

# УЧЕБНОЕ ПОСОБИЕ

ДЛЯ ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕНИЙ

СПЕЦИАЛЬНОСТЬ



## Сборник примеров и задач по программированию



Москвитина О. А.  
Новичков В. С.  
Пылькин А. Н.

2-е издание

О. А. Москвитина  
В. С. Новичков  
А. Н. Пылькин

# Сборник примеров и задач по программированию

*2-е издание, стереотипное*

Допущено УМО по образованию  
в области прикладной информатики  
в качестве учебного пособия для студентов  
высших учебных заведений, обучающихся  
по специальности «Прикладная информатика  
(по областям)» и другим междисциплинарным  
специальностям»

Москва  
Горячая линия – Телеком  
2014

ББК 32.973  
УДК 681.33  
М82

**Москвитина О. А., Новичков В. С., Пылькин А. Н.**

**М82** Сборник примеров и задач по программированию. Учебное пособие для вузов. – 2-е изд., стереотип. – М.: Горячая линия–Телеком, 2014. – 245 с.: ил.  
**ISBN 978-5-9912-0411-8**

Цель книги – помочь изучающим язык Турбо Паскаль 7.0 приобрести практические навыки разработки алгоритмов и программ с использованием принципов структурного программирования путем решения специально подобранных задач. В систематизированном виде приведены необходимые теоретические сведения и примеры решения типовых задач. По каждой теме приведены наборы заданий различной степени сложности.

Для учащихся высших учебных заведений.

**ББК 32.973**

Адрес издательства в Интернет WWW.TECHBOOK.RU

*Учебное издание*

**Москвитина** Оксана Александровна, **Новичков** Валентин Семенович,  
**Пылькин** Александр Николаевич

**СБОРНИК ПРИМЕРОВ И ЗАДАЧ ПО ПРОГРАММИРОВАНИЮ**

**Учебное пособие**

*2-е издание, стереотипное*

Художник В. Г. Ситников  
Подготовка оригинал-макета И. М. Чумаковой

Подписано к печати 05.02.2014. Формат 60×88 1/16. Усл. печ. л. 15,25. Изд. № 140411.  
Тираж 300 экз. (1-й завод 100 экз.)

ISBN 978-5-9912-0411-8

© О. А. Москвитина, В. С. Новичков,  
А. Н. Пылькин, 2007, 2014

© Издательство «Горячая линия–Телеком», 2014

## ПРЕДИСЛОВИЕ

Задания предназначены для практических и лабораторных работ, основной целью выполнения которых является приобретение навыков алгоритмизации задач и конструирования программ на алгоритмическом языке Паскаль в среде программирования Turbo Pascal 7.0.

Перед выполнением каждой работы рекомендуется ознакомиться с теоретическим материалом (Новичков В. С., Парфилова Н. И., Пылькин А. Н. Алгоритмизация и программирование на Турбо Паскале: Учеб. пособие – М.: Горячая линия – Телеком, 2005. – 462 с.: ил.) и разобрать примеры конкретных программ по каждой теме.



# Тема 1. ПРОГРАММИРОВАНИЕ ЛИНЕЙНЫХ АЛГОРИТМОВ

## ПРИМЕР СОСТАВЛЕНИЯ ПРОГРАММЫ

Пусть требуется составить программу вычисления общей поверхности и объема круглого конуса по заданным радиусу основания  $R$  и длине образующей  $L$ . При вычислениях использовать равенства

$$S = \pi R^2 + \pi RL; \quad V = \frac{1}{3} \pi R^2 H,$$

где  $H$  – высота конуса, определяемая по формуле

$$H = \sqrt{L^2 - R^2}.$$

Программа на языке Турбо Паскаль в общем случае содержит следующие основные элементы:

- вводные комментарии;
- заголовок;
- раздел определения констант;
- раздел описания переменных;
- раздел операторов.

Определение исходных данных может быть осуществлено с помощью операторов присваивания либо с помощью специальных операторов ввода. Вычисления по формулам реализуются в порядке определения числовых значений переменных  $H$ ,  $S$  и  $V$  соответственно, после чего значения  $S$  и  $V$  выводятся на экран.

Для удобства пользования введем комментарии, которые позволяют записать нужную учетную и поясняющую информацию.

При вычислениях будет использована константа  $\pi$ , которой в Турбо Паскале соответствует имя `PI`.

В качестве имен переменных будем использовать переменные, обозначения которых максимально совпадают с именами и обозначениями переменных самой задачи.

Общий вид схемы алгоритма для рассматриваемого примера показан на рис. 1.1 и представляет последовательность простейших действий (линейный алгоритм).

Программа на языке Турбо Паскаль также состоит из последовательности операторов, выполняемых в порядке их следования. С целью сокращения объема вычислений дважды повторяющееся выражение  $\pi R^2$  вычислим один раз, присвоим полученное значение вспомогательной переменной  $T$  и затем будем использовать его при вычислении более сложных выражений.

```

{*****}
{Программа: CONUS.}
{Цель: вычисление площади и объема конуса.}
{Описание параметров и переменных:}
{ R - радиус; L - длина образующей; }
{ H - высота; S - площадь поверхности; }
{ V - объем; T - вспомогательная }
{ переменная. }
{Требуемые подпрограммы: нет.}
{Метод: вычисление по формулам.}
{Программист: Старкова Т.В.}
{Дата написания: 25 октября 2005 г.}
{*****}

```

**Program** CONUS;

**Var**

H, L, R, S, T, V: Real;

**Begin** {CONUS}

WriteLn('Введите радиус и длину',  
' образующей');

Read(R,L);

WriteLn('R=',R,' L=',L); {Эхо-печать}

H:=Sqrt(L\*L-R\*R); {Вычисление высоты}

**T:=PI\*R\*R;** {Площадь основания}

S:=T+PI\*R\*L; {Полная площадь поверхности}

V:=T\*H/3; {Вычисление объема конуса}

WriteLn('Параметры конуса:');

WriteLn('H=',H,' V=',V,' S=',S)

**End.** {CONUS}

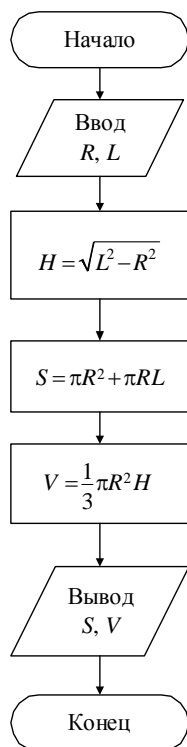


Рис. 1.1. Алгоритм вычисления площади и объема конуса

Исходные данные для просчета контрольного варианта выбираем таким образом, чтобы вычисления были достаточно простыми и в то же время обеспечивали полные вычисления по всем формулам.

Например,  $R = 3$  см,  $L = 5$  см. Тогда получим  $H = 4$  см,  $S = 75,36$  см<sup>2</sup>,  $V = 37,68$  см<sup>3</sup>.

В результате выполнения программы при заданных исходных данных на экран дисплея будет выведена следующая информация:

Введите радиус и длину образующей

3 5

R = 3.000000E+00 L = 5.000000E+00

Параметры конуса:

H = 4.000000E+00 V = 3.767990E+01 S = 7.539820E+01

Данные контрольного просчета подтвердили правильность выполнения программы.

## ЗАДАНИЯ

Составить линейный алгоритм и программу для предложенного варианта, в которой вводятся исходные данные, вычисляются заданные арифметические выражения и выводятся на экран дисплея результаты вычислений (исходные данные выбираются произвольно, исходя из области допустимых значений функции).

Вариант 1

$$y = \sqrt{a_0 + a_1 x^2} + 2x^5 + \frac{0,13725 \cdot 10^{-3} + a_0^2}{2 + \sin^2 3x}; z = a_0 \sqrt[3]{x} + \ln |1 + a_0 + a_1 x^3|$$

Вариант 2

$$y_1 = \ln |x^3| + \operatorname{tg} \alpha - e^{\alpha x^2 + x}; y_2 = \lg |a^7| + \operatorname{arctg} x^2 + \frac{\pi + 4,56 \cdot 10^{-4}}{\sqrt[4]{|a + x|}}$$

Вариант 3

$$\alpha = 5a^{2x}(a + x) - \sqrt{|\cos x^2|}; y = 18ax + \sqrt[3]{y^2} + \sin \frac{\alpha}{2}$$

Вариант 4

$$\varpi = 5b^3 + \sin(ax^2 + b); z = \frac{x - a}{(1 + 2x)^{a^3}} - e^{\sqrt{1 + \varpi^2}}$$

Вариант 5

$$y = \sqrt[3]{a^2 + b^2}; z = \left| \sin^2 \frac{a}{2} \right| \cdot \cos \frac{b}{3}; x = \frac{(y^2 + z^3)}{e^{ay + b^2 z}}$$

Вариант 6

$$x = \sqrt[5]{\frac{(a + b)}{(1 + a)}}; z = e^{\sqrt{x-1}}; y = \frac{2 \sin x + \cos \frac{x}{2}}{3 + \cos^2 x}$$

Вариант 7

$$y = \frac{x + x^3 + \sqrt{x}}{e^{ax+3}}; \quad z = \left| \sin \frac{y}{2} \right| \cdot \cos \frac{a}{2} + \sqrt{a^2 + x^3}$$

Вариант 8

$$\alpha = \frac{\pi + (a + x^2)^3}{\ln |a + x| + 3,45 \cdot 10^{-3}}; \quad \beta = \frac{1}{1 + \frac{1+x}{1 + \frac{1+x}{1+x^2}}}$$

Вариант 9

$$a = \sin x; \quad b = \cos^2 x; \quad c = 1 + \operatorname{tg} x;$$

$$y = 4a^2 + x \left( b^3 + x \left( c^2 + x \left( ab + (a+b)^x \right) \right) \right)$$

Вариант 10

$$\alpha = 1 + e^{-\sqrt{|2x|}}; \quad \beta = \frac{e^x + e^{-x}}{1 + e^{-2x}}; \quad y = \operatorname{ctg} \frac{\alpha}{3} + \frac{\cos \beta + \sin^2 3\alpha}{1 + \sqrt[3]{1+x}}$$

Вариант 11

$$y = \frac{a \cdot \sqrt{\sin x + 3} \cdot e^{-ax}}{\sqrt[3]{\ln(2b + c^2) + b^{ax}}}; \quad z = \left( \frac{a^2}{a+b} + \frac{c}{ax^2 + bx} \right)^5 + \sin \frac{y}{3}$$

Вариант 12

$$\alpha = e^{2x} - e^{-2x}; \quad y = \left( \sqrt{\frac{ax^2 + b}{a^2x + b^3}} + \operatorname{tg} x \right)^{2/3}$$

Вариант 13

$$y = \frac{(\operatorname{arctg} x^3 + \cos \sqrt{x})^{2x}}{e^x + \ln |2,4x^2|}; \quad z = ay^5 + b \cdot \cos |y| + \operatorname{arctg} y^x$$

Вариант 14

$$y = \frac{3x^5 + 25e^{x^2}}{|x^{10}| + \sqrt{ax^3 + 2}} + \ln(x+1); \quad z = \pi x^2 - a^2x + a \cdot \sin \left( \frac{x}{3} + \frac{\pi}{8} \right)$$

Вариант 15

$$z = \frac{x^2 + \frac{x}{2}}{e^x + \sin^3 x} + 16 \cdot e^{x^2} \cdot \ln x^2; \quad y = a + \frac{x}{7,5 - 3,2x^2} + \frac{x^3 \cdot (a-1)}{\ln |x^3 - 6|}$$

Вариант 16

$$z = \frac{b + b^3}{e^x + \sin^2 2x} + \frac{3,45 \cdot 10^{-6} + x^2}{\sqrt{bx^2 + x + 1}}; \quad y = \frac{\pi}{8} \cdot x^2 + \frac{\pi}{4} \cdot x + \frac{\pi}{2}$$

Вариант 17

$$z = -a^2 x^5 + b \cdot \sin x + \frac{x^2}{x + e^x} - \ln |x^2 + 2x + 3|; \quad y = a^3 \cdot \left( \frac{x}{a+x} \right)^{2/3} - 2 \cdot \sin x^2$$

Вариант 18

$$t = \frac{1}{2\pi} \cdot \sqrt{\frac{m}{c^2}}; \quad z = e^{mc^2} + t^3 \cdot \sin \frac{t}{8} - \operatorname{tg}^2 \frac{t \cdot \pi}{12}$$

Вариант 19

$$x = \operatorname{arctga} + \frac{\sin^2 a}{2} - e^{\sqrt{|2a|}}; \quad y = ax^3 + \frac{a}{2} \cdot x^2 + \sqrt{a} \cdot x + e^{-a}$$

Вариант 20

$$s = \frac{\varpi^2 \cdot t}{2} \cdot \sin \frac{\varpi}{3} + \varpi t \cdot \cos \frac{\varpi}{2}; \quad \vartheta = \frac{\varpi}{3} \cdot \cos \varpi + \frac{\pi + 2}{3} \cdot \operatorname{tg} \frac{\varpi}{4}$$

Вариант 21

$$a = \sin 2x; \quad b = \sqrt{a^2 - 4x}; \quad c = e^{4x+0,5}; \quad y = a^2 \cdot x^2 + bx + c^3$$

Вариант 22

$$c = \frac{a+b^2}{a^2+b}; \quad x = \left( \frac{a+b}{c} \right)^7 + \left( \frac{a+b}{c} \right)^5 + \frac{\pi}{3}$$

Вариант 23

$$x_1 = \frac{-b + \sqrt{D}}{2a}; \quad x_2 = \frac{-b - \sqrt{D}}{2a}, \text{ где } D = b^2 - 4ac$$

Вариант 24

$$y = a^x - e^x + a^3 + \cos(4x - 0,2); \quad z = \operatorname{tg} 4,5x + \frac{x + 84,85 \cdot 10^{-3}}{2 + \sin 0,5x}$$

Вариант 25

$$z = \ln x^2 - e^{ax+b} + \frac{x + 2,3 \cdot 10^{-3}}{x + 3,4 \cdot 10^{-3}}; \quad y = \frac{ax^2 + b}{ax^5 + b^2} + \ln |1 + z^3|$$

## Тема 2. СОСТАВЛЕНИЕ И ПРИМЕНЕНИЕ ПРОСТЕЙШИХ ФУНКЦИЙ ПОЛЬЗОВАТЕЛЯ

### ПРИМЕРЫ СОСТАВЛЕНИЯ ПРОГРАММ

**Пример 2.1.** Пусть необходимо составить программу для вычисления площади выпуклого четырехугольника  $ABCD$ , заданного длинами своих сторон  $a, b, c, d$  и одной из диагоналей  $e$  (рис. 2.1).

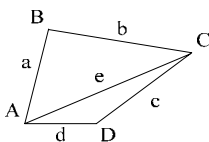


Рис. 2.1. Выпуклый четырехугольник

Площадь четырехугольника в данном случае можно получить как сумму площадей треугольников  $ABC$  и  $ACD$ , вычисленных по формуле Герона

$$S = \sqrt{p(p-a)(p-b)(p-c)},$$

где  $p = \frac{a+b+c}{2}$  – полупериметр;  $a, b, c$  – длины сторон.

Поскольку здесь дважды необходимо вычислять значение площади треугольника, то целесообразно в программу ввести описание функции и дважды обратиться к ней при вычислении площади четырехугольника.

Схема алгоритма решения поставленной задачи приведена на рис. 2.2, где наряду с алгоритмом основной программы дается и схема алгоритма вычисления функции  $PITr$ , вычисляющей площадь треугольника.

Схема алгоритма вычисления функции начинается блоком «вход» и заканчивается блоком «выход», в которых указываются имя функции, и список параметров  $a, b, c$ , являющихся исходными данными, необходимыми для вычисления площади треугольника. Вычисленное значение площади присваивается непосредственно имени функции  $PITr$  и возвращается в основную программу. В основной программе происходит обращение к описанию функции для вычисления площади первого треугольника,  $ABC$ , и второго,  $ACD$ .

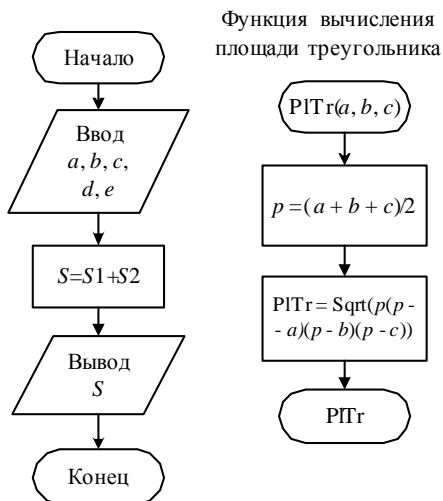


Рис. 2.2. Алгоритм вычисления площади четырехугольника

Ниже приведен текст программы.

```

{*****}
{Цель: вычисление площади четырехугольника.}
{Переменные: a,b,c,d – длины сторон четырехугольника;}
{e – длина диагонали;}
{S – площадь четырехугольника.}
{Подпрограммы: PlTr – функция вычисления площади}
{треугольника.}
{Метод: вычисление по формуле Герона.}
{Программист: Богомолов О.М.}
{Дата написания: 1 ноября 2005 г.}
{*****}

```

**Program Pl;**

**Var**

a, b, c, d, e, S :real;

{Функция вычисления площади треугольника. }

{Переменные: a, b, c – стороны треугольника; }

{ p – полупериметр. }

**Function PlTr(a, b, c :real):real;**

**Var**

p:real;

**begin** {PlTr}

p := (a+b+c)/2;

PlTr := Sqrt(p\*(p-a)\*(p-b)\*(p-c))

**end;** {PlTr}



```

Begin {P1}
  WriteLn('Введите длины сторон и диагонали');
  Read(a, b, c, d, e);
  WriteLn('При a=',a:6:2,' b=',b:6:2,' c=',c:6:2,
    ' d=',d:6:2,' e=',e:6:2);
  S := PlTr(a, b, e) + PlTr(c, d, e);
  WriteLn('Площадь четырехугольника равна',S:8:2)
End. {P1}

```

Исходные данные при выполнении программы должны выбираться таким образом, чтобы их значения удовлетворяли условию существования треугольника.

**Пример 2.2.** Пусть необходимо составить программу вычисления выражения

$$y = \frac{2\operatorname{tg}\frac{\pi x}{2} + 3\operatorname{tg}\frac{x+1}{2}}{\sqrt{4 + \operatorname{tg}^2\frac{x-1}{4}}}.$$

В заданном выражении трижды встречается тригонометрическая функция тангенса, которая отсутствует в списке стандартных функций языка Паскаль. Поэтому с целью сокращения объема вычислений ее целесообразно оформить в виде функции пользователя, определяющей значение тангенса по известной формуле

$$\operatorname{tg} x = \frac{\sin(x)}{\cos(x)},$$

а затем использовать при определении значения заданного выражения.

Схема алгоритма решения поставленной задачи будет иметь вид, аналогичный примеру 2.1, поэтому сразу приведем текст программы, где для обозначения константы  $\pi$  будем использовать стандартную действительную константу языка программирования с именем `Pi`.

```

{Цель: вычисление значения функции.          }
{Переменные: x - аргумент; y - значение выражения; }
{  Pi - константа «пи».                        }
{Подпрограммы: tg - функция вычисления тангенса.  }
{Метод: вычисление по формулам.                  }
{Программист: Стрельцов Т.Н.                    }
{Дата написания: 6 ноября 2005 г.                }
Program Expression;
Var
  x, y : real;

```

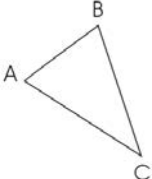
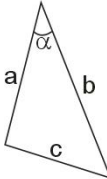
```

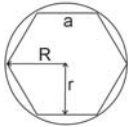
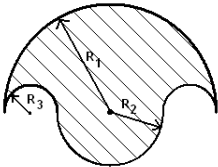
{ Функция вычисления значения тангенса. }
{ Переменные: x - аргумент. }
Function tg(x :real):real;
begin {tg}
  tg := sin(x)/cos(x)
end; {tg}
Begin {Expression}
  WriteLn('Введите значение x');
  Read(x);
  WriteLn('При x=',x:6:2);
  y:=(2*tg(Pi*x/2)+3*tg((x+1)/2))/Sqrt(4+Sqr(tg((x-1)/4)));
  WriteLn('Значение функции y = ',y:8:2)
End. {Expression}

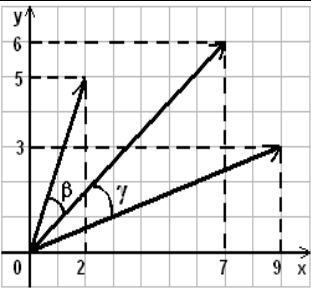
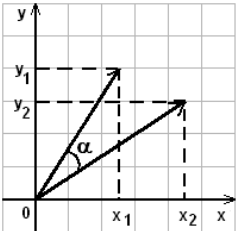
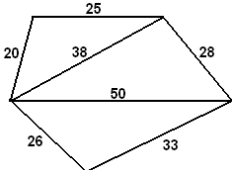
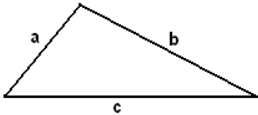
```

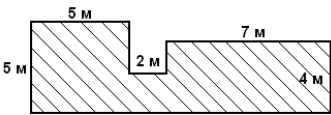
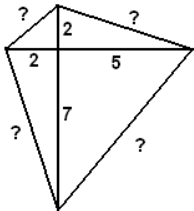
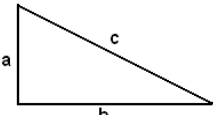
## ЗАДАНИЯ

Для каждого варианта разработать алгоритм вычисления выражения, заданного в первом столбце, введя функцию пользователя, описанную во втором столбце.

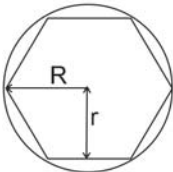
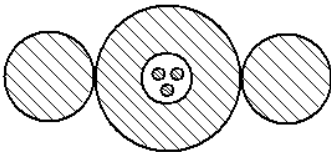
Вариант	Вычисляемые выражения	Функции пользователя
1	 <p>Заданы стороны треугольника <math>ABC</math>.</p> <p>Определить углы <math>\angle A, \angle B, \angle C</math></p>	 $\cos \alpha = \frac{a^2 + b^2 - c^2}{2ab} = x;$ $\alpha = \arccos x = \frac{\pi}{2} - \arcsin x;$ $\arcsin x = \arctg \frac{x}{\sqrt{1-x^2}}$ $(-1 < x < 1) \text{ или } \alpha = \arctg \frac{\sqrt{1-x^2}}{x}$

Вариант	Вычисляемые выражения	Функции пользователя
2	$y = \frac{2^{3x+2} + 3^{4x}}{\left(\frac{1}{2}\right)^x + \left(\frac{1}{3}\right)^{2x}};$ $z = \sin \frac{x}{2} + (x+2)^{x+1}$	$a^x = e^{x \ln a}, \quad a > 0$
3	$y = \frac{2 \operatorname{tg} \frac{x}{2} + 3 \operatorname{ctg} \frac{x}{2}}{4 + \operatorname{tg} \frac{x}{4}};$ $z = \frac{x^2 + 3,4 \cdot 10^{-3}}{x^3 + \operatorname{tg} x}$	$\operatorname{tg} x = \frac{\sin(x)}{\cos(x)}, \quad \operatorname{ctg} x = \frac{\cos(x)}{\sin(x)}$ $\left( x \neq \frac{\pi}{2}; \quad x \neq 0; \quad x \neq \pi \right)$
4	<p>Для произвольно заданного радиуса окружности <math>R</math> определить площади правильных фигур, вписанных в окружность: 10-угольника; 50-угольника; 100-угольника и площадь круга</p>	 <p>Правильный <math>n</math>-угольник вписан в окружность с радиусом <math>R</math></p> $a = 2R \sin \frac{180^\circ}{n}, \quad r = R \cos \frac{180^\circ}{n}$ <p>Площадь <math>n</math>-угольника <math>S = \frac{1}{2} Pr</math>, где <math>P = an</math> (периметр)</p>
5	 <p>Вычислить площадь заштрихованной фигуры при <math>R_1 = 25</math> см; <math>R_2 = 15</math> см; <math>R_3 = 5</math> см</p>	<p>Площадь круга <math>S = \pi R^2</math></p>

Вариант	Вычисляемые выражения	Функции пользователя
6	 <p>Определить углы <math>\beta</math> и <math>\gamma</math> между векторами</p>	 $x = \cos \alpha = \frac{x_1 x_2 + y_1 y_2}{\sqrt{x_1^2 + y_1^2} \cdot \sqrt{x_2^2 + y_2^2}};$ $\alpha = \arccos x = \frac{\pi}{2} - \arcsin x;$ $\arcsin x = \arctg \frac{x}{\sqrt{1-x^2}}$ $(-1 < x < 1) \text{ или}$ $\alpha = \arctg \frac{\sqrt{1-x^2}}{x}$
7	 <p>Определить площадь многоугольника</p>	 $S = \sqrt{p(p-a)(p-b)(p-c)},$ <p>где <math>P = \frac{a+b+c}{2}</math></p>
8	$y = \frac{\sin \frac{4\pi}{3} + 7 \sin \frac{5\pi}{2}}{5 \sin \frac{8\pi}{5} + \sin \frac{7\pi}{3}};$ $z = \frac{y^2 + \sin \frac{\pi}{2}}{1 + y}$	$f(n, k) = \sin \frac{\pi \cdot n}{k}$
9	Определить длину границы для фигуры из варианта 5	Длина окружности $C = 2\pi R$

Вариант	Вычисляемые выражения	Функции пользователя
10	$y = \frac{2 + \cos \frac{3\pi}{5} + \cos \frac{4\pi}{7}}{8 + \cos \frac{3\pi}{7} + \cos \frac{5\pi}{11}};$ $z = \frac{\cos \frac{8\pi}{7} + \cos \frac{7\pi}{8}}{1 + \cos \frac{5\pi}{6}}$	$f(m, n) = \cos \frac{\pi \cdot m}{n}$
11	 <p>Определить площадь участка</p>	<p>Площадь прямоугольника</p> $S(a, b) = a \cdot b$
12	$y = \frac{2,5x^2 + 3,4x + 8,1}{3,6x^2 - 1,8x - 5,2};$ $z = \frac{3x^2 - 2x + 1}{3x^2 - 2x + 2\pi}$	$f(a, b, c, x) = ax^2 + bx + c$
13	 <p>Определить длину периметра</p>	 $c = \sqrt{a^2 + b^2}$
14	$y = 4 \sin x + \frac{5 \sin x + 3 \cos x}{8 \sin x + 4 \cos x};$ $y = \operatorname{tg} x + \frac{1,3 \sin x + 1,8 \cos x}{2,5 \sin x - 4,5 \cos x}$	$f(a, b, x) = a \sin x + b \cos x$
15	$P(x) = \frac{8x^4 + 7,5x^2 - 4}{5,3x^3 + x + 3};$ $Q(x) = \frac{5x^5 + 3}{x^2 + 1}$	$f(a, n, x) = a \cdot x^n$

Вариант	Вычисляемые выражения	Функции пользователя
16	<p>Определить площадь четырехугольника</p>	$S = \frac{x \cdot y}{2}$
17	<p>Определить длину</p>	<p>Длина половины окружности</p> $C = \pi R$
18	$y = 2 + \sqrt[3]{\left(\frac{1+x+x^2}{1+3x}\right)^2};$ $z = \frac{1}{2} \sqrt[5]{\left(\frac{x^8+x^7+3}{x+1}\right)^3}$	$f(n, m, x) = \sqrt[n]{x^m}$
19	<p>Определить площадь кольца  <math>r = 2,87 \text{ см}</math>, <math>R = 8,51 \text{ см}</math></p>	<p>Площадь круга <math>S = \pi \cdot R^2</math></p>
20	<p>Определить заштрихованную площадь</p>	$S = \sqrt{p(p-x)(p-y)(p-z)}, \text{ где}$ $P = \frac{x+y+z}{2}$

Вариант	Вычисляемые выражения	Функции пользователя
21	 <p>Определить периметр правильного <math>n</math>-угольника при <math>R = 3</math>, <math>n = 50</math>; <math>n = 100</math></p>	<p>Периметр <math>P = 2R \cdot n \cdot \sin \frac{180^\circ}{n}</math> (см. задание к варианту 4)</p>
22	 <p>Определить площадь фигур, используемых при изображении Чебурашки:</p> <ul style="list-style-type: none"> <li>▪ радиус уха равен 1,5;</li> <li>▪ радиус головы – 2;</li> <li>▪ радиус лица – 1;</li> <li>▪ радиус глаза и носа – 0,1</li> </ul>	<p>Площадь круга <math>S = \pi \cdot R^2</math></p>
23	$y = \frac{\sin(2x+1) + \sin(3x+0,5)}{1 + \sin \frac{\pi}{8}};$ $z = \frac{1 + \sin(3x)}{2 + \sin\left(3x + \frac{\pi}{9}\right)}$	<p><math>f(a, b, x) = \sin(ax + b)</math></p>

Вариант	Вычисляемые выражения	Функции пользователя
24	 <p>Определить площадь фигуры</p>	 <p>Площадь трапеции <math>S = \frac{a+b}{2}h</math></p>
25	$y_1 = \frac{x^3 + (1+x)^3}{3x^3 + (1+2x+x^2)^3};$ $y_2 = \sqrt{2+(2x+1)^3}$	$f(x) = x^3 = x \cdot x \cdot x$



### Тема 3. ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ РАЗВЕТВЛЯЮЩЕЙСЯ СТРУКТУРЫ С ИСПОЛЬЗОВАНИЕМ УСЛОВНОГО ОПЕРАТОРА

#### ПРИМЕРЫ СОСТАВЛЕНИЯ ПРОГРАММ

**Пример 3.1.** Даны три числа  $a$ ,  $b$  и  $c$ . Вычислить и вывести на экран дисплея значение  $z$ , равное квадрату большего из них.

Проектирование схемы алгоритма решения поставленной задачи по технологии нисходящего структурного программирования показано на рис. 3.1.

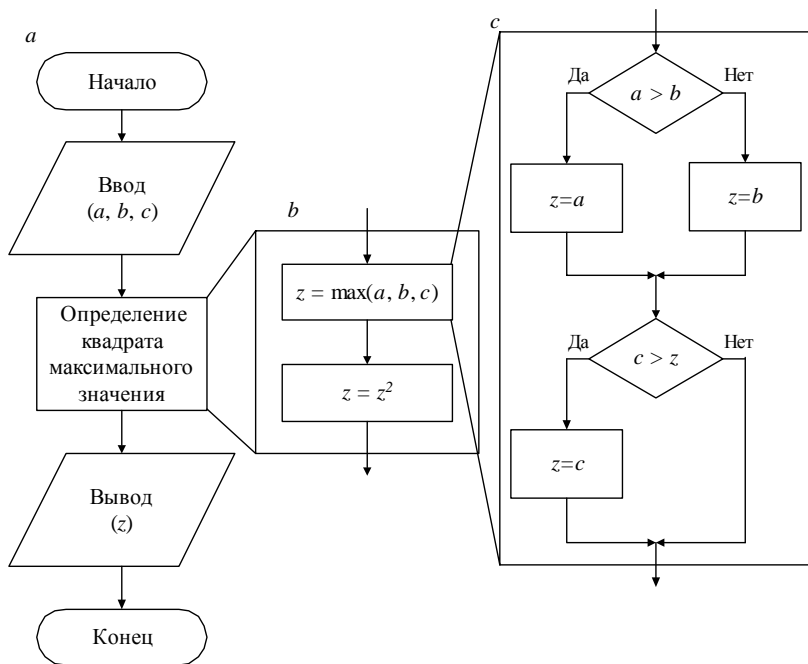


Рис. 3.1. Алгоритм нахождения квадрата максимального значения

На первом этапе любой сложный алгоритм можно представить состоящим из трех шагов: ввода данных, обработки, вывода результата

(рис. 3.1, *a*). Первый и последний шаги могут быть представлены непосредственно операторами ввода-вывода, поэтому на втором этапе детализируется только блок обработки, который разбивается на два более элементарных шага – поиск максимального из трех чисел и возведение найденного значения в квадрат (рис. 3.1, *b*). Поиск максимального значения (рис. 3.1, *c*) можно осуществить, сравнив вначале два числа  $a$  и  $b$  и выбрав наибольшее из них, а затем сравнить полученное значение с третьим числом  $c$  и выбрать в качестве максимального именно его, если оно окажется больше любого из двух первых чисел. Для реализации первого разветвления используется условный оператор в полной форме записи, а для второго – в сокращенной. Окончательная схема алгоритма, полученная после вложения соответствующих структур, изображена на рис. 3.2.

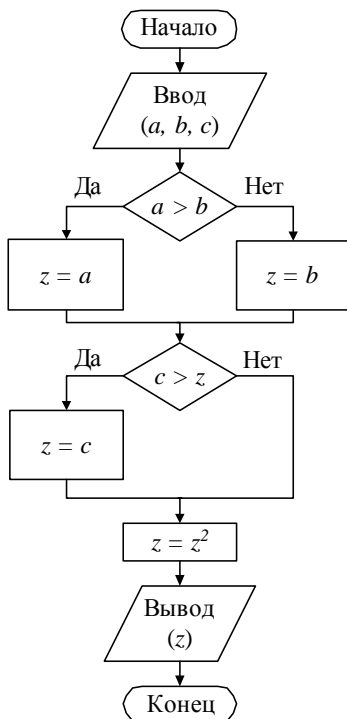


Рис. 3.2. Окончательная схема алгоритма нахождения квадрата максимального значения

Программа на языке Турбо Паскаль, соответствующая окончательной схеме, имеет следующий вид:

```
{Цель: вычисление квадрата максимального из трех чисел.}
{Переменные: a, b, c - исходные данные; z - результат.}
{Подпрограммы: нет.}
{Метод: вычисление по формулам.}
{Программист: Антипов О.В.}
{Дата написания: 11 ноября 2005 г.}
```

```
Program Square_maximal;
Var
    a, b, c, z : real;
Begin {Square_maximal}
    WriteLn('Введите a '); {Ввод исходных данных}
    Read(a);
    WriteLn('Введите b ');
    Read(b);
    WriteLn('Введите c ');
    Read(c);
    {Эхо-печать}
    WriteLn('Из чисел a=', a:4:1, ' b=', b:4:1,
        ' c=', c:4:1);
    if a > b {Выбор наибольшего из двух чисел}
    then
        z := a
    else
        z := b;
    if c > z {Уточнение максимального числа}
    then
        z := c;
    z := z*z; {Возведение в квадрат}
    {Вывод результата}
    WriteLn('квадрат максимального числа равен ', z:8:2)
End. {Square_maximal}
```

При решении практических задач разветвления могут иметь более сложную структуру.

**Пример 3.2.** Сложное разветвление можно проиллюстрировать примером вычисления значения функции по одной из предложенных формул:

$$y = \begin{cases} \frac{1}{2}\sqrt{x}, & \text{если } x \geq 1; \\ \frac{1}{3}\sqrt[3]{x}, & \text{если } 0 < x < 1; \\ \frac{1}{4}\sqrt[4]{x}, & \text{если } x < 0. \end{cases}$$

При  $x = 0$  функция не определена.

При проектировании алгоритма решения подобной задачи необходимо проанализировать прежде всего то условие, выполнение которого сразу дает однозначный ответ. В данном примере таких условий три:  $x \geq 1$ ,  $x < 0$ ,  $x = 0$ . Поскольку при выполнении отношения  $x = 0$  необходимо выдать сообщение о неопределенности функции, то в качестве первого проверяемого условия целесообразно выбрать именно его (рис. 3.3). Если условие окажется ложным, то далее анализируется любое из двух оставшихся, например первое  $x \geq 1$ , выполнение которого определяет значение  $y = 1/2\sqrt{x}$  в первой ветви алгоритма. Если же данное отношение оказалось ложным, то, следовательно, истинным будет альтернативное условие  $x < 1$  и для выбора ветви, определяющей  $y = 1/3\sqrt[3]{x}$ , достаточно проанализировать отношение  $x > 0$ . Когда ни одно из проверяемых условий не является истинным, это будет означать, что выполнилось условие  $x < 0$ , определяющее последнюю ветвь алгоритма  $y = 1/4\sqrt[4]{|x|}$ . После вычисления значения функции по одной из выбранных формул должен быть предусмотрен общий для этих ветвей вывод результата. Следует отметить, что обе альтернативные ветви каждого разветвления должны сходиться к общей точке.

