_C(t) + U_C(t) = X(t).$$

Записав передаточную функцию звена, представленного данной схемой, получим:

$$W(t) = \frac{U_C(t)}{X(t)} = \frac{1}{R \cdot C \cdot p + 1}.$$

Это передаточная функция апериодического звена первого порядка. Постоянная времени, характеризующая инерционность этого звена,  $T = R \cdot C$ .

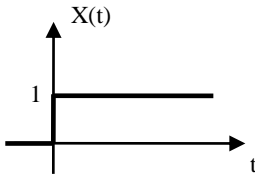


Рис. 2. Единичная ступенчатая функция.

Если на вход последовательной RC-цепи (рис. 1) с полностью разряженным конденсатором подать постоянное напряжение в виде единичной ступенчатой функции, или функции Хевисайда (рис. 2), то напряжение на конденсаторе достигнет значения  $\approx 1$  за время, равное  $3 \cdot T$ , то есть  $3 \cdot R \cdot C$ .

Проинтегрировав выражение (8), получим

$$U_C(t) = \frac{1}{R \cdot C} \int (X(t) - U_C(t)) dt.$$

### 2.1.3. Математическая модель RC-цепи. Пример 3

Пусть теперь выходным параметром будет напряжение на резисторе  $U_R(t)$ . Воспользуемся равенством токов (5), подставив в него выражение для тока на резисторе, полученное из (1) и на конденсаторе (6):

$$\frac{U_R(t)}{R} = C \cdot \frac{dU_C(t)}{dt}. \quad 9)$$

Дальнейшие действия сводятся к тому, чтобы из 2-го правила Кирхгофа исключить неизвестные функции, не являющиеся выходными параметрами модели, то есть,  $U_C(t)$ . Следует отметить, что из выражения (6) можно выразить не саму неизвестную функцию  $U_C(t)$ , а её производную. А для того, чтобы эту производную можно было подставить во второе правило Кирхгофа (3), его необходимо продифференцировать:

$$\frac{dU_R(t)}{dt} + \frac{dU_C(t)}{dt} = \frac{dX(t)}{dt}. \quad (10)$$

Подставив сюда выражение для производной функции  $U_C(t)$  из (9), получим:

$$\frac{dU_R(t)}{dt} + \frac{U_R(t)}{R \cdot C} = \frac{dX(t)}{dt}. \quad (11)$$

Уравнение содержит одну неизвестную функцию  $U_R(t)$  и её производную, производную от входного напряжения и параметры цепи. Таким образом, получена третья математическая модель для одной и той же электрической цепи.

### 2.2.1. RLC-цепь (колебательный контур). Пример 1

Рассмотрим электрическую схему, состоящую из последовательно соединённых резистора, конденсатора и катушки, которую принято называть колебательным контуром.

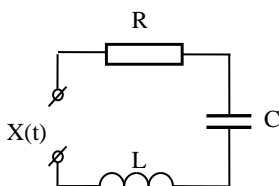


Рис. 3. RLC - цепь

Для этой цепи также можно получать разные математические модели, если выбирать в качестве выходных разные параметры. Например, можно в качестве выходных параметров рассматривать такие функции времени, как напряжение на резисторе, или на конденсаторе, или

на катушке и т.д.

Рассмотрим математическую модель, для которой выходным параметром будет напряжение на резисторе.

Основные физические законы, используемые для вывода модели, это закон Ома (1), связь между зарядом конденсатора и напряжением на нём (2), и соотношение, справедливое для катушки:

$$U_L(t) = L \cdot \frac{di_L(t)}{dt}. \quad (12)$$

Связь между элементами цепи выражается вторым законом Кирхгофа:

$$U_R(t) + U_C(t) + U_L(t) = X(t) \quad (13)$$

Задача получения математической модели в виде дифференциального уравнения сводится в нашем случае к исключению из выражения (13) неизвестных функций, не являющихся выходным напряжением  $U_R(t)$ . Ключом к решению этой задачи является равенство токов:

$$I_C(t) = I_R(t) = I_L(t). \quad (14)$$

Из первой части выражения (14) получим

$$C \cdot \frac{dU_C(t)}{dt} = \frac{U_R(t)}{R}. \quad (15)$$

Из второй части выражения (14) невозможно исключить ток  $I_L(t)$ , так как в выражении (12) мы имеем производную от тока. Поэтому продифференцируем вторую часть равенства токов

$$\frac{dI_R(t)}{dt} = \frac{dI_L(t)}{dt} \text{ и} \quad (16)$$

подставим сюда производную от тока на катушке из (12) и продифференцированный закон Ома (1):

$$\frac{1}{R} \cdot \frac{dU_R(t)}{dt} = \frac{U_L(t)}{L}. \quad (17)$$

Теперь необходимо исключить из 2-го правила Кирхгофа неизвестные функции, не являющиеся выходными параметрами модели, то есть,  $U_C(t)$  и  $U_L(t)$ . Надо отметить, что в выражении (15) содержится производная от напряжения на конденсаторе, а не сама эта функция. Это значит, что нужно будет продифференцировать второе правило Кирхгофа (13)



$$\frac{dU_R(t)}{dt} + \frac{dU_C(t)}{dt} + \frac{dU_L(t)}{dt} = \frac{dX(t)}{dt}, \text{ а} \quad (18)$$

также выражение (17), так как теперь потребуется еще и производная от напряжения на катушке:

$$\frac{dU_L(t)}{dt} = \frac{L}{R} \cdot \frac{d^2U_R(t)}{dt^2}. \quad (19)$$

Выполнив подстановку (15) и (19) в (18) получим математическую модель:

$$\frac{L}{R} \frac{d^2U_R(t)}{dt^2} + \frac{dU_R(t)}{dt} + \frac{1}{R \cdot C} U_R(t) = \frac{dX(t)}{dt}. \quad (20)$$

### 2.2.2. RLC-цепь. Пример 2

Рассмотрим ту же самую схему (рис. 3), состоящую из последовательно соединённых резистора, конденсатора и катушки.

Составим математическую модель, для которой выходным параметром будет напряжение на конденсаторе.

По-прежнему, основной задачей при построении математической модели будет исключение неизвестных функций из второго правила Кирхгофа.

$$U_R(t) + U_C(t) + U_L(t) = X(t)$$

Таковыми функциями будут в нашем случае зависимости напряжения на резисторе и на катушке от времени.

Исключить их из этого выражения можно, пользуясь тем, что токи на различных элементах цепи одинаковы. Так как

$$I_C(t) = I_R(t),$$

то на основе выражения (6) можно записать, что

$$C \cdot \frac{dU_C(t)}{dt} = \frac{U_R(t)}{R}. \quad (21)$$

Также равны токи на конденсаторе и катушке:

$$I_C(t) = I_L(t).$$

Для того, чтобы перейти к выражению, содержащему напряжения, потребуется продифференцировать это равенство:

$$\frac{dI_C(t)}{dt} = \frac{dI_L(t)}{dt}. \quad (22)$$

Выразим необходимые производные из выражений (12) и (6):

$$\frac{dI_C(t)}{dt} = C \frac{d^2U_C(t)}{dt^2}, \quad \frac{dI_L(t)}{dt} = \frac{U_L(t)}{L}.$$

Подставив их в (22), получим

$$C \frac{d^2U_C(t)}{dt^2} = \frac{U_L(t)}{L}.$$

Подставив выражения для неизвестных функций из (21) и (22) во второе правило Кирхгофа, получим математическую модель интегрирующей RLC-цепи.

$$CL \frac{d^2U_C(t)}{dt^2} + RC \frac{dU_C}{dt} + U_C(t) = X(t) \quad (23)$$

### 2.2.3. RLC-цепь. Построение математической модели с использованием преобразования Лапласа

Есть другой способ получить математическую модель, используя преобразование Лапласа.

Для этого перепишем выражения для тока на конденсаторе и для производной тока на катушке в преобразованиях Лапласа:

$$I_C = CpU_C(t), \quad pI_L = \frac{U_L(t)}{L}$$

Так как токи на конденсаторе и катушке равны, то

$$pI_C = pI_L, \quad Cp^2U_C(t) = \frac{U_L(t)}{L}$$

Отсюда можно выразить напряжение на катушке через вторую производную от напряжения на конденсаторе

$$U_L(t) = CLp^2U_C(t). \quad (24)$$

Далее нужно выразить напряжение на резисторе через напряжение на конденсаторе. Это можно сделать, учитывая равенство токов:

$$U_R(t) = RCpU_C(t) \quad (25)$$

Подставим выражения (24) и (25) во второе правило Кирхгофа:

$$CLp^2U_C(t) + RCpU_C(t) + U_C(t) = X(t) \quad (26)$$

Заменяя символ  $p$  в соответствующей степени нужной производной, получим ту же модель:

$$CL \frac{d^2U_C(t)}{dt^2} + RC \frac{dU_C}{dt} + U_C(t) = X(t)$$

### 2.3. Задача автоматического подбора параметров цепи

При проектировании различных электро- и радиотехнических устройств важно обеспечить определенные параметры их функционирования, которые соответствовали бы техническому заданию. При наличии математической модели проектируемого устройства одним из способов решения такой задачи может быть автоматический подбор параметров, осуществляемый при помощи компьютера.

Рассмотрим простейший вариант задачи автоматического подбора параметров дифференцирующей RC-цепи, на вход которой подается напряжение  $X(t)$  в виде единичной ступенчатой функции, или функции Хевисайда.

Суть задачи состоит в том, чтобы подобрать с помощью компьютерного моделирования один из параметров цепи ( $R$  или  $C$ ) так, чтобы на выходе цепи за некоторое конечное время  $t_k$  установилось напряжение, равное по величине  $U_{tr}$ .

Первое, что необходимо выяснить при решении этой задачи: изменение какого из параметров оказывает более существенное влияние на выходное напряжение.

В случае RC-цепи (рис. 1) таким параметром оказывается ёмкость конденсатора  $C$ . Чтобы убедиться в этом, достаточно выполнить три компьютерных эксперимента. Первый – выбрав конкретные значения ёмкости конденсатора  $C$ , сопротивления резистора  $R$ , задав единичное входное напряжение и установив некоторое конечное время наблюдения и шаг ведения расчёта. Второй эксперимент проводится при изменении одного единственного параметра, например, при увеличенной в 2 раза ёмкости конденсатора  $C$ . Третий – при увеличенном в 2 раза сопротивлении резистора  $R$ . Во всех трёх случаях фиксируются значения выходного напряжения в конечный

момент времени. Их сравнение позволит понять, какой из параметров схемы наиболее существенно влияет на выходное напряжение, а также понять логику достижения цели.

Теперь обсудим, как составить алгоритм автоматического подбора параметра. Очевидно, что необходимо задать какое-то начальное значение ёмкости. Назовём его  $C_0$ . Также следует задать начальное значение шага  $hC$ , с которым будет подбираться требуемое значение  $C$ .