Для программной реализации таких разветвлений используется условный оператор **if-then-else**, имеющий вложенную структуру. При этом ветвь **else** в данном примере (так же как и всегда) относится к ближайшему **if**, не имеющему **else**. Ниже приведена структурированная программа, реализующая данный алгоритм. В результате структурирования соответствующие элементы условного оператора **if-then-else** записаны друг под другом. Операторы, ограниченные операторными скобками **begin-end**, если их несколько в какой-либо ветви, записываются также друг под другом и должны смещаться вправо относительно **begin** и **end**.

```
{Цель: вычисление значения сложной функции.      }
{Переменные: x – аргумент функции;                }
{      y – значение функции.                      }
{Подпрограммы: нет.                                }
{Метод: реализация разветвляющегося алгоритма.    }
{Программист: Шестопалов А.В.                     }
{Дата написания: 13 ноября 2005 г.                 }
```

```
Program Choice;
```

```
  Var
```

```
    x, y :real;
```

```
  Begin {Choice}
```

```
    {Ввод исходных данных}
```

```
    WriteLn('Введите аргумент x');
```

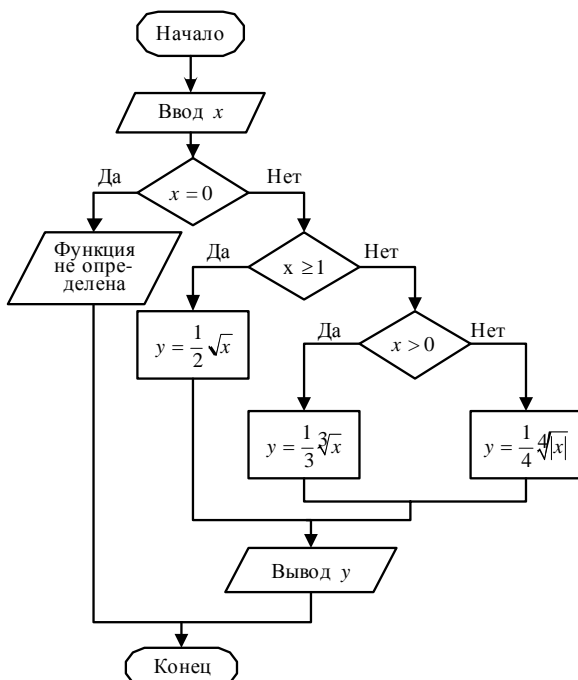


Рис. 3.3. Схема алгоритма сложного разветвления

```

Read(x);
WriteLn('При x=', x:6:2);      {Эхо-печать}
if x = 0                        {Проверка существования функции}
then
    WriteLn('Функция не определена')
else
    begin
        if x >= 1              {Выбор первой ветви}
        then
            y := Sqrt(x)/2
        else if x > 0 {Выбор второй ветви}
        then
            y := Exp(1/3*Ln(x))/3
        else
            y := Exp(1/4*Ln(Abs(x)))/4;
        {Вывод результата}
        WriteLn('Функция равна ', y:8:2)
    end
End. {Choice}
    
```

**Пример 3.3.** Вычислить значение функции, график которой изображен на рис. 3.4.

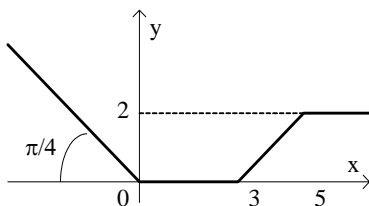


Рис. 3.4. График функции

Для решения данной задачи необходимо прежде всего записать аналитическое выражение функции. В соответствии с графиком область определения функции разбивается на 4 участка:

$$y = \begin{cases} -x, & \text{если } x \leq 0; \\ 0, & \text{если } 0 < x \leq 3; \\ x - 3, & \text{если } 3 < x \leq 5; \\ 2, & \text{если } x > 5. \end{cases}$$

Для построения схемы алгоритма используем вложенную конструкцию команд ветвления (рис. 3.5). Проверяем условия последовательно. Первым проверим условие  $x \leq 0$ . Следующее условие,  $0 < x \leq 3$ , будет проверяться только в том случае, если первое не соблюдается и  $x > 0$ . Следовательно, часть второго условия  $0 < x$  можно не проверять: если дело дошло до проверки этого условия, то заведомо  $0 < x$ . Аналогично исключается проверка  $3 < x$  из условия  $3 < x \leq 5$ , а также проверка последнего условия,  $x > 5$ . Тогда текст программы будет иметь следующий вид:

```
{Цель: вычисление значения функции, заданной }
{ графиком. }
{Переменные: x – аргумент функции; }
{ y – значение функции. }
{Подпрограммы: нет. }
{Метод: вычисление по графику. }
{Программист: Шилов В.П. }
{Дата написания: 15 ноября 2005 г. }
```

**Program Diagram;**

**Var**

x, y :real;

**Begin** {Diagram}

WriteLn('Введите аргумент x '); {Ввод исходных данных}  
Read(x);

```

WriteLn('При x=',x:6:2);           {Эхо-печать}
if x <= 0                          {Проверка первого условия}
then
    y := -x
else if x <= 3                    {Проверка второго условия}
then
    y := 0
else if x <= 5 {Проверка третьего условия}
then
    y := x - 3
else
    y := 2;
{Вывод результата}
WriteLn('Функция равна ', y:8:2)
End. {Diagram}

```

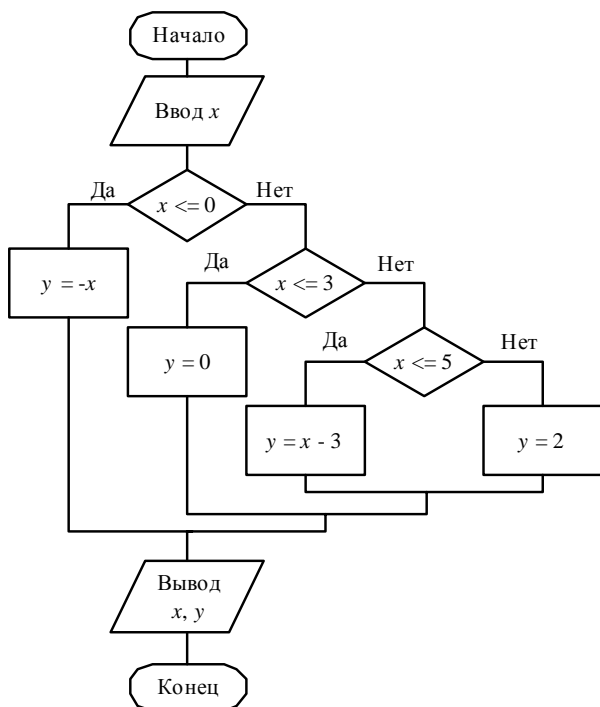


Рис. 3.5. Алгоритм вычисления значения функции по графику

**Пример 3.4.** Пусть необходимо определить, принадлежит ли точка  $M(x, y)$  к заштрихованной области (рис. 3.6).

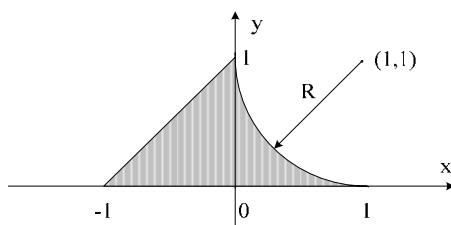


Рис. 3.6. Область определения

Для решения данной задачи необходимо определить, какие именно значения переменных  $x$  и  $y$  удовлетворяют граничным условиям заданной области. Поскольку она расположена в первом и втором квадрантах, то обязательным условием принадлежности точки с координатами  $(x, y)$  к изображенной на рисунке области является истинность отношения  $y > 0$ , что позволяет выбрать верхнюю полуплоскость (рис. 3.7, а).

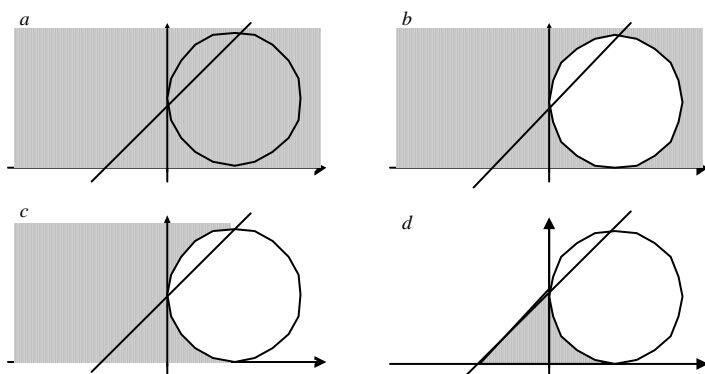


Рис. 3.7. Процесс отделения области

Уравнение окружности радиусом  $R$  с координатами центра  $(a, b)$  задается выражением

$$(x-a)^2 + (y-b)^2 = R^2.$$

Следовательно, координаты точки, находящейся *вне* круга (рис. 3.7, b), ограниченного окружностью радиусом  $R = 1$  с центром в точке с координатами  $a = 1$  и  $b = 1$ , должны удовлетворять неравенству

$$(x-1)^2 + (y-1)^2 > 1.$$

Однако при этом в выделенную область будет входить *все* пространство верхней полуплоскости, находящееся вне круга. Нам же необходимо



ограничиться областью, *прилегающей к центру* координат. Поэтому следует наложить еще одно ограничение,  $x < 1$  (рис. 3.7, *с*). Для того чтобы точка попала в область, расположенную *ниже* прямой, которая определяется уравнением  $y = x + 1$ , необходимо выполнение неравенства  $y < x + 1$  (рис. 3.7, *д*).

Тогда окончательное условие попадания точки с координатами  $(x, y)$  в заданную область будет определяться *одновременным* выполнением совокупности следующих неравенств:

$$\begin{aligned} y &> 0; \\ (x-1)^2 + (y-1)^2 &> 1; \\ x &< 1; \\ y &< x+1. \end{aligned}$$

Схема алгоритма сводится к простому ветвлению по двум направлениям, которую обучаемый легко может изобразить самостоятельно. Программная же реализация решения поставленной задачи будет иметь следующий вид:

```
{Цель: определить, попадает ли точка      }
{   с координатами (x, y) в заданную      }
{   область.                               }
{Переменные: x,y - координаты точки.      }
{Подпрограммы: нет.                        }
{Метод: определение области.               }
{Программист: Клинов Ю.Т.                 }
{Дата написания: 17 ноября 2005 г.        }
Program Area;
Var
    x, y : real;
Begin {Area}
    WriteLn('Введите координаты точки x и y');
    Read(x,y);
    {Эхо-печать}
    WriteLn('Точка с координатами x=',x:6:2,' и y=',y:6:2);
    Write('в заданную область ');
    if (y>0)and(Sqr(x-1)+Sqr(y-1)>1)and(x<1)and(y<x+1)
    then
        WriteLn('попадает')
    else
        WriteLn('не попадает')
    End. {Area}
```

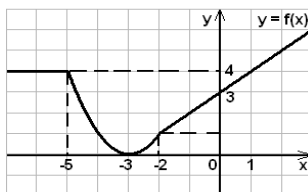
## ЗАДАНИЯ

Вариант 1. Составить алгоритм и программу вычисления функции  $f(x)$  при произвольном  $x$ :

$$f(x) = \begin{cases} 2x+4, & \text{если } -2 \leq x \leq -1; \\ 2x^2, & \text{если } -1 < x \leq 1; \\ -2x+2, & \text{если } 1 < x \leq 2. \end{cases}$$

При  $x < -2$  и  $x > 2$  функция  $f(x)$  не определена.

Вариант 2. Составить алгоритм и программу вычисления функции  $y = f(x)$ , график которой изображен на рисунке.

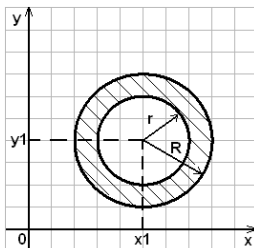


Вариант 3. Составить алгоритм и программу вычисления функции  $f(x)$  при произвольном  $x$ :

$$f(x) = \begin{cases} 1, & \text{если } x \leq -1; \\ 2x^2 - 1, & \text{если } -1 < x \leq 1; \\ x^2, & \text{если } 1 < x \leq 2. \end{cases}$$

При  $x > 2$  функция  $f(x)$  не определена.

Вариант 4. Составить алгоритм и программу, которые определяют, принадлежит ли точка с некоторыми координатами  $(x, y)$  к заштрихованной области, изображенной на рисунке.



Вариант 5. Составить функцию пользователя, определяющую максимальное значение из двух аргументов. Применяя функцию, определить максимальное значение из заданных  $x_1, x_2, x_3, x_4$ .

Вариант 6. Заданы длины трех отрезков  $x_1, x_2$  и  $x_3$ . Разработать алгоритм и программу, которые по результатам анализа вводимых длин отрезков выводят на экран дисплея одно из следующих сообщений:

- «треугольник построить нельзя»;
- «разносторонний треугольник»;
- «равнобедренный треугольник»;
- «равносторонний треугольник».

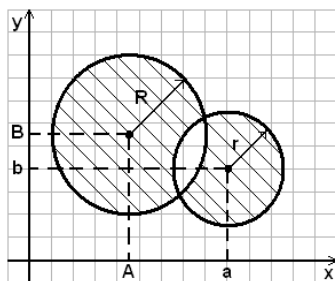
Вариант 7. Заданы следующие параметры геометрических фигур:

- $x, y, z$  – стороны треугольника;
- $a$  – стороны квадрата;
- $r$  – радиус круга.

Вывести на экран дисплея наименование и числовое значение площади фигуры с максимальной площадью.

Вариант 8. Для отрезков  $a, b$  и  $c$  определить, можно ли из них построить треугольник и является ли этот треугольник прямоугольным ( $a, b, c$  – целые числа).

Вариант 9. Составить алгоритм и программу, которые определяют, принадлежит ли точка с координатами  $(x, y)$  к заштрихованной области, изображенной на рисунке.



Вариант 10. Составить алгоритм и программу вычисления функции  $f(x)$  при произвольном  $x$ :

$$f(x) = \begin{cases} 2 \sin \frac{3x}{4}, & \text{если } x \leq \frac{\pi}{2}; \\ \frac{x}{2} \operatorname{tg} \frac{x+1}{3}, & \text{если } x > \frac{\pi}{2}. \end{cases}$$

В тех случаях, когда тангенс не имеет значений, вывести сообщение «функция не существует».

Вариант 11. Дана функция  $f(x) = \sqrt{x}$ . Проверьте, что для любых произвольно выбранных аргументов  $x_1 > 0$  и  $x_2 > 0$  имеет место неравенство  $f\left(\frac{x_1 + x_2}{2}\right) \geq \frac{f(x_1) + f(x_2)}{2}$ .

Вариант 12. Дана функция  $y = \frac{x}{4x^2 + 9x}$ . Найти значение функции  $y$  при произвольно заданных значениях аргумента  $x_1, x_2$  и  $x_3$ . На экран дисплея вывести минимальное значение функции.

Вариант 13. Составить алгоритм и программу вычисления функции  $f(x)$  при произвольном  $x$ :

$$f(x) = \begin{cases} \frac{8}{x}, & \text{если } x \leq -2; \\ x^3 + 4, & \text{если } -2 < x \leq 0 \\ \frac{4}{x^2 + 1}, & \text{если } x > 0. \end{cases}$$

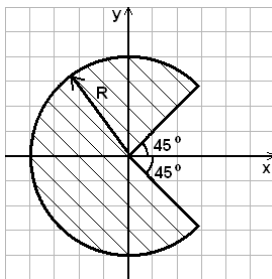
Вариант 14. Функция  $y = 2 + \sqrt{3 - 5x}$  убывает на  $(-\infty; 0,6)$ . Выберите произвольно три значения аргумента  $x_1 > x_2 > x_3$  и убедитесь, что  $f(x_1) > f(x_2) > f(x_3)$ .

Вариант 15. Определите знак следующих выражений:

$$\sin \frac{5\pi}{6} \cos \frac{5\pi}{7} \operatorname{tg} \frac{5\pi}{8} \operatorname{ctg} \frac{5\pi}{9};$$

$$\sin \frac{4\pi}{7} \cos \left(-\frac{4\pi}{9}\right) \operatorname{tg} \frac{4\pi}{9} \operatorname{ctg} \left(-\frac{4\pi}{11}\right).$$

Вариант 16. Составить алгоритм и программу, которые определяют, принадлежит ли точка с некоторыми координатами  $(x, y)$  к заштрихованной области, изображенной на рисунке.



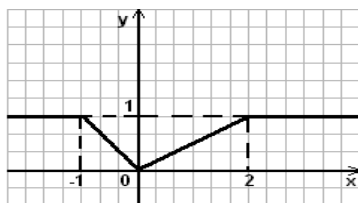
Вариант 17. Определить минимальное значение среди заданных  $x_1, x_2, x_3$  и  $x_4$ .

Вариант 18. Составить алгоритм и программу вычисления функции  $f(x)$  при произвольном  $x$ :

$$f(x) = \begin{cases} \frac{x}{2} \sqrt[3]{1+x}, & \text{если } 1 \leq x \leq 3; \\ \frac{\sin 2x}{2 + \cos 3x}, & \text{если } x > 3. \end{cases}$$

При  $x < 1$  функция  $f(x)$  не определена.

Вариант 19. Составить алгоритм и программу вычисления значения функции (см. рисунок) при произвольном значении аргумента  $x$ .



Вариант 20. Заданы стороны двух треугольников:  $(a_1, b_1, c_1)$  и  $(a_2, b_2, c_2)$ . Определить треугольник с максимальной площадью.

Вариант 21. Составить алгоритм и программу вычисления функции  $f(x)$  при произвольном  $x$ :

$$f(x) = \begin{cases} 3x + \sqrt{|x+1|}, & \text{если } x < -2; \\ \frac{\sin x + \cos x}{2x+1}, & \text{если } -2 \leq x \leq -1; \\ e^{-3x^2+2x-1}, & \text{если } -1 < x \leq 1. \end{cases}$$

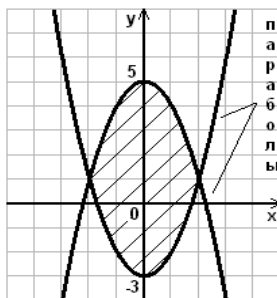
При  $x > 1$  функция  $f(x)$  не определена.

Вариант 22. Составить алгоритм и программу, которые производят анализ дискриминанта квадратного трехчлена  $ax^2 + bx + c$  и выводят на экран дисплея одно из следующих сообщений:

- «корней нет»;
- «корни одинаковые»;
- «корни разные».

Вариант 23. Составить алгоритм и программу определения минимального значения среди трех произвольно заданных  $x_1$ ,  $x_2$  и  $x_3$ .

Вариант 24. Составить алгоритм и программу, которые определяют, принадлежит ли точка с некоторыми координатами  $(x, y)$  к заштрихованной области, изображенной на рисунке.



Вариант 25. Составить алгоритм и программу вычисления функции  $f(x)$  при произвольном  $x$ :

$$f(x) = \begin{cases} \frac{(x+5)(x-6)}{(x-2)(x+3)}, & \text{если } x > 1,75; \\ \frac{x^2 + 2x + 4}{x^2 - 2x + 1}, & \text{если } x \leq 1,75. \end{cases}$$

Точки разрыва исключить.

## Тема 4. ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ РАЗВЕТВЛЯЮЩЕЙСЯ СТРУКТУРЫ С ИСПОЛЬЗОВАНИЕМ ОПЕРАТОРА ВАРИАНТА

### ПРИМЕР СОСТАВЛЕНИЯ ПРОГРАММЫ

Примером, иллюстрирующим применение оператора варианта, может служить программа выполнения следующего задания.

Составить программу, моделирующую работу простейшего калькулятора, выполняющего 4 арифметические операции. Для выбора операции предусмотреть вывод меню следующего вида:

Введите номер операции:

- сложение 1
- вычитание 2
- умножение 3
- деление 4

Схема алгоритма решения данной задачи приведена на рис. 4.1. Здесь вначале вводятся значения двух операндов  $a$  и  $b$ . Затем после вывода предусмотренного меню пользователь вводит номер требуемой операции  $k$  в диапазоне от 1 до 4. В зависимости от значения  $k$  происходит выбор оператора, производящего соответствующую операцию. В случае, когда значение  $k$  выходит за допустимый диапазон, выводится сообщение об ошибке. После корректного выполнения выбранной операции производится вывод результата.

Программа решения задачи приведена ниже.

```
{ ***** }  
{ Цель: моделирование работы калькулятора. }  
{ Переменные : a, b – операнды; }  
{ y – результат операции; }  
{ k – номер операции. }  
{ Программист: Поляков К. }  
{ Дата: 19 ноября 2005 г. }  
{ ***** }
```

**Program** Calculator;

**Var**

a, b, k:integer;  
y:real;

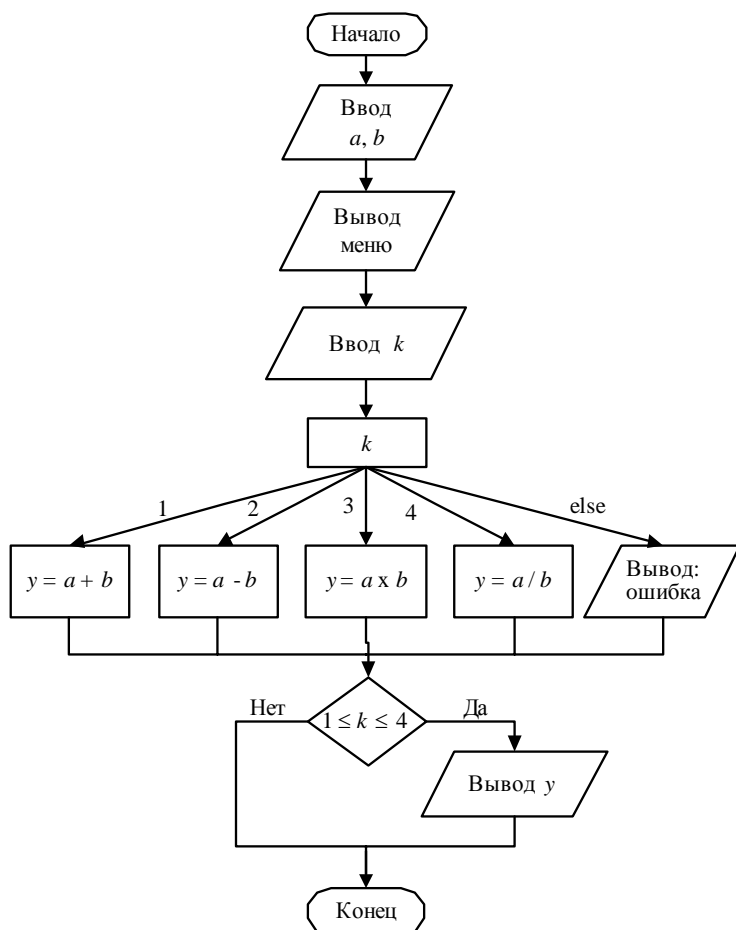


Рис. 4.1. Алгоритм моделирования калькулятора

```

Begin {Calculator}
  WriteLn('Введите два целых числа a, b');
  Read(a,b);
  WriteLn('Введите номер операции:'); {Меню}
  WriteLn('- сложение      1');
  WriteLn('- вычитание     2');
  WriteLn('- умножение     3');
  WriteLn('- деление      4');
  Read(k);
  case k of           {Оператор варианта}

```



```

1: begin
    y := a + b;
    Write('Сумма чисел:')
end;
2: begin
    y := a - b;
    Write('Разность чисел:')
end;
3: begin
    y := a * b;
    Write('Произведение чисел:')
end;
4: begin
    y := a/b;
    Write('Частное чисел:')
end
else
    WriteLn('Ошибочный номер операции')
end;
if (k >= 1) and (k <= 4)
then
    WriteLn(y:8:2)
End. {Calculator}

```

## ЗАДАНИЯ

Вариант 1. Разработать программу, которая вычисляет одну из функций в указанной точке.

Вид простейшего меню:

Введите номер формулы:

- 1)  $y = x$
- 2)  $y = \text{SQRT}(x)$
- 3)  $y = \text{EXP}(1/3 * \text{Ln}(x))$
- 4)  $y = \text{SQRT}(\text{SQRT}(x))$

Вариант 2. Разработать программу, которая выводит на экран дисплея значения функции  $\sin t$  при различных значениях аргумента  $t$ .

Код аргумента	1	2	3	4	5
Строка меню	$\sin(0)$	$\sin(\pi/6)$	$\sin(\pi/4)$	$\sin(\pi/3)$	$\sin(\pi/2)$
Выводимая строка	'0'	'1/2'	'SQRT(2)/2'	'SQRT(3)/2'	'1'

Вариант 3. Разработать программу, которая вычисляет значение одной из заданных функций в указанной точке.

Вид простейшего меню:

Введите номер функции:

1)  $y = \text{EXP}(1)$

2)  $y = \text{EXP}(2)$

3)  $y = \text{EXP}(3)$

4)  $y = \text{EXP}(4)$

Вариант 4. Разработать программу, которая выводит на экран дисплея значения функции  $\cos t$  при различных значениях аргумента  $t$ .

Код аргумента	1	2	3	4	5
Строка меню	$\cos(0)$	$\cos(\pi/6)$	$\cos(\pi/4)$	$\cos(\pi/3)$	$\cos(\pi/2)$
Выводимая строка	'1'	'SQRT(3)/2'	'SQRT(2)/2'	'1/2'	'0'

Вариант 5. Разработать программу, которая вычисляет значение одной из заданных функций в указанной точке.

Вид простейшего меню:

Введите номер функции:

1)  $y = \text{SQRT}(2)$

3)  $y = \text{SQRT}(3)$

1)  $y = \text{SQRT}(5)$

Вариант 6. Разработать программу, которая выводит на экран дисплея сообщение о значениях функции  $\text{tg } t$  при различных значениях  $t$ .

Код аргумента	0	6	4	3	2
Строка меню	$\text{tg}(0)$	$\text{tg}(\pi/6)$	$\text{tg}(\pi/4)$	$\text{tg}(\pi/3)$	$\text{tg}(\pi/2)$
Выводимая строка	'0'	'SQRT(3)/3'	'1'	'SQRT(3)'	'не сущ.'

Вариант 7. Разработать программу, которая вычисляет значение одной из заданных функций в указанной точке.

Вид простейшего меню:

Введите номер функции:

11)  $y = \text{SQRT}(11)$

12)  $y = \text{SQRT}(12)$

13)  $y = \text{SQRT}(13)$

14)  $y = \text{SQRT}(14)$

Вариант 8. Разработать программу, которая выводит на экран дисплея название ансамбля из некоторого числа исполнителей.

Вид простейшего меню:

Введите номер ансамбля:

1) 2 человека 'дуэт'

2) 3 человека 'трио'

- 3) 4 человека 'квартет'  
4) 5 человек 'квинтет'

Вариант 9. Разработать программу, которая выводит на экран дисплея сообщение о значениях функции  $\text{ctg } t$  при различных значениях  $t$ .

Код аргумента	0	6	4	3	2
Строка меню	$\text{ctg } (0)$	$\text{ctg } (\text{Pi}/6)$	$\text{ctg } (\text{Pi}/4)$	$\text{ctg } (\text{Pi}/3)$	$\text{ctg } (\text{Pi}/2)$
Выводимая строка	'не сущ.'	'SQRT(3)'	'1'	'SQRT(3)/3'	'0'

Вариант 10. Разработать программу, которая выводит на экран дисплея соотношение между старыми русскими мерами длины и современными.

Вид простейшего меню:

Введите номер меры длины:

- 1) миля '7,4676 км'  
2) верста '1,068 км'  
3) сажень '2,1336 м'  
4) аршин '0,7112 м'  
5) фунт '30,48 см'  
6) дюйм '2,54 см'

Вариант 11. Разработать программу, которая вычисляет значение одной из заданных функций в указанной точке.

Вид простейшего меню:

Введите номер функции:

- 1)  $y = x + 1$   
2)  $y = x * x + 1$   
3)  $y = x * x * x + 1$

Вариант 12. Разработать программу, которая выводит на экран дисплея сообщения о значениях функции при различных значениях аргумента.

Вид простейшего меню:

Введите номер функции:

- 2)  $y = \sin (\text{Pi}/2)$   
3)  $y = \sin (\text{Pi}/3)$   
4)  $y = \sin (\text{Pi}/4)$

Вариант 13. Разработать программу, которая вычисляет число байтов в килобайте, мегабайте и гигабайте (соответственно  $2^{10}$ ,  $2^{20}$ ,  $2^{30}$ ).

Вид простейшего меню:

Введите степень двойки:

- 10) килобайт
- 20) мегабайт
- 30) гигабайт

Вариант 14. Разработать программу, которая выводит на экран дисплея сообщения о значении кратной приставки, используемой в метрической системе.

Вид простейшего меню:

Введите номер приставки:

- 1) дека 1E+1
- 2) гекто 1E+2
- 3) кило 1E+3
- 6) мега 1E+6
- 9) гига 1E+9

Вариант 15. Разработать программу, которая вычисляет значения одной из заданных функций при указанных значениях.

Вид простейшего меню:

Введите номер функции:

- 1)  $y = x$
- 2)  $y = x * x$
- 3)  $y = x * x * x$
- 4)  $y = x * x * x * x$

Вариант 16. Разработать программу, которая выводит на экран дисплея сообщение о значении дольной приставки, используемой в метрической системе.

Вид простейшего меню:

Введите пункт меню:

Выводимое значение:

- |          |      |
|----------|------|
| 1) деци  | 1E-1 |
| 2) санти | 1E-2 |
| 3) милли | 1E-3 |
| 6) микро | 1E-6 |
| 9) нано  | 1E-9 |

Вариант 17. Разработать программу, которая выводит на экран дисплея значение единицы длины метрической системы, выбранной с помощью простейшего меню.

Вид простейшего меню:

Введите пункт меню:

Выводимое значение:

- |              |                    |
|--------------|--------------------|
| 1) декаметр  | декаметр = 10 м    |
| 2) гектометр | гектометр = 100 м  |
| 3) километр  | километр = 1 000 м |

4) мириаметр

мириаметр = 10 000 м

5) мегаметр

мегаметр = 1 000 000 м

Вариант 18. Разработать программу, которая вычисляет значение одной из заданных функций в указанной точке.

Вид простейшего меню:

Введите номер функции:

1)  $y = 1/x$

2)  $y = 1/(x*x)$

3)  $y = 1/(x*x*x)$

Вариант 19. Разработать программу, которая выводит на экран дисплея значения  $\cos t$  при различных значениях аргумента  $t$ .

Код аргумента	1	2	3	4	5
Строка меню	$\cos (Pi/10)$	$\cos (Pi/11)$	$\cos (Pi/13)$	$\cos (Pi/14)$	$\cos (Pi/15)$
Выводимая строка	$\cos Pi/10$	$\cos Pi/11$	$\cos Pi/13$	$\cos Pi/14$	$\cos Pi/15$

Вариант 20. Разработать программу, которая вычисляет значение одной из заданных функций в указанной точке.

Вид простейшего меню:

Введите номер функции:

1)  $y = \sin(x)$

2)  $y = \cos(x)$

3)  $y = \operatorname{tg}(x)$

4)  $y = \operatorname{ctg}(x)$

5)  $y = 1/\sin(x)$

Вариант 21. Разработать программу, которая выводит на экран дисплея значение одной из выбранных единиц веса.

Вид простейшего меню:

Введите номер пункта меню:

Выводимое значение:

1) дециграмм

дг = 0,1 г

2) миллиграмм

мг = 0,001 г

3) микрограмм

мкг = 0,000001 г

4) карат

к = 0,2 г

Вариант 22. Разработать программу, которая вычисляет значение одной из заданных функций в указанной точке.

Вид простейшего меню:

Введите номер функции:

1)  $y = x$

2)  $y = 2*x + 1$

3)  $y = 3*x^2 + 2*x + 1$

Вариант 23. Разработать программу, которая вычисляет значение стороны правильного  $n$ -угольника, вписанного в круг радиусом  $R$ .

Вид простейшего меню:

Введите номер пункта меню:

3)  $A3 = R * \text{SQRT}(3)$

4)  $A4 = R * \text{SQRT}(2)$

5)  $A5 = \text{SQRT}((5 - \text{SQRT}(5))/2)$

6)  $A6 = R$

Вариант 24. Разработать программу, которая вычисляет значение числа Архимеда  $\pi$  с помощью различных приближений, используемых в математике.

Вид простейшего меню:

Введите номер приближения:

1)  $Pi = \text{SQRT}(10)$  (Китай)

2)  $Pi = ((3 + 10/71) + (3 + 1/7))/2$  (Архимед)

3)  $Pi = 3,1415926535...$  (Транслятор Паскаля)

Вариант 25. Разработать программу, которая вычисляет значение одной из заданных функций в указанной точке.

Вид простейшего меню:

Введите номер функции:

1)  $y = \sin x$

2)  $y = \cos x$

3)  $y = \text{tg } x$

4)  $y = \text{ctg } x$

## КОНТРОЛЬНЫЕ ЗАДАНИЯ

При выполнении программы на ЭВМ исходные данные выбрать таким образом, чтобы получить результаты по каждой из ветвей программы.

Вариант 1. Составить программу вычисления значений функции

$$y = \begin{cases} x^2 - 3 - \sqrt[3]{\pi - x} & \text{при } x < 0; \\ (x^2 + 3)^2 - \sqrt{0,5\pi + x} & \text{при } 0 \leq x < 1; \\ x(x^2 + 3) + \ln(\pi + x) & \text{при } x \geq 1. \end{cases}$$

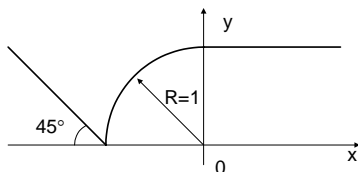
Вариант 2. Даны три числа  $a$ ,  $b$ ,  $c$ . Если хотя бы одно из них равно нулю с погрешностью  $\varepsilon = 0,01$ , то вычислить сумму этих чисел, в противном случае – их произведение.

Вариант 3. Составить программу вычисления значения функции

$$y = \begin{cases} \sin x + 2 & \text{при } k = 3 \text{ или } k = 5; \\ \cos x^2 & \text{при } k = 20; \\ \operatorname{tg} x + \sin^3 x & \text{при } k = 10 \text{ или } k = 15. \end{cases}$$

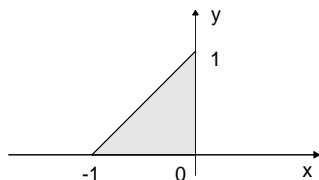
В остальных случаях значение  $y$  не определено.

Вариант 4. Составить программу вычисления значений функции, заданной графиком при произвольном значении  $x$ .



Вариант 5. Даны отрезки  $A, B, C$ . Составить программу для определения 1) возможности построения из этих отрезков треугольника, 2) равносторонности построенного треугольника и к печати соответствующего сообщения.

Вариант 6. Определить, принадлежит ли точка  $M(x, y)$  для заштрихованной области.



Вариант 7. Составить программу вычисления значений функции

$$y = \begin{cases} \sqrt{\left| \frac{\sin x + \operatorname{tg}^2 x}{3,5 \cos x} \right|} & \text{при } -\frac{\pi}{2} < x < \frac{\pi}{2}; \\ \frac{\cos(x/3)}{\sin x + \operatorname{tg}^2 x} & \text{при } \frac{\pi}{2} < x < \pi. \end{cases}$$

В остальных случаях значение  $y$  равно нулю.

Вариант 8. Даны три числа  $a, b, c$ . Составить программу нахождения значения минимального отклонения каждого из них от их среднего арифметического. Данные выбрать произвольно.