Также следует задать точность  $\varepsilon$ , с которой мы будем оценивать, достигнуто ли при текущем значении ёмкости  $C$  необходимое значение выходного напряжения  $U_{tr}$  с допустимой погрешностью.

Примерную схему решения задачи смотрите на рисунке 4.

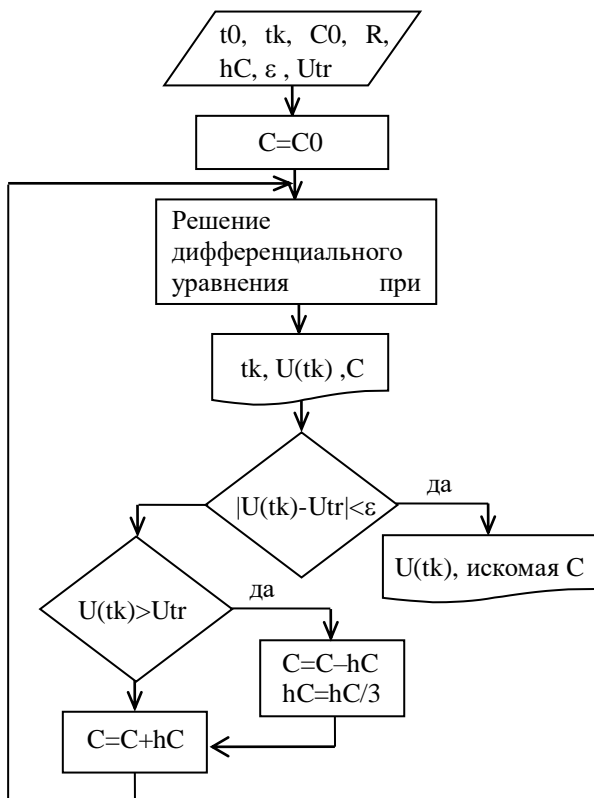


Рис. 4. Схема алгоритма автоподбора ёмкости конденсатора  $C$ .

Ниже смотрите листинг программы, решающей задачу автоподбора емкости конденсатора  $C$  с построением графиков получаемых решений.

```

#define USE MATH DEFINES
#include "graphics.h"
#include <math.h>
#include <stdio.h>
//объявление глобальных переменных
double xmin, xmax, ymin, ymax;
//функция пересчета математической
//координаты x в физическую (пиксель)
double sx(double x)
{
    return(x-xmin)/(xmax-xmin)*getmaxx();
}
//функция пересчета математической
//координаты y в физическую (пиксель)
double sy(double y)
{
    return getmaxy()-(y-ymin)/(ymax-ymin)*getmaxy();
}
//функция, вычисляющая правую часть
//дифференциального уравнения
double f(double x,double y,double R,double C)
{
    return -y/(R*C);
}
//"главная" функция
int main(void)
{
    double x, y, h, y0, k1, k2, k3, k4, z, R, C, a, HC, UTR=0.2;
    double mx[10000], my[10000];
    int n, i;
//ввод начального и конечного времени (xmin, xmax)
//и шага изменения времени h
    printf("Input the range and step:xmin,xmax,h\n");
    scanf("%lf,%lf,%lf", &xmin, &xmax, &h);
//ввод параметров электрической цепи
    printf("Input R,C\n");
    scanf("%lf,%lf", &R, &C);
//ввод начального значения функции y0
    printf("Input beginning function value y0\n");

```

```

scanf ("%lf", &y0);
y0=1;xmin=0;xmax=0.25;h=0.001;R=20000;C=0.000004;
//открытие графического окна
initwindow(400,200);
setcolor(WHITE);
setlinestyle(SOLID_LINE,0,NORM_WIDTH);
//рисование осей: горизонтальной и вертикальной
line(sx(xmin),sy(0),sx(xmax),sy(0));
line(sx(0),sy(ymin),sx(0),sy(1));
//подготовка к рисованию графика функции
setcolor(RED);
setlinestyle(SOLID_LINE,0,THICK_WIDTH);
//значение шага изменения ёмкости
HC=C/3;
//количество точек на интервале
n=(xmax-xmin)/h;
//подготовка к определению минимального
//и максимального значений решения
ymin=y0;
ymax=ymin;
//цикл, который работает, если расчётное
//значение напряжения y по модулю отличается
//от требуемого UTR более, чем на 0.0001
while(fabs(y-UTR)>0.0001)
{
y=y0;
x=xmin;
//цикл решения дифференциального уравнения
//при конкретном текущем значении C
for(i=0;i<=n;i++)
{
mx[i]=x;
my[i]=y;
if(y>ymax)ymax=y;
if(y<ymin)ymin=y;
//метод Рунге-Кутты
k1=h*f(x,y,R,C);
k2=h*f(x+h/2,y+k1/2,R,C);
k3=h*f(x+h/2,y+k2/2,R,C);

```

```

        k4=h*f(x+h, y+k3,R,C);
//значение решения в следующий
//момент времени
        y=y+(k1+2*k2+2*k3+k4)/6;
        x=x+h;
    }
//прорисовываем решение при
//конкретном значении C
x=mx[0];
y=my[0];
moveto(sx(x),sy(y));
    for(i=1;i<=n;i++)
        {
            x=mx[i];
            y=my[i];
            lineto(sx(x),sy(y));
        }
//реализация алгоритма автоподбора
if(y>UTR)
{
    C=C-NC;
    NC=NC/3;
}
C=C+NC;
//распечатка выходного напряжения
//в конечный момент времени и текущего C
printf("t=%lf Uvux=%lf C=%.9lf\n",mx[n],my[n],C);
delay(800);
}
printf("podobrali C=%.9lf\n",C);
getch();
closegraph();
return 0;
}

```

Результаты работы программы представлены в виде двух окон, одно из которых содержит значения выходного напряжения в конечный момент времени при различных значениях емкости конденсатора  $C$  (рис. 5), а другое – соответствующие графики решений (рис. 6).



```

Input the range and step:xmin,xmax,h
0,0.5,0.01
Input R,C
10000,0.000003
Input beginning function value y0
0
t=0.250000 Uvux=0.043937 C=0.000005333
t=0.250000 Uvux=0.095967 C=0.000006667
t=0.250000 Uvux=0.153355 C=0.000008000
t=0.250000 Uvux=0.209611 C=0.000007111
t=0.250000 Uvux=0.172422 C=0.000007556
t=0.250000 Uvux=0.191204 C=0.000008000
t=0.250000 Uvux=0.209611 C=0.000007704
t=0.250000 Uvux=0.197386 C=0.000007852
t=0.250000 Uvux=0.203522 C=0.000007753
t=0.250000 Uvux=0.199436 C=0.000007802
t=0.250000 Uvux=0.201482 C=0.000007770
t=0.250000 Uvux=0.200119 C=0.000007759
t=0.250000 Uvux=0.199664 C=0.000007764
t=0.250000 Uvux=0.199891 C=0.000007770
t=0.250000 Uvux=0.200119 C=0.000007766
t=0.250000 Uvux=0.199967 C=0.000007768
podobrali C=0.000007768

```

Рис. 5. Численные результаты работы программы автоподбора.

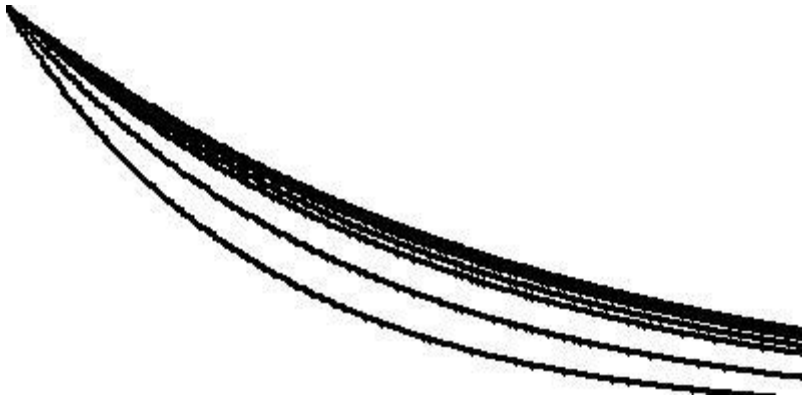


Рис. 6. Графики решений при различных значениях емкости конденсатора C.

## 2.4. Математические модели разветвленных цепей.

### 2.4.1. Математическая модель разветвлённой RC-цепи.

Рассмотрим разветвленную электрическую схему, состоящую из двух резисторов и конденсатора (рис. 7).

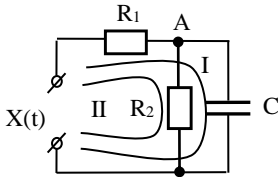


Рис. 7. Пример разветвленной RC - цепи

Получим математическую модель, в которой неизвестной функцией времени будет напряжение на конденсаторе.

Основные физические законы, используемые для вывода модели, это закон Ома (1), а также связь между зарядом конденсатора и напряжением на нём (2).

Запишем для контуров I и II второй закон Кирхгофа:

$$U_{R1}(t) + U_C(t) = X(t) \quad (27)$$

$$U_{R1}(t) + U_{R2}(t) = X(t)$$

Сравнивая полученные выражения, находим подтверждение того факта, что напряжения на параллельно соединенных элементах цепи равны:

$$U_{R2}(t) = U_C(t) \quad (28)$$

Задача получения математической модели сводится к исключению из выражения (27) неизвестной функции  $U_{R1}(t)$ . Для этого запишем первый закон Кирхгофа относительно узла A:

$$I_{R1}(t) = I_{R2}(t) + I_C(t) \quad (29)$$

Выразим токи в выражении (29) через напряжения

$$\frac{U_{R1}(t)}{R1} = \frac{U_{R2}(t)}{R2} + C \frac{dU_C(t)}{dt} \quad (30)$$

Получим отсюда  $U_{R1}(t)$  с учетом (28):

$$U_{R1}(t) = \frac{R1}{R2} U_C(t) + CR1 \frac{dU_C(t)}{dt} \quad (31)$$

Теперь подставим выражение (31) в выражение (27) и получим математическую модель разветвленной цепи (рис. 7):

$$CR1 \frac{dU_C(t)}{dt} + \left( \frac{R1}{R2} + 1 \right) U_C(t) = X(t) \quad (32)$$

## 2.5. Математические модели электрических цепей с нелинейностями. Однополупериодный выпрямитель

Математические модели рассмотренных ранее электрических цепей представляют собой линейные дифференциальные уравнения. Их решение может легко быть получено не только численными методами, но и аналитически. Однако огромное количество динамических систем являются по природе своей нелинейными и описываются нелинейными дифференциальными уравнениями. Нелинейности часто представляют собой кусочно-линейные или разрывные характеристики, которые не могут быть представлены в виде одного аналитического выражения.

В качестве конкретного примера рассмотрим описание процессов в схеме однополупериодного выпрямителя, нагруженного активным сопротивлением (рис. 8).