Вариант 9. Даны три целых положительных числа  $a, b, c$ . Найти остаток  $k$  от деления на 3 величины  $M$ :

$$M = \frac{a + b^2}{c};$$

вычислить значение функции

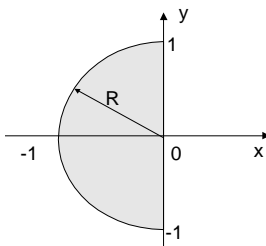
$$y = \begin{cases} e^{M+C} & \text{при } k = 1; \\ \ln(a/b) & \text{при } k = 0; \\ \sqrt{|(a+b)^2 + c|} & \text{при } k = 2. \end{cases}$$

Вариант 10. Составить программу, печатающую одно из следующих сообщений:

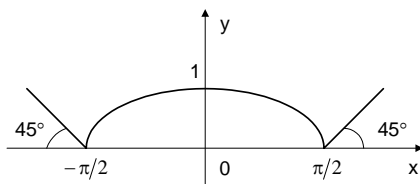
- «Корни действительные и равные»;
  - «Корни действительные и различные»;
  - «Корни мнимые»,
- в зависимости от вида корней квадратного уравнения

$$ax^2 + bx + c = 0. \text{ Принять } a, b, c \neq 0.$$

Вариант 11. Определить, принадлежит ли точка  $A(x, y)$  к заштрихованной области.



Вариант 12. Составить программу вычисления значения функции, заданной графиком, при произвольном значении  $x$ .





Вариант 13. Составить программу вычисления значения функции

$$y = \begin{cases} 1 - e^{-ax} \sin(ax + b) & \text{при } x > \pi; \\ 1 - e^{-ax} (ax + b) & \text{при } -\pi \leq x \leq \pi; \\ 1 - (e^{-ax} + e^{-bx}) & \text{при } x < -\pi, \end{cases}$$

где  $x = ab^2 - \sin b^2 + mb$ .

Вариант 14. Даны три числа  $A, B, C$ . Если все числа положительны, вычислить  $Z = A + B + C$ , если все отрицательны –  $Z = (A + B) * C$ , в противном случае  $Z = ABC$ .

Вариант 15. Составить программу преобразования вещественного вектора  $X(x_1, x_2, x_3)$  по правилу: если  $x_1 < x_2 < x_3$ , то всем компонентам присвоить значение наибольшей из них  $x_3$ , если  $x_1 > x_2 > x_3$ , то вектор оставить без изменения, в противном случае все компоненты заменить их квадратами.

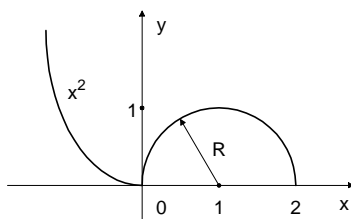
Вариант 16. Составить программу для вычисления функции

$$a = \cos \frac{\sin 1,25 + \operatorname{tg} x + x}{2 \operatorname{tg}^2 x},$$

где  $x = \begin{cases} \omega - \pi & \text{при } \pi/2 < \omega < 3\pi/4; \\ \omega + \pi & \text{при } -3\pi/4 < \omega < -\pi/2. \end{cases}$

В остальных случаях принять  $a = 0$ . Для просчета значение  $\omega$  выбрать произвольно.

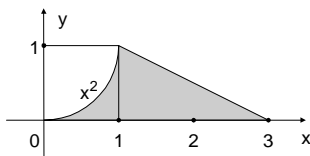
Вариант 17. Составить программу вычисления значений функции, заданной графиком. Значение  $x$  взять произвольно.



Вариант 18. Даны три числа  $a, b, c$ . Составить программу вычисления экспоненты числа, значение которого ближе всего к значению функции

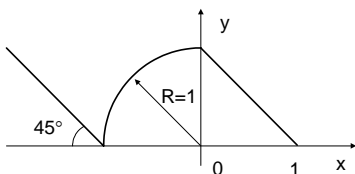
$$y = \frac{\operatorname{tg}(\pi * b) + \cos a}{\ln(c + 2)}.$$

Вариант 19. Определить принадлежность точки  $B(x, y)$  к заштрихованной области.



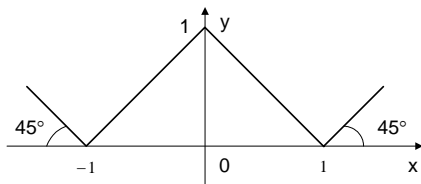
Вариант 20. Даны произвольные отрезки  $a$ ,  $b$ ,  $c$  и  $d$ . Составить программу определения возможности построения из них параллелограмма и печати соответствующего сообщения.

Вариант 21. Составить программу вычисления значения функции, заданной графиком при произвольном значении  $x$ .



Вариант 22. Даны три числа  $a$ ,  $b$ ,  $c$ . Составить программу определения, являются ли они пифагоровыми числами. Данные выбрать произвольно.

Вариант 23. Составить программу вычисления значения функции, заданной графиком, при произвольном значении  $x$ .



Вариант 24. Составить программу для вычисления функции

$$y = \ln \frac{\sin 0,25 + \operatorname{tg} x + x}{\operatorname{tg}^2 x},$$

определив область ее существования. Для просчета значение  $x$  выбрать произвольно.

Вариант 25. Даны произвольные отрезки  $a$ ,  $b$ ,  $c$  и  $d$ . Составить программу определения возможности построения из них геометрической фигуры и печати сообщения о ее типе.

## Тема 5. ТАБУЛИРОВАНИЕ ФУНКЦИЙ С ИСПОЛЬЗОВАНИЕМ ЦИКЛИЧЕСКИХ АЛГОРИТМОВ С ПОСТУСЛОВИЕМ

### ПРИМЕР СОСТАВЛЕНИЯ ПРОГРАММЫ

Пусть необходимо протабулировать функцию  $f(x)$ , т. е. получить таблицу ее значений, заданную графиком (рис. 5.1), на отрезке  $[x_0, x_n]$  с постоянным шагом  $h_x$ .

*Рис. 5.1. График функции*

Для вычисления значений функции необходимо представить ее в аналитическом виде:

$$f(x) = \begin{cases} 3, & \text{если } x < -1; \\ -x + 2, & \text{если } -1 \leq x < 2; \\ 1,5x - 3, & \text{если } x \geq 2 \end{cases}$$

и обеспечить изменение ее аргумента  $x$  от начального значения  $x_0$  до конечного  $x_n$  с шагом  $h_x$  по формуле  $x = x + h_x$ . Перед входом в цикл аргументу  $x$  необходимо присвоить его начальное значение  $x_0$ . Для обеспечения возможности повторного выполнения тела цикла или прекращения его выполнения текущее значение аргумента должно сравниваться с правой границей заданного интервала  $x_n$ . Во избежание возможных ошибок округления правую границу интервала целесообразно увеличить на величину полушага  $h_x/2$ . Для каждого значения аргумента вычисляется функция и производится вывод текущих значений. Тогда схема алгоритма будет иметь вид, представленный на рис. 5.2. Ниже приводится текст программы.

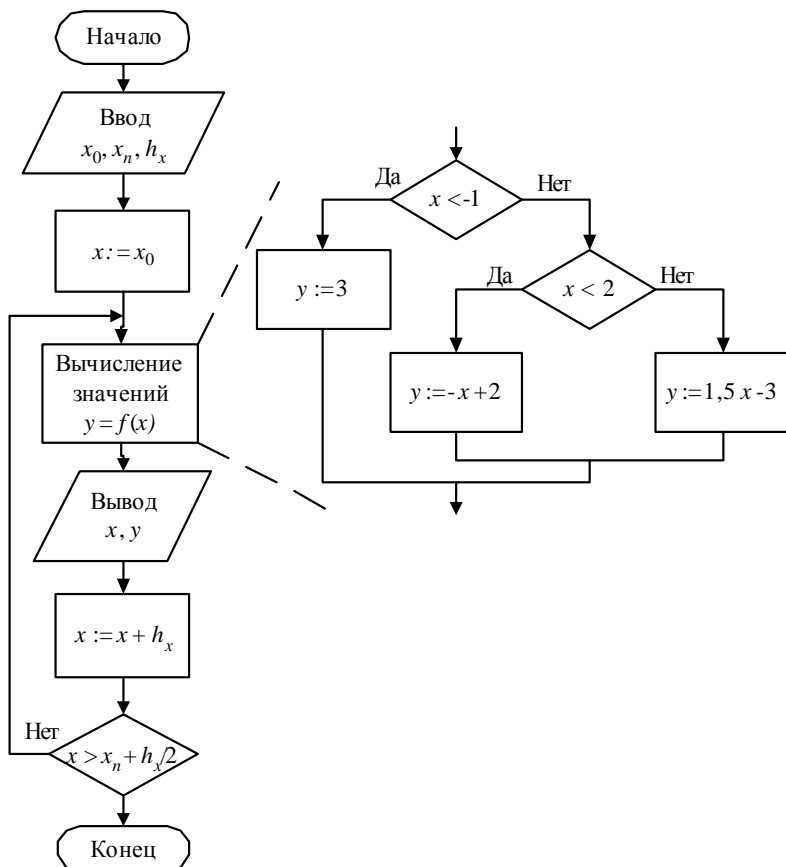


Рис. 5.2. Схема алгоритма табулирования функции

```

{ ***** }
{ Цель: табулирование функции y=F(x) с }
{ помощью оператора цикла с }
{ постусловием. }
{ Переменные: }
{ x - аргумент функции; }
{ x0, xn - начальное и конечное значения; }
{ hx - шаг изменения. }
{ Программист: Федоров Ф.Ф. }
{ Дата написания: 21 ноября 2005 г. }
{ ***** }
    
```

```
Program Tab;
Var
  hx,x,x0,xn,y:real;
Begin {Tab}
  {Ввод и эхо-печать исходных данных }
  Write('x0=');
  Read(x0);
  Write('hx=');
  Read(hx);
  Write('xn=');
  Read(xn);
  WriteLn('X0=',x0,' HX=',hx,' XN=',xn);
  {Табулирование функции }
  x:=x0;
  repeat {начало цикла}
    if x<=-1
    then
      y:=3
    else if x<2
    then
      y:=-x+2
    else
      y:=1.5*x-3;
    WriteLn('X=',x,' Y=',y);
    x:=x+hx
  until x>xn+hx/2 {конец цикла}
End. {Tab}
```

## ЗАДАНИЯ

Для всех вариантов заданий разработать алгоритм и программу, имеющих структуру цикла с постусловием и осуществляющих табулирование заданной функции при изменении аргумента  $x$  от начального значения  $x_0$  до конечного значения  $x_n$  с постоянным шагом  $h_x$ , т. е.  $x = x_0(h_x)x_n$ .

В каждом из вариантов дана сложная функция  $f(x)$ .

Вариант 1.

$$f(x) = \begin{cases} 2x+4, & \text{если } -2 \leq x \leq -1; \\ 2x^2, & \text{если } -1 < x \leq 1; \\ -2x+2, & \text{если } 1 < x \leq 2. \end{cases}$$

При  $x < -2$  и  $x > 2$  функция  $f(x)$  не определена.

Вариант 2.

$$f(x) = \begin{cases} 1, & \text{если } x \leq -1; \\ 2x^2 - 1, & \text{если } -1 < x \leq 1; \\ x^2, & \text{если } 1 < x \leq 2. \end{cases}$$

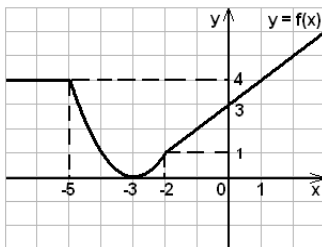
При  $x > 2$  функция  $f(x)$  не определена.

Вариант 3.

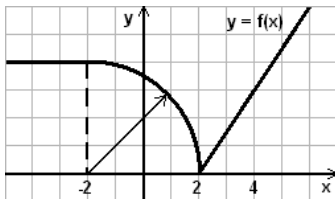
$$f(x) = \begin{cases} 2 \sin \frac{3x}{4}, & \text{если } x \leq \frac{\pi}{2}; \\ \frac{x}{2} \operatorname{tg} \frac{x+1}{3}, & \text{если } x > \frac{\pi}{2}. \end{cases}$$

В тех случаях, когда тангенс не имеет значений, вывести сообщение «функция не существует».

Вариант 4.



Вариант 5.



Вариант 6.

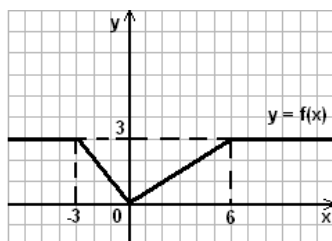
$$f(x) = \begin{cases} \frac{8}{x}, & \text{если } x \leq -2; \\ x^3 + 4, & \text{если } -2 < x \leq 0; \\ \frac{4}{x^2 + 1}, & \text{если } x > 0. \end{cases}$$

Вариант 7.

$$f(x) = \begin{cases} \frac{x}{2} \sqrt[3]{1+x}, & \text{если } 1 \leq x \leq 3; \\ \frac{\sin 2x}{2 + \cos 3x}, & \text{если } x > 3. \end{cases}$$

При  $x < 1$  функция  $f(x)$  не определена.

Вариант 8.



Вариант 9.

$$f(x) = \begin{cases} 3x + \sqrt{|x+1|}, & \text{если } x < -2; \\ \frac{\sin x + \cos x}{2x+1}, & \text{если } -2 \leq x < -1,5; \\ e^{-3x^2+2x-1}, & \text{если } -1 < x \leq 1. \end{cases}$$

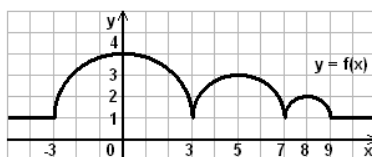
При других значениях  $x$  функция  $f(x)$  не определена.

Вариант 10.

$$f(x) = \begin{cases} \frac{(x+5)(x-6)}{(x-2)(x+3)}, & \text{если } x > 1,75; \\ \frac{x^2 + 2x + 4}{x^2 - 2x + 1}, & \text{если } x \leq 1,75. \end{cases}$$

Точки разрыва исключить.

Вариант 11.



Вариант 12.

$$f(x) = \begin{cases} \frac{2x+1}{x}, & \text{если } x < -1; \\ 2-x^2, & \text{если } -1 \leq x < 2; \\ -3, & \text{если } x \geq 2. \end{cases}$$

Вариант 13.

$$f(x) = \begin{cases} 2 \sin \frac{x}{2}, & \text{если } x < \frac{\pi}{2}; \\ \frac{x}{2} \cos 2x, & \text{если } \frac{\pi}{2} \leq x \leq \pi; \\ \sin x + \cos x, & \text{если } x > \pi. \end{cases}$$

Вариант 14.

$$f(x) = \begin{cases} \frac{2|x|-1}{x-3}, & \text{если } x < 2; \\ \frac{3x+5}{1+2x}, & \text{если } 2 \leq x < 4. \end{cases}$$

При  $x \geq 4$  функция  $f(x)$  не определена.

Вариант 15.

$$f(x) = \begin{cases} 2 \sin^2 \left( \frac{\pi}{4} - \frac{x}{2} \right), & \text{если } -\frac{\pi}{2} \leq x \leq -\frac{\pi}{4}; \\ \frac{1 - \cos 2x}{\sin 2x}, & \text{если } -\frac{\pi}{4} < x < -\frac{\pi}{8}; \\ \frac{x}{2 + \cos x}, & \text{если } -\frac{\pi}{8} \leq x \leq 0. \end{cases}$$

При других значениях  $x$  функция  $f(x)$  не определена.



Вариант 16.



Вариант 17.

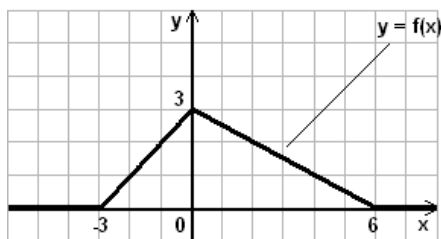
$$f(x) = \begin{cases} |x|, & \text{если } |x| \leq 1; \\ x^2, & \text{если } 1 < |x| \leq 2; \\ 4 & \text{при других } x. \end{cases}$$

Вариант 18.

$$f(x) = \begin{cases} x^2, & \text{если } x < 1; \\ \sqrt{x+1}, & \text{если } 1 \leq x \leq 3; \\ \sin x, & \text{если } 3 < x < 5. \end{cases}$$

При  $x \geq 5$  функция  $f(x)$  не определена.

Вариант 19.



Вариант 20.

$$f(x) = \begin{cases} x, & \text{если } 0 \leq x \leq 1; \\ x^2, & \text{если } 1 < x \leq 2; \\ \sqrt{x}, & \text{если } 2 < x \leq 3. \end{cases}$$

При других значениях  $x$  функция  $f(x)$  не определена.

Вариант 21.

$$f(x) = \frac{3x^3 + 2x^2 + 4x - 5}{(x^2 - 9)(x + 2)(x - 1)}.$$

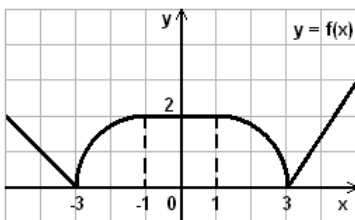
Точки разрыва функции  $f(x)$  исключить.

Вариант 22.

$$f(x) = \begin{cases} \sin \frac{x}{2}, & \text{если } -\frac{\pi}{2} < x < \frac{\pi}{2}; \\ \frac{x}{8} \cos 3x, & \text{если } \frac{\pi}{2} \leq x \leq \pi; \\ \sin x + \cos x, & \text{если } \pi < x < \frac{3}{2}\pi. \end{cases}$$

При других значениях  $x$  функция  $f(x)$  не определена.

Вариант 23.



Вариант 24.

$$f(x) = \begin{cases} \frac{x}{2} \sqrt[4]{1+x}, & \text{если } 0 \leq x \leq 1; \\ \frac{\sin 2x}{\sin x + \sin 3x}, & \text{если } x > 1. \end{cases}$$

При  $x < 0$  функция  $f(x)$  не определена.

Вариант 25.

$$f(x) = \begin{cases} 1 - e^{-ax} \sin(ax + b) & \text{при } x > \pi; \\ 1 - e^{-ax} (ax + b) & \text{при } -\pi \leq x \leq \pi; \\ 1 - (e^{-ax} + e^{-bx}) & \text{при } x < -\pi, \end{cases}$$

где  $x = ab^2 - \sin b^2 + mb$ .

## Тема 6. ПРОЕКТИРОВАНИЕ АЛГОРИТМОВ И ПРОГРАММ С ИСПОЛЬЗОВАНИЕМ ЦИКЛА С ПРЕДУСЛОВИЕМ. ОБРАБОТКА ФУНКЦИЙ

### ПРИМЕР СОСТАВЛЕНИЯ ПРОГРАММЫ

Пусть значения функций  $f_1(x) = 2 + \frac{\sin 3x}{\cos 2x + 1}$  и  $f_2(x) = \cos \frac{x}{3} \sin \frac{x}{4}$  заданы в точках  $x = x_0(h_x)x_n$ . Определить, при каком  $x$  расстояние между  $f_1(x)$  и  $f_2(x)$  максимально.

Для реализации данного алгоритма предположим, что максимальное расстояние между функциями  $d_{\max}$  имеет место в начальной точке  $x_0$ . Затем, изменяя в цикле значение аргумента  $x$ , будем определять текущую разность функций  $d$  и сравнивать ее с  $d_{\max}$ . Если текущее значение  $d$  окажется больше  $d_{\max}$ , то именно оно и будет принято в качестве максимального. После завершения цикла выведем значение максимальной разности и ее координату  $x_{\max}$  (рис. 6.1).

Текст программы будет иметь следующий вид:

```
{*****}  
{Цель: нахождение максимального отклонения}  
{  двух функций}  
{Переменные:  
{  x –аргумент функции;  
{  x0, xn – начальное и конечное значения;  
{  hx – шаг изменения;  
{  dmax, d – максимальное и текущее  
{    отклонения функций;  
{  xmax – положение максимума.  
{Подпрограммы: f1, f2 – заданные функции}  
{Программист: Столбов В.Ф.  
{Дата написания: 28 ноября 2005 г.  
{*****}
```

**Program** Lab6;

**Var**

d, dmax, hx, x, x0, xmax, xn:real;

**Function** f1(x:real):real;

**begin**{f1}

f1 := 2 + sin(3\*x)/(cos(2\*x) + 1)

**end**;{f1}

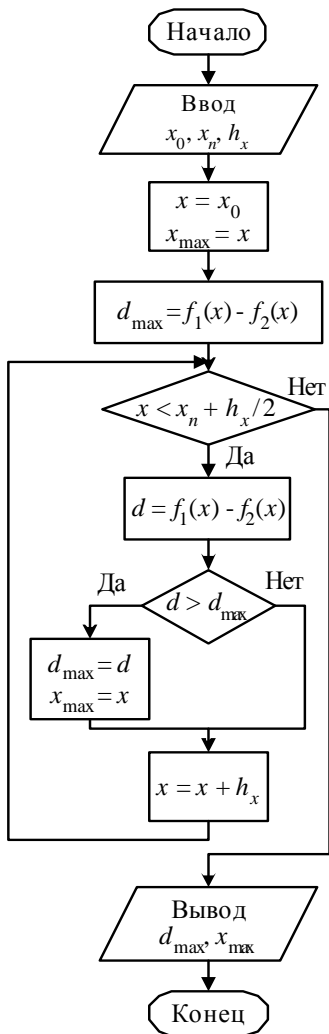


Рис. 6.1. Алгоритм поиска максимального отклонения

```

Function f2(x:real):real;
begin{f2}
    f2 := cos(x/3)*sin(x/4)
end;{f2}
Begin {Lab6}
{Ввод и эхо-печать исходных данных}
    
```

```

WriteLn('Введите пределы и шаг изменения x');
Read(x0, xn, hx);
WriteLn('x0=', x0:6:2, ' xn=', xn:6:2, ' hx=', hx:5:2);
{Инициализация переменных}
x := x0;
xmax := x;
dmax := f1(x) - f2(x);
while x < xn + hx/2 do {Начало цикла}
begin
    {Поиск максимального отклонения}
    d := f1(x) - f2(x);
    if d > dmax
    then
        begin
            dmax := d;
            xmax := x
        end;
    x := x + hx
end; {Конец цикла}
WriteLn('Максимальная разность функций ', dmax,
' в точке x=', xmax:6:2)
End. {Lab6}

```

## ЗАДАНИЯ

Вариант 1. Значения функции  $f(x) = 2\sin\frac{3}{4}x$  определены при различных значениях аргумента  $x = x_0(h_x)x_n$ . Определить координаты точки  $(x; f(x))$ , которая ближе всего расположена к началу координат, т. е. к точке  $(0; 0)$ .

Вариант 2. Значения функций  $f_1(x) = 2 + \sin 3x$  и  $f_2(x) = \cos\frac{x}{3}$  определены в точках  $x = x_0(h_x)x_n$ . Определить, при каком  $x$  расстояние между  $f_1(x)$  и  $f_2(x)$  минимально.

Вариант 3. Значения функции  $f(x) = \frac{1}{2} + \sin\frac{x}{2}$  определены при  $x = x_0(h_x)x_n$ . Определить, сколько значений функции  $f(x)$  при заданных  $x$  являются отрицательными и сколько положительными.

Вариант 4. Значения функции  $f(x) = \sin(x) + \cos(x)$  определены при  $x = x_0(h_x)x_n$ . Определить, сколько значений функции  $f(x)$  принадлежит к окружности радиусом  $R$ , центр которой расположен в точке  $(a; b)$ .

Вариант 5. Значения функции  $f(x) = 3x^2 + 2x + 1$  определены при  $x = x_0(h_x)x_n$ . Определить, сколько точек функции  $f(x)$  при заданных  $x$  расположены выше прямой  $y = x + 3$ .

Вариант 6. Значения функции  $f(x) = x^2 + 2x - 3$  определены при  $x = x_0(h_x)x_n$ . Определить значение функции, которое ближе всего расположено к точке с координатами  $(x_1; y_1)$ . Значения  $x_0, h_x, x_n, x_1, y_1$  задать произвольно.

Вариант 7. Значения функций  $f_1(x) = 3 + \sin^2 \frac{x}{2}$  и  $f_2(x) = 1 + \cos^2 \frac{x}{3}$  определены в точках  $x = x_0(h_x)x_n$ . Определить, при каком  $x$  расстояние между  $f_1(x)$  и  $f_2(x)$  максимально.

Вариант 8. Значения функции  $f(x) = 0,348 + \cos \frac{x}{4}$  определены при  $x = x_0(h_x)x_n$ . Определить, сколько значений функции  $f(x)$  при заданных  $x$  расположены выше и ниже прямой  $y = 0,555$ .

Вариант 9. Значения функций  $f_1(x) = 2,5 + \sin 4x$  и  $f_2(x) = 2 \cos \frac{x}{3}$  определены в точках  $x = x_0(h_x)x_n$ . Найти среднее значение расстояния между  $f_1(x)$  и  $f_2(x)$  в заданных точках.

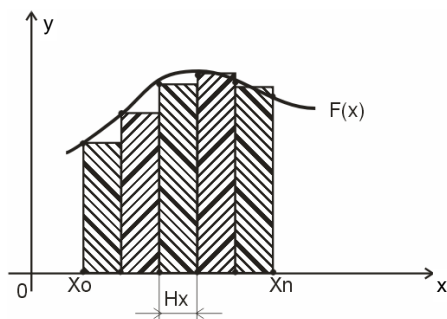
Вариант 10. Значения функции  $f(x) = \sin x$  заданы в точках  $x = x_0(h_x)x_n$ . Определить количество точек функции  $f(x)$ , расположенных между прямыми  $y = 0,5$  и  $y = -0,5$ .

Вариант 11. Значения функции  $f(x) = \cos x$  заданы в точках  $x = x_0(h_x)x_n$ . Определить количество точек функции  $f(x)$ , расположенных выше прямой  $y = 0,5$  и ниже прямой  $y = -0,5$ .

Вариант 12. Значения функции  $f(x) = 2 \cos \frac{5}{4}x$  заданы в точках  $x = x_0(h_x)x_n$ . Определить среднее значение положительных значений функции  $f(x)$ .

Вариант 13. Значения функции  $f(x) = x^2 + 2x + 2$  заданы в точках  $x = x_0(h_x)x_n$ . Определить количество точек функции  $f(x)$ , расположенных от вершины параболы на расстоянии не менее  $a$ . Значения  $x_0, h_x, x_n, a$  задать произвольно.

Вариант 14. Определить площадь заштрихованной области для функции  $f(x) = \frac{x}{2} \cos 3x$  при заданных произвольных  $x_0, h_x, x_n$ .



Вариант 15. Значения функций  $f_1(x) = 3x^3 + 2x^2 + x + 5$  и  $f_2(x) = 2x^3 - 4x^2 + 2x - 5$  определены в точках  $x = x_0(h_x)x_n$ . Определить минимальное и максимальное расстояние между функциями  $f_1(x)$  и  $f_2(x)$  в заданных точках.

Вариант 16. Значения функций  $f_1(x) = \frac{x+1}{2} \cos \frac{x}{3}$  и  $f_2(x) = \frac{x+2}{3} \sin \frac{x}{4}$  определены в точках  $x = x_0(h_x)x_n$ . Определить расстояние между максимальными значениями функций  $f_1(x)$  и  $f_2(x)$ .

Вариант 17. Значения функции  $f(x) = 4x^3 - 3x^2 + 2x - 1$  определены в точках  $x = x_0(h_x)x_n$ . Определить расстояние между максимальным и минимальным значениями функции  $f(x)$ .

Вариант 18. Значения функций  $f_1(x) = 4x^2 + x - 3$  и  $f_2(x) = 8x^3 + x^2 + 2x - 4$  определены в точках  $x = x_0(h_x)x_n$ . Определить расстояние между минимальными значениями функций  $f_1(x)$  и  $f_2(x)$ .

Вариант 19. Значения функции  $f(x) = 2 \sin\left(\frac{x}{10} + 0,5\right)$  определены в точках  $x = x_0(h_x)x_n$ . Определить среднее значение для отрицательных значений функции  $f(x)$ .

Вариант 20. Значения функции  $f(x) = \sin(x) + \cos(x)$  определены при  $x = x_0(h_x)x_n$ . Определить минимальное значение среди положительных значений функции  $f(x)$ .

Вариант 21. Значения функции  $f(x) = \frac{\sin 3x + \cos 2x}{x^2 + 1}$  заданы в точках  $x = x_0(h_x)x_n$ . Определить число точек функции  $f(x)$ , значения которых превосходят величину  $b = 2,345$ .

Вариант 22. Значения функции  $f(x) = x^4 + x^2 + x - 3$  заданы в точках  $x = x_0(h_x)x_n$ . Определить среднее значение для заданных значений функции  $f(x)$ .

Вариант 23. Значения функций  $f_1(x) = \sin(3x + 1)$ ,  $f_2(x) = \frac{1}{2} \cos\left(\frac{x}{2} + \frac{1}{3}\right)$  и  $f_3(x) = x^2 + x + 1$  заданы в точках  $x = x_0(h_x)x_n$ . Определить максимальные значения функций  $f_1(x)$ ,  $f_2(x)$  и  $f_3(x)$ .

Вариант 24. Значения функции  $f(x) = 3x^4 + 2x^3 - 3x^2 - 3$  заданы в точках  $x = x_0(h_x)x_n$ . Определить, каких значений функции  $f(x)$  больше, положительных или отрицательных.

Вариант 25. Значения функции  $f(x) = 3 \sin \frac{x}{3} + 2 \cos \frac{x}{2}$  заданы в точках  $x = x_0(h_x)x_n$ . Определить максимальное и минимальное значения функции  $f(x)$ .



## Тема 7. ОБРАБОТКА ЧИСЛОВЫХ ДАННЫХ

### ПРИМЕР СОСТАВЛЕНИЯ ПРОГРАММЫ

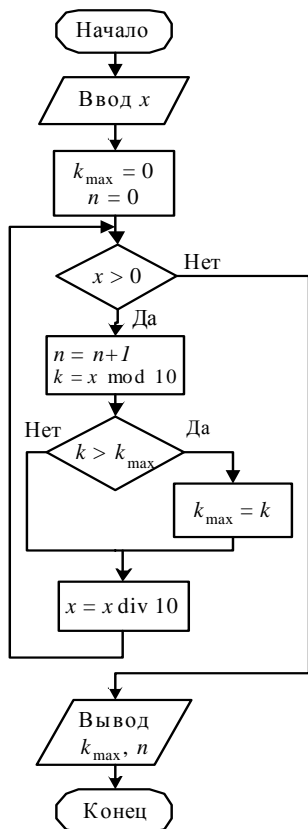


Рис. 7.1. Схема алгоритма обработки числа

Пусть задано некоторое целое натуральное число  $x$  типа **longint**. Необходимо определить значение максимальной цифры и общее количество цифр числа.

Для решения данной задачи необходимо последовательно выделять цифры заданного числа и сравнивать их между собой до тех пор, пока не будет получена самая старшая цифра. При этом на каждом шаге алгоритма должен увеличиваться счетчик количества цифр числа  $n$ . Каждая выделенная цифра сравнивается с предыдущими и выбирается максимальная из них. Схема алгоритма представлена на рис. 7.1.

Здесь для определения цифры младшего разряда  $k$  используется операция получения остатка от целочисленного деления на основание 10 десятичной системы счисления – операция **mod**. Затем с помощью операции целочисленного деления **div** отбрасывается младшая цифра, и исходное число  $x$  уменьшается в 10 раз. После чего осуществляется переход на выделение цифры следующего разряда. И так до тех пор, пока не будут выделены все цифры числа и значение  $x$  не станет равным нулю.

Ниже приведен текст программы.

```

{ ***** }
{ Цель:   нахождение максимальной цифры }
{         и количества разрядов числа.   }
{ Переменные: }
{  x - заданное число; }
{  k - текущая цифра числа; }

```

```

{ kmax – максимальная цифра; }
{ n – количество разрядов числа. }
{Подпрограммы: нет }
{Программист: Белов М.А. }
{Дата написания: 05 декабря 2005 г. }
{***** }

```

**Program** Number;

**Var**

  x : **longint**;  
  k, kmax : 0..9;  
  n : **integer**;

**Begin** {Number}

{Ввод и эхо-печать исходных данных}  
  WriteLn('Введите целое число x');  
  Read(x);  
  WriteLn('В числе', x:11);

{Инициализация переменных}

  kmax := 0;  
  n := 0;

**while** x > 0 **do** {Начало цикла}

**begin**

      {Выделение младшего разряда числа}

      n := n + 1;

      k := x **mod** 10;

      {Поиск максимальной цифры}

**if** k > kmax

**then**

        kmax := k;

      {Уменьшение исходного числа}

      x := x **div** 10

**end**; {Конец цикла}

  WriteLn('Максимальная цифра ', kmax);

  WriteLn('Из ', n, ' разрядов')

**End.** {Number}

## ЗАДАНИЯ

Дано  $n$ -значное целое число типа **longint**. Произвести с ним действия согласно номеру варианта.

Вариант 1. Подсчитать количество четных цифр в числе.

Вариант 2. Подсчитать количество нечетных цифр в числе.

Вариант 3. Подсчитать количество цифр в числе, делящихся на 3 без остатка.

Вариант 4. Подсчитать в числе количество цифр, не превосходящих заданной цифры, введенной с клавиатуры.

Вариант 5. В исходном числе, содержащем нечетное количество цифр, вывести центральную цифру.

Вариант 6. Вывести введенное число в зеркальном отображении.

Вариант 7. Вывести только те цифры числа, которые не делятся на 3 без остатка.

Вариант 8. Подсчитать количество цифр, значения которых больше заданной.

Вариант 9. Вывести на экран только нечетные цифры в порядке их расположения в числе.

Вариант 10. Вывести на экран только повторяющиеся цифры и общее количество их повторений.

Вариант 11. Вывести наименьшую цифру числа.

Вариант 12. Вывести наибольшую цифру числа.

Вариант 13. Вывести цифры, лежащие в интервале  $[a, b]$ .

Вариант 14. Вывести те цифры, которые при делении на  $p$  дают в остатке  $q$  ( $p > q \geq 0$ ).

Вариант 15. Заменить нулями все четные цифры.

Вариант 16. Вывести только те цифры, значения которых совпадают с местом (разрядом) этой цифры в числе. Для упрощения задачи вводите каждую цифру только один раз.

Вариант 17. Вывести только цифры самого младшего и самого старшего разряда.

Вариант 18. Вывести сумму всех четных цифр числа.

Вариант 19. Вывести сумму всех нечетных цифр числа.

Вариант 20. Вывести сумму наибольшей и наименьшей цифр числа.

Вариант 21. Вывести сумму всех цифр числа.

Вариант 22. Вывести разницу между цифрами старшего и младшего разряда числа.

Вариант 23. Если цифра старшего разряда больше цифры младшего разряда, то исходное число увеличить вдвое, в противном случае вычесть из него число, равное цифре старшего разряда.

Вариант 24. Если цифра младшего разряда четная, то вывести все четные цифры числа, в противном случае – все нечетные.

Вариант 25. Вывести разницу между максимальной и минимальной цифрами числа.

## Тема 8. ВЫЧИСЛЕНИЕ ЧИСЛОВЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ С ПОМОЩЬЮ ЦИКЛА С ПАРАМЕТРОМ

### ПРИМЕР СОСТАВЛЕНИЯ ПРОГРАММЫ

Пусть необходимо вычислить первые 15 членов ряда Фибоначчи, определяемые рекуррентным соотношением

$$f_{i+2} = f_{i+1} + f_i \text{ при } f_0 = 0, f_1 = 1,$$

и найти их сумму.

Особенностью решения данной задачи является необходимость сдвига получаемых значений вправо, для возможности вычисления каждого нового значения, и накопление суммы по рекуррентному соотношению

$$S_{i+2} = S_{i+1} + f_{i+2}.$$

Начальное значение суммы должно быть принято равным  $f_0 + f_1$ . Схема алгоритма представлена на рис. 8.1, а ниже дается текст программы.

```

{*****}
{Цель:  вычисления последовательности }
{      чисел Фибоначчи и их суммы.    }
{Переменные: f0, f1, f2 - числа Фибоначчи; }
{ n - количество чисел; S - их сумма; }
{ i - параметр цикла. }
{Подпрограммы: нет. }
{Программист: Светлов Н.А. }
{Дата написания: 12 декабря 2005 г. }
{*****}

```

**Program Fib;**

**Var**

i, n: **integer**;  
f0, f1, f2, S : **integer**;

**Begin {Fib}**

{Ввод и эхо-печать исходных данных}

WriteLn('Введите число членов последовательности');

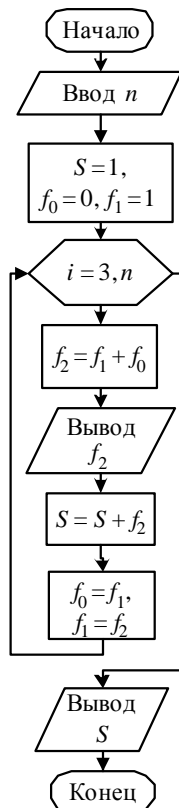


Рис. 8.1. Алгоритм  
вычисления  
последовательности

```
Read(n);
WriteLn('Сформировано следующих', n:3, ' чисел:');
{Инициализация переменных}
S:=1;
f0:=0;
f1:=1;
Write(f0:4, f1:4);
for i := 3 to n do    {Начало цикла}
begin
  {Формирование и суммирование чисел Фибоначчи}
  f2 := f1 + f0;
  Write(f2:4);
  S := S + f2;
  f0 := f1;
  f1 := f2
end;                  {Конец цикла}
WriteLn;
WriteLn('Их сумма равна:', S:5)
End. {Fib}
```

## ЗАДАНИЯ

Вариант 1. Составить алгоритм и программу вычисления и вывода первых 25 членов последовательности и их суммы. Последовательность задана формулой  $n$ -го члена  $a_n = 1/n^2$ .

Вариант 2. Последовательность задана рекуррентной формулой  $b_{n+2} = b_{n+1} + b_n$  и условиями  $b_1 = 1, b_2 = 3$ . Разработать алгоритм и программу вычисления и вывода на экран дисплея первых 20 членов последовательности.

Вариант 3. Последовательность задана рекуррентной формулой  $a_{n+1} = \sin\left(\frac{\pi}{2} a_n\right)$  и условием  $a_1 = 0,5$ . Разработать алгоритм и программу вычисления и вывода первых 15 членов последовательности и их суммы.

Вариант 4. Составить алгоритм и программу вычисления и вывода первых восьми членов последовательности, которая задана рекуррентной формулой  $a_{n+2} = a_n^2 + a_{n+1}$  и условиями  $a_1 = 2, a_2 = 3$ .

Вариант 5. Разработать алгоритм и программу вычисления и вывода первых 20 членов последовательности  $1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \dots$  и их суммы.

Вариант 6. Разработать алгоритм и программу вычисления суммы первых 20 членов последовательности  $S = 1 + \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{3}} + \dots + \frac{1}{\sqrt{n}}$ .

Вариант 7. Последовательность задана рекуррентной формулой  $a_{n+1} = \frac{1}{2} + \sqrt{a_n - a_n^2}$  и условием  $a_1 = \frac{1}{2}$ . Разработать алгоритм и программу вычисления и вывода на экран дисплея первых 20 членов последовательности.

Вариант 8. Разработать алгоритм и программу вычисления и вывода первых 15 членов последовательности, которая задана формулой

$$a_n = \begin{cases} n^2 + 1, & \text{если } n \text{ четное;} \\ n^2 + 2n, & \text{если } n \text{ нечетное.} \end{cases}$$

Вариант 9. Разработать алгоритм и программу вычисления и вывода первых 12 членов последовательности, заданной рекуррентной формулой

$$a_{n+2} = \frac{1 - a_{n+1}}{1 + a_n} \text{ и условиями } a_1 = 1, a_2 = 2.$$

Вариант 10. Разработать алгоритм и программу вычисления и вывода первых 15 членов и их суммы последовательности  $a_n = \frac{n}{n+1}$ ,  $n = 1, 2, \dots$

Вариант 11. Разработать алгоритм и программу вычисления и вывода первых 20 членов последовательности  $a_{n+1} = \cos\left(\frac{\pi}{3} \cdot a_n\right)$ ,  $a_1 = 1$ .

Вариант 12. Разработать алгоритм и программу вычисления и вывода первых 25 членов и их суммы последовательности, которая задана рекуррентным соотношением  $a_{n+1} = \frac{1}{2 + a_n}$  и условием  $a_1 = 1$ .

Вариант 13. Дана последовательность  $a_{n+2} = \frac{a_{n+1}}{a_n}$ ;  $a_1 = 1, a_2 = 2$ . Составить алгоритм и программу вычисления первых 15 членов последовательности.

Вариант 14. Дана последовательность  $a_n = \sqrt{n^2 + 1}$ ,  $n = 1, 2, \dots$ . Составить алгоритм и программу вычисления и вывода членов последовательности начиная с  $a_{10}$  и заканчивая  $a_{20}$ .

Вариант 15. Разработать алгоритм и программу вычисления и вывода первых 15 членов и их суммы последовательности  $a_n = \sqrt[n]{1+b}$  при произвольном  $b$  ( $b > 0$ ).

Вариант 16. Разработать алгоритм и программу вычисления и вывода на экран дисплея первых 15 членов последовательности, которая задана формулой

$$a_n = \left(\frac{1}{2}\right)^n + \left(\frac{1}{3}\right)^n; \quad n = 1, 2, \dots$$

Вариант 17. Разработать алгоритм и программу вычисления и вывода на экран дисплея членов последовательности с 10-го по 25-й. Последовательность задана формулой

$$a_n = \frac{1}{n} \sin\left(\frac{\pi}{4} \cdot n\right) + \frac{1}{n^2} \cos\left(\frac{\pi}{4} \cdot n\right); \quad n = 1, 2, \dots$$

Вариант 18. Последовательность задана рекуррентным соотношением  $a_{n+1} = \frac{a_n}{n+1}$  и условием  $a_1 = 1$ . Разработать алгоритм и программу вычисления суммы первых 15 членов последовательности.

Вариант 19. Разработать алгоритм и программу вычисления и вывода на экран дисплея первых 15 членов последовательности  $a_n = \frac{2n+1}{n-1}$ ;  $n = 2, 3, \dots$

Вариант 20. Разработать алгоритм и программу вычисления и вывода на экран дисплея первых 15 членов последовательности и их суммы. Последовательность задана формулой  $a_n = \frac{\sqrt{4n^2 + 1}}{n}$ ;  $n = 1, 2, \dots$

Вариант 21. Составить алгоритм и программу вычисления и вывода на экран дисплея первых 20 членов последовательности, которая задана рекуррентной формулой  $a_{n+2} = \frac{2 + a_{n+1}}{4 + a_n}$  при  $a_1 = a_2 = 0$ .

Вариант 22. Разработать алгоритм и программу вычисления и вывода на экран дисплея первых 15 членов последовательности, которая задана формулой

$$a_n = \frac{1+n^2}{1+n^3}; \quad n = 1, 2, \dots$$

Вариант 23. Последовательность задана формулой:

$$a_n = \begin{cases} \frac{1}{n}, & \text{если } n \text{ нечетное;} \\ \sqrt{n}, & \text{если } n \text{ четное.} \end{cases}$$

Разработать алгоритм и программу вычисления суммы первых 12 членов последовательности.

Вариант 24. Разработать алгоритм и программу вычисления и вывода на экран дисплея первых 15 членов последовательности

$$a_n = \frac{1}{n^2 + 2}; \quad n = 1, 2, \dots$$

Вариант 25. Разработать алгоритм и программу вычисления членов  $a_{10}, a_{11}, \dots, a_{20}$  последовательности, которая задана формулой

$$a_n = \sqrt{n+2}.$$



## Тема 9. ВЫЧИСЛЕНИЕ КОНЕЧНЫХ СУММ И ПРОИЗВЕДЕНИЙ

### ПРИМЕР СОСТАВЛЕНИЯ ПРОГРАММЫ

Составим программу вычисления значений функции

$$z = \begin{cases} \frac{x+1}{2} \sum_{n=1}^{10} \left(\frac{x}{n}\right)^n, & \text{если } x \leq 2; \\ \frac{\sin x + \cos x}{2 + \sin x} \prod_{n=0}^{15} \left(1 + \frac{x}{n+2}\right), & \text{если } x > 2. \end{cases}$$

В зависимости от значения переменной  $x$  реализуется вычисление суммы или произведения (рис. 9.1). Вычисление суммы целесообразно произвести с помощью оператора цикла с параметром  $n$ . В теле цикла необходимо вычислить значение очередного слагаемого  $u_n = (x/n)^n$  при текущем  $n$  и осуществить накопление суммы по формуле  $S_n = S_{n-1} + u_n$ . Подобные операции требуется выполнить для  $n = 1(1)10$ . Так как нет необходимости запоминать значения всех слагаемых  $u_1, u_2, \dots, u_{10}$  и конечных сумм  $S_1, S_2, \dots, S_{10}$ , то в качестве  $S_n$  и  $u_n$  можно использовать скалярные переменные  $S$  и  $u$ . При этом накопление суммы можно реализовать с помощью операции  $S := S + u$ . Перед выполнением цикла значение переменной  $S$  должно быть нулевым ( $S := 0$ ).

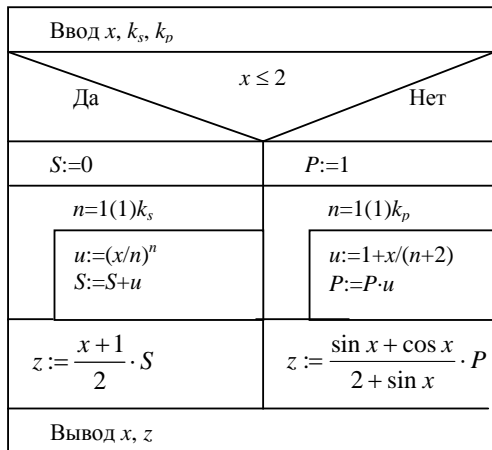


Рис. 9.1. Структурограмма алгоритма вычисления суммы и произведения

Вычисление произведения организуем с помощью аналогичной циклической структуры с параметром. В данном случае необходимо вычислить сомножитель  $u := 1 + x/(n + 2)$  и произведение по формуле  $p := p \cdot u$ . Перед выполнением цикла переменной  $p$  должно быть присвоено значение 1 ( $p := 1$ ).

Для обеспечения большей универсальности алгоритма обозначим предел суммирования через  $ks$ , а предел произведения через  $kp$  и обеспечим их ввод в программе в качестве исходных данных. Запишем программу.

```

{*****}
{Цель:  вычисление сложной функции (конечная      }
{        сумма и произведение).                     }
{Переменные:                                       }
{  z - значение функции; x - аргумент функции      }
{  S - сумма; P - произведение                     }
{  n - переменная суммирования и произведения;    }
{  u - слагаемое (сомножитель);                    }
{  KS - число слагаемых; KP - число сомножителей.  }
{Программист: Сергеев С.С.                         }
{Дата написания: 19 декабря 2005 г.                 }
{*****}

```

**Program** Sum\_Mul;

**Var**

P, S, T, u, x, z: **real**;

n, KP, KS : **integer**;

**Begin** {Sum\_Mul}

{Ввод и эхо-печать исходных данных}

Write('X=');

Read(x);

Write('KS=');

Read (KS);

Write('KP=');

Read(KP);

WriteLn ('X=', x, ' KS=', KS, ' KP=', KP);

**if** x<=2 {Выбор формулы}

**then**

**begin** {Вычисление суммы}

S:=0;

**for** n:=1 **to** KS **do**

**begin**

u:=Exp(n\*Ln(x/n));

S:=S+u

**end**;

z:=S\*(x+1)/2

**end**

```

else
  begin      {Вычисление произведения}
    P:=1;
    for n:=0 to KP do
      P:=P*(1+x/(n+2));
    T:=Sin(x);
    z:=(T+Cos(x))*P/(2+T)
  end;
  WriteLn ('X=',x, ' Z=',z)
End. {Sum_Mul}

```

Необходимо обратить внимание на то, что значение переменной  $x$  в данной программе может быть только положительным, так как при возведении  $x$  в степень используется стандартная функция  $Ln$ .

## ЗАДАНИЯ

Вариант 1. Вычислить сумму

$$S = \sum_{n=0}^{10} \frac{x^{4n+1}}{4n+1} \quad \text{при } x = 1, 2.$$

Вариант 2. Вычислить сумму

$$S = \sum_{n=1}^{20} (a^n + 1) \ln x; \quad a = \begin{cases} 0,5, & \text{если } n \geq 12 \text{ и } x \geq 3,5; \\ 7,5 & \text{в остальных случаях.} \end{cases}$$

Вариант 3. Вычислить произведение

$$P = \prod_{i=1}^{10} \frac{2x+1}{(2i)^2+1} \quad \text{при } x = 2, 3.$$

Вариант 4. Вычислить сумму

$$S = \sum_{n=1}^{10} n^2 + \sum_{n=1}^{12} n^3.$$

Вариант 5. Вычислить произведение

$$P = \prod_{i=1}^{10} \frac{i(i+a)}{i^2+a^2} + \prod_{k=2}^4 \frac{k^3}{k+a} \quad \text{при } a = 2.$$

Вариант 6. Вычислить

$$W = \begin{cases} \sum_{k=1}^{10} x^k \sin \frac{k\pi}{4}, & \text{если } x \geq a; \\ \prod_{m=1}^5 (a^m - x^m), & \text{если } x < a. \end{cases}$$

Для контрольного просчета принять  $x = 7,5$ ;  $a = 1,7$ .

Вариант 7. Вычислить

$$\lambda = (l \cdot k)!, \quad l = \begin{cases} 2, & \text{если } k \text{ четное;} \\ 1, & \text{если } k \text{ нечетное;} \end{cases}$$

$$\text{причем } n! = 1 \cdot 2 \cdot \dots \cdot n = \prod_{m=1}^n m.$$

Для контрольного просчета принять  $k = 5$ .

Вариант 8. Вычислить

$$y = \frac{\sin x + 2}{3 + \cos x} \sum_{n=0}^{20} ax^n; \quad a = \begin{cases} 2n, & \text{если } x \leq 0,5; \\ \frac{n}{2}, & \text{если } x > 0,5. \end{cases}$$

Для контрольного просчета принять  $x = 0,5$ ;  $a = 1,7$ .

Вариант 9. Вычислить

$$y = \frac{ax^2}{\sqrt{x+a}}; \quad a = \begin{cases} \sum_{k=1}^{12} \frac{2kx}{x+k^2}, & \text{если } x < 1; \\ 1, & \text{если } x \geq 1. \end{cases}$$

Для контрольного просчета принять  $x = 7,5$ ;  $a = 0,7$ .

Вариант 10. Вычислить

$$z = \sum_{k=0}^{10} \ln x \cdot \sin k(x-a); \quad a = \begin{cases} \pi/4, & \text{если } x \leq 1; \\ \pi, & \text{если } x > 1. \end{cases}$$

Для контрольного просчета принять  $x = 1,5$ ;  $a = 2,7$ .

Вариант 11. Вычислить

$$z = \begin{cases} \sum_{n=1}^{10} \frac{a^2}{a^n - 5}, & \text{если } a < 4; \\ \frac{a+1}{a} \prod_{n=1}^8 \frac{a-1}{n}, & \text{если } a \geq 4. \end{cases}$$

Для контрольного просчета принять  $a = 1,7$ .

Вариант 12. Вычислить

$$P = (xt)!; \quad n! = 1 \cdot 2 \cdot \dots \cdot n; \quad t = \begin{cases} 1,5, & \text{если } x - \text{четное}; \\ 2, & \text{если } x - \text{нечетное}. \end{cases}$$

Для контрольного просчета принять  $x = 3$ .

Вариант 13. Вычислить

$$z = \begin{cases} \ln(1-x), & \text{если } x \leq 0; \\ \ln(1+x), & \text{если } x > 0. \end{cases}$$

Для вычисления  $\ln(1-x)$  воспользоваться равенством

$$\ln(1-x) = -\sum_{n=1}^{50} \frac{x^n}{n}.$$

Для контрольного просчета принять  $x = 0,5$ .

Вариант 14. Вычислить

$$F = \frac{a+x}{3} \sum_{n=0}^6 (x+a)^{n/2}.$$

Для контрольного просчета принять  $x = 7,5; a = 1,7$ .

Вариант 15. Вычислить

$$z = \begin{cases} a \cdot \ln x, & \text{если } x \geq a; \\ x \cdot \ln a, & \text{если } x < a. \end{cases}$$

Для вычисления  $\ln x$  воспользоваться равенством

$$\ln x \approx 2 \sum_{n=0}^{10} \frac{(x-1)^{2n-1}}{(2n+1)(x+1)^{2n+1}}.$$

Для контрольного просчета принять  $x = 1,5; a = 1,7$ .

Вариант 16. Вычислить произведение

$$y = \prod_{n=1}^8 \left( p - \frac{x^n}{2n+1} \right); \quad p = \begin{cases} 1, & \text{если } n \leq 5; \\ 2, & \text{если } n > 5. \end{cases}$$

Для контрольного просчета принять  $x = 0,7$ .

Вариант 17. Вычислить сумму

$$t = a \sum_{i=1}^{10} \frac{i}{i + a^i}.$$

Для контрольного просчета принять  $a = 1,7$ .

Вариант 18. Вычислить

$$y = \begin{cases} \sum_{k=1}^{10} a^k x^k, & \text{если } a \leq x; \\ \prod_{n=1}^8 (a^k - x^k), & \text{если } a > x. \end{cases}$$

Для контрольного просчета принять  $x = 2,5$ ;  $a = 1,7$ .

Вариант 19. Вычислить

$$z = \begin{cases} \prod_{n=1}^8 \left( \frac{x}{2} \right)^n, & \text{если } x \leq 2; \\ \sum_{n=0}^5 (1 + xn), & \text{если } x > 2. \end{cases}$$

Для контрольного просчета принять  $x = 2,3$ .

Вариант 20. Вычислить

$$F = e^{-x} - x!; \quad x! = \prod_{n=1}^x n; \quad 0! = 1.$$

Для контрольного просчета принять  $x = 5$ .

Вариант 21. Вычислить произведение

$$\lambda = \prod_{n=1}^5 (1/n + a \cdot \sin x); \quad a = \begin{cases} 1, & \text{если } x \geq 0; \\ -1, & \text{если } x < 0. \end{cases}$$

Для контрольного просчета принять  $x = 1,5$ .

Вариант 22. Вычислить произведение

$$y = \prod_{n=1}^5 \left( n + \frac{\sin x}{n} \right).$$

Для контрольного просчета принять  $x = 0,7$ .

Вариант 23. Вычислить сумму

$$x = \sum_{n=0}^4 (k+n)a^{k+n}.$$

Для контрольного просчета принять  $k = 5; a = 1,7$ .

Вариант 24. Вычислить значения функции

$$y = \begin{cases} \sum_{n=1}^{10} \left( \frac{x}{n} \right)^n, & \text{если } x \leq 2; \\ \prod_{n=0}^8 \left( 1 + \frac{x}{n+2} \right), & \text{если } x > 2. \end{cases}$$

Значения  $x$  выбрать таким образом, чтобы обеспечить тестирование всех ветвей программы.

Вариант 25. Вычислить произведение

$$P = \prod_{i=1}^{10} \frac{i(i+a)}{i^2+a^2} + \prod_{k=2}^4 \frac{k^3}{k+a} \quad \text{при } a = 2.$$

## Тема 10. ВЫЧИСЛЕНИЕ СУММ ПРОСТЕЙШИХ ЧИСЛОВЫХ РЯДОВ

### ПРИМЕРЫ СОСТАВЛЕНИЯ ПРОГРАММ

**Пример 10.1.** Составим программу вычисления значения числового ряда, определяющего значение основания натурального логарифма  $e$ , равного примерно 2,718282:

$$1 + 1/1! + 1/2! + 1/3! + \dots + 1/n! = e.$$

Накопление суммы ряда будем производить по рекуррентной формуле

$$S_i = S_{i-1} + u_i,$$

где  $u_i$  – текущий член ряда, определяемый выражением

$$u_i = \frac{1}{i!}.$$

Для того чтобы не вводить дополнительного цикла вычисления факториала  $i!$ , с целью сокращения объема вычислений следует использовать рекуррентное соотношение

$$u_i = u_{i-1}/i.$$

Принимая начальные значения суммы  $S_0$  и текущего члена ряда  $u_0$  равными единице, схему алгоритма решения задачи можно представить рис. 10.1. Текст программы имеет следующий вид:

```
{ ***** }
{ Цель: вычисление суммы ряда. }
{ Переменные: }
{ S – сумма; i – параметр цикла; }
{ u – член ряда; }
{ n – число членов ряда. }
{ Программист: Сычев В.К. }
{ Дата написания: 26 декабря 2005 г. }
{ ***** }
```

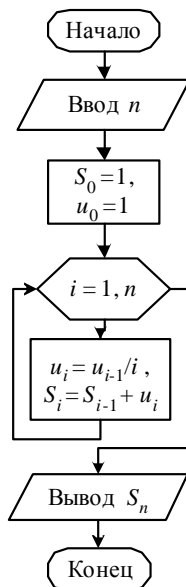


Рис. 10.1. Алгоритм вычисления суммы ряда



```

Program Line;
Var
    S, u : real;
    i, n : integer;
Begin {Line}
    {Ввод и эхо-печать исходных данных}
    Write('Введите число членов ряда');
    Read(n);
    Write ('Сумма',n:4,' слагаемых равна');
    {Инициализация переменных}
    S:=1;
    u:=1;
    for i:=1 to n do {Накопление суммы ряда}
        begin
            u:=u/i;
            S:=S+u
        end;
    {Вывод результатов}
    WriteLn(S);
    WriteLn('Точное значение ряда: e = ', Exp(1))
End. {Line }

```

**Пример 10.2.** Составим программу вычисления значения числового ряда, определяющего значение константы  $1/4$ :

$$1/(1 \cdot 2 \cdot 3) + 1/(2 \cdot 3 \cdot 4) + 1/(3 \cdot 4 \cdot 5) + \dots \\ \dots + 1/(n(n+1)(n+2)) = 1/4.$$

Накопление суммы ряда, как и в предыдущем примере, будем производить по рекуррентной формуле

$$S_i = S_{i-1} + u_i,$$

где  $u_i$  – текущий член ряда, определяемый выражением

$$u_i = \frac{1}{i(i+1)(i+2)}.$$

В отличие от предыдущего примера здесь текущий член ряда может быть вычислен непосредственно и его необязательно обозначать отдельной переменной. Поэтому схему алгоритма решения задачи можно представить рис. 10.2. Текст же программы примет следующий вид:

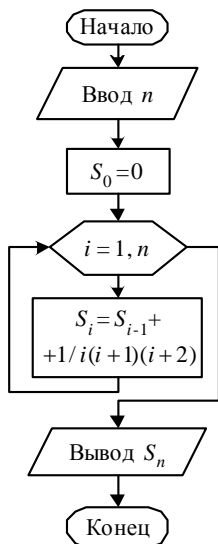


Рис. 10.2. Алгоритм вычисления второй суммы

```

{ ***** }
{ Цель: вычисление второй суммы ряда }
{ Переменные: }
{ S - сумма; i - параметр цикла; }
{ n - число членов ряда. }
{ Программист: Краснов В.К. }
{ Дата написания: 28 декабря 2005 г. }
{ ***** }
Program Line2;
Var
  S :real;
  i, n :integer;
Begin {Line2}
  {Ввод и эхо-печать исходных данных}
  Write('Введите число членов ряда');
  Read(n);
  Write ('Сумма',n:4,' слагаемых равна');
  {Инициализация переменной суммы}
  S:=1;
  for i:=1 to n do {Накопление суммы ряда}
    S:=S+1/(i*(i+1)*(i+2));
  {Вывод результата}
  WriteLn(S);
End. {Line2}

```

## ЗАДАНИЯ

Составить алгоритм и программу вычисления суммы ряда в соответствии с приведенным вариантом. Справа от знака равенства дано для контроля математическое значение ряда.

Вариант 1.

$$1 - 1/2 + 1/3 - 1/4 + \dots \pm 1/n \approx \ln 2.$$

Вариант 2.

$$1 + 1/2 + 1/4 + 1/8 + \dots + 1/2^n \approx 2.$$

Вариант 3.

$$1 - 1/2 + 1/4 - 1/8 + \dots \pm 1/2^n \approx 2/3.$$

Вариант 4.

$$1 - 1/3 + 1/5 - 1/7 + 1/9 - \dots \pm 1/(2n-1) \approx \pi/4.$$

Вариант 5.

$$1/(1 \cdot 2) + 1/(2 \cdot 3) + 1/(3 \cdot 4) + \dots + 1/(n(n+1)) \approx 1.$$

Вариант 6.

$$1/(1 \cdot 3) + 1/(3 \cdot 5) + 1/(5 \cdot 7) + \dots + 1/((2n-1)(n+1)) \approx 1/2.$$

Вариант 7.

$$1/(1 \cdot 3) + 1/(2 \cdot 4) + 1/(3 \cdot 5) + \dots + 1/((n-1)(n+1)) \approx 2/3.$$

Вариант 8.

$$1/(3 \cdot 5) + 1/(7 \cdot 9) + 1/(11 \cdot 13) + \dots + 1/((4n-1)(4n+1)) \approx 1/2 - \pi/8.$$

Вариант 9.

$$1/(1 \cdot 2 \cdot 3) + 1/(2 \cdot 3 \cdot 4) + 1/(3 \cdot 4 \cdot 5) + \dots + 1/(n(n+1)(n+2)) \approx 1/4.$$

Вариант 10.

$$1 + 1/2^2 + 1/3^2 + 1/4^2 + \dots + 1/n^2 \approx \pi^2/6.$$

Вариант 11.

$$1/1^2 + 1/3^2 + 1/5^2 + \dots + 1/(2n+1)^2 \approx \pi^2/8.$$

Вариант 12.

$$1 + 1/2^4 + 1/3^4 + 1/4^4 + \dots + 1/n^4 \approx \pi^4/90.$$

Вариант 13.

$$1 - 1/2^4 + 1/3^4 - \dots \pm 1/n^4 \approx 7\pi^4/720.$$

Вариант 14.

$$1/1^4 + 1/3^4 + 1/5^4 + \dots + 1/(2n+1)^4 \approx \pi^4/96.$$

Вариант 15.

$$1 - 1/1! + 1/2! - 1/3! + \dots \pm 1/n! \approx 1/e.$$

Вариант 16.

$$1 + 1/1! + 1/2! + 1/3! + \dots + 1/n! \approx e.$$

Вариант 17.

$$1 + 2 + 3 + \dots + n = n(n+1)/2.$$

Вариант 18.

$$p + (p+1) + (p+2) + \dots + q = (2p + q)(q+1)/2.$$

Вариант 19.

$$1 + 3 + 5 + \dots + (2n-1) = n^2.$$

Вариант 20.

$$2 + 4 + 6 + \dots + 2n = n(n+1).$$

Вариант 21.

$$1^2 + 2^2 + 3^2 + \dots + n^2 = n(n+1)(2n+1)/6.$$

Вариант 22.

$$1^3 + 2^3 + 3^3 + \dots + n^3 = n^2 (n+1)^2 / 4.$$

Вариант 23.

$$1^2 + 3^2 + 5^2 + \dots + (2n-1)^2 = n(4n^2-1)/3.$$

Вариант 24.

$$1^3 + 3^3 + 5^3 + \dots + (2n-1)^3 = n^2(2n^2-1).$$

Вариант 25.

$$1^4 + 2^4 + 3^4 + \dots + n^4 = n(n+1)(2n+1)(3n^2+3n-1)/30.$$

## Тема 11. ТАБУЛИРОВАНИЕ ФУНКЦИИ С ПОМОЩЬЮ ОПЕРАТОРА ЦИКЛА С ПАРАМЕТРОМ

### ПРИМЕР СОСТАВЛЕНИЯ ПРОГРАММЫ

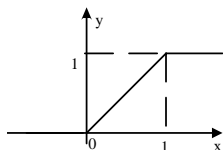


Рис. 11.1. График функции

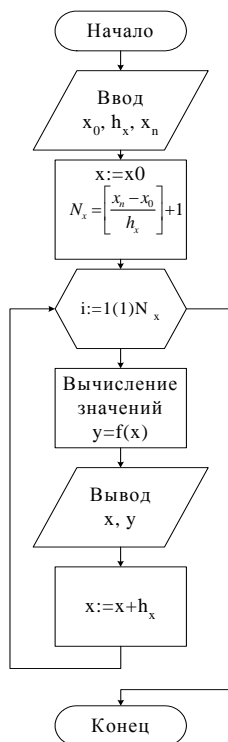


Рис. 11.2. Схема алгоритма табулирования функции

Пусть необходимо протабулировать функцию  $f(x)$ , т. е. получить таблицу ее значений, заданную графиком (рис. 11.1), на отрезке  $[x_0, x_n]$  с постоянным шагом  $h_x$ .

Поскольку в качестве параметра в операторе **for** не может быть использована переменная действительного типа (например, переменная  $x$ ), то введем дополнительную переменную  $i$ , значение которой будем изменять от 1 до  $N_x$  с постоянным шагом 1. Значение  $N_x$  равно числу повторений цикла при законе изменения параметра  $x = x_0(h_x)x_n$  и определяется формулой

$$N_x = \left[ \frac{x_n - x_0}{h_x} \right] + 1,$$

где  $[z]$  означает целую часть числа  $z$ .

Для переменной  $x$  перед циклом зададим ее начальное значение  $x_0$ , а в теле цикла будем производить ее модификацию (изменение). Закон изменения параметра  $i$  цикла укажем в заголовке цикла. В результате получаем схему алгоритма циклической структуры с заголовком (рис. 11.2), для которой запишем программу табулирования функции при  $x = x_0(h_x)x_n$  в виде

$$y = \begin{cases} 0, & \text{если } x \leq 0; \\ x, & \text{если } 0 < x < 1; \\ 1, & \text{если } x \geq 1. \end{cases}$$

```
{ ***** }
{ Цель:   табулирование функции  $y=F(x)$  с   }
{         помощью оператора цикла с         }
{         параметром.                         }
{ Переменные:                               }
{   x - переменная цикла;                   }
{   x0, xn - начальное и конечное значения; }
{   hx - шаг изменения;                     }
{   i - параметр цикла;                     }
{   nx - число повторений тела цикла.       }
{ Программист: Федоров Ф.Ф.                 }
{ Дата написания: 3 января 2006 г.          }
{ ***** }
```

**Program** Tab;

**Var**

        hx,x,x0,xn,y:**real**;

        i,nx:**integer**;

**Begin** {Tab}

        {Ввод и эхо-печать исходных данных }

            Write('x0=');

            Read(x0);

            Write('hx=');

            Read(hx);

            Write('xn=');

            Read(xn);

            WriteLn('X0=',x0,' HX=',hx,' XN=',xn);

        {Табулирование функции }

            x:=x0;

            nx:=Trunc((xn-x0)/hx+1e-6)+1;

**for** i:=1 **to** nx **do**

**begin** {начало цикла}

**if** x<=0

**then**

                            y:=0

**else if** x<1

**then**

                            y:=x

**else**

                            y:=1;

                    WriteLn('X=',x,' Y=',y);

                    x:=x+hx

**end** {конец цикла}

**End.** {Tab}

В программе использована стандартная функция Trunc(x), результат которой есть наибольшее целое число, меньшее или равное x. Аргумент

функции Trunc дополнен слагаемым 1E-6, который, не изменяя полученного результата, позволяет избежать ошибки представления вещественных значений  $x_0$ ,  $hx$  и  $xn$ .

## ЗАДАНИЕ

Выполнить соответствующий вариант из заданий к теме 5, используя оператор цикла с параметром.

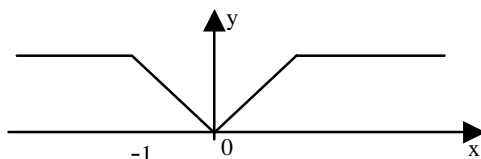
## КОНТРОЛЬНЫЕ ЗАДАНИЯ

Вариант 1. Вычислить

$$z = \begin{cases} 2 \sin x, & \text{если } x \leq -\pi/2; \\ A \sin x + B, & \text{если } -\pi/2 < x < \pi/2; \\ \cos x, & \text{если } x \geq \pi/2, \end{cases}$$

где  $x = -2(0,2)2$ ;  $A = -5$ ;  $B = 12$ .

Вариант 2. Вычислить значение функции  $f(x)$  по указанному графику для значения аргумента  $x = x_0(h_x)x_n$ , где  $x_0 = -2$ ;  $h_x = 0,5$ ;  $x_n = 2$ .



Вариант 3. Вычислить значение функции одной переменной

$$z = \frac{\sin^2 x}{x^2 - 4} + \frac{\cos x^2}{(x - 2)(x - 5)}$$

в интервале  $-3 \leq x \leq 6$  с шагом  $h_x = 0,5$ . Точки разрыва исключить.

Вариант 4. Вычислить сумму

$$S = \sum_{n=0}^{10} \frac{x_n^{4n+1}}{4n+1} \quad \text{при } x_n = 1(0,1)2.$$

Вариант 5. Вычислить сумму

$$S = \sum_{n=1}^{10} n^2 + \sum_{n=1}^{12} n^3.$$

Вариант 6. Вычислить произведение

$$P = \prod_{i=1}^{10} \frac{2x_i + 1}{(2i)^2 + 1} \quad \text{при } x_i = 2, 1(0, 1)3.$$

Вариант 7. Вычислить значение интеграла

$$J = \int_a^b \frac{1 + \operatorname{tg}^2 x}{1 + \operatorname{tg} x} dx$$

по формуле прямоугольников

$$J \approx h \sum_{i=1}^n f(x_i), \quad \text{где } h = (b-a)/n, \quad x_i = a + ih,$$

$f(x_i)$  – подынтегральная функция. Принять  $a = 0$ ,  $b = \pi/4$ ,  $n = 30$ .

Вариант 8. Вычислить сумму

$$S = \sum_{n=1}^{20} (a^n + 1) \ln x; \quad a = \begin{cases} 0,5, & \text{если } n \geq 12 \text{ и } x \geq 3,5; \\ 7,5 & \text{в остальных случаях.} \end{cases}$$

Для контрольного просчета принять  $x = 1,75$ .

Вариант 9. Вычислить значение интеграла

$$J = \int_a^b \frac{x^3}{x^4 + 1} dx$$

по формуле трапеций

$$J \approx h \left[ \frac{f(a) + f(b)}{2} + \sum_{k=1}^{n-1} f(x_k) \right],$$

где  $h = (b-a)/n$ ;  $x_k = a + i \cdot h$ ;  $f(x_k)$  – подынтегральная функция.

Принять  $a = 1$ ,  $b = 4$ ,  $n = 40$ .

Вариант 10. Вычислить произведение

$$P = \prod_{i=1}^{10} \frac{i(i+a)}{i^2 + a^2} + \prod_{k=2}^4 \frac{k^3}{k+a} \quad \text{при } a = 2.$$



Вариант 11. Вычислить

$$W = \begin{cases} \sum_{k=1}^{10} x^k \sin \frac{k\pi}{4}, & \text{если } x \geq a; \\ \prod_{m=1}^5 (a^m - x^m), & \text{если } x < a. \end{cases}$$

Для контрольного просчета принять  $x = 7,5$ ;  $a = 1,7$ .

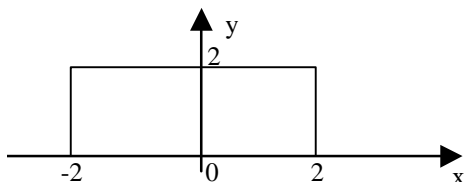
Вариант 12. Вычислить

$$\lambda = (l \cdot k)!, \quad l = \begin{cases} 10, & \text{если } k \text{ четное;} \\ 1, & \text{если } k \text{ нечетное;} \end{cases}$$

$$\text{причем } n! = 1 \cdot 2 \cdot \dots \cdot n = \prod_{m=1}^n m.$$

Для контрольного просчета принять  $k = 5$ .