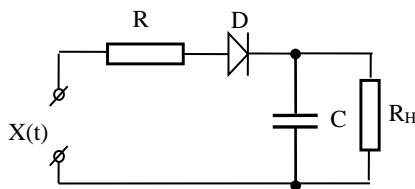


Рис. 8. Однополупериодный выпрямитель.

Нелинейная характеристика диода в упрощенном, идеализированном, виде может быть описана с помощью единичной функции Хевисайда:

$$i_D = f(U_D) = \frac{U_D(t)}{R_D} \cdot I(U_D), \quad \text{где } I(U_D) = \begin{cases} 0, & \text{если } U_D < 0 \\ 1, & \text{если } U_D \geq 0 \end{cases} \quad (33)$$

Соотношения, справедливые для линейных элементов нам известны:

$$i_R = \frac{U_R}{R}, \quad i_C = C \cdot \frac{dU_C(t)}{dt}, \quad i_H = \frac{U_{R_H}}{R_H} \quad (34)$$

Согласно второму закону Кирхгофа получим соотношение для напряжений:

$$U_R(t) + U_D(t) + U_{R_H}(t) = X(t); \quad U_{R_H}(t) = X(t) - U_R(t) - U_D(t) \quad (35)$$

Согласно первому закону Кирхгофа получим соотношение для токов:

$$i_D = i_C + i_{R_H} \quad (36)$$

Получим математическую модель, описывающую процессы в выпрямителе и выражающую зависимость выходного напряжения  $U_{R_H}(t)$  от входного  $X(t)$ .

В силу того, что резистор и диод соединены последовательно,

$$I_R(t) = I_D(t) \quad \text{и} \quad f(U_D) = \frac{U_R(t)}{R} \quad (37)$$

Так как конденсатор подключен параллельно нагрузке

$$U_C = U_{R_H} \quad (38)$$

Заменим токи в первом законе Кирхгофа (36) на соответствующие напряжения и их производные с учетом (34) и (38):

$$f(U_D) = C \cdot \frac{dU_{R_H}}{dt} + \frac{U_{R_H}}{R_H} \quad (39)$$

Нашей целью является исключение из второго закона Кирхгофа неизвестных функций  $U_R(t)$  и  $U_D(t)$ . Но в данном случае сделать это затруднительно, так как  $U_D(t)$  является аргументом нелинейной характеристики диода (33). Поэтому в математической модели в качестве неизвестной функции будем рассматривать  $U_D(t)$ .

Рассчитав из математической модели  $U_D(t)$  в каждый момент времени и подставив его в выражение (35), мы найдем выходное напряжение  $U_{R_H}(t)$ .

Итак, получим зависимость между входным напряжением и напряжением на диоде. Продифференцируем выражение (35):

$$\frac{dU_{R_H}(t)}{dt} = \frac{dX(t)}{dt} - \frac{dU_R(t)}{dt} - \frac{dU_D(t)}{dt} \quad (40)$$

Перепишем (40) с учетом (33) и (37):

$$\frac{dU_{R_H}(t)}{dt} = \frac{dX(t)}{dt} - \left( 1 + \frac{R}{R_D} \cdot \mathbf{1}(U_D) \right) \cdot \frac{dU_D(t)}{dt} \quad (41)$$

Подставим в уравнение (39) функцию  $U_{R_H}(t)$  и ее производную из выражений (35) и (41):

$$f(U_D) = C \left( \frac{dX}{dt} - \left( 1 + \frac{R}{R_D} \mathbf{1}(U_D) \right) \frac{dU_D}{dt} \right) + \frac{1}{R_H} (X - Rf(U_D) - U_D) \quad (42)$$

Разрешим полученное уравнение относительно производной искомой функции  $U_D(t)$ , то есть представим его в форме Коши, а также подставим в получаемое выражение нелинейную характеристику диода  $f(U_D)$  из (33):

$$\frac{dU_D}{dt} = \frac{\frac{dX}{dt} + \frac{X - U_D}{R_H \cdot C} - \frac{U_D}{R_D \cdot C} \cdot \left( 1 + \frac{R}{R_D} \right) \cdot \mathbf{1}(U_D)}{1 + \frac{R}{R_D} \cdot \mathbf{1}(U_D)} \quad (43)$$

На вход выпрямителя обычно подается синусоидальное напряжение:

$$X(t) = A \cdot \sin(\omega \cdot t).$$

Выходное напряжение, получаемое на нагрузке, зависит от параметров входного напряжения, параметров элементов схемы самого выпрямителя и нагрузки и может иметь вид, представленный на рис. 9.

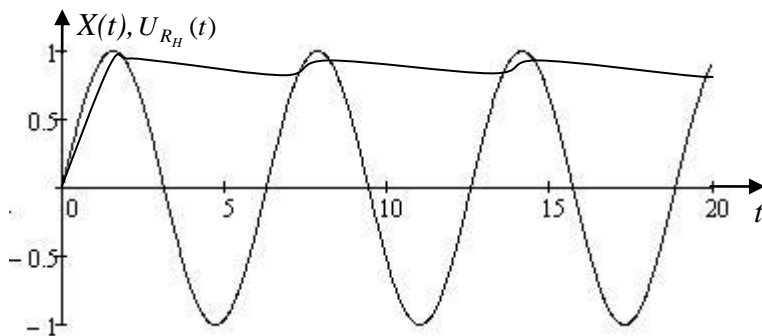


Рис. 9. Напряжение на выходе однополупериодного выпрямителя.

Время интегрирования следует выбирать достаточным для наблюдения не только переходного процесса, но и установившегося режима.

## 2.6. Моделирование электрических цепей в пакете MathCAD 14

В пакете MathCAD существуют несколько способов решения дифференциальных уравнений, являющихся математическими моделями динамических систем. Рассмотрим некоторые из способов.

В качестве **первого примера** рассмотрим решение в MathCADe математической модели (8) цепи, рассмотренной в п. 2.1.1.

В примере использован блок решения Given для формулировки задачи – задания дифференциального уравнения и начальных условий в момент времени  $t_0=0$ , а также решающая функция Odesolve. Параметры цепи следует задавать до блока решения, так как использование в нем команды Definition (обозначаемой знаком :=) запрещено.

$R := 20000$      $C := 0.000001$      $X(t) := 1$

Given

$$\frac{d}{dt}U_c(t) = \frac{X(t) - U_c(t)}{R \cdot C}$$

$$U_c(0) = 0$$

$U_c := \text{Odesolve}(t, 0.2)$

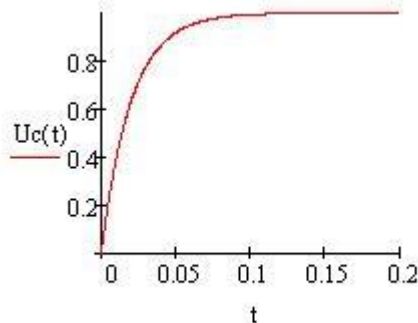


Рис. 10. График зависимости напряжения на конденсаторе от времени.

**Замечания.** Для ввода производной следует использовать шаблон с панели инструментов Calculus. Первый параметр функции Odesolve – независимая переменная (время), второй – конечное время наблюдения.

**Второй пример** – решение математической модели (8) цепи, рассмотренной в п. 2.1.1. с помощью функции rkfixed (метод Рунге-Кутты с фиксированным шагом).

Параметры цепи:  $R := 20000$      $C := 0.000001$

Входной сигнал:  $X(t) := 1$  Конечное время  $t_k := 0.2$

Количество точек решения на интервале  $[t_0, t_k]$   $N := 1000$

Необходимо решить дифференциальное уравнение:

$$\frac{d}{dt}U_c(t) = f(t, U_c) \quad , \text{ где}$$

$$f(t, U_c) := \frac{X(t) - U_c}{R \cdot C}$$

Начальные условия:  $U_{c0} := 0 \quad t0 := 0$

Для получения решения нужно задать вектор начальных условий:

$$nu_0 := U_{c0}$$

Задаем функцию, равную правой части дифференциального уравнения, второй аргумент которой является вектором искомых значений функции  $U_c$ :

$$D(t, U_c) := f(t, U_{c0})$$

В качестве решения получаем матрицу  $U$ :

$$U := rkfixed(nu, t0, tk, N, D)$$

$T := U^{(0)}$  - вектор - столбец значений времени

$U_c := U^{(1)}$  - вектор - столбец значений напряжения на конденсаторе

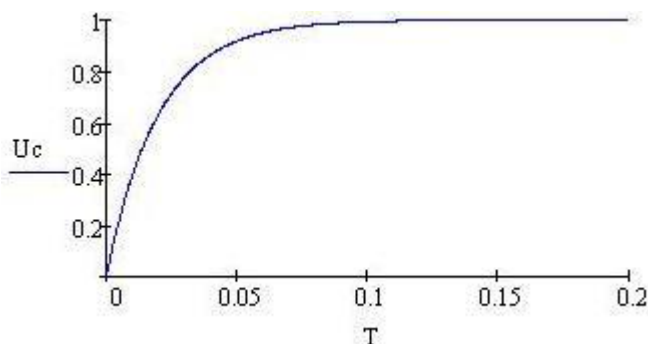


Рис. 11. График зависимости напряжения на конденсаторе от времени, полученный с помощью решающей функции `rkfixed`.



**Замечание.** Эту задачу можно решить в MathCADe и с помощью других функций, параметры которых аналогичны параметрам функции `rkfixed`. Такой функцией, например, является функция `Rkadapt`, реализующая метод Рунге-Кутты с адаптивным подбором шага. Эта функция позволяет получать решение с мелким шагом в областях быстрого изменения искомой функции и с более крупным шагом там, где решение меняется медленно, что и повышает точность, и сокращает время, требуемое для решения уравнения.

Также может быть применена функция, решающая дифференциальные уравнения методом Адамса (Adams), или функция `Radau`, реализующая метод Радая 5. В случаях, когда известно, что решение является гладкой функцией, имеет смысл использовать функцию `Vulstoer`, решающую дифференциальные уравнения методом Булирша-Штёра.

**Третий пример** – решение математической модели колебательного контура из п. 2.2.1.

В решении также использована функция `Odesolve`. Так как модель представляет собой дифференциальное уравнение второго порядка, в блоке `Given` задано два начальных условия – значение искомой функции, а также её производной в момент времени  $t_0=0$ .

$$\underline{R} := 500000 \quad \underline{C} := 0.000001 \quad \underline{L} := 0.004 \quad X(t) := \sin(2 \cdot t)$$

Given

$$\frac{L}{R} \frac{d^2}{dt^2} U_R(t) + \frac{d}{dt} U_R(t) + \frac{1}{R \cdot C} U_R(t) = \frac{d}{dt} X(t)$$

$$U_R(0) = 0$$

$$U_R'(0) = 0$$

$$U_R := \text{Odesolve}(t, 20)$$

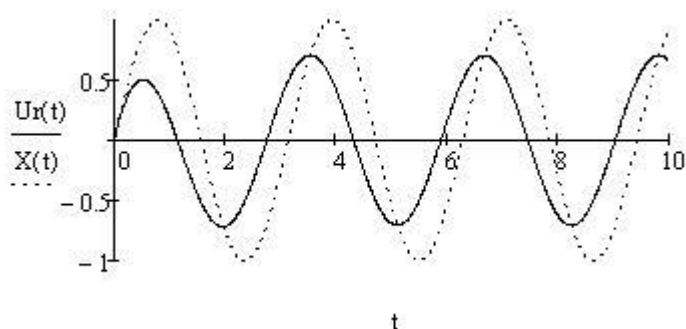


Рис. 12. Графики входного  $X(t)$  и выходного  $U_R(t)$  напряжений на резисторе RLC-цепи (колебательного контура).