Вариант 13. Определить количество заданных точек  $(x, y)$ , попавших в указанную область, включая ее границы, где  $x = x_0 + i \cdot h$ ;  $y = y_0 + i \cdot h$ ;  $x_0 = -1,5$ ;  $y_0 = 0,5$ ;  $h = 0,1$ ;  $i = 1(1)10$ .



Вариант 14. Определить, сколько четных целых чисел лежит в интервале  $(a, b)$ , где  $a < \sin x^2$ ;  $b = x^4$ ;  $x = 3$ .

Вариант 15. Определить максимальное целое число  $n$ , удовлетворяющее условию  $3n^2 - 730n < 5$ .

Вариант 16. Вычислить первые 20 членов последовательности чисел Фибоначчи:  $u_1 = 1$ ;  $u_2 = 2$ ;  $u_n = u_{n-1} + u_{n-2}$ , а также значение золотого сечения

$$V_n = \frac{u_n}{u_{n-1}}.$$

Вариант 17. Вычислить

$$\alpha = \begin{cases} a \sin x, & \text{если } x > 0; \\ a \ln |x|, & \text{если } x \leq 0, \end{cases}$$

для  $x = -3(0,5)3$ ;  $a = 1,35$ . Точку  $x = 0$  исключить.

Вариант 18. Вычислить значение интеграла

$$J = \int_{1,5}^5 \frac{\sin x}{\sqrt{c + x^3}} dx$$

по формуле трапеций при  $n = 30$  (см. вариант 9), где  $c = 2,1$ .

Вариант 19. Вычислить значение интеграла

$$J = \int_0^1 \frac{1 - e^x}{x - c^2} dx$$

по формуле трапеций при  $n = 20$  (см. вариант 9), где  $c = 1,5$ .

Вариант 20. Вычислить

$$Z = \frac{(a + \sqrt{c^3}) \sin x + c^3}{(a^3 - c^3)(a - 4)}$$

для  $x = 0,5(0,2)1,7$ . Точки разрыва исключить.

Вариант 21. Вычислить

$$Z = \frac{(a + x) \sin^2 x}{(a^2 - x^3)(a - 2)}$$

для  $x = 0,3(0,1)1,2$ . Точки разрыва исключить.

Вариант 22. Вычислить значение функции одной переменной

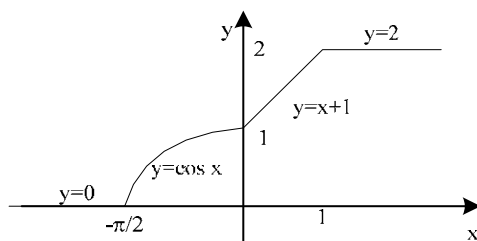
$$z = \frac{\cos x^2}{(x - 2)(x - 5)}$$

в интервале  $-3 \leq x \leq 6$  с шагом  $h_x = 1$ . Точки разрыва исключить.

Вариант 23. Вычислить сумму

$$S = \sum_{k=1}^8 \frac{k^3}{k + a}$$

Вариант 24. Вычислить значение функции  $f(x)$  по указанному графику для значения аргумента  $x = x_0(h_x)x_n$ , где  $x_0 = -2$ ;  $h_x = 0,5$ ;  $x_n = 2$ .



Вариант 25. Вычислить значение интеграла

$$J = \int_{-3}^2 \frac{\sin^2 x}{x^2 - 4} dx$$

по формуле прямоугольников (см. вариант 7). Принять  $n = 20$ .

## Тема 12. ИТЕРАЦИОННЫЕ ЦИКЛЫ. УТОЧНЕНИЕ КОРНЕЙ УРАВНЕНИЙ МЕТОДОМ ПОЛОВИННОГО ДЕЛЕНИЯ

### ПРИМЕР СОСТАВЛЕНИЯ ПРОГРАММЫ

Пусть необходимо найти корень уравнения

$$5x^3 + 10x^2 + 5x - 1 = 0$$

на интервале  $[0; 1]$  с погрешностью  $\varepsilon$  методом половинного деления.

Схема алгоритма, реализующего данный метод, представлена на рис. 12.1, где  $f(x) = 5x^3 + 10x^2 + 5x - 1$ .

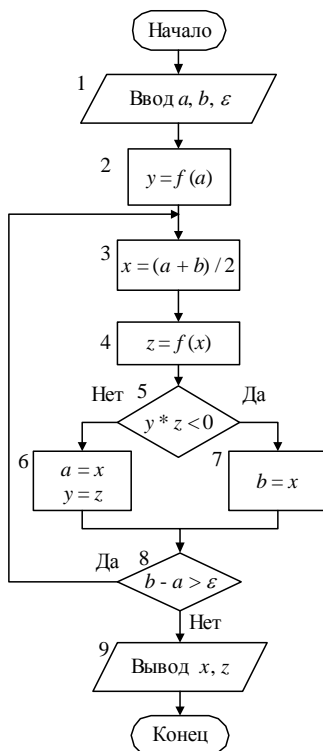


Рис. 12.1. Алгоритм нахождения корня методом половинного деления

Как показано на схеме, в начале алгоритма обеспечивается ввод начальных границ интервала и заданной погрешности вычислений (блок 1). Затем перед входом в цикл (блок 2) вычисляется значение функции в точке  $a$ . В начале цикла (блок 3) вычисляется координата середины отрезка и значение функции в точке  $x$  (блок 4). В логическом блоке 5 определяется половина интервала, содержащая корень. Затем (блоки 6, 7) задается новая граница интервала. Когда в точку  $x$  сдвигается левая граница  $a$  интервала  $[a, b]$ , то значение функции  $y = f(a)$  можно не вычислять, а принять его равным ранее вычисленному  $z = f(x)$ . Если величина полученного интервала превышает заданную погрешность, то обеспечивается повторение цикла (блок 8). В качестве результата выводится (блок 9) найденное значение корня  $x$  и контрольное значение функции в данной точке  $z$ , которое по определению должно быть близко к нулю.

Текст программы для решения уравнения

$$5x^3 + 10x^2 + 5x - 1 = 0$$

методом половинного деления будет иметь следующий вид:

```
{Цель: уточнение корня.}
{Переменные:}
{  eps - допустимая погрешность;}
{  x - приближение корня;}
{  a, b - границы интервала;}
{  y - значение функции на левой границ;}
{  z - значение функции в середине отрезка.}
{Метод: половинное деление.}
{Подпрограммы: F - исследуемая функция.}
{Программист: Петров П.П.}
{Дата: 10 января 2006 г.}
```

**Program** Halving;

**Var**

a, b, eps, x, y, z :real;

{Описание исследуемой функции}

{ F - имя функции}

{ x - аргумент функции}

**Function** F(x :real) :real;

**Begin** {F}

F := 5\*x\*x\*x + 10\*x\*x + 5\*x - 1

**End;** {F}

**Begin** {Halving}

{Ввод и эхо-печать исходных данных}

WriteLn('Введите границы интервала и погрешность');

Read(a, b, eps);

```

WriteLn('В интервале: ', a:6:2, ', ', b:6:2);
WriteLn('с погрешностью: ', eps:9:6);
{Реализация метода половинного деления}
y := F(a);
repeat {Начало цикла}
  x := (a+b)/2;
  z := F(x)
  if y*z < 0
  then
    b := x
  else
    begin
      a := x;
      y := z
    end;
until b - a <= eps; {Конец цикла}
WriteLn('корень: ', x:10:6, ' f(x)=', z)
End. {Halving}

```

## ЗАДАНИЯ

Для заданного уравнения и заданного интервала, содержащего один корень, используя метод половинного деления, уточнить корень нелинейного уравнения с точностью до  $\varepsilon = 10^{-5}$ .

На экран дисплея вывести границы интервала, получаемые при его делении пополам (для всех вариантов число таких интервалов не превышает 18).

После выхода из итерационного цикла корень считать равным середине интервала и провести проверку путем подстановки в исходное уравнение, чтобы убедиться в правильности нахождения корня.

Исходное уравнение  $f(x) = 0$ . Целесообразно  $f(x)$  оформить в виде функции пользователя.

Вариант	Уравнение	Область, содержащая корень
1	$3 \sin \sqrt{x} + 0,35x - 3,8 = 0$	[2; 3]
2	$0,25x^3 + x - 1,2502 = 0$	[0; 2]
3	$x + \sqrt{x} + \sqrt[3]{x} - 2,5 = 0$	[0,4; 1]
4	$x - \frac{1}{3 + \sin 3,6x} = 0$	[0; 0,85]
5	$0,1x^2 - x \ln x = 0$	[1; 2]

Вариант	Уравнение	Область, содержащая корень
6	$\operatorname{tg} x - \frac{1}{3} \operatorname{tg}^3 x + \frac{1}{5} \operatorname{tg}^5 x - \frac{1}{3} = 0$	[0; 0,8]
7	$x - 2 + \sin \frac{1}{x} = 0$	[1,2; 2]
8	$e^x + \ln x - 10x = 0$	[3; 4]
9	$\cos x - e^{-x^2/2} + x - 1 = 0$	[1; 2]
10	$1 - x + \sin x - \ln(1+x) = 0$	[0; 1,5]
11	$3x - 14 + e^x - e^{-x} = 0$	[1; 3]
12	$\sqrt{1-x} - \operatorname{tg} x = 0$	[0; 1]
13	$x + \cos(x^{0,52} + 2) = 0$	[0,5; 1]
14	$3\ln^2 x + 6\ln x - 5 = 0$	[1; 3]
15	$\sin x^2 + \cos x^2 - 10x = 0$	[0; 1]
16	$\ln x - x + 1,8 = 0$	[2; 3]
17	$x \cdot \operatorname{tg} x - \frac{1}{3} = 0$	[0,2; 1]
18	$\operatorname{tg} \frac{x}{2} - \operatorname{ctg} \frac{x}{2} + x = 0$	[1; 2]
19	$0,4 + \arctg \sqrt{x} - x = 0$	[1; 2]
20	$\sqrt{1-x} - \cos \sqrt{1-x} = 0$	[0; 1]
21	$0,6 \cdot 3^x - 2,3x - 3 = 0$	[2; 3]
22	$\frac{1}{3}x^3 - x^2 + \frac{1}{3} = 0$	[0; 1]
23	$e^x (\cos x + \sin x) - 2 = 0$	[0; 1,5]
24	$\sqrt[3]{(\sqrt{x^3} - 2)^2} - 1 = 0$	[0; 2]
25	$e^{\frac{x}{2}} + e^{-\frac{x}{2}} - 3 = 0$	[1; 3]

## ТЕМА 13. ИТЕРАЦИОННЫЕ ЦИКЛЫ. УТОЧНЕНИЕ КОРНЕЙ УРАВНЕНИЙ МЕТОДОМ ИТЕРАЦИЙ

### ПРИМЕР СОСТАВЛЕНИЯ ПРОГРАММЫ

Пусть, например, необходимо найти с погрешностью  $\varepsilon = 10^{-4}$  корень уравнения

$$5x^3 + 10x^2 + 5x - 1 = 0$$

методом итераций.

Приведем исходное уравнение к виду

$$x = \frac{1}{5(x+1)^2}.$$

Для определения  $x_0$  применим графический метод отделения корней, а именно построим график функций

$$y_1 = x; y_2 = \frac{1}{5(x+1)^2}.$$

Нетрудно убедиться, что корень (точка пересечения этих графиков) принадлежит отрезку  $[0, 1]$ . Поэтому для всех  $x \in [0, 1]$  метод итераций применим:

$$|\varphi'(x)| = \frac{2}{5(x+1)^3} < 1.$$

Схема алгоритма представлена на рис. 13.1.

Программа будет иметь следующий вид:

```
{Цель: уточнение корня.           }
{Переменные:                       }
{  eps - допустимая погрешность; }
{  x0 - начальное приближение;   }
{  x1, x2 - последовательные     }
{    приближения корня.         }
{Метод: метод итераций.           } }
```

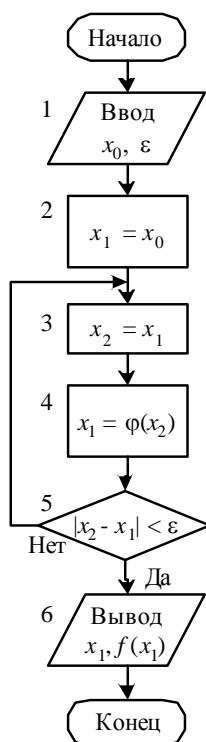


Рис. 13.1. Алгоритм нахождения корня методом итераций



```
{Программист: Петров П.П.           }
{Дата: 13 января 2006 г.             }
Program Iter;
  Var
    eps,x0,x1,x2: real;
  Begin {Iter}
    {Ввод и эхо-печать исходных данных }
    WriteLn('Введите начальное',
      ' приближение и погрешность');
    Read(eps,x0);
    WriteLn('Погрешность: ',eps);
    WriteLn(' начальное приближение: ',x0:6:2);
    {Реализация метода итераций}
    x1=x0;
    repeat                                {Начало цикла}
      x2:=x1;
      x1:= 1/(5*(x2+1)*(x2+1))
    until Abs (x1-x2) < eps; {Конец цикла}
    WriteLn('корень: ',x1:9:5)
  End. {Iter}
```

## ЗАДАНИЯ

Уточнить корень уравнения из предыдущего задания методом итераций с погрешностью  $\varepsilon = 10^{-5}$ . Определить число повторений цикла при методе половинного деления и методе итераций и сравнить эффективность этих методов.

## Тема 14. ИТЕРАЦИОННЫЕ ЦИКЛЫ. ВЫЧИСЛЕНИЕ СУММЫ БЕСКОНЕЧНОГО РЯДА

### ПРИМЕРЫ СОСТАВЛЕНИЯ ПРОГРАММ

**Пример 14.1.** Рассмотрим задачу вычисления функции  $y = \cos(x)$  с погрешностью  $\varepsilon$ , используя разложение косинуса в ряд:

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots = \sum_{n=0}^{\infty} t_n(x).$$

Накопление суммы производим по рекуррентной формуле

$$S_n = S_{n-1} + t_n(x).$$

Здесь общий вид слагаемого суммы может быть записан в следующей форме:

$$t_n(x) = (-1)^n \frac{x^{2n}}{(2n)!}.$$

Для сокращения объема вычислений его целесообразно вычислять по формуле

$$t_n(x) = t_{n-1}(x) \varphi_n(x).$$

Сомножитель  $\varphi_n(x)$ , называемый переходным коэффициентом, можно определить из соотношения

$$\varphi_n(x) = \frac{t_n(x)}{t_{n-1}(x)}.$$

Учитывая, что

$$t_{n-1}(x) = (-1)^{n-1} \frac{x^{2(n-1)}}{[2(n-1)]!} = (-1)^{n-1} \frac{x^{2n-2}}{(2n-2)!},$$

получим

$$\varphi_n(x) = \frac{t_n(x)}{t_{n-1}(x)} = \frac{(-1)^n \frac{x^{2n}}{(2n)!}}{(-1)^{n-1} \frac{x^{2n-2}}{(2n-2)!}} = -\frac{x^2(2n-2)}{(2n)!} = -\frac{x^2}{2n(2n-1)}.$$

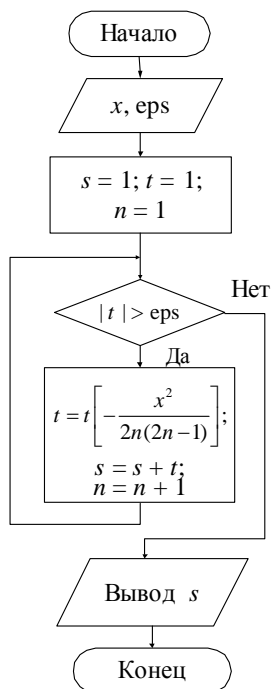


Рис. 14.1. Алгоритм вычисления суммы бесконечного ряда

Тогда текущий член ряда будет вычисляться по формуле

$$t_n(x) = -\frac{x^2}{2n(2n-1)} t_{n-1}(x).$$

Начальное значение  $t_0(x)$  находим подстановкой в формулу для общего члена  $t_n(x)$  значения  $n = 0$ :

$$t_0(x) = t_n(x)|_{n=0} = (-1)^0 \frac{x^{2 \cdot 0}}{(2 \cdot 0)!} = 1.$$

Начальное значение суммы можно принять равным первому члену ряда, т. е.  $S_0 = 1$ . Учитывая, что значения текущего члена ряда и суммы при  $n = 0$  уже определены, номер члена ряда  $n$  примем равным единице.

На рис. 14.1 приведен алгоритм вычисления бесконечного ряда с заданной погрешностью  $\text{eps}$ .

Программа вычисления такой суммы имеет следующий вид:

```

{*****}
{Цель: вычисление суммы с заданной}
{  погрешностью по итерационному}
{  алгоритму.}
{Переменные:}
{  x - аргумент функции}
{  n - переменная суммирования}
{  eps - погрешность вычисления суммы}
{  S - сумма}
{  t - слагаемое.}
{Программист: Карлов К.К.}
{Дата написания: 16 января 2006 г.}
{*****}

```

Program Serie;

Var

eps, S, t, x:real;  
n:integer;

```

Begin {Serie}
{Ввод исходных данных}
  Write('x=');
  Read(x);
  Write('eps=');
  Read(eps);
{Эхо-печать}
  WriteLn('Для x=', x:6:2, ' с погрешностью ', eps);
{Вычисление суммы}
  t:=1;
  S:=1;
  n:=1;
  while abs(t)>eps do {Условие продолжения цикла}
    begin
      t:=-t*x*x/(2*n*(2*n-1));
      S:=S+t;
      n:=n+1
    end;
  WriteLn('n=', n, ' S=', S:12:9)
  WriteLn('Контрольное значение: ', cos(x):12:9)
End. {Serie}

```

Заметим, что переменной цикла является параметр  $n$ , однако он напрямую не участвует в управлении циклом.

**Пример 14.2.** Пусть необходимо вычислить значение суммы ряда

$$S = \frac{1}{1 \cdot 2} + \frac{1}{2 \cdot 3} + \frac{1}{3 \cdot 4} + \dots$$

с заданной погрешностью  $\varepsilon$ .

Здесь при накоплении суммы по формуле

$$S_n = S_{n-1} + t_n$$

общий член ряда имеет вид

$$t_n = \frac{1}{n(n+1)}$$

и может быть вычислен непосредственно без определения переходного коэффициента  $\varphi_n$ . Принимая начальные значения суммы и члена ряда равными  $1/2$ , алгоритм решения данной задачи можно представить в виде рис. 14.2.

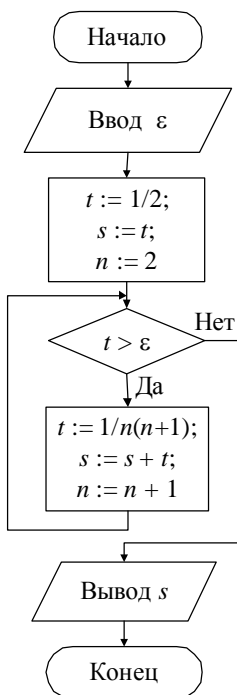


Рис. 14.2. Схема алгоритма вычисления суммы ряда

Программа вычисления такой суммы простого ряда будет иметь следующий вид:

```

Program Serie_simple;
{*****}
{Цель: вычисление суммы с заданной}
{  погрешностью по итерационному}
{  алгоритму.}
{Переменные:}
{  n - переменная суммирования}
{  t - член ряда}
{  eps - погрешность вычисления суммы}
{  S - сумма.}
{Программист: Краснов Ф.Е.}
{Дата написания: 19 января 2006 г.}
{*****}

Var
    eps, S, t:real;
    n:integer;
    
```

```

Begin {Serie_simple}
{Ввод исходных данных}
  Write('eps=');
  Read(eps);
{Эхо-печать}
  WriteLn('Сумма ряда с погрешностью ', eps);
{Вычисление суммы}
  t:=1/2;
  S:=t;
  n:=2;
  while t>eps do {Условие продолжения цикла}
    begin
      t:=1/(n*(n+1));
      S:=S+t;
      n:=n+1;
    end;
  WriteLn('S=', S:12:9, ' за ', n-2, ' шагов')
End. {Serie_simple}

```

**Пример 14.3.** Рассмотрим пример вычисления суммы ряда, текущий член которого представляется в виде двух сомножителей, один из которых вычисляется по рекуррентному соотношению, а другой непосредственно:

$$S = x + \frac{x^3}{3} + \frac{x^5}{5} + \dots + \frac{x^{2n+1}}{2n+1} + \dots$$

Здесь при накоплении суммы по формуле

$$S_n = S_{n-1} + t_n$$

общий член ряда может быть представлен в виде произведения

$$t_n = v_n u_n,$$

где сомножитель  $v_n = x^{2n+1}$  целесообразно вычислить по рекуррентному соотношению

$$v_n = x^2 v_{n-1},$$

а сомножитель  $u_n = \frac{1}{2n+1}$  – непосредственно по приведенной формуле.

Принимая начальные значения  $S_0, v_0, t_0$  равными  $x$ , а  $n=1$ , схему алгоритма можно представить в виде рис. 14.3.

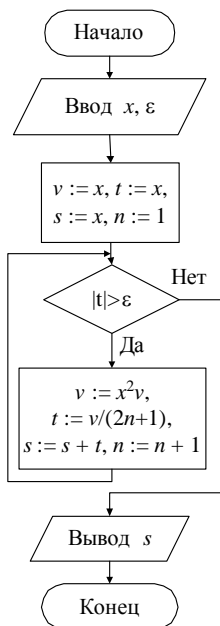


Рис. 14.3. Схема алгоритма вычисления суммы комбинированного ряда

Программа же вычисления суммы комбинированного ряда будет иметь следующий вид:

```

Program Serie_complex;
{*****}
{Цель: вычисление суммы с заданной}
{  погрешностью по итерационному}
{  алгоритму.}
{Переменные:}
{  n – переменная суммирования;}
{  t – член ряда;}
{  x – аргумент;}
{  v – сомножитель;}
{  eps – погрешность вычисления суммы;}
{  S – сумма.}
{Программист: Сизов Б.М.}
{Дата написания: 21 января 2006 г.}
{*****}
Var
    eps, S, t, v:real;
    
```

```

n:integer;
Begin {Serie_complex}
{Ввод исходных данных}
  Write('Погрешность eps=');
  Read(eps);
  Write('Аргумент x=');
  Read(x);
{Эхо-печать}
  WriteLn('Сумма ряда с погрешностью ', eps);
  WriteLn('при x= ', x:6:3);
{Вычисление суммы}
  v:=x;
  t:=x;
  S:=x;
  n:=1;
  while Abs(t)>eps do {Условие продолжения цикла}
    begin
      v:=x*x*v
      t:=v/(2*n+1);
      S:=S+t;
      n:=n+1
    end;
  WriteLn('S=', S:12:9, ' за ', n-1, ' шагов')
End. {Serie_complex}

```

## ЗАДАНИЯ

Разработать алгоритм и программу, определяющие сумму ряда с заданной погрешностью  $\varepsilon$  (ипсilon) для произвольно заданного аргумента  $x$ . Определить число повторений итерационного цикла.

Вариант	Бесконечная сумма	Диапазон изменения $x$	Выражение для проверки
1	$1 + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots + \frac{x^n}{n!} + \dots$	$-\infty < x < \infty$	$e^x$
2	$1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots + (-1)^n \frac{x^{2n}}{(2n)!} + \dots$	$-\infty < x < \infty$	$\cos x$
3	$2 \left[ \frac{1}{x} + \frac{1}{3x^3} + \frac{1}{5x^5} + \dots + \frac{1}{(2n+1)x^{2n+1}} + \dots \right]$	$ x  > 1$	$\ln \left( \frac{x+1}{x-1} \right)$
4	$x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} + \dots$	$-\infty < x < \infty$	$\sin x$



Вариант	Бесконечная сумма	Диапазон изменения $x$	Выражение для проверки
5	$1 - \frac{1}{x} + \frac{1}{x^2} - \frac{1}{x^3} + \dots + \left(-\frac{1}{x}\right)^n + \dots$	$ x  > 1$	$\frac{x}{x+1}$
6	$2 \left[ x + \frac{x^3}{3} + \frac{x^5}{5} + \frac{x^7}{7} + \dots + \frac{x^{2n+1}}{2n+1} + \dots \right]$	$ x  < 1$	$\ln \left( \frac{1+x}{1-x} \right)$
7	$1 + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \dots + \frac{x^{2n+1}}{(2n+1)!} + \dots$	$-\infty < x < \infty$	$\frac{e^x - e^{-x}}{2}$
8	$1 + \frac{x \ln a}{1!} + \frac{(x \ln a)^2}{2!} + \dots + \frac{(x \ln a)^n}{n!} + \dots$	$-\infty < x < \infty$	$a^x$
9	$x + \frac{x^2}{2} + \frac{x^3}{3} + \frac{x^4}{4} + \dots + \frac{x^n}{n} + \dots$	$-1 \leq x < 1$	$-\ln(1-x)$
10	$\cos x + \frac{1}{2} \cos 2x + \frac{1}{3} \cos 3x + \dots$ $\dots + \frac{1}{n} \cos nx + \dots$	$0 < x \leq \pi$	$-\ln \left( 2 \sin \frac{x}{2} \right)$
11	$1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \frac{x^6}{6!} + \dots + \frac{x^{2n}}{(2n)!} + \dots$	$-\infty < x < \infty$	$\frac{e^x + e^{-x}}{2}$
12	$\cos x - \frac{1}{2} \cos 2x + \frac{1}{3} \cos 3x - \frac{1}{4} \cos 4x + \dots$ $\dots + (-1)^{n+1} \frac{1}{n} \cos nx + \dots$	$0 \leq x < \pi$	$\ln \left( 2 \cos \frac{x}{2} \right)$
13	$x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots + (-1)^{n+1} \frac{x^n}{n} + \dots$	$-1 < x \leq 1$	$\ln(1+x)$
14	$(x-1) - \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3} - \frac{(x-1)^4}{4} + \dots$ $\dots + (-1)^{n+1} \frac{(x-1)^n}{n} + \dots$	$0 < x \leq 2$	$\ln x$
15	$x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots + (-1)^n \frac{x^{2n+1}}{2n+1} + \dots$	$ x  < 1$	$\operatorname{arctg}(x)$

Вариант	Бесконечная сумма	Диапазон изменения $x$	Выражение для проверки
16	$\cos x + \frac{1}{3} \cos 3x + \frac{1}{5} \cos 5x + \dots$ $\dots + \frac{1}{2n+1} \cos[(2n+1)x] + \dots$	$0 < x < \pi$	$\frac{1}{2} \ln \left( \operatorname{ctg} \frac{x}{2} \right)$
17	$1 + x + x^2 + x^3 + \dots + x^n + \dots$	$ x  < 1$	$\frac{1}{1-x}$
18	$1 + x \cos x + x^2 \cos 2x + \dots$ $\dots + x^n \cos nx + \dots$	$ x  < 1$	$\frac{1 - x \cos x}{1 - 2x \cos x + x^2}$
19	$1 + \frac{1}{2} \left[ 2 \cdot 3x + 3 \cdot 4x^2 + 4 \cdot 5x^3 + \dots \right]$ $\dots + (n+1) \cdot (n+2)x^n + \dots$	$ x  < 1$	$\frac{1}{(1-x)^3}$
20	$1 + \frac{1}{6} \left[ 2 \cdot 3 \cdot 4x + 3 \cdot 4 \cdot 5x^2 + \dots \right]$ $\dots + (n+1) \cdot (n+2) \cdot (n+3)x^n + \dots$	$ x  < 1$	$\frac{1}{(1-x)^4}$
21	$x \sin x + x^2 \sin 2x + \dots + x^n \sin nx + \dots$	$ x  < 1$	$\frac{x \sin x}{1 - 2x \cos x + x^2}$
22	$1 - (1-x) + (1-x)^2 -$ $-(1-x)^3 + \dots + (-1)^n (1-x)^n + \dots$	$0 < x < 1$	$\frac{1}{2-x}$
23	$1 - x + x^2 - x^3 + \dots \pm x^n \mp \dots$	$ x  < 1$	$\frac{1}{1+x}$
24	$1 + 2x + 3x^2 + 4x^3 + \dots + nx^{n-1} + \dots$	$ x  < 1$	$\frac{1}{(1-x)^2}$
25	$x + x^3 + x^5 + \dots + x^{2n+1} + \dots$	$ x  < 1$	$\frac{x}{1-x^2}$

## КОНТРОЛЬНЫЕ ЗАДАНИЯ

Вариант 1. Корень  $n$ -й степени

$$y = \sqrt[n]{x}$$

из числа  $x$  является пределом последовательности  $y_0, y_1, y_2, \dots, y_k, \dots$ , каждый член которой определяется итерацией

$$y_{k+1} = y_k + \frac{1}{n} \left( \frac{x}{y_k^{n-1}} - y_k \right), \quad y_0 = x.$$

Определить с точностью до  $\varepsilon = 10^{-3}$  корень 2-й и 4-й степени из числа  $\pi/3$  и число итераций, необходимых при этом.

Вариант 2. Вычислить значение функции  $\sin(z)$ , используя разложение ее в степенной ряд:

$$\sin z = \sum_{n=0}^{\infty} (-1)^n \frac{z^{2n+1}}{(2n+1)!}.$$

Определить число итераций, необходимых для вычисления  $\sin(0,3)$  с точностью до  $\varepsilon = 10^{-4}$ .

Вариант 3. Вычислить значение функции  $\sin(z)$ , используя разложение ее в бесконечное произведение:

$$\sin z = \prod_{k=1}^{\infty} \left[ 1 - \left( \frac{z}{k\pi} \right)^2 \right].$$

Определить число итераций, необходимых для вычисления  $\sin(0,3)$  с точностью до  $\varepsilon = 10^{-4}$ .

Вариант 4. Вычислить значение функции  $\cos x$ , используя разложение ее в степенной ряд:

$$\cos x = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!}.$$

Определить число итераций, необходимых для вычисления  $\cos(0,3)$  с точностью до  $\varepsilon = 10^{-4}$ .

Вариант 5. Для уравнения  $x^2 - \cos x = 0$  уточнить корень методом итераций на интервале  $[0, 2]$  для  $\varepsilon = 10^{-4}$ .

Вариант 6. Вычислить значение функции  $\cos x$ , используя разложение ее в бесконечное произведение:

$$\cos x = \prod_{k=0}^{\infty} \left\{ 1 - \left[ \frac{2x}{\pi(2k+1)} \right]^2 \right\}.$$

Определить число итераций, необходимых для вычисления  $\cos(0,3)$  с точностью до  $\varepsilon = 10^{-4}$ .

Вариант 7. Вычислить значение гиперболического синуса, используя разложение его в степенной ряд

$$\operatorname{sh} x = \sum_{n=0}^{\infty} \frac{x^{2n+1}}{(2n+1)!}.$$

Задание выполнить для  $x = 1,75$  с погрешностью  $\varepsilon = 10^{-3}$ .

Вариант 8. Вычислить значение гиперболического синуса по формуле

$$\operatorname{sh} x = \frac{e^x - e^{-x}}{2},$$

используя разложение функции  $e^x$  в степенной ряд

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}.$$

Задание выполнить для  $x = 1,75$  с погрешностью  $\varepsilon = 10^{-3}$ .

Вариант 9. Вычислить значение гиперболического косинуса по формуле

$$\operatorname{ch} x = \sum_{n=0}^{\infty} \frac{x^{2n}}{(2n)!}.$$

Задание выполнить для  $x = 1,75$  с погрешностью  $\varepsilon = 10^{-3}$ .

Вариант 10. Вычислить значение интегрального косинуса

$$Ci(x) = -\int_x^{\infty} \frac{\cos t}{t} dt,$$

используя разложение

$$Ci(x) = C + \ln x - \frac{1}{2!} \frac{x^2}{2} + \frac{1}{4!} \frac{x^4}{4} \mp \dots$$

для аргумента  $x = 3$  с погрешностью  $\varepsilon = 10^{-3}$ . Постоянную Эйлера – Маскерони принять равной  $C = 0,577216$ .

Вариант 11. По определению Эйлера гамма-функция

$$\Gamma(x) = \lim_{n \rightarrow \infty} \frac{n!}{x(x+1)(x+2) \cdots (x+n)} n^x.$$

Составить рекуррентное соотношение и вычислить  $\Gamma(2,5)$  с погрешностью  $\varepsilon = 10^{-7}$ .

Вариант 12. Вычислить интеграл Френеля для  $x = 1,55$  с погрешностью  $\varepsilon = 10^{-4}$  по формуле

$$\frac{1}{2\sqrt{x}} \int_0^x \frac{\sin t}{\sqrt{t}} dt = \frac{x}{3} - \frac{1}{7} \frac{x^3}{3!} + \frac{1}{11} \frac{x^5}{5!} \mp \dots$$

Вариант 13. Вычислить интеграл Френеля для  $x = 1,55$  с погрешностью  $\varepsilon = 10^{-4}$  по формуле

$$\frac{1}{2\sqrt{x}} \int_0^x \frac{\cos t}{\sqrt{t}} dt = 1 - \frac{1}{5} \frac{x^2}{2!} + \frac{1}{9} \frac{x^4}{4!} - \frac{1}{13} \frac{x^6}{6!} \pm \dots$$

Вариант 14. Вычислить значение функции  $\arctg(x)$  с погрешностью  $\varepsilon = 10^{-3}$  и определить число требуемых для этого итераций при  $x = 0,5$ ;  $x = 0,7$ .

$$\arctg x = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)}.$$

Вариант 15. Вычислить значение функции  $\arcsin(x)$  с погрешностью  $\varepsilon = 10^{-3}$  и определить число требуемых для этого итераций при  $x = 0,5$ ;  $x = 0,7$ .

$$\arcsin x = x + \frac{1}{2} \frac{x^3}{3} + \frac{1}{2} \frac{3}{4} \frac{x^5}{5} + \frac{1}{2} \frac{3}{4} \frac{5}{6} \frac{x^7}{7} + \dots$$

Вариант 16. Для уравнения приведенного в задании 5, и найденного интервала  $(a, b)$  уточнить корень  $x$  с погрешностью  $\varepsilon = 10^{-4}$  методом половинного деления.

Полученный корень  $x$  проверить путем подстановки его значения в уравнение  $f(x) = 0$ .

Вариант 17. Для уравнения, приведенного в задании 5, и найденного интервала  $(a, b)$  уточнить корень  $x$  с погрешностью  $\varepsilon = 10^{-4}$  методом хорд (пропорциональных частей).

Через точки  $(a, f(a))$  и  $(b, f(b))$  проводим прямую и находим значение  $x$  точки пересечения данной прямой с осью  $Ox$  по формуле

$$x = a - \frac{f(a)(b-a)}{f(b)-f(a)}.$$

Если  $f(a)f(x) < 0$ , то правую границу  $b$  переносят в точку  $x$ , т. е. выполняют присваивание  $b := x$ , в противном случае уменьшают интервал

за счет присвоения  $a := x$ . Искомый корень  $x$  определяется с погрешностью, равной разности двух его соседних значений. Если полученная погрешность больше заданной, то вновь применяют вышеприведенную формулу.

Вариант 18. Выполнить задание 5 для уравнения  $1 - x^2 + \sin x = 0$  на интервале  $(0, \pi/2)$ .

Вариант 19. Вычислить значение функции

$$\ln z = -2 \sum_{k=1}^{\infty} \frac{1}{2k-1} \left( \frac{1-z}{1+z} \right)^{2k-1}$$

с допустимой погрешностью  $\varepsilon = 10^{-4}$ . Процесс суммирования прекращается, как только выполнится неравенство  $|u_k| < 4\varepsilon$ , где  $u_k$  – текущий член ряда суммирования. Найти  $\ln(5)$ ,  $\ln(0,5)$  и определить число слагаемых, учитываемых в обоих случаях.

Вариант 20. Определить значение функции  $y = f(x)$  при  $x=2$ , если эта функция задана неявно в виде

$$(x-1)^2 + (y-2)^2 - 9 = 0.$$

Применить метод итераций, при котором уравнение  $F(x,y)=0$  представляется следующей итерационной формулой:

$$y_{n+1} = y_n - \frac{F(x, y_n)}{F'_y(x, y_n)}.$$

Начальное приближение задать  $y_0 = 1$ . Процесс итерации продолжать до тех пор, пока два последовательных приближения  $y_{n+1}$  и  $y_n$  не совпадут между собой с точностью до  $\varepsilon = 10^{-2}$ .