**Четвертый пример** – решение этой же задачи в параметрическом виде. Здесь решение запрашивается в виде параметрической функции  $fU_R(R)$ , зависящей не только от времени, но и от одного из параметров самой цепи, в данном примере – от величины сопротивления резистора  $R$ .

После получения параметрического решения в общем виде с помощью решающей функции `Odesolve`, находятся решения при значениях  $R=500$  кОм и  $R=50$  кОм. Эти решения записываются в новые функции, имена которых  $U_{R500000}$  и  $U_{R50000}$ . Графики этих функций изображены на рисунке 13.

$R := 500000$        $C := 0.000001$        $L := 0.004$        $X(t) := \sin(2 \cdot t)$

$t0 := 0$

Given

$$\frac{L}{R} \cdot \frac{d^2}{dt^2} U_R(t) + \frac{d}{dt} U_R(t) + \frac{1}{R \cdot C} \cdot U_R(t) = \frac{d}{dt} X(t)$$

$$U_R(t0) = 0$$

$$U_R'(t0) = 0$$

$fU_R(R) := \text{Odesolve}(t, 40)$

$U_{R500000} := fU_R(500000)$        $U_{R50000} := fU_R(50000)$

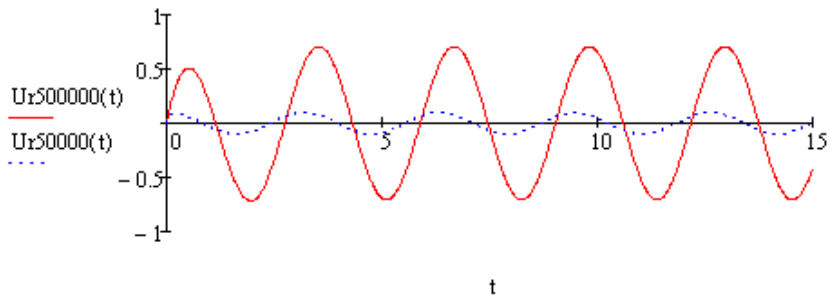


Рис. 13. Графики выходного напряжения на резисторе RLC-цепи при разных значениях сопротивления  $R$ .

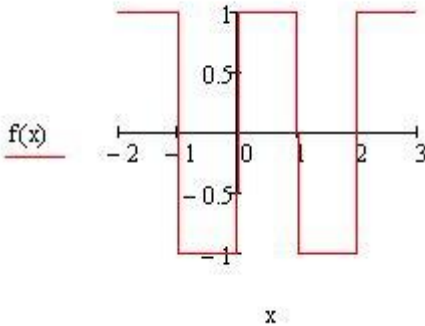
## 2.7. Моделирование электрических сигналов в пакете MathCAD 14.

Сигналы, подаваемые на вход электрической цепи, могут быть весьма разнообразными. Для компьютерного моделирования этих сигналов могут использоваться комбинации стандартных математических функций пакета MathCAD, а также известные математические методы, например, разложение произвольной математической функции в ряд Фурье.

Рассмотрим несколько примеров моделирования.

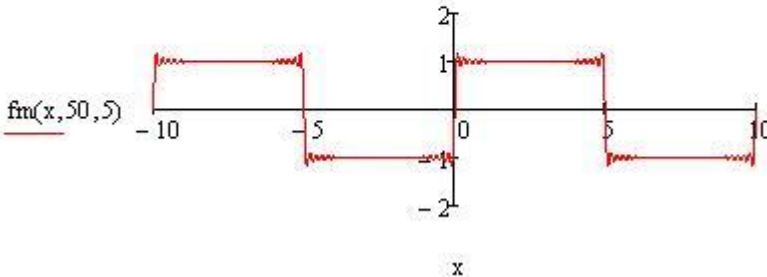
**Пример 1.** Пусть требуется смоделировать сигнал «меандр». Это можно сделать, используя функцию MathCAD  $\text{sign}(x)$ , которая принимает значение 1, если аргумент  $x > 0$ , значение -1, если  $x \leq 0$  и равна 0 при  $x = 0$ .

$$f(x) := \text{sign}(\sin(x \cdot \pi))$$



Другой вариант построения меандра – с помощью ряда Фурье:

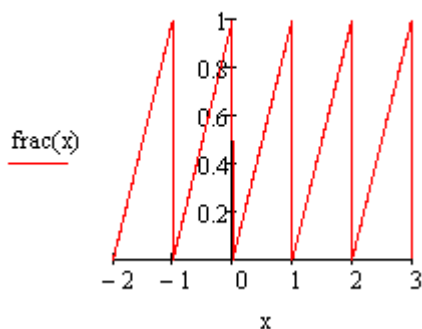
$$f_m(x, N, T) := \frac{4}{\pi} \cdot \left[ \sum_{n=1}^N \left( \frac{1}{n} \cdot \sin\left(\frac{n \cdot \pi \cdot x}{T}\right) \cdot \text{mod}(n, 2) \right) \right]$$



Здесь применена функция  $\text{mod}(n, 2)$ , которая возвращает в качестве результата остаток от деления  $n$  на 2, то есть равна 0 для четных и 1 для нечетных значений  $n$ .

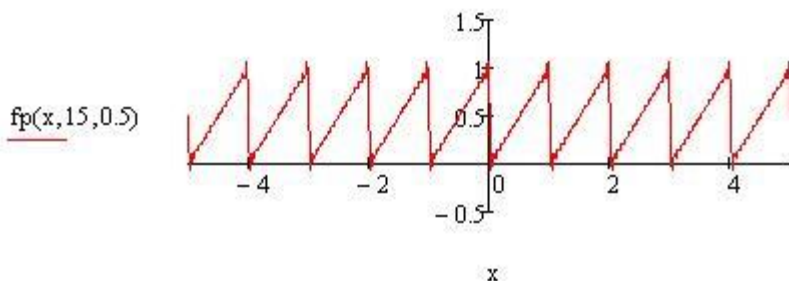
**Пример 2.** Пусть требуется смоделировать пилообразный сигнал. Для его получения используется функция  $\text{floor}(x)$ , результат вычисления которой – наибольшее целое число, меньшее либо равное  $x$ .

$$\text{frac}(x) := x - \text{floor}(x)$$



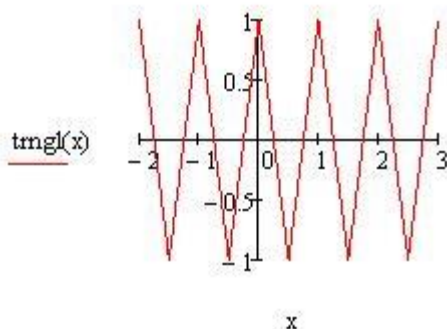
Аналогичный сигнал, смоделированный с помощью ряда Фурье:

$$\text{fp}(x, N, T) := \frac{1}{2} - \frac{1}{\pi} \left[ \sum_{n=1}^N \left( \frac{1}{n} \cdot \sin\left(\frac{n \cdot \pi \cdot x}{T}\right) \right) \right]$$



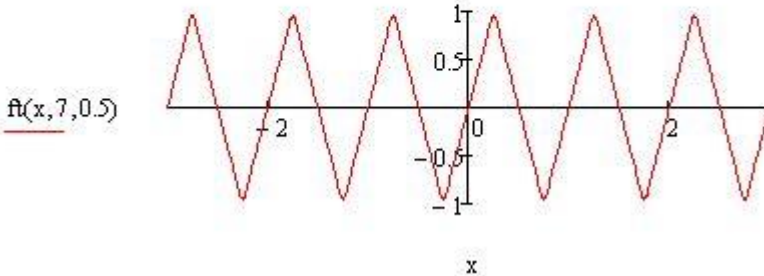
**Пример 3.** Смоделируем треугольный сигнал, применив функцию  $\text{frac}(x)$ , заданную в примере 2.

$$\text{trngl}(x) := |\text{frac}(x) \cdot 2 - 1| \cdot 2 - 1$$



Другой треугольный сигнал, смоделированный с помощью ряда Фурье:

$$f(x, N, L) := \frac{8}{\pi^2} \cdot \left[ \sum_{n=1}^N \left[ \frac{(-1)^{\frac{n-1}{2}}}{n^2} \cdot \sin\left(\frac{n \cdot \pi \cdot x}{L}\right) \cdot \text{mod}(n, 2) \right] \right]$$



**Пример 4.** Пусть требуется смоделировать произвольный электрический сигнал, вид которого на интервале  $[0, T]$  задан кусочно-линейной функцией  $f(t, T)$ .

Пусть период  $T := 1$

Определим кусочно-линейную функцию следующим образом:

$$f(t, T) := \begin{cases} t & \text{if } 0 \leq t \leq 0.4T \\ 0.4 & \text{if } 0.4T \leq t \leq 0.6T \\ -t + 1 & \text{if } 0.6T \leq t \leq T \end{cases}$$

Пусть  $w := 2 \cdot \frac{\pi}{T}$

Вычислим коэффициенты ряда Фурье:

$$a_0 := \frac{\int_0^T f(t, T) dt}{0.5T}$$

$$A(k, f, T) := \frac{\int_0^T f(t, T) \cdot \cos(k \cdot w \cdot t) dt}{0.5 \cdot T}$$

$$B(k, f, T) := \frac{\int_0^T f(t, T) \cdot \sin(k \cdot w \cdot t) dt}{0.5 \cdot T}$$

Зададим параметр  $Nf$  - количество гармонических членов ряда, сумма которых достаточно хорошо приближается к функции  $f(t, T)$ . Определим переменную-диапазон (или ранжированную переменную)  $k$  и вычислим гармонические члены ряда при различных значениях  $k$ :

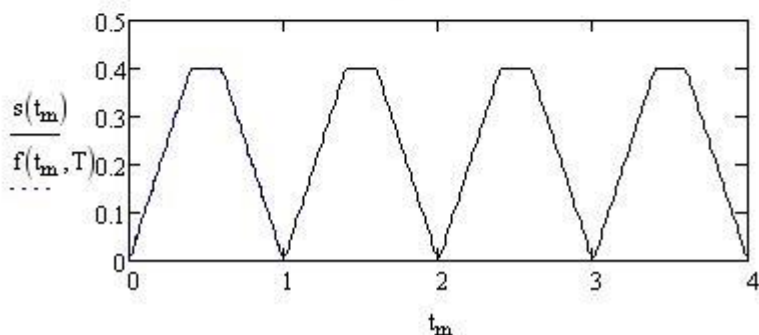
$$Nf := 10 \quad k := 1..Nf \quad a_k := A(k, f, T) \quad b_k := B(k, f, T) \quad +$$

Используя ранее вычисленный член ряда  $a_0$  сформируем новую функцию  $s(t)$  по формуле ряда Фурье:

$$s(t) := \frac{a_0}{2} + \left[ \sum_k (a_k \cdot \cos(k \cdot w \cdot t) + b_k \cdot \sin(k \cdot w \cdot t)) \right]$$

Построим новую функцию  $s(t)$  на интервале  $[0, 4]$  с шагом  $0,01$ :

$$Np := 100 \quad m := 0..4 \cdot Np \quad t_m := \frac{T \cdot m}{Np}$$



Видно, что графики практически полностью совпадают.

**Пример 5.** Рассмотрим еще одну периодическую кусочно-линейную функцию, определенную на интервале  $[-T, T]$  период которой равен  $2T$ .

В данном примере для вычисления коэффициентов ряда Фурье используется функция  $FurCoeff(f, N, T)$ , которая вычисляет значения коэффициентов при помощи программы, причем записывает их в

матрицу, количество столбцов которой равно  $N+1$ . Первый (точнее нулевой) столбец заполняется в самом начале, а затем в цикле подсчитываются остальные столбцы. В самом конце функции выполняется операция транспонирования матрицы.

Параметр  $N$  определяет количество членов ряда Фурье, достаточное для хорошего приближения к функции  $f(x, T)$ .

Определение кусочно-линейной периодической функции:

$$f(x, T) := \begin{cases} 1 & \text{if } 0 \leq x \leq T \\ -x & \text{if } -T \leq x < 0 \\ f(x - 2, T) & \text{if } x > T \end{cases}$$

Пусть правая граница интервала  $T := 1$

Программа для вычисления коэффициентов ряда Фурье:

$$\text{FurCoeff}(f, N, T) := \begin{array}{l} M^{(0)} \leftarrow \begin{pmatrix} \frac{1}{2 \cdot T} \int_{-T}^T f(x, T) dx \\ 0 \end{pmatrix} \\ \text{for } n \in 1..N \\ M^{(n)} \leftarrow \begin{pmatrix} \frac{1}{T} \int_{-T}^T f(x, T) \cdot \cos\left(\frac{n \cdot \pi \cdot x}{T}\right) dx \\ \frac{1}{T} \int_{-T}^T f(x, T) \cdot \sin\left(\frac{n \cdot \pi \cdot x}{T}\right) dx \end{pmatrix} \\ M^T \end{array}$$

Порядок аппроксимации, обеспечивающий хорошее приближение к функции  $f(x, T)$ :

$Nt := 20$

Вычисляем матрицу коэффициентов Фурье:  $\text{res} := \text{FurCoeff}(f, Nt, T)$

Выделяем вектор-столбцы коэффициентов  $A$  и  $B$ :

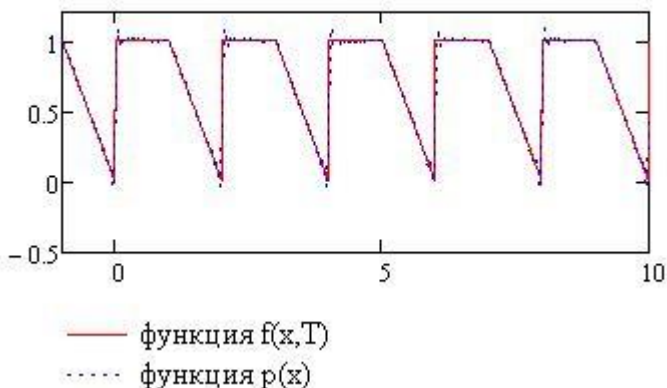
$A := \text{res}^{(0)}$        $B := \text{res}^{(1)}$



Определим функцию  $p(x)$  - ряд Фурье:

$$p(x) := A_0 + \sum_{n=1}^{Nt} \left( A_n \cdot \cos\left(\frac{n \cdot \pi \cdot x}{T}\right) + B_n \cdot \sin\left(\frac{n \cdot \pi \cdot x}{T}\right) \right)$$

Построим графики функций  $p(x)$  и  $f(x, T)$ :



Видно, что ряд Фурье при  $Nt=20$  обеспечивает достаточно хорошее приближение к функции  $f(x, T)$ .

## Задания для лабораторных работ

### Лабораторная работа 1

Задание на лабораторную работу:

- 1) Используя математическую модель интегрирующей RC-цепи, решить задачу автоматического подбора параметра. Величина выходного напряжения, снимаемого на конденсаторе к моменту времени  $t_k$  после подачи на вход прямоугольного единичного сигнала (рис. 2) должна достигнуть значения  $U_{тр}$ . Подобрать такое значение сопротивления резистора  $R$  при определенной заданной ёмкости конденсатора, которое обеспечило бы решение

поставленного технического задания. Значения конечного времени  $t_k$  и ёмкости конденсатора  $C$  взять из таблицы по указанию преподавателя. Обратит внимание на табличные единицы измерения и учесть, что в программах следует задавать ёмкость в фарадах.

Таблица 1. Значения  $t_k$ , ёмкости  $C$  и требуемого выходного напряжения.

Вариант	$t_k$	$C$ , мкФ	$U_{тр}$ , В
1	0.8	0.8	0.7
2	0.82	0.85	0.71
3	0.84	0.9	0.72
4	0.86	0.95	0.73
5	0.88	1.00	0.74
6	0.9	1.05	0.75
7	0.92	1.1	0.76
8	0.94	1.15	0.77
9	0.96	1.2	0.78
10	0.98	1.25	0.79
11	1.00	1.3	0.8
12	1.02	1.35	0.81

- 2) Используя математическую модель дифференцирующей RC-цепи, решить задачу автоматического подбора параметра. Величина выходного напряжения, снимаемого на резисторе к моменту времени  $t_k$  после подачи на вход прямоугольного единичного сигнала (рис. 2) должна достигнуть значения  $U_{тр}$ . Подобрать такое значение ёмкости конденсатора  $C$  при определенном заданном сопротивлении резистора, которое обеспечило бы решение поставленного технического задания. Значения конечного времени  $t_k$  и

сопротивления резистора  $R$  взять из таблицы по указанию преподавателя. Обратите внимание на табличные единицы измерения и учесть, что в программах следует задавать сопротивление в омах.

Таблица 2. Значения  $t_k$ , сопротивления  $R$  и выходного напряжения  $U_{Tr}$ .

Вариант	$t_k$	$R$ (МОм)	$U_{Tr}$ (В)
1	0.2	0.81	0.3
2	0.25	0.84	0.28
3	0.3	0.87	0.26
4	0.35	0.9	0.24
5	0.4	0.93	0.22
6	0.45	0.96	0.2
7	0.5	1.00	0.18
8	0.55	1.13	0.16
9	0.6	1.2	0.14
10	0.65	1.23	0.12
11	0.7	1.27	0.1
12	0.75	1.3	0.08

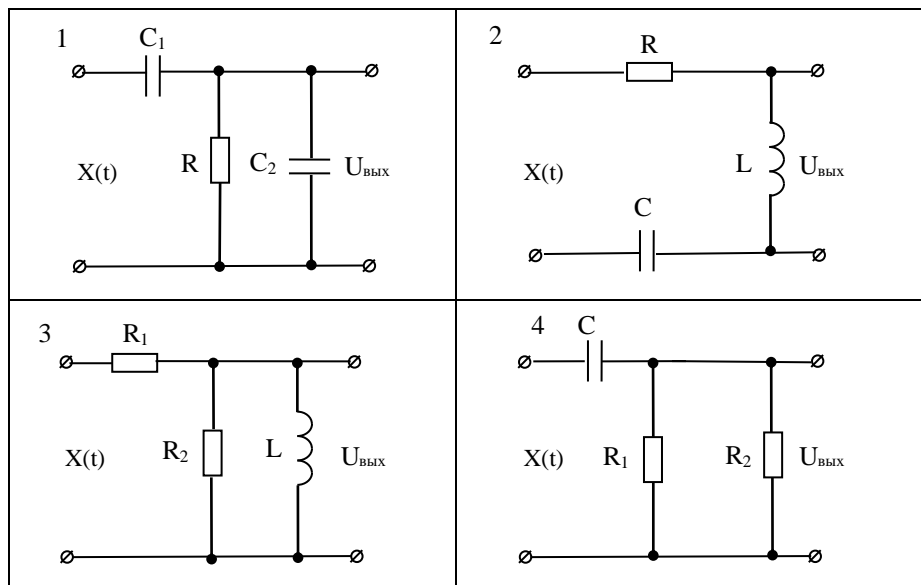
3) В качестве отчета по лабораторной работе представить преподавателю программу на языке Си, строящую графики выходного напряжения и выводящую на экран значения выходного напряжения в конечный момент времени при различных значениях  $R$  для первого и  $C$  для второго заданий, соответственно. Программа также должна выводить на экран автоматически подобранные значения сопротивления резистора  $R$  для первой задачи и ёмкости конденсатора  $C$  для второй.