Вариант 21. Вычислить значение функции  $\sqrt[3]{x}$  с погрешностью  $\varepsilon = 10^{-3}$  и определить число требуемых для этого итераций при  $x = 0,5$ ;  $x = 0,7$ .

$$\sqrt[3]{x} = 1 + \frac{1}{3}(x-1) - \frac{1}{2!} \frac{2}{3^2}(x-1)^2 + \frac{1}{3!} \frac{2 \cdot 5}{3^3}(x-1)^3 + \frac{1}{4!} \frac{2 \cdot 5 \cdot 8}{3^4}(x-1)^4 + \dots$$

Вариант 22. Для уравнения  $x^2 - \cos x = 0$  уточнить корень методом половинного деления на интервале  $[0, 2]$  для  $\varepsilon = 10^{-4}$ .

Вариант 23. Вычислить значение интеграла  $\int_0^x e^{-x^2} dx$  с точностью

до  $\varepsilon = 0,5 \cdot 10^{-4}$ , используя его разложение в ряд:

$$\int_0^x e^{-x^2} dx = x - \frac{1}{1!} \frac{x^3}{3} + \frac{1}{2!} \frac{x^5}{5} - \dots + (-1)^n \frac{1}{n!} \frac{x^{2n+1}}{2n+1} + \dots$$

Вариант 24. Вычислить с погрешностью  $\varepsilon = 10^{-5}$  ряд Фурье для функции  $f(x) = x$ , имеющий следующий вид:

$$2 \left[ \frac{1}{1} \sin x - \frac{1}{2} \sin 2x + \frac{1}{3} \sin 3x - \frac{1}{4} \sin 4x + \dots + \frac{(-1)^{n+1}}{n} \sin nx + \dots \right],$$

для  $x = 1$ .

Вариант 25. Вычислить с погрешностью  $\varepsilon = 10^{-5}$  ряд Фурье для функции  $f(x) = |x|$ , имеющий следующий вид:

$$\frac{\pi}{2} - \frac{4}{\pi} \left[ \frac{\cos x}{1^2} + \frac{\cos 3x}{3^2} + \dots + \frac{\cos(2n-1)x}{(2n-1)^2} + \dots \right],$$

для  $x = -1$ .

## Тема 15. КОНСТРУИРОВАНИЕ ПРОГРАММ СО СТРУКТУРОЙ ВЛОЖЕННЫХ ЦИКЛОВ

### ПРИМЕР СОСТАВЛЕНИЯ ПРОГРАММЫ

Пусть необходимо вычислить значение функции

$$y = \frac{x+1,5}{3,75} \sum_{k=0}^{10} \sum_{n=0}^5 (k+n)x^{k+n}$$

при некоторых значениях аргумента  $x = x_0(h_x)x_n$ .

Проектирование алгоритма решения этой задачи начинаем с составления циклической структуры с постусловием, которая решает в общем виде табулирование функции  $y = f(x)$  (рис. 15.1). Значение двойной суммы вычислим с помощью вложенного цикла по параметрам  $k$  и  $n$ , являющимися одновременно переменными суммирования. В качестве начального значения для суммы  $S$  берем нулевое значение. Оба цикла ( $k$  и  $n$ ) являются циклическими структурами с заголовком и целиком расположены внутри тела цикла по параметру  $x$ . Следует обратить внимание на то, что задание начальных значений переменных внутреннего цикла производится непосредственно *перед* входом в *данный* цикл.

Составим программу:

```
{ ***** }
{ Цель:   вычисление двойной суммы при          }
{         различных значениях аргумента x.      }
{ Переменные:                                     }
{   S - сумма; S1 - промежуточная сумма;         }
{   ns, ns1 - пределы внешней и внутренней сумм; }
{   k,n - переменные суммирования; xk - x в степени }
{   k; x1 - x в степени n; x= x0(hx)xn - параметры }
{   аргумента; y - функция аргумента x.          }
{ Программист: Федоров Ф.Ф.                      }
{ Дата написания: 23 января 2006                  }
{ ***** }
```

**Program** Lab15;

**Var**

hx, S, S1, x, x0, x1, xk, xn, y:real;  
k, n, ns, ns1:byte;

**Begin** {Lab15}

{Ввод и эхо-печать исходных данных}



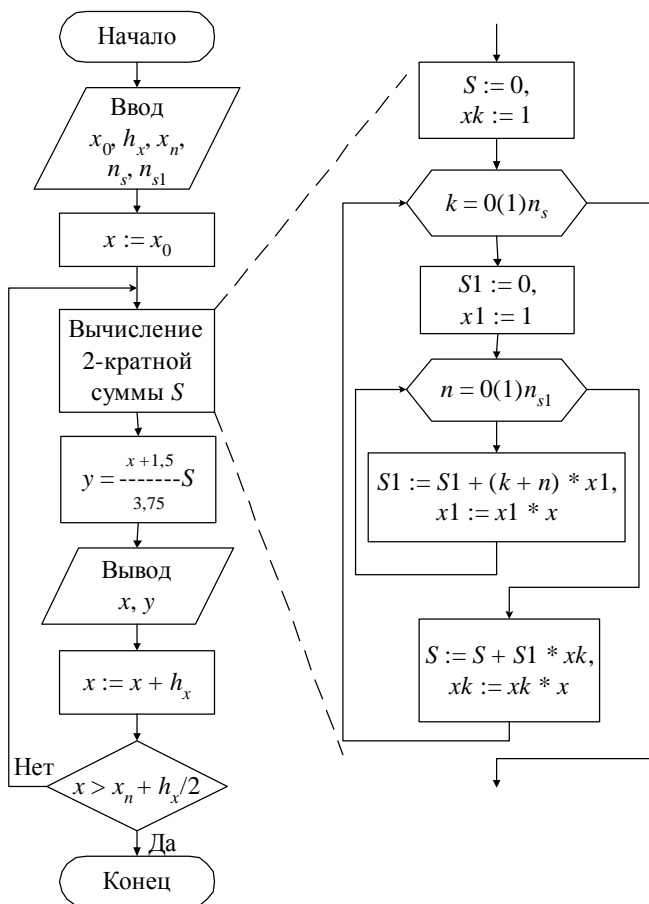


Рис. 15.1. Схема алгоритма вычисления кратной суммы

```

Write('x0=');
Read(x0);
Write('hx=');
Read(hx);
Write('xn=');
Read(xn);
Write('ns=');
Read(ns);
Write('nsl=');
Read(nsl);
WriteLn('x0=', x0, ' hx=', hx, ' xn=', xn);
    
```

```

WriteLn('ns=', ns, ' ns1=', ns1);
{Табулирование функции}
x:=x0;
repeat
  {Вычисление двойной суммы}
  S:=0;
  xk:=1;
  for k:=0 to ns do
    begin
      S1:=0;
      x1:=1;
      for n:=0 to ns1 do
        begin
          S1:=S1+(k+n)*x1;
          x1:=x1*x
        end;
      S:=S+S1*xk;
      xk:=xk*x
    end;
  y:=(x+1.5)*s/3.75;
  WriteLn('x=', x, ' y=', y);
  x:=x+hx;
until x>xn+hx/2
End. {Lab15}

```

## ЗАДАНИЯ

В процессе выполнения задания необходимо осуществить нисходящее проектирование алгоритма табулирования сложной функции и по разработанному алгоритму записать текст программы.

Вариант 1.

$$Y = \frac{ax}{\sqrt{x+a}},$$

$$\text{где } a = \begin{cases} \sum_{k=1}^{12} \frac{2kx}{x+k^2}, & \text{если } x < 1, \\ 1, & \text{если } x \geq 1; \end{cases} \quad x = 0,2(0,2)1,6.$$

Вариант 2.

$$Z = \begin{cases} \sum_{n=1}^{10} \frac{a^2}{a^n - 5}, & \text{если } a < 4; \\ \frac{a+1}{a} \sum_{n=1}^8 \frac{a^n}{2n}, & \text{если } a \geq 4; \end{cases} \quad a = 2(0,5)8.$$

Вариант 3.

$$Y = \begin{cases} \frac{x^2 + 3x + 4}{5x + 1} \sum_{n=0}^{\infty} \frac{x^{2n}}{4n + 1}, & \text{если } x < 0.5; \\ \frac{x}{3} \sum_{n=1}^{10} \frac{x^n}{n}, & \text{если } x \geq 0.5. \end{cases}$$

$x = 0,1(0,15)1,6$ ; погрешность вычисления бесконечной суммы  $\varepsilon = 10^{-4}$ .

Вариант 4.

$$S = \begin{cases} \sum_{n=1}^8 \frac{nx}{n^2 x + 1}, & \text{если } x \leq 0,5; \\ \frac{x+1}{4} \sum_{k=0}^6 \frac{x}{k+1}, & \text{если } x > 0,5; \end{cases} \quad x = 0,1(0,1)0,9.$$

Вариант 5.

$$Z = \frac{x}{2} \sum_{n=1}^8 \frac{x^n}{2n} + \frac{x^2 + 1}{3} \sum_{n=1}^6 \frac{x^{2n}}{4n}; \quad x = -2(0,5)3,5.$$

Вариант 6.

$$Y = Ax^2 + Bx + C,$$

где  $A = \sum_{k=0}^{10} \frac{x^k}{k+1}$ ;  $B = \sum_{n=1}^8 \frac{x^{2n}}{2n+1}$ ;  $C = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{n!}$ ;  $x = 0,1(0,05)0,5$ .

Вариант 7.

$$Z = x^2 \cos(ax + t),$$

где  $a = \begin{cases} 2,5, & \text{если } |x| \leq 1; \\ 0,5, & \text{если } |x| > 1; \end{cases}$   $t = \sum_{n=0}^{15} \frac{x^{2n}}{3n+2}$ ;  $x = 0,1(0,1)0,8$ .

Вариант 8.

$$Y = \frac{x}{2} \sum_{n=1}^8 \frac{x^{n+1}}{n+2} + \frac{x+1}{3} \sum_{n=1}^7 \frac{x^{2n}}{n+3}; \quad x = 0,1(0,1)1.$$

Вариант 9.

$$Y = a \sum_{i=1}^{10} \sum_{j=1}^i \frac{i}{j + a^j}; \quad a = 1, 1(0, 1)1, 6.$$

Вариант 10.

$$Y = \begin{cases} \sum_{n=0}^{\infty} \frac{x^{2n}}{(2n)!}, & \text{если } x < 1; \\ \sum_{n=0}^{\infty} \frac{x}{(3n)!}, & \text{если } x \geq 1; \end{cases}$$

$x = 0, 1(0, 1)0, 9$ , а погрешность вычисления бесконечных сумм  $\varepsilon = 10^{-4}$ .

Вариант 11.

$$Z = \begin{cases} \sum_{n=1}^{10} \left(\frac{x}{n}\right)^n, & \text{если } x \leq 2; \\ \sum_{n=0}^8 \frac{x^n}{n+2}, & \text{если } x > 2; \end{cases} \quad x = 0, 5(0, 25)3.$$

Вариант 12.

$$Y = \begin{cases} \frac{x}{2} \sum_{n=1}^{20} \frac{n}{n+1}, & \text{если } x \leq 0, 5; \\ \sum_{n=1}^8 \frac{x^n}{n}, & \text{если } x > 0, 5; \end{cases} \quad x = -0, 5(0, 1)3.$$

Вариант 13.

$$Y = \begin{cases} \frac{x}{x+1} \prod_{n=1}^8 \left(1 + \frac{x^n}{n}\right), & \text{если } x > 0; \\ \sum_{n=0}^7 \frac{x^n}{n+1}, & \text{если } x \leq 0; \end{cases} \quad x = -0, 5(0, 1)0, 5.$$

Вариант 14.

$$Y = \frac{x^2 + 1}{x} \prod_{n=1}^{10} \left(1 + \frac{x^n}{n+1}\right) + \frac{x^3 + 2}{4} \sum_{n=1}^{10} \frac{x^n}{n}; \quad x = 0, 5(0, 5)4.$$

Вариант 15.

$$Y = \begin{cases} \sum_{n=0}^{\infty} (-1)^n \frac{x^n}{n}, & \text{если } |x| \leq 1; \\ \sin^2 x + \cos^3(x+1) & \text{при других } x; \end{cases} \quad x = 0,5(0,1)1,6.$$

Вариант 16.

$$Y = \begin{cases} \frac{x+3}{x^2+4} \sum_{n=1}^{10} \frac{(2x+1)^n}{2n+1}, & \text{если } |x| < 0,5; \\ \frac{x+1}{5} \prod_{n=1}^8 \left( 1 + \frac{(x+1)^n}{n+1} \right) & \text{при других } x; \end{cases} \quad x = 0(0,1)1,2.$$

Вариант 17.

$$Z = \frac{x+2,5 \cdot 10^{-3}}{x^2+3,8 \cdot 10^{-4}} \sum_{n=1}^{10} \frac{(x+2)^2}{n+4} + \frac{x+1}{x+2} \sum_{n=1}^7 \frac{(x+1)^n}{n}; \quad x = 0,1(0,05)0,5.$$

Вариант 18.

$$P = \prod_{n=1}^{10} \left( 1 + \frac{x^n}{n+2} \right) + \frac{x+1}{2,5} \sum_{n=1}^{\infty} \frac{x^{2n}}{(2n)!}, \quad x = 0,1(0,1)1.$$

Вариант 19.

$$Y = \begin{cases} \sum_{n=0}^{10} \frac{x^{n+1}}{2n+3}, & \text{если } 0,1 < x < 0,5; \\ \frac{x+1}{3} \sum_{n=1}^7 \frac{x^n + x^{2n}}{8n+1}, & \text{если } 0,5 \leq x < 1; \end{cases}$$

при других  $x$  функция  $y = y(x)$  не определена;  $x = 0,05(0,05)0,6$ .

Вариант 20.

$$Y = \begin{cases} \sum_{n=0}^{50} \frac{1}{(2n+1)^2}, & \text{если } x < 0,5; \\ \sum_{n=0}^{10} \ln(x) \cdot \sin k(x-a), & \text{если } x \geq 0,5; \end{cases}$$

$a$  – задается произвольно;  $x = 0,4(0,2)1,8$ .

Вариант 21.

$$Y = \sum_{k=1}^{10} (1 + k \ln x) + \frac{x+1}{3} \sum_{n=1}^7 \frac{x^n}{2n+1}; \quad x = 1, 1(0,1)1, 2.$$

Вариант 22.

$$y = \prod_{k=0}^8 \prod_{n=1}^5 (k + \frac{\sin x}{n}); \quad x = -0,5(1)4,5.$$

Вариант 23.

$$Q = \begin{cases} \frac{a}{3} \sum_{n=1}^{10} \frac{a^2}{a^n - 5}, & \text{если } a < 0,4; \\ \frac{a+1}{a} \sum_{n=1}^8 \left[ 1 + \frac{(a-1)^n}{n} \right], & \text{если } a \geq 0,4; \end{cases} \quad a = 0,2(0,05)0,8.$$

Вариант 24.

$$S = \begin{cases} \sum_{n=1}^8 \frac{nx}{nx^2 + 1}, & \text{если } x \leq 0,5; \\ \sum_{k=0}^8 \frac{x^k}{k+1}, & \text{если } x > 0,5; \end{cases} \quad x = 0,1(0,1)0,9.$$

Вариант 25.

$$P = \frac{x}{2} \prod_{n=1}^8 \left( 1 + \frac{x^n}{n} \right) + \frac{x+1}{x^2+1} \prod_{n=1}^7 \left( 1 + \frac{x^2}{2n} \right); \quad x = 0,1(0,1)1, 2.$$

## Тема 16. ОРГАНИЗАЦИЯ ВВОДА-ВЫВОДА. ПРЕДСТАВЛЕНИЕ РЕЗУЛЬТАТОВ РЕШЕНИЯ

### ПРИМЕРЫ СОСТАВЛЕНИЯ ПРОГРАММ

**Пример 16.1.** Рассмотрим пример табулирования функций

$$s1 = 4\pi r^2;$$

$$s2 = \pi r^2 / 2$$

и организации вывода результатов в виде таблицы, где параметр  $r$  изменяется от 1.00 до 1.4 с шагом 0.05.

Таблица должна состоять из заголовка, строк и окончания. По мере вычисления значений функции  $s1$  и  $s2$  в цикле по параметру  $r$  выполняется печать строки таблицы.

Схема алгоритма достаточно проста и приведена на рис. 16.1.

Для определения формы печати результатов строится стилизованная форма таблицы с использованием символов языка Турбо Паскаль. Горизонтальные линии можно представить символом '-' (минус), вертикальные – символом '|' (вертикальная черта). Учитывая возможные значения параметра  $r$ , будем выводить его значения в отдельном столбце таблицы в форме  $d.dd$ , где  $d$  – десятичная цифра. Значения функций  $s1$  и  $s2$  заранее неизвестны, поэтому будем выводить их в форме  $dd.ddd$ .

С целью упрощения операторов вывода таблицы введем вспомогательную константу  $P$ , содержащую два пробела. Первую строку заголовка таблицы реализуем печатью 31 символа '-', вторую – печатью следующего текста:

```
|-----R-----|-----S1-----|-----S2-----|
```

Третья строка повторяет первую. Четвертая строка реализуется оператором вывода

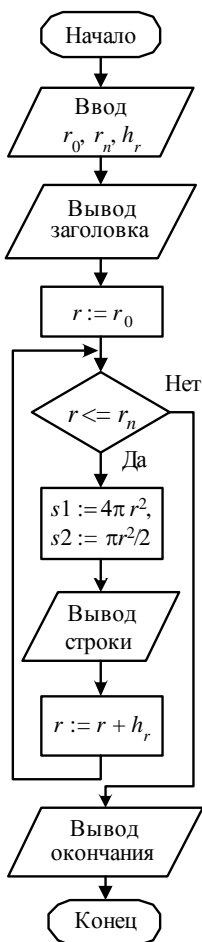


Рис. 16.1. Алгоритм  
вывода таблицы

```
WriteLn(' | ',P,r:4:2,P,' | ',P,s1:6:3,P,' | ',P,s2:5:3,P,' | ');
```

Печать окончания таблицы реализуется аналогично печати первой и третьей строк таблицы.

Для организации позиционирования таблицы на экране вводится константа *T*, содержащая 12 пробелов.

В целом программа табулирования функций и печати результатов в виде таблицы имеет следующий вид:

```
Program tabl;
Const
    P='    ';
    S='-----';
    T='                ';
Var
    hr,r,r0,rn,s1,s2:real;
Begin{tabl}
    WriteLn(T,'ИВАНОВ И.И., ВАР.15');
    WriteLn;
    WriteLn('ВВОД ИСХОДНЫХ ДАННЫХ');
    Write('?');
    Read(r0,hr,rn);
    WriteLn(T,'ТАБЛИЦА РЕЗУЛЬТАТОВ');
    WriteLn;
    {Вывод заголовка таблицы}
    WriteLn(T,S);
    WriteLn(T,' | ', 'R':5, ' | ':4, 'S1':6, ' | ':5, 'S2':5, ' | ':5);
    WriteLn(T,S);
    r:=r0;
    while r<rn do
        begin
            s1:=4*pi*r*r;
            s2:=pi*r*r/2;
        {Вывод строки таблицы}
        WriteLn(T,' | ',P,r:4:2,P,' | ',P,s1:6:3,P,' | ',P,s2:5:3,P,' | ');
            r:=r+hr
        end;
    {Окончание таблицы}
    WriteLn(T,S)
End.{tabl}
```

Результат работы программы представлен на рис. 16.2.

Используя символы псевдографики, коды которых приведены в табл. 16.1, можно представить таблицу в более привычном графическом виде. Для этого необходимо помещать в оператор **Write** функцию **Chr**, имеющую формат **Chr(<код символа>)**, которая обеспечит вывод на эк-



ран именно того символа псевдографики, код которого является аргументом функции.

```
ИВАНОВ И.И., ВАРИАНТ 15
ВВОД ИСХОДНЫХ ДАННЫХ
? 1.,.05,1.4
ТАБЛИЦА РЕЗУЛЬТАТОВ

-----
| R | S1 | S2 |
-----
| 1.00 | 12.566 | 1.571 |
| 1.05 | 13.854 | 1.732 |
| 1.10 | 15.205 | 1.901 |
| 1.15 | 16.619 | 2.077 |
| 1.20 | 18.096 | 2.262 |
| 1.25 | 19.635 | 2.454 |
| 1.30 | 21.237 | 2.655 |
| 1.35 | 22.902 | 2.863 |
| 1.40 | 24.630 | 3.079 |
-----
```

Рис. 16.2. Результат работы программы Tabl

Таблица 16.1

DEC/	0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240	
/HEX	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0	
0	0	NUL	DLE		0	@	P		p	A	P	a		Л	л	р	ё
1	1	SON	DC1	!	1	A	Q	a	q	Б	С	б		┐	└	с	ё
2	2	STX	DC2	“	2	B	R	b	r	В	Т	в		┘	┌	т	€
3	3	ETX	DC3	#	3	C	S	c	s	Г	У	г		└	┘	у	€
4	4	EOT	DC4	\$	4	D	T	d	t	Д	Ф	д	└	┐	ф	ї	
5	5	ENQ		%	5	E	U	e	u	Е	Х	е	└	┐	х	ї	
6	6	ACK	SYN	&	6	F	V	f	v	Ж	Ц	ж	└	┐	ц	ў	
7	7	BEL	ETB	^	7	G	W	g	w	З	Ч	з	└	┐	ч	ў	
8	8	BS	CAN	(	8	H	X	h	x	И	Ш	и	└	┐	ш	°	
9	9	HT	EM	)	9	I	Y	i	y	Й	Щ	й	└	┐	щ	я	
10	A	LF	SUB	*	:	J	Z	j	z	К	Ъ	к	└	┐	ъ	?	
11	B	VT	ESC	+	;	K	[	k	{	Л	Ы	л	└	┐	ы	_	
12	C	FF	FS	,	<	L	\	l		М	Ь	м	└	┐	ь	№	
13	D	CR	?	-	=	M	]	m	}	Н	Э	н	└	┐	э	№	
14	E	SO	RS	.	>	N	^	n	~	О	Ю	о	└	┐	ю	_	
15	F	SI		/	?	O	_	o	DEL	П	Я	п	└	┐	я		

Ниже приведена программа, обеспечивающая вывод таблицы в графическом виде, а на рис. 16.3 – результаты ее работы.

```
Program tabl2;
Const
  P='  ';
  T='          ';
  i:integer;
Var
  hr,r,r0,rn,s1,s2:real;
Begin{tabl2}
  WriteLn(T,'ИВАНОВ И.И., ВАР.15');
  WriteLn;
  WriteLn('ВВОД ИСХОДНЫХ ДАННЫХ');
  Write('?');
  Read(r0,hr,rn);
  WriteLn(T,'ТАБЛИЦА РЕЗУЛЬТАТОВ');
  WriteLn;
  {Вывод заголовка таблицы}
  Write(T,Chr(219));
  for i:=1 to 8 do
    Write(Chr(196));
  Write(Chr(194));
  for i:=1 to 10 do
    Write(Chr(196));
  Write(Chr(194));
  for i:=1 to 10 do
    Write(Chr(196));
  WriteLn(Chr(191));
  Write(T,Chr(179),'R':5,Chr(179):4,'S1':6);
  WriteLn(Chr(179):5,'S2':5,Chr(179):5);
  Write(T,Chr(195));
  for i:=1 to 8 do
    Write(Chr(196));
  Write(Chr(197));
  for i:=1 to 10 do
    Write(Chr(196));
  Write(Chr(197));
  for i:=1 to 10 do
    Write(Chr(196));
  WriteLn(Chr(180));
  r:=r0;
  while r<rn do
    begin
      s1:=4*pi*r*r;
      s2:=pi*r*r/2;
```

```

{Вывод строки таблицы}
  Write(T,Chr(179),P,r:4:2,P,Chr(179),P,s1:6:3,P,Chr(179));
  WriteLn(P,s2:5:3,P,Chr(179));
  r:=r+hr
end;
{Окончание таблицы}
  Write(T, Chr(192));
  for i:=1 to 8 do
    Write(Chr(196));
  Write(Chr(193));
  for i:=1 to 10 do
    Write(Chr(196));
  Write(Chr(193));
  for i:=1 to 10 do
    Write(Chr(196));
  WriteLn(Chr(217));
End.{tabl2}

```

ИВАНОВ И.И., ВАРИАНТ 15

ВВОД ИСХОДНЫХ ДАННЫХ

? 1..05.1.4

ТАБЛИЦА РЕЗУЛЬТАТОВ

R	S1	S2
1.00	12.566	1.571
1.05	13.854	1.732
1.10	15.205	1.901
1.15	16.619	2.077
1.20	18.096	2.262
1.25	19.635	2.454
1.30	21.237	2.655
1.35	22.902	2.863
1.40	24.630	3.079

Рис. 16.3. Результат работы программы Tabl2

Однако, как видно из приведенной программы, данный способ формирования таблицы является достаточно громоздким. Более компактного решения можно добиться, если использовать вместо функции Chr помещение в строку вывода непосредственно изображения самого графического символа. Для этого необходимо при нажатой клавише <Alt> набрать код символа на дополнительной клавиатуре справа. Тогда объем программы будет сравним с ее первоначальным вариантом:

```

Program tabl3;
Const
  P='  ';
  T='          ';
Var
  hr,r,r0,rn,s1,s2:real;
Begin{tabl3}
  WriteLn(T,'ИВАНОВ И.И., ВАР.15');
  WriteLn;
  WriteLn('ВВОД ИСХОДНЫХ ДАННЫХ');
  Read(r0,hr,rn);
  WriteLn(T,'ТАБЛИЦА РЕЗУЛЬТАТОВ');
  WriteLn;
  {Вывод заголовка таблицы}
  WriteLn(T,'          ');
  WriteLn(T,'          ');
  WriteLn(T,'          ');
  r:=r0;
  while r<rn do
    begin
      s1:=4*pi*r*r;
      s2:=pi*r*r/2;
    {Вывод строки таблицы}
      WriteLn(T,'|',P,r:4:2,P,'|',P,s1:6:3,P,'|',P,s2:5:3,P,'|');
      r:=r+hr
    end;
  {Окончание таблицы}
  WriteLn(T,'          ');
End.{tabl3}

```

**Пример 16.2.** В качестве примера составим программу организации вывода на экран фигуры, представленной на рис. 16.4.

Для упрощения введем вспомогательные константы, позволяющие изобразить повторяющиеся элементы фигуры:

```

P='00000000';
C='0      0';

```

```

      3
      4
      5
      6
      7
      8
0000000
0      0
0      0
0      0
0      0
0      0
0000000

```

Рис. 16.4. Изображение фигуры  
цифры шесть

Первые шесть строк фигуры изображаются в цикле по параметру  $i = 1(1)6$ , где печатается значение выражения  $i + 2$  в позиции, определяемой выражением  $7 - i$ . Затем на печать выводится значение константы  $P$ . Последующее построение фигуры предполагает многократное повторение печати значения константы  $C$ . Текст программы приведен ниже.

```

Program Fig;
Const
  T='                ';
  P='0000000';
  C='0      0';
Var
  i:integer;
Begin{Fig}
  for i:=1 to 6 do
    WriteLn(T,' ':(7-i),(i+2):1);
    WriteLn(t,p);
    for i:=1 to 5 do
      WriteLn(T,C);
    WriteLn(T,P);
End.{Fig}

```

**Пример 16.3.** Составим программу построения и вывода изображения параллелепипеда (рис. 16.5), используя рассмотренные подходы. Для упрощения процесса построения введем вспомогательные константы:

```

R='I-----I'
K='/ ...../'
V='I ...../'

```

Изображение боковых ребер реализовано с помощью цикла по параметру  $i$ . В теле цикла печатается значение константы  $K$ , номер начала позиции которой задается как  $5 - i$ . Одновременно изображается дальняя вертикальная грань символом  $T$ , номер позиции которого определяется значением параметра  $i$ . Далее выводятся значения константы  $V$  и формируется боковая грань с помощью символа  $'/'$ , значение позиции которого определяется значением параметра цикла.

Для позиционирования объемного изображения в центре экрана используется константа  $P$ , содержащая 10 пробелов. Программа построения объемного изображения представлена ниже.

```

      I----- I
      /...../ I
      /...../ I
      /...../ I
      /...../ I
      /...../ I
I----- I      I
I.....I      /
I.....I      /
I.....I      /
I.....I      /
I.....I      /
I-----I/

```

Рис. 16.5.  
Изображение  
параллелепипеда

```

Program Cub;
Const
    P= '          ';
    R='I-----I';
    K=' / ..... /';
    V='I ..... I';
Var
    i:integer;
Begin{Cub}
    WriteLn(P, ' ':6,R);
    WriteLn(p, ' ':5,K,'I');
    for i:=1 to 4 do
        WriteLn(P, ' ':(5-i),K, ' ':i,'I');
    WriteLn(p,r, ' ':5,'I');
    for i:=1 to 5 do
        WriteLn(P,V, ' ':(6-i), '/' );
    WriteLn(P,R);
End.{Cub}

```

**Пример 16.4.** Пусть необходимо построить график функции

$$y = f(x)$$

в диапазоне от начального значения  $x_n$  до конечного  $x_k$  с шагом  $h_x$ .

Для вывода графика на экран необходимо выполнить следующие шаги:

- 1) задание точности построения и определения числа точек на оси ординат;
- 2) определить минимальное и максимальное значения функции;
- 3) найти период квантования;
- 4) вывести вертикальную ось (ординат);
- 5) вывести горизонтальную ось и график функции.

В общем случае для организации вывода графика функции типа  $f(x)$ , аргумент которой изменяется от  $x_n$  до  $x_k$  с шагом  $h_x$ , можно рекомендовать алгоритм, приведенный на рис. 16.6. Для организации печати графика функции введена дополнительная переменная  $p$ , вычисляемая по формуле  $p = [(f(x) - y_{\min})/dy]$ , где  $[x]$  означает округление  $x$  до ближайшего целого) оператором

```
WriteLn('!':round(f(x)-ymin/dy), '*':p).
```

В качестве примера ниже приведен листинг программы печати значения функции вида  $y = x^2$ .

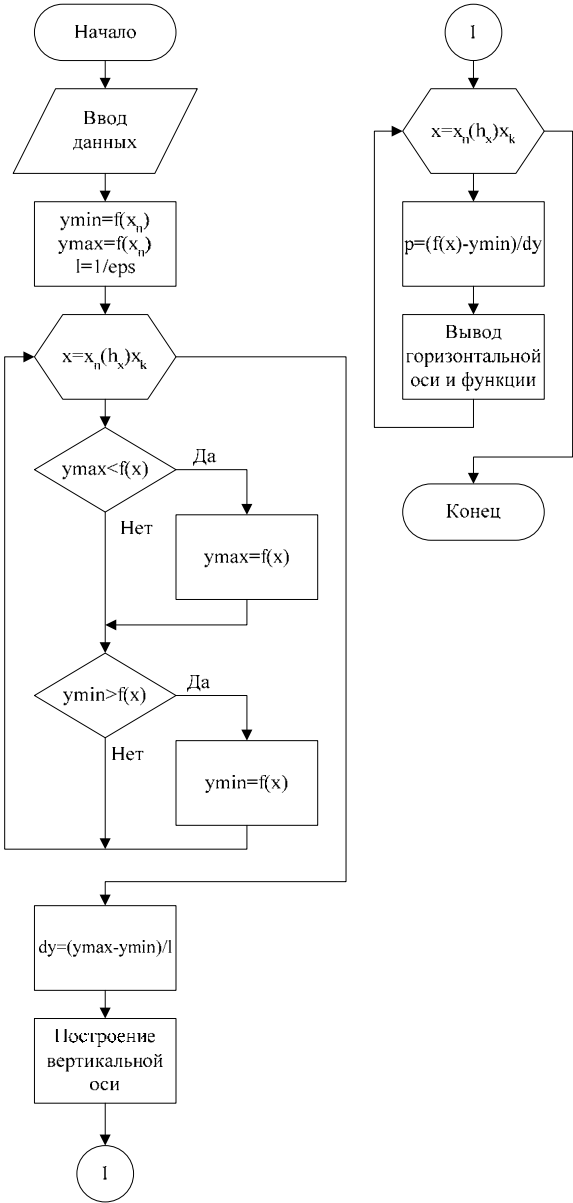


Рис. 16.6. Алгоритм вывода графика функции

```

{Цель: вывод графика функции.}
{Описание параметров и переменных:}
{  eps - погрешность построения; dy - период квантования; }
{  xn,xk - начальное и конечное; i - параметр цикла }
{          значения аргумента;    p - значение функции; }
{  hx - шаг изменения;          k - вспомогательная }
{  l - максимальное число точек;      переменная. }
{          на оси ординат; }
{Программист: Емельяненко С.А.}
{Дата: 30 января 2006 г.}
Program Vef;{Вывод графика}
  Var
    dy, eps, hx, x, xk, xn, y,ymax, ymin:real;
    i, k, l, p:integer;
  {Заданная функция}
  Function F(x:real):real;
    Begin {F}
      F:=x*x
    End; {F}
  Begin{Vef}
    {Ввод данных}
    WriteLn('Точность построения-');
    Write('?');
    Read(eps);
    WriteLn('Ввод xn, hx, xk');
    Write('?');
    Read(xn, hx, xk);
    {Число точек на оси ординат}
    l:=Round(1/eps);
    {Определение min/маx значения функции}
    ymin:=F(xn);
    ymax:=F(xn);
    for i:=1 to round((xk-xn)/hx+1) do
      begin
        y:=F(x)
        if ymax<y
        then
          ymax:=y;
        if ymin>y
        then
          ymin:=y;
        x:=x+hx
      end;
    {Определение периода квантования}
    dy:=(ymax-ymin)/l;
    WriteLn('eps-',eps:4:2,' ymin-',ymin:4:2,

```



```

        ' ymax-',ymax:4:2, ' dy-',dy:4:2);
{Вывод вертикальной оси}
    WriteLn ( ' ':30,'Ось ординат');
    WriteLn ( '0-----> y');
{Вывод графика и горизонтальной оси}
    x:=xn;
    k:=round(-ymin/dy);
    for i:=1 to round((xk-xn)/hx+1) do
        begin
            x:=xn+(i-1)*hx;
            y:=F(x);
            p:=round((y-ymin)/dy);
            WriteLn(' | ':k, '*' :p);
        end;
    WriteLn('v x');
End.{Vef}

```

Задавая  $\text{eps} = 0.03$ ;  $x_n = 1$ ;  $x_k = 4$ ;  $h_x = 0.5$ , получаем следующий график (рис. 16.7).

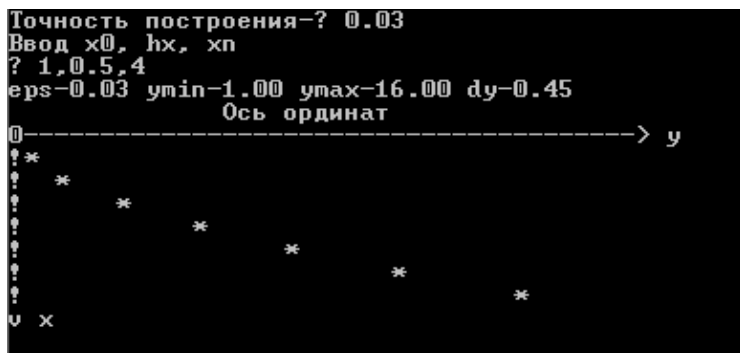


Рис. 16.7. График функции, выведенный программой

Рассмотренный подход построения графика целесообразно применять с использованием массивов, что существенно облегчит процесс организации печати и позволит конструировать более оптимальные программы.

## ЗАДАНИЯ

Каждое задание состоит из двух частей: первая часть определяет вид табулируемых функций, показанных в таблице; вторая – фигуру или график, которые необходимо «нарисовать». Результаты табулирования должны быть представлены в табличной форме.