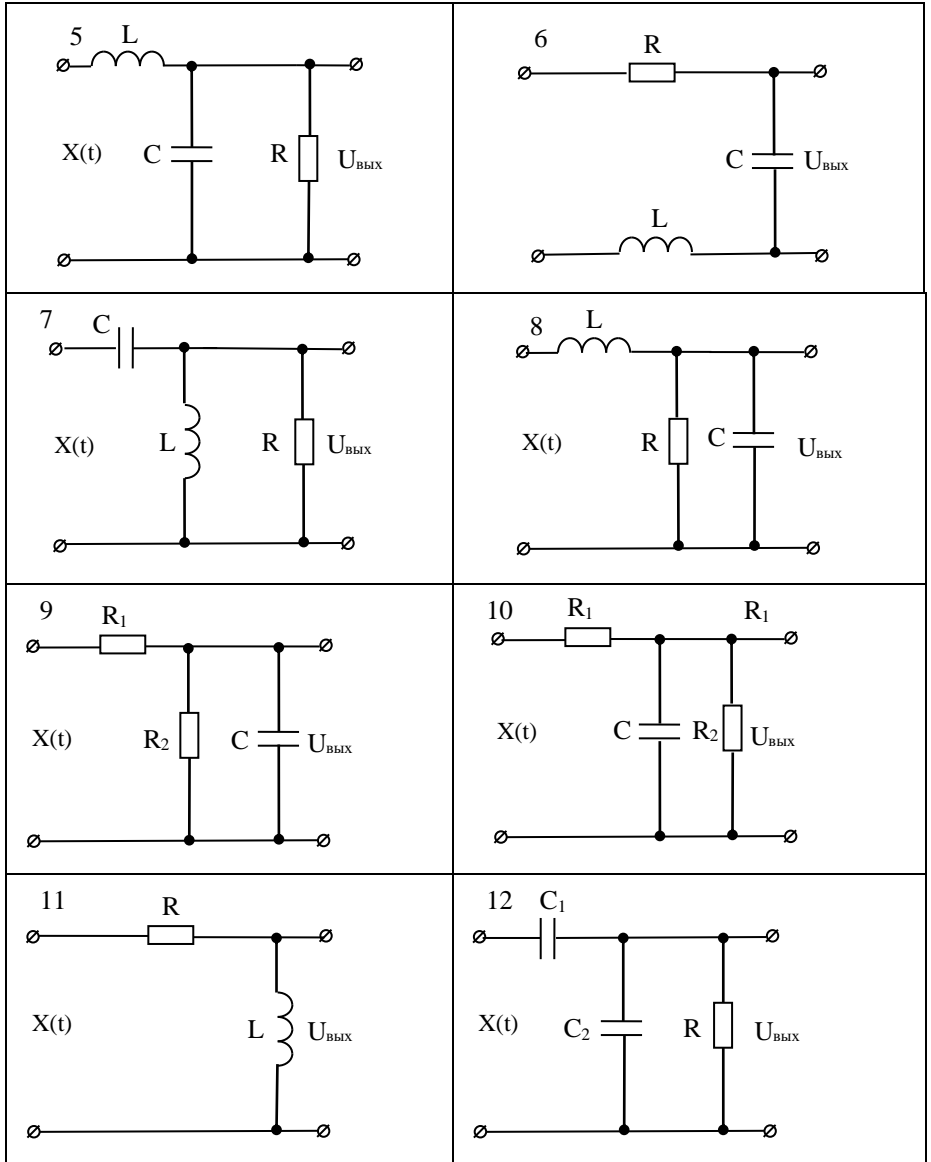
## Лабораторная работа 2

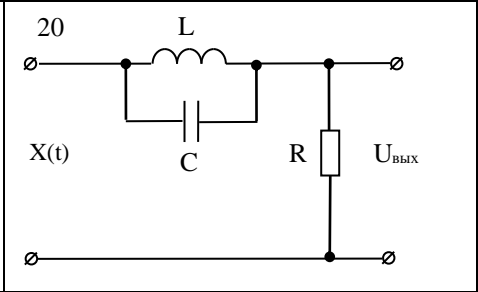
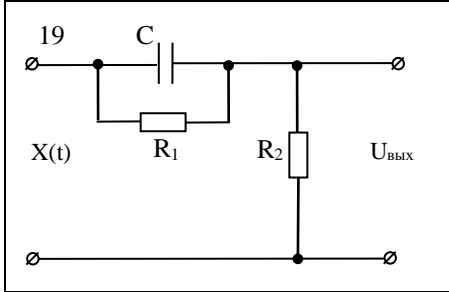
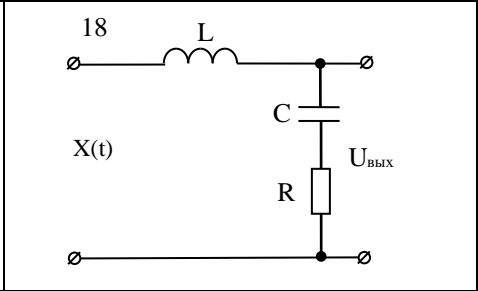
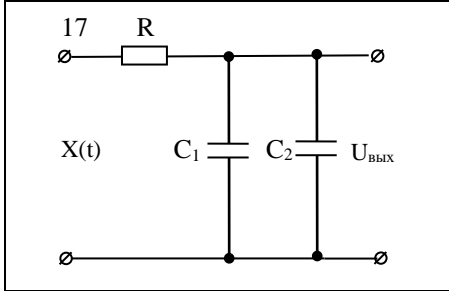
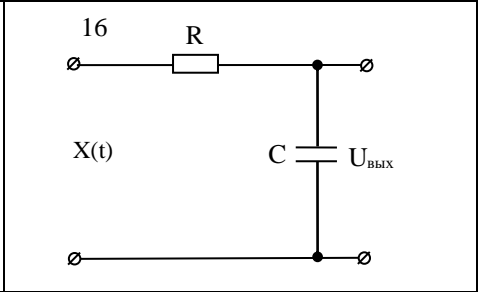
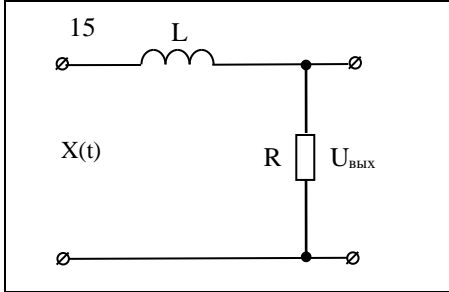
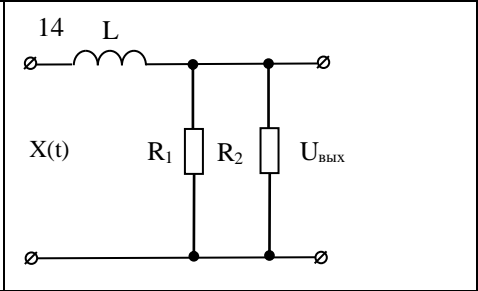
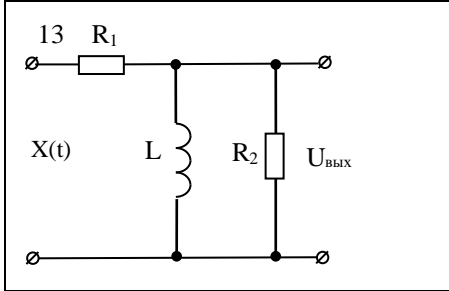
Задание на лабораторную работу:

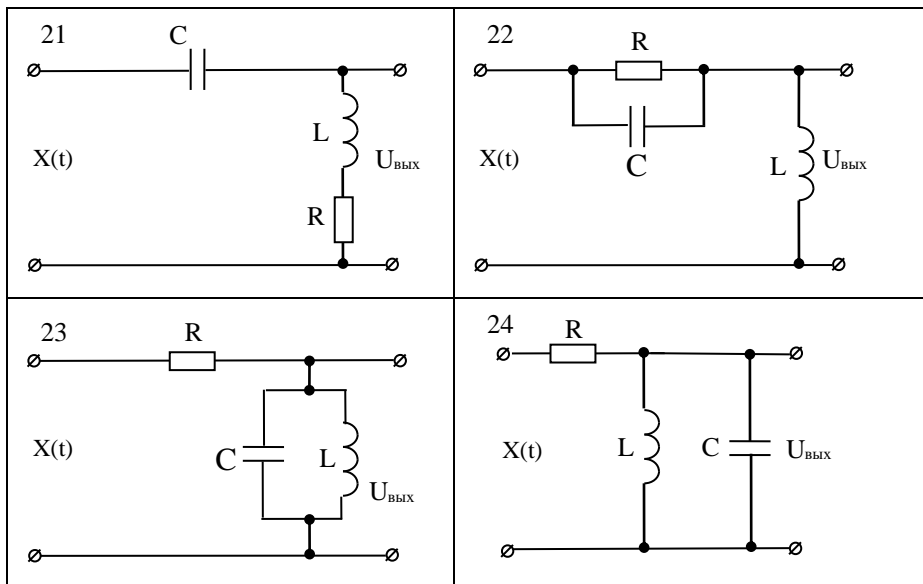
- 1) составить математическую модель динамической системы в виде дифференциального уравнения с неизвестной функцией  $U_{\text{ВЫХ}}$  по варианту из таблицы 3, выданному преподавателем. Учесть, что  $X(t)$  – функция, задающая входное напряжение цепи. Вид входного напряжения также следует уточнить у преподавателя;
- 2) решить полученное дифференциальное уравнение в пакете MathCAD или в программе на языке Си;
- 3) исследовать зависимость выходного напряжения  $U_{\text{ВЫХ}}$  от параметров цепи.
- 4) Представить преподавателю отчет, содержащий вывод уравнения, программу на языке Си или решение, полученное в пакете MathCAD.
- 5) Защитить работу.

Таблица 3. Варианты электрических схем.