Вариант	Закон изменения параметра (параметров)	Табулируемые функции
1	$x = 0(\pi / 8)\pi$	$y = \begin{cases} \sin x, & x \leq \pi / 2; \\ \cos x, & x > \pi / 2. \end{cases}$
2	$x = -0.22(0.02)0.3$	$y = x^2; z = x^3$
3	$a = 0(0.55)5.5$	$z = a \cdot \sin x + b \cdot \cos x;$ $x = a + 0, 3;$ $b = \begin{cases} a / 2, & a \leq 2; \\ 2a, & a > 2 \end{cases}$
4	$x = 0.01(0.01)0.1$	$y = e^Z; Z = e^x$
5	$x = 1(1)8$	$y = \ln x; p = \lg x; s = \log_2 x$
6	$g = 0.5(0.1)0.8$ $q = 1(1)3$	$P = \frac{1 + q^2}{1 + \frac{1 + g^3}{1 + g^4}}$
7	$z = 0.99(0.01)1.09$	$y = 1 + z^2;$ $\lambda = (1 + y)^{1.25} + y$
8	$x = 0.1(0.1)0.4$ $y = 1.1(0.2)1.5$	$z = 3x + 4\cos y$
9	$x = 0.25(0.05)0.65$	$y = e^{-ax} \sin ax; a = \begin{cases} 1, & x < 0.5; \\ 2, & x \geq 0.5 \end{cases}$
10	$a = -3.1(0.1)3.4$ $b = 2(3)9$	$x = \begin{cases} \frac{a+b}{3} + b, & a \leq b; \\ 3.7, & a > b \end{cases}$
11	$y = 0.7(0.1)1.5$	$z = f(y) - 6.3;$ $f(y) = \begin{cases} y^2 - 0.3, & y < 0; \\ 0, & 0 \leq y \leq 1; \\ y^2 + y, & y > 1 \end{cases}$
12	$x = 0.15(0.15)1.5$	$y_1 = \operatorname{sh} x; y_2 = \operatorname{ch} x; y_3 = \operatorname{th} x$
13	$x = 1.5(0.5)5.5$	$\alpha = \sqrt{x}; \beta = \sqrt[3]{x}$



6.	6	7.	1111111	8.	1	1	9.	654321
	5 5		22222		2	2		22
	4 4		333		3	3		3 3
	3 3		4		4 4			4 4
	2 2		555		55			5 5
	1 1		66666		654321			6 6
			7777777					

10. Построить домик, используя символы '/', '\', '-', '!'.  
 11. Составить программу построения куба с рисованием невидимых линий фигуры.

12. В поле вывода, состоящем из 10 строк и 20 позиций, построить фигуру, соответствующую цифре «7».

13. В поле вывода, состоящем из 10 строк и 20 позиций, построить фигуру, соответствующую цифре «9».

14.	123456	15.	6	16.	1	1	17.	55
	22 5		55		2	2		4 4
	3 3 4		4 4		3	3		3 3
	4 4 3		3 3		4 4			2 2
	5 52		2 2		55			1 1
	654321		111111		123456			

18. Построить график функции  $y = \ln x$  при изменении аргумента  $x$  от 1,5 до 3,5 с шагом 0,25.

19. Построить фигуру, соответствующую спичечному коробку (параллелепипед).

20.	1	21.	6***6	22.	6	23.	*
	2 2		5*5		5*5		***
	3 * 3		44		4***4		*****
	4 * * 4		33		3*****3		*****
	3 * 3		2*3		2*****2		*****
	2 2		1***1		1*****1		*****
	1						

24. Построить график функции  $y = \sqrt[3]{x}$  для значений аргумента, изменяющегося от 1 до 8 с шагом 1.

25. Построить фигуру, соответствующую изометрии правильной четырехгранной пирамиды.

## Тема 17. ОБРАБОТКА ОДНОМЕРНЫХ МАССИВОВ ДАННЫХ

### ПРИМЕРЫ ПРОГРАММИРОВАНИЯ ЗАДАЧ

**Пример 17.1.** Пусть необходимо составить программу определения количества вхождений каждой строчной буквы латинского алфавита в текст, состоящий из 100 символов.

Для решения данной задачи достаточно создать целочисленный массив  $k$ , индексами которого являются значения букв латинского алфавита от  $a$  до  $z$ , и затем при анализе исходного символьного массива  $Stroka$  увеличивать на единицу содержимое элемента массива  $k$ , индекс которого совпадает со значением элемента массива  $Stroka$ . Полученный массив  $k$  и будет содержать количество повторений каждой буквы.

Схема алгоритма решения задачи приведена на рис. 17.1.

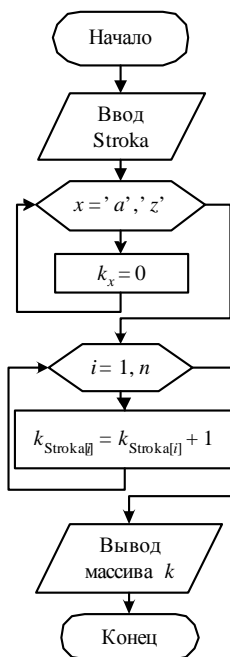


Рис. 17.1. Схема алгоритма подсчета количества букв, входящих в текст

Текст программы будет иметь следующий вид:

```
{Цель: подсчет количества вхождений          }
{   строчных букв в текст.                    }
{Метод: обработка массивов.                  }
{Переменные: Stroka - массив символов;       }
{   k - массив количества вхождения букв;    }
{   i, x - параметры циклов;                 }
{   n - количество обрабатываемых            }
{   символов.                                }
{Программист: Федоров В.Ф.                   }
{Дата: 6 февраля 2006 г.                     }
Program Letters;
Const
    Nmax = 200;
Type
    Latter = 'a' .. 'z';
Var
    Stroka: array [1 .. Nmax] of Latter;
    k: array [Latter] of byte;
    i,n: byte;
    x: Latter;
Begin {Letters}
    WriteLn('Введите количество обрабатываемых символов');
    Read(n);
    WriteLn('Введите', n:4, ' латинских строчных букв');
    for i := 1 to n do
        Read(Stroka[i]);
    WriteLn('В вашем тексте:');
    for i := 1 to n do
        Write(Stroka[i]); {Эхо-печать}
    WriteLn;
    {Обнуление массива k}
    for x := 'a' to 'z' do
        k[x] := 0;
    {Подсчет количества каждого из символов}
    for i := 1 to n do
        Inc(k[Stroka[i]]);
    {Вывод результата}
    for x := 'a' to 'z' do
        WriteLn('число букв', x:2, ' равно', k[x]:4)
End. {Letters}
```

**Пример 17.2.** Имеется массив  $A$ , содержащий  $n$  элементов. Разместить элементы массива в порядке возрастания их значений.

При решении этой задачи воспользуемся сортировкой по методу пузырька. Суть этого метода состоит в организации упорядоченного списка элементов, в который на соответствующие им места добавляются один за другим неотсортированные элементы. На рис. 17.2 дана схема описания алгоритма сортировки методом пузырька.

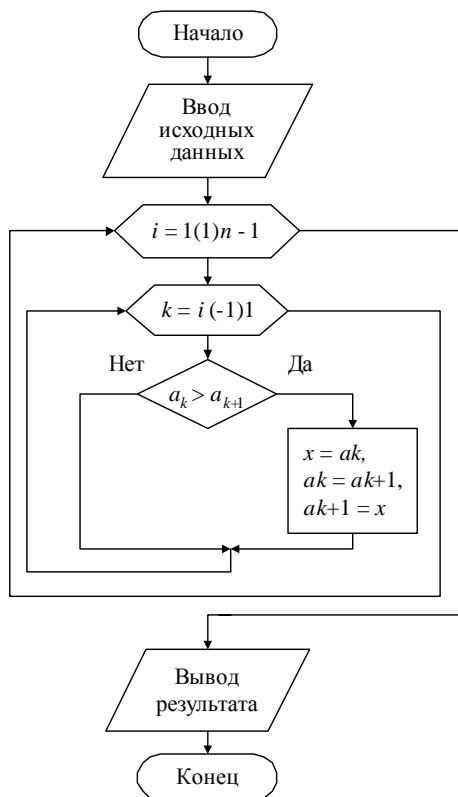


Рис. 17.2. Схема алгоритма сортировки методом пузырька

Текст программы имеет следующий вид:

```

{Цель: сортировка элементов одномерного массива          }
{  в порядке возрастания их значений.                    }
{Метод: сортировка массива методом пузырька              }
{Переменные: a – исходный массив;                        }
{               n – количество обрабатываемых элементов массива; }
{               i, k – параметры циклов;                  }
{               x – вспомогательная переменная.          }
    
```

```

{Программист: Свиридов Е.П.}
{Дата: 8 февраля 2006 г.}
}
Program Sort;
  Const
    Nmax=100;
  Var
    a: array [1..Nmax] of real;
    i, k, n: integer;
    x: real;
  Begin {Sort}
    WriteLn('Задайте количество элементов массива');
    Read(n);
    WriteLn('Введите ', n, ' чисел');
    for i := 1 to n do
      Read(a[i]);
    WriteLn;
    {Эхо-печать}
    WriteLn('Исходный массив:');
    for i := 1 to n do
      Write(a[i]:7:2);
    WriteLn;
    {Сортировка массива}
    for i := 1 to n - 1 do
      for k := i downto 1 do
        if a[k] > a[k + 1]
        then
          begin
            x := a[k];
            a[k] := a[k + 1];
            a[k + 1] := x;
          end;
        else
          continue;
        end;
      end;
    {Вывод результата}
    WriteLn('Отсортированный массив:');
    for i := 1 to n do
      Write(a[i]:7:2);
    WriteLn;
  End. {Sort}

```

**Пример 17.3.** Определить минимальное значение элемента и его номер в одномерном массиве из 20 элементов.

Задача может быть решена последовательным сравнением «текущего» минимального значения со значением элемента и запоминанием минимального значения.

Схема алгоритма показана на рис. 17.3.



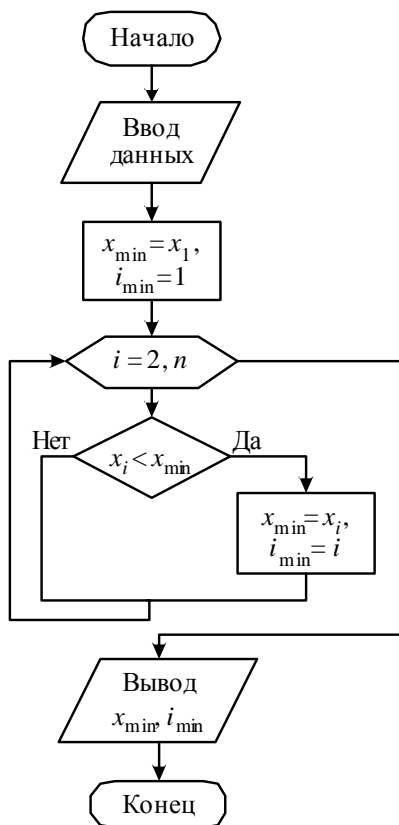


Рис. 17.3. Схема алгоритма определения минимального элемента массива

Программа имеет следующий вид:

```

{Цель: нахождение минимального элемента одномерного массива.}
{Метод: метод минимального элемента.}
{Переменные: x – исходный массив; }
{   n – количество обрабатываемых элементов массива; }
{   i – параметр цикла; xmin – минимальный элемент }
{ массива; imin – его номер. }
{Программист: Сотков Д.Е.}
{Дата: 10 февраля 2006 г.}

```

**Program** MinEl;

**Const**

Nmax = 20;

```

Var
    i, imin, n:integer;
    xmin:real;
    x:array[1..Nmax] of real;
Begin {MinEl}
    WriteLn('Введите количество элементов массива');
    Read(n);
    WriteLn('Введите', n:3, 'элементов массива');
    for i :=1 to n do
        begin
            Write('x[', i, ']=');
            Read(x[i])
        end;
    WriteLn('В массиве:');
    for i :=1 to n do
        Write(x[i]:6:2);
    WriteLn;
    xmin := x[1];
    imin :=1;
    for i := 2 to n do
        if x[i] < xmin
        then
            begin
                xmin := x[i];
                imin := i
            end;
    WriteLn('минимальный элемент', xmin:6:2,
        ' с номером', i:3)
End. {MinEl}

```

**Пример 17.4.** Составить программу вычисления скалярного произведения  $S$  двух векторов  $V$  и  $U$ , состоящих из пяти элементов каждый, по формуле

$$S = \sum_{i=1}^5 v_i u_i.$$

Вычислить длину  $D_v$  вектора  $V$

$$D_v = \sqrt{\sum_{i=1}^5 v_i^2}.$$

На рис. 17.4 приведена блок-схема алгоритма, а ниже – программа.

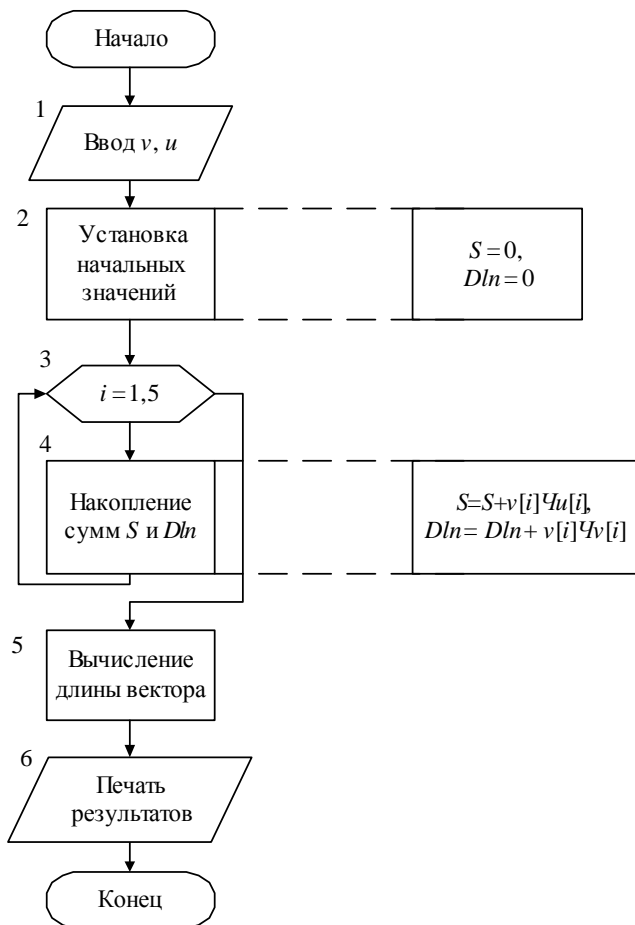


Рис. 17.4. Схема алгоритма вычисления длины вектора и скалярного произведения

```

{Цель: определение длины вектора Dln и скалярного      }
{произведения S двух векторов.                        }
{Параметры и переменные:                              }
{  v,u - одномерные массивы ( векторы );              }
{  Dln,S - переменные для результатов;                }
{  i,j - вспомогательные переменные.                  }
{Программист: Стасов К.Т.                             }
{Дата: 12 февраля 2006 г.                             }
Program Scalar;
    
```

```

Const
  pr3='  ';
  nmax=10;
Var
  u, v:array [1..nmax] of real;
  Dln, S:real;
  i, j:integer;
Begin{Scalar}
  WriteLn(pr3,' введите размер массивов');
  Read(n);
  WriteLn(pr3,' введите массивы v и u размером ',n);
  for i := 1 to n do
    Read (v[i],u[i]);
  WriteLn(pr3,' исходные данные ');
  WriteLn(' массив v ');
  for i:=1 to n do
    Write (pr3,v[i]:7:3);
  WriteLn;
  WriteLn(' массив u ');
  for i:=1 to n do
    Write(pr3,u[i]:7:3);
  S:=0;
  Dln:=0;
  for i:=1 to n do
    begin
      S:=S+v[i]*u[i];
      Dln:=dln+v[i]*v[i]
    end;
  Dln:=Sqrt(Dln);
  WriteLn;
  WriteLn;
  WriteLn(pr3,'результаты вычислений ');
  WriteLn;
  WriteLn(pr3,' скалярное произведение --',s:12);
  WriteLn(pr3,' длина вектора v -- ',dln:12);
End.{Scalar}

```

Результат работы программы:

исходные данные

массив v

1.100	1.300	3.400	4.100	3.300
-------	-------	-------	-------	-------

массив u

1.200	2.500	3.700	4.200	5.100
-------	-------	-------	-------	-------

результаты вычислений

скалярное произведение -- 5.120000e+01

длина вектора v -- 6.493070e+00

В приведенном примере максимальный размер индекса  $n$ тах массивов  $v$  и  $u$  задан в разделе описания констант. Если размеры массивов изменяются, то достаточно будет ввести новое значение переменной  $n$ , не изменяя остальной части программы. Элементы векторов  $v$  и  $u$  вводятся парами в цикле ввода. В цикле вычислений происходит накопление сумм для вычисления скалярного произведения  $S$  и длины вектора  $Dln$ . В заключительной части программы осуществляется вывод исходных данных и результатов вычислений.

## **ЗАДАНИЯ**

Вариант 1. В произвольно заданном одномерном массиве определить число отрицательных, нулевых и положительных элементов.

Вариант 2. В произвольно заданном одномерном массиве определить минимальный и максимальный элементы и поменять их значения местами.

Вариант 3. В произвольно заданном одномерном массиве определить два элемента с наибольшими значениями и обнулить все элементы, расположенные между найденными значениями.

Вариант 4. В произвольно заданном одномерном массиве определить местоположение первого и последнего из всех отрицательных элементов.

Вариант 5. В произвольно заданном одномерном массиве определить элемент, сумма которого с первым элементом максимальна.

Вариант 6. В произвольно заданном одномерном массиве целых чисел определить, есть ли в этом массиве одинаковые элементы.

Вариант 7. В произвольно заданном одномерном массиве определить три элемента с наибольшими значениями. Могут ли быть найденные значения сторонами треугольника?

Вариант 8. Из значений произвольно заданного одномерного массива сформировать массив из положительных и массив из отрицательных элементов исходного массива.

Вариант 9. В произвольно заданном одномерном массиве целых чисел определить элементы, сумма цифр в записи которых максимальна и минимальна. Поместить найденные элементы в начало и в конец соответственно.

Вариант 10. Первый и второй элементы одномерного массива равны единице. Каждый последующий элемент является суммой двух предыдущих элементов. По данному правилу сформировать массив из 50 эле-

ментов. Определить и вывести «простые» элементы, т. е. элементы, которые делятся только на единицу и сами на себя.

Вариант 11. Из элементов произвольно заданного одномерного массива сформировать массив, в котором в начале расположены отрицательные, а далее – положительные элементы исходного массива.

Вариант 12. В произвольно заданном одномерном массиве определить номера двух элементов с наименьшими значениями. Обнулить значения элементов, расположенных между найденными номерами в исходном массиве.

Вариант 13. В произвольно заданном одномерном массиве определить два элемента с наибольшими значениями и два элемента с наименьшими значениями. Сократить число элементов в исходном массиве на 4 найденных элемента.

Вариант 14. В произвольно заданных двух одномерных массивах определить максимальные элементы и поменять их местами.

Вариант 15. В произвольно заданном одномерном массиве определить число положительных и число отрицательных элементов. Сформировать новый массив из элементов одного знака, число которых больше.

Вариант 16. Произвольно заданы три одномерных массива с одинаковым числом элементов. Сформировать массив, каждый элемент которого является максимальным элементом соответственно в каждом из трех исходных массивах. Определить местоположение максимального и минимального элементов в сформированном массиве.

Вариант 17. В произвольно заданном одномерном массиве определить среднее значение всех элементов, значение которых превышает среднее значение.

Вариант 18. В произвольно заданном одномерном массиве определить максимальную последовательность из положительных элементов и вывести ее на экран дисплея.

Вариант 19. В произвольно заданном одномерном массиве определить максимальную последовательность из отрицательных элементов и вывести ее на экран дисплея.

Вариант 20. В произвольно заданном одномерном массиве определить элементы, слева и справа от которых расположены меньшие значения.

Вариант 21. В произвольно заданном одномерном массиве все нулевые элементы заменить максимальным элементом.

Вариант 22. В произвольно заданном одномерном массиве все отрицательные элементы заменить значением минимального элемента, а все положительные – максимальным значением.

Вариант 23. В произвольно заданном одномерном массиве определить два элемента с минимальными значениями и уменьшить исходный массив на элементы, расположенные между найденными значениями.

Вариант 24. Произвольно заданы три одномерных массива. Сформировать новый массив, состоящий из 10 элементов с наибольшими значениями исходных массивов.

Вариант 25. В произвольно заданном одномерном массиве определить 4 элемента с наибольшими значениями. Определить, сколько отрицательных значений оказалось среди найденных.

# Тема 18. ОБРАБОТКА ДВУМЕРНЫХ МАССИВОВ ДАННЫХ. МАТРИЦЫ

## ПРИМЕРЫ СОСТАВЛЕНИЯ ПРОГРАММ

**Пример 18.1.** Пусть необходимо найти произведение двух матриц  $A$  и  $B$  размера  $2 \times 3$  и  $3 \times 3$  соответственно.

Элементы результирующей матрицы  $C$  (размера  $2 \times 3$ ) определяются по формуле

$$c_{ij} = \sum_{k=1}^m a_{ik} b_{kj}; \quad i=1, 2, \dots, n; j=1, 2, \dots, p,$$

где  $m$  – число столбцов матрицы  $A$  и число строк матрицы  $B$ ;  $n$  – число строк матрицы  $A$ ;  $p$  – число столбцов матрицы  $B$ . В общем случае результирующая матрица имеет размер  $n \times p$ .

Схема алгоритма умножения двух прямоугольных матриц представлена на рис. 18.1.

Программа имеет следующий вид:

```

{*****}
{Цель: Определение произведения матриц.}
{Переменные:  a, b – исходные двумерные }
{               массивы (матрицы);        }
{      c – произведение матриц;           }
{      S – вспомогательная переменная;  }
{      i, j, k – параметры циклов;        }
{      n – число строк матрицы a;         }
{      m – число столбцов матрицы a       }
{            и строк матрицы b;           }
{      p – число столбцов матрицы b.      }
{Программист: Сергеев Т.И.                }
{Дата: 14 февраля 2006 г.                 }
{*****}

```

**Program** Mult;

**Const**

Nmax = 10;

**Type**

Matr = array[1..Nmax] of integer;

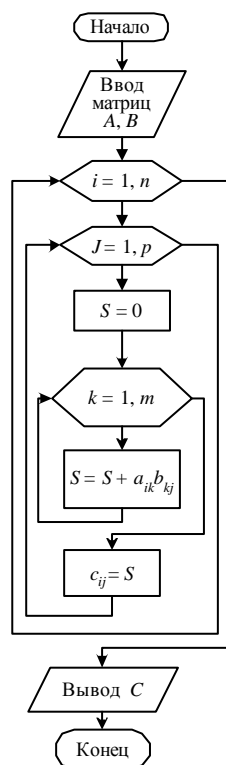


Рис. 18.1.  
Схема алгоритма  
умножения матриц



```
Var
  a, b, c :Matr;
  i, j, k, m, n, p, S :integer;
Begin {Mult}
  WriteLn('Введите число строк матрицы A:');
  Read(n);
  WriteLn('столбцов:');
  Read(m);
  WriteLn('и столбцов матрицы B:');
  Read(p);
  {Ввод элементов матрицы A}
  WriteLn('Введите', n*m:4, ' элементов матрицы A:');
  for i := 1 to n do
    for j := 1 to m do
      begin
        Write('a[', i, ', ', j, ']=');
        Read(a[i,j])
      end;
    {Ввод элементов матрицы B}
    WriteLn('Введите', m*p:4, ' элементов матрицы B:');
    for i := 1 to m do
      for j := 1 to p do
        begin
          Write('b[', i, ', ', j, ']=');
          Read(b[i,j])
        end;
      {Эхо-печать}
      WriteLn('Исходные');
      WriteLn('матрица A:');
      for i := 1 to n do
        begin
          for j := 1 to m do
            Write(a[i,j]:6);
          WriteLn
        end;
      WriteLn('матрица B:');
      for i := 1 to m do
        begin
          for j := 1 to p do
            Write(b[i,j]:6);
          WriteLn
        end;
      {Умножение матриц}
      for i := 1 to n do
        for j := 1 to p do
```

```

begin
  S := 0;
  for k := 1 to m do
    S := S + a[i,k]*b[k,j];
  c[i,j] := S
end;
{Вывод результирующей матрицы C}
WriteLn('их произведение:');
for i := 1 to n do
  begin
    for j := 1 to p do
      Write(b[i,j]:6);
    WriteLn
  end
end. {Mult}

```

**Пример 18.2.** Для данной матрицы, размер которой не превышает  $10 \times 10$  элементов, найти максимум среди сумм элементов диагоналей, параллельных главной. Блок-схема алгоритма вычислений приведена на рис. 18.2. После нее дан текст программы.

Для вычисления сумм элементов диагоналей в программе зарезервирован массив  $S$ . Если размер исходной матрицы  $A$  равен  $n \times n$ , то в ней содержится  $2n - 1$  диагоналей, параллельных главной. Ввод элементов матрицы  $A$  осуществляется после ввода переменной  $n$ , определяющей размер данной матрицы. По условию задачи  $n$  должно быть меньше 10. Перед вычислением нужных величин осуществляется обнуление элементов массива  $S$ .

**Program** SumDg;

```

{*****}
{Цель: для заданной матрицы найти максимум среди сумм }
{ элементов диагоналей, параллельных главной. }
{Параметры и переменные: }
{ A - двумерный массив для исходной матрицы; }
{ S - одномерный массив сумм диагональных элементов; }
{ max - максимальная сумма. }
{Программист Ларин В.И. }
{Дата: 16 февраля 2006 г. }
{*****}

```

**Const**

pr3=' ';

**Var**

```

A:array[1..10,1..10] of integer;
S:array[1..20] of integer;
i,j,k,n,max:integer;

```

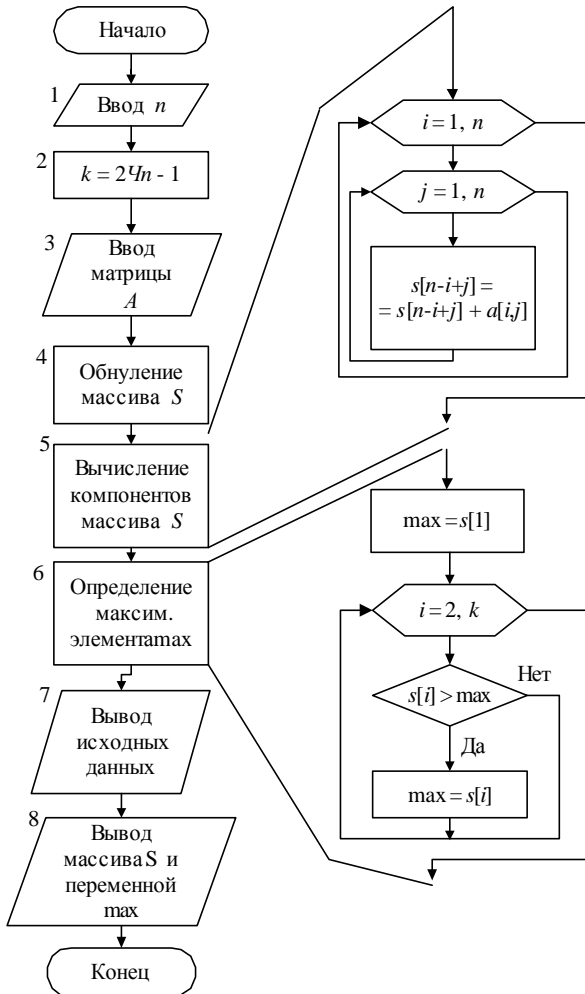


Рис. 18.2. Схема алгоритма вычисления сумм диагональных элементов

```

Begin{SumDg}
  WriteLn(pr3, 'введите число <=10 ,характеризующее ',
    ' размерность матрицы');
  Read(n);
  k:=2*n-1;
  WriteLn(pr3, 'введите исходную матрицу из', n*n:5,
    ' элементов');
  
```

```

for i:=1 to n do
  for j:=1 to n do
    Read(a[i,j]);
  WriteLn(,pr3,' исходная матрица ');
for i:=1 to n do
  begin
    for j:=1 to n do
      Write(a[i,j]:5);
    WriteLn
  end;
{*** цикл вычисления сумм диагональных элементов ***}
for i:=1 to k do
  s[i]:=0;
for i:=1 to n do
  for j:=1 to n do
    s[n-i+j]:=s[n-i+j]+a[i,j];
{*** определение максимальной суммы ***}
max:=s[1];
for i:=2 to k do
  if s[i]>max
  then
    max:=s[i];
WriteLn(pr3,pr3,' результаты вычисления ');
WriteLn(pr3,'суммы диагональных элементов ');
for i:=1 to n do
  Write(s[i]:5);
WriteLn;
for i:=n+1 to k do
  Write(s[i]:5);
WriteLn;
WriteLn(pr3,'максимальная сумма диагональных
элементов ', max:5);
End.{SumDg}

```

Для вычисления сумм диагональных элементов организован вложенный цикл по перебору всех элементов исходной матрицы  $A$ . Нужный индекс элементов массива  $S$  определяется из параметров циклов выражением  $n - i + j$ . После формирования элементов массива  $S$  определяется его максимальный компонент. Для этого переменной  $max$  присваивается значение первого элемента массива. Далее организован цикл по просмотру всех остальных элементов массива. Если какой-либо компонент массива  $S$  превысил значение переменной  $max$ , то последней присваивается значение этого компонента. В заключительной части программы осуществляется вывод элементов исходной матрицы  $A$ , сформированного массива сумм диагональных элементов  $S$  и максимального элемента массива  $max$ .

Для контрольного просчета была выбрана матрица размера  $7 \times 7$ . Результаты вычислений приведены ниже.

введите число  $\leq 10$ , характеризующее размерность матрицы  
7

введите исходную матрицу из 49 элементов

исходная матрица

1	2	3	4	5	6	7
11	12	13	14	15	16	17
21	22	23	24	25	26	27
31	32	33	34	35	36	37
41	42	43	44	45	46	47
51	52	53	54	55	56	57
61	62	63	64	65	66	67

результаты вычисления

суммы диагональных элементов

61	113	156	190	215	231	238
177	125	82	48	23	7	

максимальная сумма диагональных элементов 238

## Задания

Вариант 1. Сформировать матрицу  $A$  по правилу

$$a_{ij} = \frac{2b_{ij} + b_{\min}}{b_{\max}},$$

где  $b_{\min}$ ,  $b_{\max}$  – минимальный и максимальный элементы произвольно заданной матрицы  $B$  размера  $3 \times 4$ .

Вариант 2. В произвольно заданных матрицах  $X$  и  $Y$  размера  $5 \times 4$  определить максимальные элементы и поменять их значения местами. В матрице  $X$  все отрицательные элементы заменить максимальным значением.

Вариант 3. Сформировать одномерный массив, состоящий из максимальных значений положительных элементов соответствующих строк произвольно заданной матрицы  $B$  размера  $5 \times 6$ .

Вариант 4. Определить максимальный элемент среди элементов квадратной матрицы размера  $5 \times 5$ , лежащих выше главной диагонали, и минимальный элемент среди элементов, лежащих ниже главной диагонали. Поменять найденные значения местами.

Вариант 5. Определить максимальный элемент в произвольно заданной матрице  $Z$  размера  $m \times n$  и обнулить все элементы строки и столбца, на пересечении которых расположено найденное значение.

Вариант 6. Сформировать одномерный массив, каждый элемент которого равен максимальному элементу соответствующего столбца произвольно заданной матрицы размера  $m \times n$ . В сформированном массиве найти минимальный элемент.

Вариант 7. Сформировать произвольный двумерный массив целочисленных значений размера  $6 \times 6$ . Определить число повторений каждого из значений первой строки.

Вариант 8. В произвольно заданном двумерном массиве размера  $4 \times 5$  определить три элемента с наибольшими значениями.

Вариант 9. В произвольно заданном двумерном массиве поменять местами строки, содержащие минимальный и максимальный элемент. Если минимальный и максимальный элементы принадлежат одной строке, то поменять местами соответствующие столбцы.

Вариант 10. В произвольно заданной матрице размера  $4 \times 6$  определить строку с максимальной суммой элементов и столбец с минимальной суммой.

Вариант 11. Из произвольно заданной матрицы размера  $5 \times 5$  сформировать построчный одномерный массив из положительных элементов исходной матрицы.

Вариант 12. В двух произвольно заданных двумерных массивах поменять местами строки, содержащие максимальные элементы. Вывести на экран исходные и измененные матрицы.

Вариант 13. Сформировать два произвольных двумерных массива размера  $5 \times 4$ . Поменять местами столбцы исходных матриц, содержащие минимальные элементы.

Вариант 14. Определить минимальный элемент в произвольно заданной матрице размера  $4 \times 6$  и заменить на это минимальное значение элементы строки и столбца, которым принадлежит найденное значение.

Вариант 15. Найти минимальное значение главной диагонали и максимальное значение вспомогательной диагонали в квадратной матрице размера  $5 \times 5$ . Найденные значения поменять местами. Если эти значения принадлежат одному элементу, то это значение присвоить всем элементам главной и вспомогательной диагоналей.

Вариант 16. Определить и поменять местами максимальное и минимальное значения среди элементов, расположенных выше главной и ниже вспомогательной диагоналей в произвольно заданной квадратной матрице размера  $6 \times 6$ .

Вариант 17. В произвольно заданной матрице  $A$  размера  $5 \times 4$  определить минимальный элемент и обнулить значение элементов, расположенных ниже и правее найденного элемента.

Вариант 18. В трех произвольно заданных положительных матрицах размера  $3 \times 3$  определить максимальные элементы. Считая найденные значения длинами отрезков, определить возможность построения из них треугольника.

Вариант 19. Сформировать одномерный массив, элементами которого являются средние значения строк произвольно заданной матрицы размера  $8 \times 3$ . Упорядочить значения одномерного массива по возрастанию.

Вариант 20. Сформировать одномерный массив из положительных элементов произвольно заданной матрицы и упорядочить отобранные значения по убыванию.

Вариант 21. Сформировать двумерный массив  $C$  размера  $5 \times 5$ , элементы которого вычисляются по формуле

$$c_{ij} = \frac{i + \sin \frac{\pi}{j+i+3}}{j + \cos \frac{\pi}{j+i+2}}.$$

Определить максимальный элемент и его значение присвоить всем отрицательным элементам массива  $C$ .

Вариант 22. Произвольно заданы две матрицы размера  $5 \times 3$ . Сформировать новую матрицу, каждый элемент которой является наибольшим значением из соответствующих элементов исходных массивов. В сформированной матрице определить сумму положительных элементов.

Вариант 23. В произвольно заданной матрице размерам  $3 \times 4$  упорядочить построчно все значения элементов по возрастанию.

Вариант 24. В произвольно заданной матрице размера  $5 \times 6$  определить элементы, которые снизу, сверху, справа и слева «окружены» меньшими значениями. Из найденных элементов сформировать одномерный массив.

Вариант 25. В двух произвольно заданных квадратных целочисленных матрицах размера  $n \times n$  определить количество одинаковых элементов, расположенных и в первой и во второй матрице.

### КОНТРОЛЬНЫЕ ЗАДАНИЯ

Вариант 1. Дана матрица размера  $3 \times 4$ . Составить программу для подсчета количества четных элементов в каждой строке матрицы.

Вариант 2. Дана матрица размера  $5 \times 6$ . Составить программу для подсчета количества нечетных элементов в каждом столбце матрицы.

Вариант 3. Проверить, имеется ли в заданной строке символов баланс открывающих и закрывающих круглых скобок.

Вариант 4. Перечислить все числа заданной последовательности чисел, которые состоят из тех же цифр, что и первое.

Вариант 5. Дана матрица  $B$  размера  $5 \times 4$ . Составить программу формирования вектора, элементы которого равны сумме элементов строк матрицы  $B$ .

Вариант 6. Дана матрица  $B$  размера  $5 \times 7$ . Составить программу формирования вектора, элементы которого равны сумме элементов столбцов матрицы  $B$ .