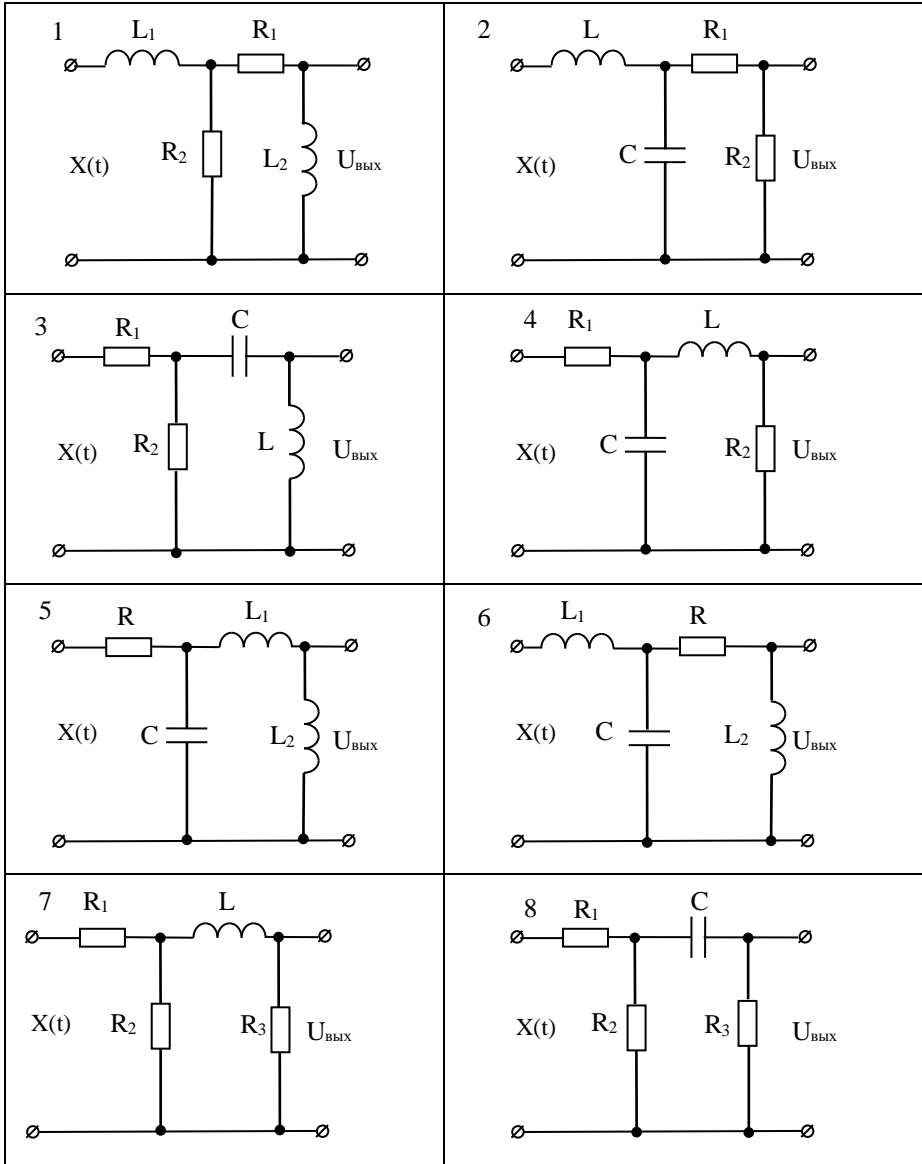


### Лабораторная работа 3

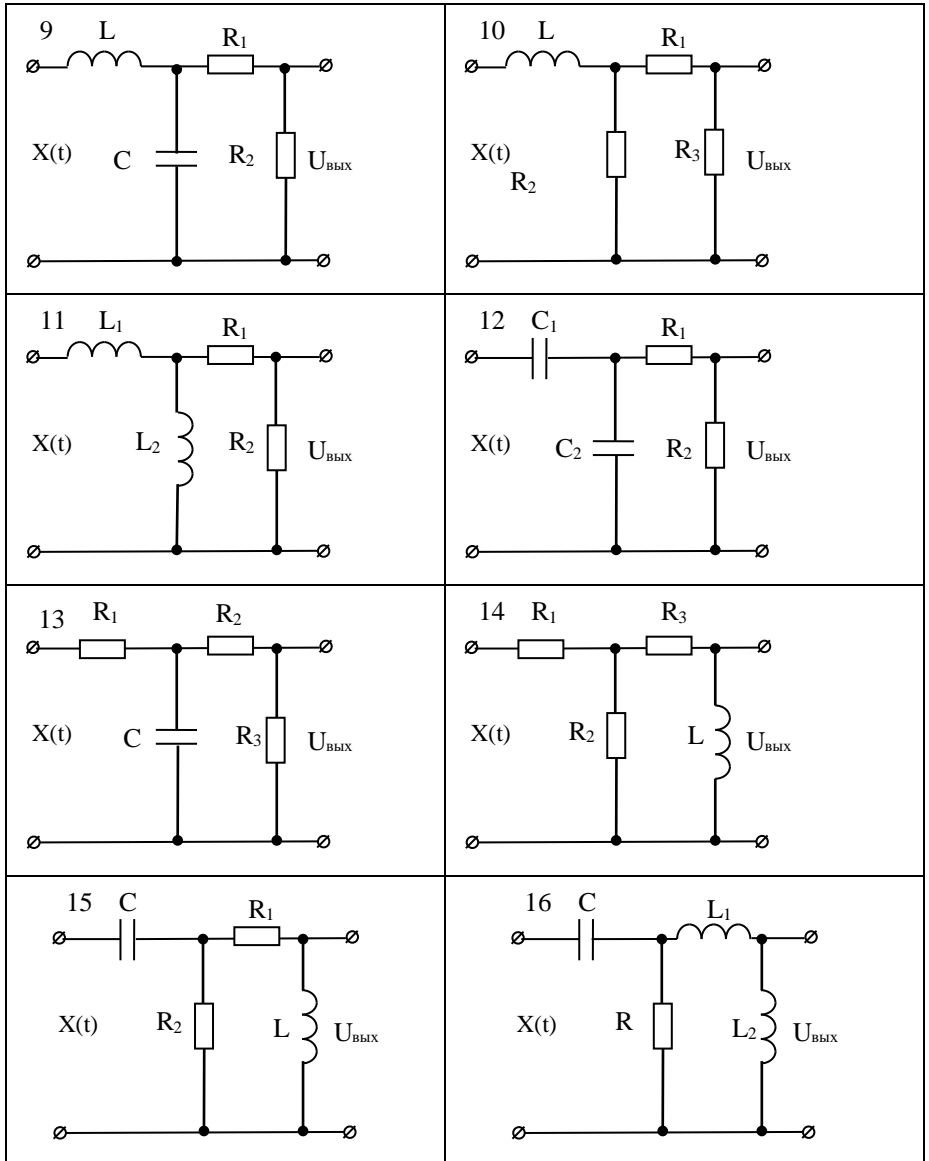
Задание на лабораторную работу:

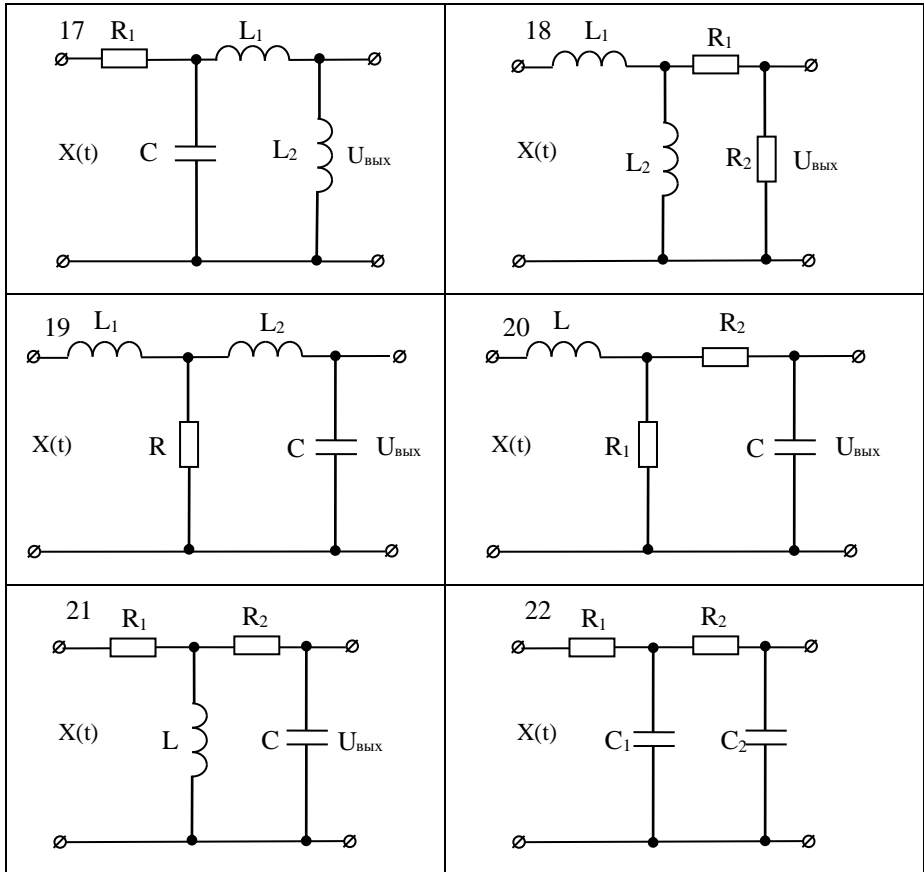
- 1) составить математическую модель динамической системы в виде дифференциального уравнения с неизвестной функцией  $U_{\text{ВЫХ}}$  по варианту из таблицы 4, выданному преподавателем. Учесть, что  $X(t)$  – функция, задающая входное напряжение цепи. Вид входного напряжения также следует уточнить у преподавателя;
- 2) решить полученное дифференциальное уравнение в пакете MathCAD или в программе на языке Си;
- 3) исследовать зависимость выходного напряжения  $U_{\text{ВЫХ}}$  от параметров цепи.
- 4) Представить преподавателю отчет, содержащий вывод уравнения, программу на языке Си или решение, полученное в пакете MathCAD.
- 5) Защитить работу.

Таблица 4. Варианты электрических схем.









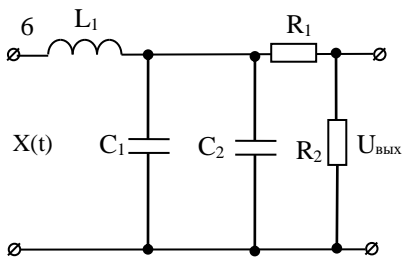
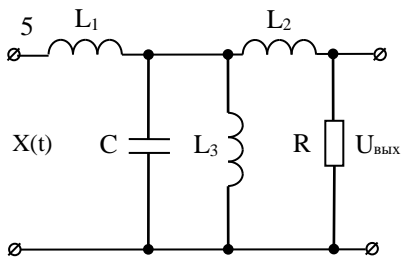
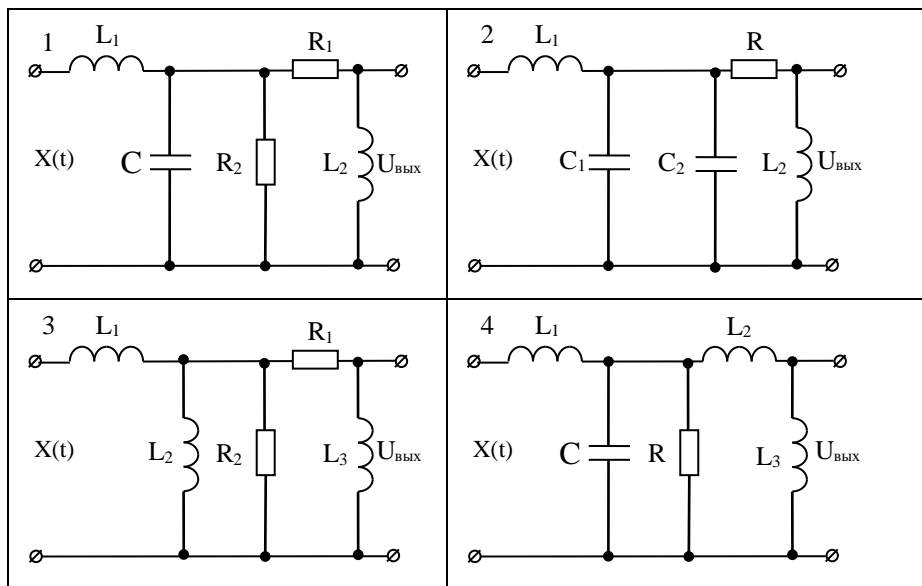
### Лабораторная работа 4

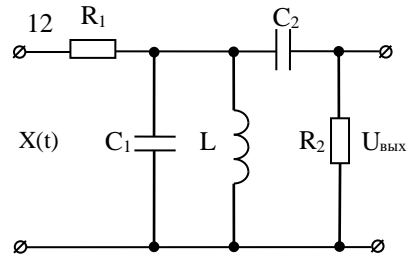
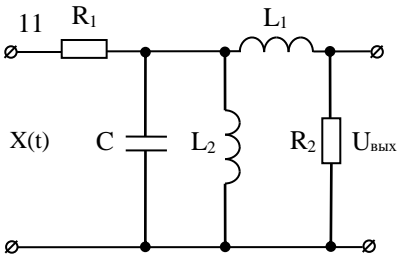
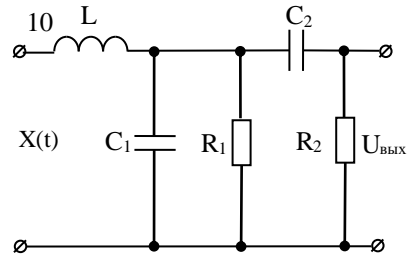
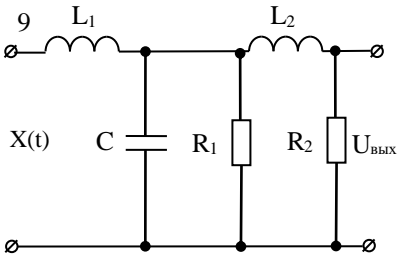
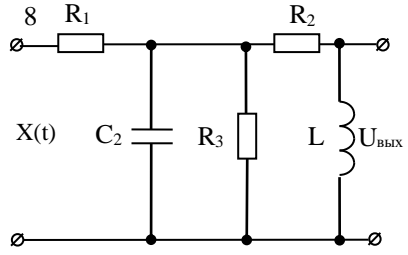
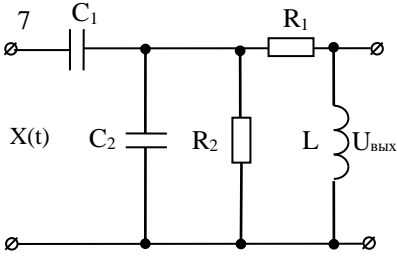
Задание на лабораторную работу:

- 1) составить математическую модель динамической системы в виде дифференциального уравнения с неизвестной функцией  $U_{\text{ВЫХ}}$  по варианту из таблицы 5, выданному преподавателем. Учесть, что  $X(t)$  – функция, задающая входное напряжение цепи. Вид входного напряжения также следует уточнить у преподавателя;

- 2) решить полученное дифференциальное уравнение в пакете MathCAD или в программе на языке Си;
- 3) исследовать зависимость выходного напряжения  $U_{\text{ВЫХ}}$  от параметров цепи.
- 4) Представить преподавателю отчет, содержащий вывод уравнения, программу на языке Си или решение, полученное в пакете MathCAD.
- 5) Защитить работу.

Таблица 5. Варианты электрических схем.





## Оглавление

Введение .....	3
1.Компьютерное моделирование .....	4
2.Получение математических моделей электротехнических цепей .....	6
2.1. Математические модели неразветвленных цепей.....	6
2.1.1. Математическая модель RC-цепи. Пример 1 .....	6
2.1.2. Математическая модель RC-цепи. Пример 2 .....	8
2.1.3. Математическая модель RC-цепи. Пример 3 .....	9
2.2.1. RLC-цепь (колебательный контур). Пример 1 .....	10
2.2.2. RLC-цепь. Пример 2 .....	12
2.2.3. RLC-цепь. Построение математической модели с использованием преобразования Лапласа.....	13
2.3. Задача автоматического подбора параметров цепи ...	14
2.4. Математические модели разветвленных цепей.....	21
2.4.1. Математическая модель разветвлённой RC-цепи. ..	21
2.5. Математические модели электрических цепей с нелинейностями. Однополупериодный выпрямитель .....	22
2.6. Моделирование электрических цепей в пакете MathCAD 14.....	25
2.7. Моделирование электрических сигналов в пакете MathCAD14 .....	30
Задания для лабораторных работ .....	36
Лабораторная работа 1 .....	36
Лабораторная работа 2 .....	39
Лабораторная работа 3 .....	42
Лабораторная работа 4 .....	45

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**

**Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)  
ул. Нестерова, 5а, Нижний Новгород,  
603951  
Тел. (831)419 – 79 – 51**

Кафедра «Систем информационной  
безопасности, управления и телекоммуникаций»

**Экзамен по дисциплине «Информатика»  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматике»**

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №1**

1. Назначение микропроцессоров. Характеристика регистров МП.
2. Решение дифференциальных уравнений в MathCAD.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**

**Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)  
ул. Нестерова, 5а, Нижний Новгород,  
603951  
Тел. (831)419 – 79 – 51**

Кафедра «Систем информационной  
безопасности, управления и телекоммуникаций»

**Экзамен по дисциплине «Информатика»  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматике»**

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №2**

1. Регистры общего назначения. Двойные регистры. Слово состояния процессора.
2. Вычисление определенных интегралов в MathCAD.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**

**Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)**

ул. Нестерова, 5а, Нижний Новгород,  
603951

Тел. (831)419 – 79 – 51

Кафедра «Систем информационной  
безопасности, управления и телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматики»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №3**

1. Регистр признаков (флагов). Примеры использования.
2. Моделирование. Виды моделирования. Классификация моделей.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**

**Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)**

ул. Нестерова, 5а, Нижний Новгород,  
603951

Тел. (831)419 – 79 – 51

Кафедра «Систем информационной  
безопасности, управления и телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматики»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №4**

1. Регистры: указатель стека, программный счетчик. Назначение.
2. Обработка данных двумерных массивов в MathCAD.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**

**Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)**

ул. Нестерова, 5а, Нижний Новгород,  
603951

Тел. (831)419 – 79 – 51

Кафедра «Систем информационной  
безопасности, управления и телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматики»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №5**

1. Структура машинной команды МП.
2. Исследование переходных процессов в электрических цепях с помощью MathCAD.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**

**Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)**

ул. Нестерова, 5а, Нижний Новгород,  
603951

Тел. (831)419 – 79 – 51

Кафедра «Систем информационной  
безопасности, управления и телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматики»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №6**

1. Команды пересылки данных МП.
2. Обработка данных одномерных массивов в MathCAD.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.



**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**  
Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)  
ул. Нестерова, 5а, Нижний Новгород,  
603951  
Тел. (831)419 – 79 – 51

Кафедра «Систем информационной безопасности,  
управления и телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматики»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №7**

1. Команды МП с косвенной адресацией. Примеры.
2. Математические и компьютерные модели неразветвленных электрических цепей.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**  
Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)  
ул. Нестерова, 5а, Нижний Новгород,  
603951  
Тел. (831)419 – 79 – 51

Кафедра «Систем информационной  
безопасности, управления и телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматики»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №8**

1. Арифметические команды МП. Примеры.
2. Способы задания сложных функций MathCAD.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**  
Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)  
ул. Нестерова, 5а, Нижний Новгород,  
603951  
Тел. (831)419 – 79 – 51

Кафедра «Систем информационной безопасности,  
управления и телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматики»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №9**

1. Команды условных и безусловных переходов МП.
2. Математические модели разветвленных электрических цепей.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**  
Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)  
ул. Нестерова, 5а, Нижний Новгород,  
603951  
Тел. (831)419 – 79 – 51

Кафедра «Систем информационной  
безопасности, управления и телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматики»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №10**

1. Логические команды МП.
2. Исследование функций MathCAD.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**  
Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)  
ул. Нестерова, 5а, Нижний Новгород,  
603951  
Тел. (831)419 – 79 – 51

Кафедра «Систем информационной безопасности,  
управления и телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматики»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №11**

1. Примеры использования логических команд МП.
2. Математические модели электрических цепей с нелинейными элементами.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**  
Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)  
ул. Нестерова, 5а, Нижний Новгород,  
603951  
Тел. (831)419 – 79 – 51

Кафедра «Систем информационной  
безопасности, управления и телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматики»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №12**

1. Принцип маскирования. Примеры использования.
2. Этапы компьютерного эксперимента с использованием математической модели. Понятие адекватности и неадекватности модели.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**  
Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)  
ул. Нестерова, 5а, Нижний Новгород,  
603951  
Тел. (831)419 – 79 – 51

Кафедра «Систем информационной безопасности,  
управления и телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматики»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №13**

1. Команды сдвига. Примеры использования. Алгоритм умножения на число, равное  $2^N$ .
2. Инструменты пакета MathCAD.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**  
Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)  
ул. Нестерова, 5а, Нижний Новгород,  
603951  
Тел. (831)419 – 79 – 51

Кафедра «Систем информационной  
безопасности, управления и телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматики»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №14**

1. Подпрограммы и регистр стека МП.
2. Основные модули судовых информационных систем, их назначение.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**  
Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)  
ул. Нестерова, 5а, Нижний Новгород,  
603951  
Тел. (831)419 – 79 – 51

Кафедра «Систем информационной  
безопасности, управления и телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматики»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №15**

1. Организация и принцип работы стека.
2. Основные принципы функционирования судовой информационной системы.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**  
Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)  
ул. Нестерова, 5а, Нижний Новгород,  
603951  
Тел. (831)419 – 79 – 51

Кафедра «Систем информационной  
безопасности, управления и телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматики»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №16**

1. Алгоритм подпрограммы временной задержки МП.
2. Построение графиков в MathCAD.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**  
**Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования**  
**«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)**  
ул. Нестерова, 5а, Нижний Новгород,  
603951  
Тел. (831)419 – 79 – 51

Кафедра «Систем информационной безопасности,  
управления и телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматике»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №17**

1. Организация индикации с помощью светодиодов. Кодировка символов.
2. Основные команды MathCAD.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**  
**Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования**  
**«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)**  
ул. Нестерова, 5а, Нижний Новгород,  
603951  
Тел. (831)419 – 79 – 51

Кафедра «Систем информационной  
безопасности, управления и телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматике»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №18**

1. Алгоритм сдвига символов на индикаторах.
2. Основные принципы функционирования судовой информационной системы.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**  
Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)  
ул. Нестерова, 5а, Нижний Новгород,  
603951  
Тел. (831)419 – 79 – 51

Кафедра «Систем информационной безопасности,  
управления и телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматики»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №19**

1. Алгоритм сложения однобайтных чисел, если их сумма больше 255.
2. Вычисление определенных интегралов в MathCAD.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

**ФЕДЕРАЛЬНОЕ АГЕНСТВО  
МОРСКОГО И РЕЧНОГО  
ТРАНСПОРТА**  
Федеральное государственное  
бюджетное образовательное  
учреждение высшего образования  
«Волжский государственный  
университет водного транспорта»  
(ФГБОУ ВО «ВГУВТ»)  
ул. Нестерова, 5а, Нижний Новгород,  
603951  
Тел. (831)419 – 79 – 51

Кафедра «Систем информационной  
безопасности, управления и телекоммуникаций»

**Экзамен по дисциплине «Информатика»**  
по направлению подготовки 26.05.07  
«Эксплуатация судового электрооборудования и  
средств автоматики»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №20**

1. Алгоритм «движущегося текста» на индикаторах. Общая блок-схема.
2. Решение дифференциальных уравнений в MathCAD.

Зав. кафедрой, д.т.н., профессор



Федосенко Ю.С.