Вариант 7. Дана матрица  $A$  размера  $7 \times 7$ . Составить программу нахождения суммы элементов, лежащих выше главной диагонали.

Вариант 8. Дана матрица  $A$  размера  $7 \times 7$ . Составить программу нахождения суммы элементов, лежащих ниже главной диагонали.

Вариант 9. В массиве слов найти пару слов, из которых одно является обращением другого.

Вариант 10. Задана матрица  $B$  размера  $7 \times 7$ . Составить программу, осуществляющую перестановку элементов в каждой строке матрицы так, чтобы 1-й элемент строки поменялся с последним, 2-й с предпоследним и т. д.

Вариант 11. Расстояние между  $k$ -й и  $m$ -й строками матрицы  $A = \|a_{ij}\|$  определяется как

$$\sum_{j=1}^n |a_{kj}| \times |a_{mj}|.$$

Указать номер строки, максимально удаленной от первой строки заданной матрицы.



Вариант 12. Расстояние между  $k$ -й и  $m$ -й строками матрицы  $A = \|a_{ij}\|$  определяется как

$$\sum_{j=1}^n |a_{kj}| \times |a_{mj}|.$$

Указать номер строки, максимально удаленной от последней строки заданной матрицы.

Вариант 13. Сформировать двумерный массив целых чисел. Проверить, содержится ли в нем заданная строка чисел.

Вариант 14. Определить норму заданной матрицы  $A = \|a_{ij}\|$ , т. е. число

$$\max_i \left( \sum_j |a_{ij}| \right).$$

Вариант 15. Сформировать два массива целых чисел определенной длины. Вывести на печать числа, встречающиеся в каждом массиве.

Вариант 16. Сформировать два массива целых чисел определенной длины. Вывести на печать числа, встречающиеся только в одном массиве.

Вариант 17. Из символов произвольного предложения сформировать массив целых чисел, соответствующих порядковому номеру литер в коде ASCII. Определить максимальный элемент этого массива.

Вариант 18. Из символов произвольного предложения сформировать массив целых чисел, соответствующих порядковому номеру литер в коде ASCII. Определить минимальный элемент этого массива.

Вариант 19. Среди столбцов заданной целочисленной матрицы  $C$  размера  $7 \times 7$ , компоненты которой не превышают 10, найти столбец с минимальным произведением элементов.

Вариант 20. Среди строк заданной целочисленной матрицы  $C$  размера  $7 \times 7$ , компоненты которой не превышают 10, найти строку с максимальным произведением элементов.

Вариант 21. Для заданной матрицы  $B$  размера  $5 \times 5$  найти такие  $k$ , для которых  $k$ -я строка матрицы совпадает с  $k$ -м столбцом.

Вариант 22. Вычислить значение полинома

$$Y = \sum_{k=0}^n a_k \times x^k,$$

используя схему Горнера:

$$y = \left( \dots \left( (a_n x + a_{n-1}) x + a_{n-2} \right) x + \dots + a_1 \right) x + a_0.$$

Значение  $n$  принять равным 10,  $x$  и элементы массива  $A$  выбрать произвольно.

Вариант 23. По заданному символьному массиву  $C$  сформировать массив логических элементов  $B$ . Компоненте  $b_i$  присвоить значение **true**, если  $c_i$  является цифрой, и присвоить **false** – в противном случае.

Вариант 24. Составить программу записи элементов прямоугольной матрицы  $A$  в одномерный массив в порядке следования столбцов. Найти наименьший элемент полученного массива. Размер и элементы массива выбрать произвольно.

Вариант 25. Составить программу формирования вектора из количества ненулевых элементов каждой строки произвольного двумерного массива.

## Тема 19. ПРОСТЕЙШИЕ ПРОЦЕДУРЫ

### ПРИМЕР СОСТАВЛЕНИЯ ПРОГРАММЫ

Пусть необходимо составить программу для вычисления площади выпуклого четырехугольника  $ABCD$ , заданного длинами своих сторон  $a, b, c, d$  и одной из диагоналей  $e$  (рис. 19.1).

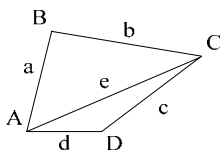


Рис. 19.1. Выпуклый четырехугольник

Площадь четырехугольника в данном случае можно получить как сумму площадей треугольников  $ABC$  и  $ACD$  по формуле Герона:

$$S = \sqrt{p(p-a)(p-b)(p-c)},$$

где  $p = \frac{a+b+c}{2}$  – полупериметр;  $a, b, c$  – длины сторон.

Поскольку здесь дважды необходимо вычислять значение площади треугольника, то алгоритм ее вычисления целесообразно оформить в виде подпрограммы, в качестве которой можно использовать и процедуру (в отличие от примера к теме 2). Затем необходимо дважды обратиться к ней и вычислить сумму площадей треугольников.

Схема алгоритма решения поставленной задачи приведена на рис. 19.2, где наряду с алгоритмом основной программы дается и схема алгоритма процедуры Str, вычисляющей площадь одного треугольника.

В начале ее алгоритма помещается блок входа в подпрограмму, где указывается имя процедуры и список формальных параметров  $a, b, c$ , являющиеся исходными данными, необходимыми для вычисления площади треугольника. Вычисленное значение площади присваивается формальному параметру результата  $S$ , служащего для возвращения вычисленного значения в основную программу. Он указывается в списке выходных параметров в блоке выхода. В основной программе происходит обращение к описанию функции для вычисления площади первого треугольника,  $ABC$ , и второго,  $ACD$ , которое отображается с помощью специального блока обращения к процедуре.

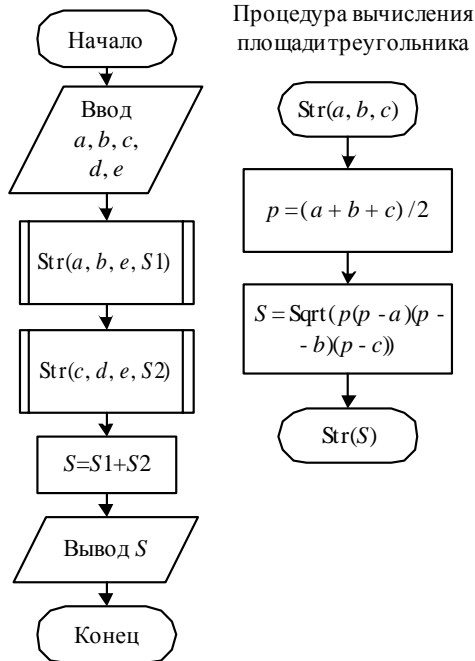


Рис. 19.2. Алгоритм вычисления площади четырехугольника

Ниже приведен текст программы:

```

{*****}
{Цель: вычисление площади четырехугольника.}
{Переменные: a, b, c, d – длины сторон четырехугольника;}
{  e – длина диагонали; S – площадь четырехугольника;}
{  S1, S2 – площади треугольников.}
{Подпрограммы: Str – процедура вычисления площади}
{  треугольника.}
{Метод: вычисление по формуле Герона.}
{Программист: Трегудов В.П.}
{Дата написания: 18 февраля 2006 г.}
{*****}
  
```

**Program P1;**

**Var**

a, b, c, d, e, S :real;

{Процедура вычисления площади треугольника.}

{Переменные: a, b, c – стороны треугольника;}

{ p – полупериметр; S – площадь треугольника.}

**Procedure** Str(a, b, c :real; S:real);

```
Var
  p:real;
Begin {Str}
  p := (a+b+c)/2;
  S := Sqrt(p*(p-a)*(p-b)*(p-c))
End; {Str}
Begin {Pl}
  WriteLn('Введите длины четырех сторон и диагонали');
  Read(a, b, c, d, e);
  WriteLn('При a=',a:6:2,' b=',b:6:2,' c=',c:6:2,
    ' d=',d:6:2,' e=',e:6:2);
  Str(a, b, e, S1);
  Str(c, d, e, S2);
  S := S1 + S2;
  WriteLn('Площадь четырехугольника равна',S:8:2)
End. {Pl}
```

### ЗАДАНИЕ

Выполнить соответствующий вариант задания к теме 2, введя вместо функции пользователя описание процедуры.

## Тема 20. ИСПОЛЬЗОВАНИЕ ПРОЦЕДУР ДЛЯ ОБРАБОТКИ МАССИВОВ

### ПРИМЕРЫ СОСТАВЛЕНИЯ ПРОГРАММ

**Пример 20.1.** Пусть необходимо, используя процедуру PrMax определения максимального элемента в массиве  $z = (z_1, z_2, \dots, z_n)$ , найти точки, в которых функции

$$y_1 = e^{-x} \cdot \cos x - 1; \quad y_2 = x \cdot \ln|x|$$

имеют максимальные значения на заданном отрезке  $[a, b]$ .

При решении данной задачи в процедуре PrMax для одномерного массива  $z$  определяются максимальный элемент Max и его индекс  $k$ . В основной программе сначала формируются массивы значений заданных функций  $y_1, y_2$ , а затем с помощью процедуры PrMax определяются их максимумы и индексы  $k_1, k_2$  соответствующих максимальных элементов в этих массивах. Индексы используются для нахождения на заданном отрезке точек  $x_1, x_2$ , в которых функции достигают максимума (при заданном разбиении).

Схема нисходящего проектирования алгоритма решения задачи в виде структурограммы представлена на рис. 20.1. Программа имеет следующий вид:

```
{ ***** }
{ Программа: MaxFun. }
{ Цель: определение максимального значения функции }
{ на заданном отрезке. }
{ Входные данные: a, b - границы отрезка; h - шаг }
{ изменения аргумента. }
{ Выходные данные: max1, max2 - максимальные значения }
{ функций; }
{ x1, x2 - точки, в которых функции имеют максимум. }
{ Программист: Иванов А.Н. }
{ Дата: 25 февраля 2006 г. }
{ ***** }
Program MaxFun ;
Const
  Nmax = 100;
Type
  massiv = array[ 1 .. Nmax ] of real;
```

```

Var
    i, k1, k2, n: integer;
    a, b, h, max1, max2, x, x1, x2: real;
    y1, y2: massiv;
{Процедура определения максимального элемента      }
{    в одномерном массиве                          }
{Параметры: n – размер массива; z – имя массива;    }
{    max – максимальный элемент;                   }
{    k – номер максимального элемента              }
Procedure PrMax(n:integer; z:massiv; Var max:real;
               Var k:integer);

    Var
        i:integer;
    Begin {PrMax}
        max:= z[1];
        k:= 1;
        for i:= 2 to n do
            if max < z[i]
            then
                begin
                    max:= z[i];
                    k:= i
                end
            end
    End; {PrMax}
Begin {MaxFun} {Основная программа}
    Write(' Введите a, b, h ');
    ReadLn(a, b, h);
    WriteLn(' Исходные данные : ');
    WriteLn(' a= ', a, ' b= ', b, ' h= ', h);
    n:=Trunc((b-a)/h + 0.5 ) + 1;
    x:= a;
    for i:= 1 to n do
        begin {Формирование массивов значений функций}
            y1[i]:= Exp(-x)*Cos(x)-1;
            y2[i]:= x*LN(Abs(x));
            x:=x+h;
        end;
        PrMax(n, y1, max1, k1);           {Вызов процедуры}
        PrMax(n, y2, max2, k2);           {Вызов процедуры}
        x1:= a+(k1-1)*h;
        x2:= a+(k2-1)*h;
        WriteLn('максимум функции y1:',max1,' в точке',x1);
        WriteLn('максимум функции y2:',max2,' в точке',x2)
    End. {MaxFun}

```

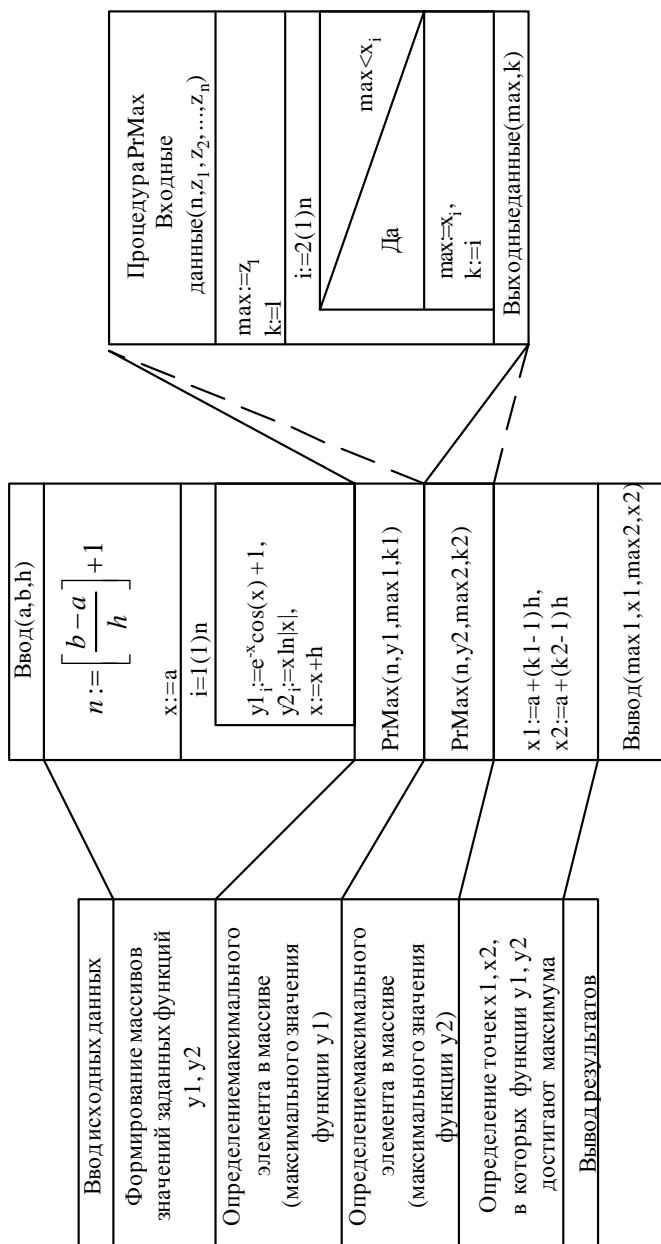


Рис. 20.1. Схема нисходящего проектирования алгоритма определения максимального значения функции на заданном интервале



**Пример 20.2.** Составить процедуру сортировки по убыванию значений элементов одномерного массива. Используя ее, отсортировать элементы в каждом столбце прямоугольной матрицы.

Пусть задана матрица  $A$ , содержащая  $n$  строк и  $m$  столбцов, описана процедура Sort сортировки одномерного массива  $C$ , например методом максимального элемента. Чтобы организовать сортировку столбцов матрицы, после ввода элементов матрицы необходимо предусмотреть цикл по номерам столбцов матрицы. Поскольку процедура сортировки обеспечивает преобразование только одномерного массива, необходимо на каждом шаге внешнего цикла с помощью дополнительного цикла по номерам строк матрицы преобразовать очередной ее столбец в одномерный массив  $C$  и обратиться к процедуре Sort. Полученный отсортированный массив записывается на место исходного столбца матрицы. В конце программы осуществляется вывод преобразованной матрицы  $A$ .

Алгоритм сортировки прямоугольной матрицы  $A$  представлен на рис. 20.2, где ввод и вывод матрицы  $A$  также оформлен в виде соответствующих процедур.

При составлении программы, в которой имеются процедуры и функции, использующие в качестве формальных параметров массивы, необходимо обратить внимание на следующую особенность. Описание массива является структурированным типом, имеющим сложную структуру, состоящую из нескольких элементов. Тип же параметров процедур и функций, как правило, определяется одним *идентификатором*. Поэтому в программе в разделе типов следует определить тип массива, обозначив его каким-либо именем, а затем уже использовать его для определения формальных параметров процедур и функций, являющихся массивами. С этой целью мы введем определение типа одномерного массива Vector и двумерного массива Matrix.

Кроме того, для обеспечения возможности вывода на экран в пояснительных надписях произвольных имен массивов в процедурах ввода InArg и вывода OutArg введем формальный параметр имени массива Name.

С учетом сделанных замечаний текст программы решения данной задачи будет иметь следующий вид:

```
{Цель: сортировка столбцов прямоугольной матрицы.      }
{Параметры и переменные: a - двумерный массив;          }
{  c - одномерный массив; n - число строк;                }
{  m - число столбцов; Nmax - максимально допустимое     }
{  число строк; Mmax - максимально допустимое число      }
{  столбцов; Vector - тип одномерного массива;            }
{  Matrix - тип двумерного массива;                       }
```

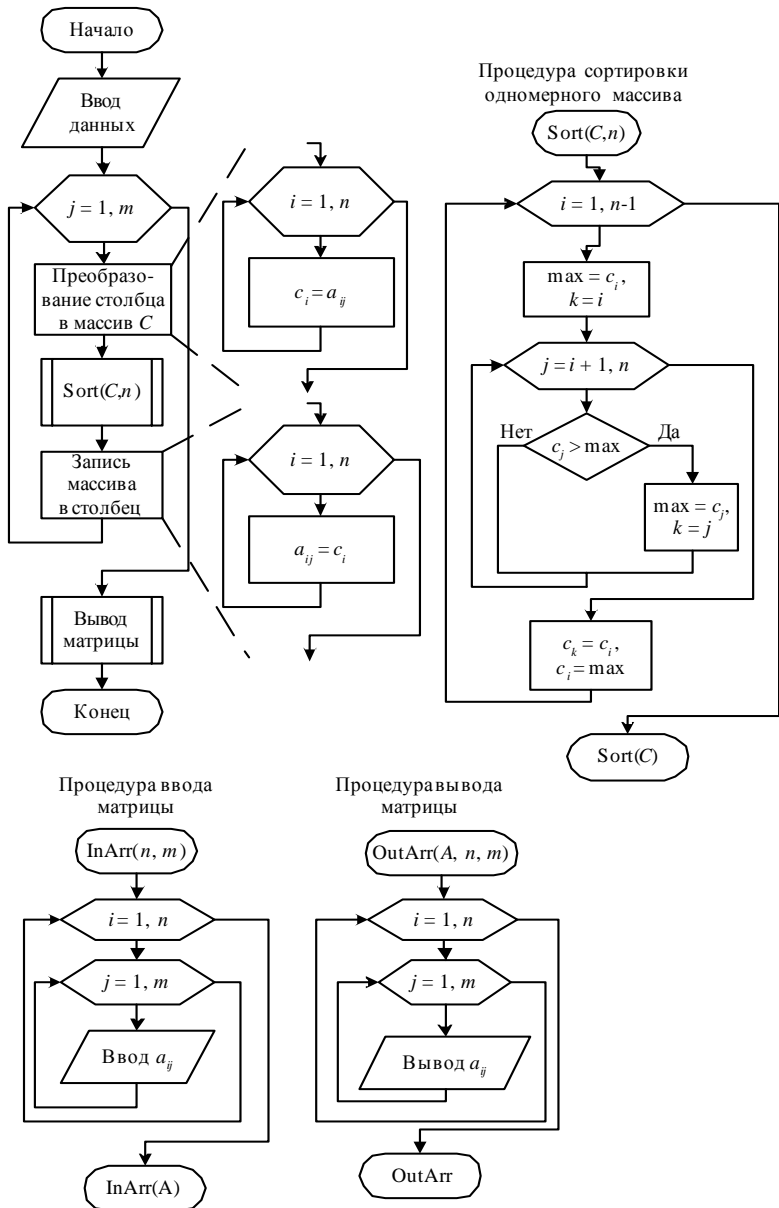


Рис. 20.2. Схема алгоритма сортировки столбцов прямоугольной матрицы

```

{      i, j - параметры циклов.                                }
{Метод: сортировка методом максимального элемента.          }
{Подпрограммы: InArr - процедура ввода матрицы;             }
{      OutArr - процедура вывода матрицы;                   }
{      Sort - процедура сортировки одномерного              }
{      массива.                                              }
{Программист: Филатов К.Е.                                   }
{Дата: 27 февраля 2006 г.                                    }
Program SortingColumn;
Const
    Nmax=20;
    Mmax=20;
Type
    Vector=array[1..Nmax] of real;
    Matrix=array[1..Nmax, 1..Mmax] of real;
Var
    a:Matrix;
    c:Vector;
    i, j, m, n:integer;
{Процедура ввода двумерного массива                          }
{Параметры:  a - двумерный массив; n - число строк;         }
{      m - число столбцов; i, j - параметры                 }
{      циклов; Name - имя массива.                           }
Procedure InArr(n, m:integer; Var a:Matrix; Name:Char);
Var
    i, j:integer;
Begin {InArr}
    WriteLn('Введите ',n*m,'элементов матрицы ',Name);
    for i:= 1 to n do
        for j:= 1 to m do
            begin
                Write(Name,['\','i','\','j','=');
                Read(a[i,j])
            end
        End; {InArr}
    {Процедура вывода двумерного массива                      }
    {Параметры:  a - двумерный массив; n - число строк;     }
    {      m - число столбцов; i, j - параметры             }
    {      циклов.                                            }
Procedure OutArr(Var a:Matrix; n, m:integer);
Var
    i, j:integer;
Begin {OutArr}
    for i:= 1 to n do
        begin
            for j:= 1 to m do

```

```

        Write(a[i,j]:6:2);
    WriteLn
end
End; {OutArr}
{Процедура сортировки одномерного массива}
{Параметры: c - одномерный массив; n - размер}
{массива; Max - максимальный элемент;}
{
    k - номер максимального элемента;
    i, j - параметры циклов.
}
Procedure Sort(Var c:Vector; n:integer);
Var
    i, j, k:integer;
    Max:real;
Begin {Sort}
    for i:= 1 to n-1 do
        begin
            Max:= c[i];
            for j:= i+1 to n do
                if c[j]>Max
                then
                    begin
                        Max:=c[j];
                        k:=j
                    end;
            c[k]:=c[i];
            c[i]:=Max
        end
    End; {Sort}
Begin {SortingColumn}
    WriteLn('Введите число строк и столбцов матрицы A');
    Read(n, m);
    InArr(n,m,a,'a');      {Ввод матрицы A}
    WriteLn('Исходная матрица:');
    OutArr(a,n,m);          {Эхо-печать}
    {Преобразование матрицы}
    for j:= 1 to m do
        begin
            {Преобразование столбца в одномерный массив}
            for i:= 1 to n do
                c[i]:=a[i,j];
            Sort(c,n);      {Сортировка столбца}
            {Запись массива в столбец матрицы}
            for i:= 1 to n do
                a[i,j]:=c[i]
            end;
            WriteLn('Преобразованная матрица:');
            OutArr(a,n,m);      {Вывод результата}
        End. {SortingColumn}

```

## ЗАДАНИЯ

Во всех вариантах заданий ввод исходных и вывод результирующих матриц производить с помощью соответствующих процедур.

Вариант 1. Составить процедуру подсчета суммы значений элементов в одномерном массиве. С помощью этой процедуры подсчитать суммы элементов столбцов произвольно заданной двумерной матрицы  $A$  и вывести номер столбца с минимальной суммой.

Вариант 2. Составить процедуру подсчета суммы значений элементов в одномерном массиве. С помощью этой процедуры подсчитать суммы элементов строк произвольно заданной двумерной матрицы  $A$  и вывести номер строки с максимальной суммой.

Вариант 3. Составить процедуру подсчета суммы значений элементов в одномерном массиве. С помощью этой процедуры подсчитать суммы элементов в главной и побочной диагоналях в произвольно заданной квадратной матрице  $B$ . Вывести сообщение, какая диагональ имеет максимальное значение.

Вариант 4. Составить процедуру подсчета суммы значений элементов в одномерном массиве. С помощью этой процедуры подсчитать суммы элементов в главной и побочной диагоналях в произвольно заданной квадратной матрице  $B$ . Вывести сообщение, какая диагональ имеет минимальное значение.

Вариант 5. Составить процедуру подсчета количества  $n$  элементов одномерного массива, значения которых больше числа  $P$ . С помощью этой процедуры подсчитать количество элементов, больших  $P$ , в каждом столбце двумерной матрицы  $C$ .

Вариант 6. Составить процедуру подсчета количества  $n$  элементов одномерного массива, значения которых меньше числа  $P$ . С помощью этой процедуры подсчитать количество элементов, меньших  $P$ , в каждой строке двумерной матрицы  $C$ .

Вариант 7. Составить процедуру определения максимального значения и его адреса в одномерном массиве. С помощью этой процедуры найти максимальные элементы в строках произвольно заданного двумерного массива и среди них найти максимальный, вывести номер строки, содержащей этот элемент.

Вариант 8. Составить процедуру определения минимального значения и его адреса в одномерном массиве. С помощью процедуры найти минимальные элементы в столбцах произвольно заданного двухмерного

массива и среди них найти минимальный. Вывести номер столбца, содержащего этот элемент.

Вариант 9. Составить процедуру упорядочения одномерного массива по возрастанию. С помощью процедуры отсортировать столбцы произвольно заданной двумерной матрицы. При этом вновь полученные строки также должны быть отсортированы по возрастанию.

Вариант 10. Составить процедуру упорядочения одномерного массива по убыванию. С помощью процедуры отсортировать столбцы произвольно заданной двумерной матрицы. При этом вновь полученные строки также должны быть отсортированы по убыванию.

Вариант 11. Составить процедуру подсчета количества элементов одномерного массива  $A$ , значения которых лежат в пределах от  $p$  до  $q$ . С помощью этой процедуры подсчитать количество таких элементов для каждой строки матрицы  $B$ .

Вариант 12. Составить процедуру подсчета количества элементов одномерного массива  $A$ , значения которых лежат в пределах от  $p$  до  $q$ . С помощью этой процедуры подсчитать количество таких элементов для каждого столбца матрицы  $B$ .

Вариант 13. Составить процедуру обнуления всех отрицательных элементов в одномерном массиве  $A[m]$ . С помощью этой процедуры обнулить все отрицательные элементы в  $n$ -й и  $k$ -й строке матрицы  $B[m, m]$ ,  $n \leq m, k \leq m$ .

Вариант 14. Составить процедуру обнуления всех положительных элементов в одномерном массиве  $A$ . С помощью этой процедуры обнулить все положительные элементы в  $n$ -м и  $k$ -м столбце произвольной матрицы  $B$ .

Вариант 15. Составить процедуру определения адресов элементов, значения которых лежат в пределах от  $p$  до  $q$ , в одномерном массиве  $A$ . С помощью этой процедуры определить и вывести на экран адреса соответствующих элементов в строках двумерной матрицы  $B$ .

Вариант 16. Составить процедуру определения адресов элементов, значения которых лежат в пределах от  $p$  до  $q$ , в одномерном массиве  $A$ . С помощью этой процедуры определить и вывести на экран адреса соответствующих элементов в столбцах двумерной матрицы  $B$ .

Вариант 17. Составить процедуру определения среднего арифметического значения элементов одномерного массива  $A$ . Используя эту процедуру, определить среднее арифметическое главной диагонали квадратной матрицы  $B$ .

Вариант 18. Составить процедуру определения среднего арифметического значения элементов одномерного массива  $A$ . Используя эту процедуру, определить среднее арифметическое побочной диагонали квадратной матрицы  $B$ .

Вариант 19. Составить процедуру умножения одномерного массива  $A$  на число  $q$  (каждый элемент массива умножается на число  $q$ ). Используя эту процедуру, умножить каждую строку двумерной матрицы  $B$  на число, соответствующее номеру строки.

Вариант 20. Составить процедуру умножения одномерного массива  $A$  на число  $q$  (каждый элемент массива умножается на число  $q$ ). Используя эту процедуру, умножить каждый столбец двумерной матрицы  $B$  на число, соответствующее номеру столбца.

Вариант 21. Составить процедуру подсчета суммы значений элементов в одномерном массиве. Используя ее, вычислить сумму элементов прямоугольной матрицы  $A$ .

Вариант 22. Составить процедуру подсчета суммы значений элементов в одномерном массиве. Используя ее, найти сумму элементов, расположенных выше главной диагонали для квадратных матриц  $A$  и  $B$ .

Вариант 23. Составить процедуру сортировки по возрастанию значений элементов одномерного массива. Используя ее, отсортировать элементы в каждой строке прямоугольной матрицы.

Вариант 24. Описать процедуру вычисления следа матрицы – суммы диагональных элементов. Определить, какая из матриц  $X$ ,  $Y$ ,  $Z$  имеет максимальный след.

Вариант 25. Составить процедуру умножения двух матриц произвольной размерности. Используя ее, вычислить  $k$ -ю степень квадратной матрицы  $A$ .

## КОНТРОЛЬНЫЕ ЗАДАНИЯ

Для каждого из вариантов составить программу с использованием процедур или функций. Данные для контрольного просчета выбрать самостоятельно.

Вариант 1. Составить подпрограмму для определения расстояния между точками  $A$  и  $B$  в  $n$ -мерном пространстве по формуле

$$\bar{\sigma} = \sqrt{\sum_{i=1}^n (a_i - b_i)^2},$$

где  $a_i, b_i$  – координаты точек  $A$  и  $B$ . Используя ее, найти минимальное из расстояний между точками  $X, Y, Z$ .

Вариант 2. Составить подпрограмму вычисления среднего арифметического элементов вектора. Используя ее, преобразовать квадратную матрицу следующим образом: диагональные элементы матрицы заменить средними арифметическими значениями элементов соответствующих строк.

Вариант 3. Составить подпрограмму умножения двух матриц произвольной размерности. Используя ее, вычислить  $k$ -ю степень квадратной матрицы.

Вариант 4. Составить подпрограммы определения максимального и минимального элементов в одномерном массиве. Используя их, найти минимум среди максимальных элементов строк матрицы.

Вариант 5. Составить подпрограмму определения баланса открывающих и закрывающих скобок в выражении. Используя ее, составить программу контроля правильности записи вводимых выражений.

Вариант 6. Составить подпрограмму вычисления нормы матрицы  $A = [a_{ij}]$ ,  $i, j = \overline{1, n}$ , по формуле

$$P = \max_{1 \leq i \leq n} \left\{ \sum_{j=1}^n |a_{ij}| \right\}.$$

Используя ее, определить матрицу из  $X, Y, Z$  с минимальной нормой.

Вариант 7. Составить требуемые подпрограммы работы с комплексными числами (сложение, вычитание, умножение и деление комплексных чисел). Используя эти подпрограммы, определить действительную и мнимую части числа

$$\omega = \frac{z_1^3 + 1}{z_1 z_2^2 - 1}.$$

Вариант 8. Составить подпрограммы определения максимального и минимального элемента в одномерном массиве. Используя их, найти седловую точку матрицы  $A = [a_{ij}]$ ,  $i, j = \overline{1, n}$ . Матрица имеет седловую точку  $a_{km}$ , если элемент  $a_{km}$  является минимальным в  $k$ -й строке и максимальным в  $m$ -м столбце или наоборот.



Вариант 9. Составить подпрограмму определения суммы элементов одномерного массива. Используя ее, вычислить сумму элементов матрицы.

Вариант 10. Составить подпрограмму, определяющую число и номера позиций, в которых встречается в тексте заданный символ. Используя ее, определить эти характеристики для символов 'Т', 'О', 'У' текста «по ту сторону добра и зла».

Вариант 11. Составить подпрограмму вычисления значения полинома  $n$ -го порядка по схеме Горнера. Используя ее, вычислить значение функции

$$z = \frac{12x^5 - 3x^3 + 2x^2 - 1}{6x^6 - 5x^4 + x - 8}$$

для различных значений  $x$ , вводимых с терминала.

Вариант 12. Составить подпрограмму, определяющую число заданных вдвоенных символов в тексте. Используя ее, подсчитать количество вдвоенных символов 'сс', 'ее', 'нн', 'лл' текста «класс, рассеянность, естественность, вееер, аллегро».

Вариант 13. Составить подпрограмму сортировки по убыванию значений элементов одномерного массива. Используя ее, отсортировать элементы в каждой строке матрицы.

Вариант 14. Используя необходимые подпрограммы, найти сумму элементов вспомогательных диагоналей квадратных матриц  $A$ ,  $B = \lambda \cdot A$ ,  $C = \lambda \cdot B + \mu \cdot A$ , где  $\lambda$ ,  $\mu$  – скалярные переменные.

Вариант 15. Используя необходимые подпрограммы, найти сумму элементов, расположенных выше главной диагонали, для матриц:  $X$ ,  $Y$ ,  $Z = X^2 + Y^2$ .

Вариант 16. Составить подпрограмму сортировки по возрастанию значений элементов одномерного массива. Используя ее, отсортировать элементы в каждой строке матрицы.

Вариант 17. Используя подпрограммы транспонирования и сложения массивов, вычислить и вывести значения следующих матриц:  $Z = A^T + B$ ,  $Y = B^T + A$ , где  $A$ ,  $B$  – исходные матрицы. Все матрицы квадратные и одинаковой размерности.

Вариант 18. Описать подпрограмму идентификации символа: буква или цифра. Используя ее, составить программу определения, является ли вводимая последовательность символов идентификатором.

Вариант 19. Описать подпрограмму вычисления следа матрицы – суммы диагональных элементов. Определить матрицу  $X$ ,  $Y$ ,  $Z$  с максимальным следом.

Вариант 20. Описать подпрограмму определения суммы цифр заданного целого числа. Определить указанные суммы для всех трехзначных целых чисел, лежащих в заданном интервале.

Вариант 21. Составить подпрограмму сортировки по убыванию значений элементов одномерного массива. Используя ее, отсортировать элементы в каждом столбце матрицы.

Вариант 22. Составить подпрограммы определения максимального и минимального элементов в одномерном массиве. Используя их, сформировать одномерные массивы, содержащие максимальные и минимальные элементы столбцов матрицы.

Вариант 23. Используя необходимые подпрограммы, найти сумму элементов, расположенных ниже побочной диагонали, для матриц  $X$ ,  $Y$ ,  $Z = X^2 + Y^2$ .

Вариант 24. Составить подпрограмму определения суммы элементов одномерного массива. Используя ее, вычислить сумму элементов матрицы, расположенных ниже главной диагонали.

Вариант 25. Составить подпрограммы определения максимального и минимального элементов в одномерном массиве. Используя их, найти максимум среди минимальных элементов столбцов матрицы.

## Тема 21. ПАРАМЕТРЫ ПРОЦЕДУР И ФУНКЦИЙ

### ПРИМЕР СОСТАВЛЕНИЯ ПРОГРАММЫ

Составить программу вычисления значения определенного интеграла

$$J = \int_a^b \frac{x^2 \ln x}{x + \sin x} dx$$

с заданной погрешностью  $\varepsilon$  по формуле трапеций

$$J = \int_a^b f(x) dx \approx h \left( \frac{y_0 + y_n}{2} + \sum_{i=1}^{n-1} y_i \right),$$

где  $h = (b - a)/n$ ;  $y_0 = f(a)$ ;  $y_n = f(b)$ ;  $y_i = f(a + i \cdot h)$ ,  $n$  – число участков разбиения интервала интегрирования.

Формула трапеций дает приближенное значение определенного интеграла. Точность приближения определяется значением числа  $n$  участков разбиения. Увеличивая  $n$ , можно обеспечить требуемую точность вычисления приближенного интеграла. Вычисление интеграла по формуле трапеций при фиксированном значении  $n$  целесообразно оформить в виде функции `Trap` с параметрами-значениями  $n$ ,  $a$ ,  $b$  и параметром-функцией  $f$ , определяющим произвольное подынтегральное выражение. Для вычислений значений определенного подынтегрального выражения введем функцию `f1`, в которой опишем заданное правило вычислений. В описании функции `f1` укажем обязательную директиву `Far` для формирования полного адреса подпрограммы. В основной программе организуем цикл по увеличению (в 2 раза) числа  $n$  отрезков разбиения и вычислению очередного приближения интеграла до тех пор, пока не будет выполнено условие достижения заданной точности вычислений. Схема проектирования алгоритма решения задачи в виде структурограммы представлена на рис. 21.1. Программа имеет следующий вид:

```
{ ***** }  
{ Программа: Интеграл. }  
{ Назначение: вычисление определенного интеграл }  
{ с заданной точностью. }  
{ Исходные данные: a, b – пределы интегрирования; }  
{ eps – погрешность вычислений. }
```

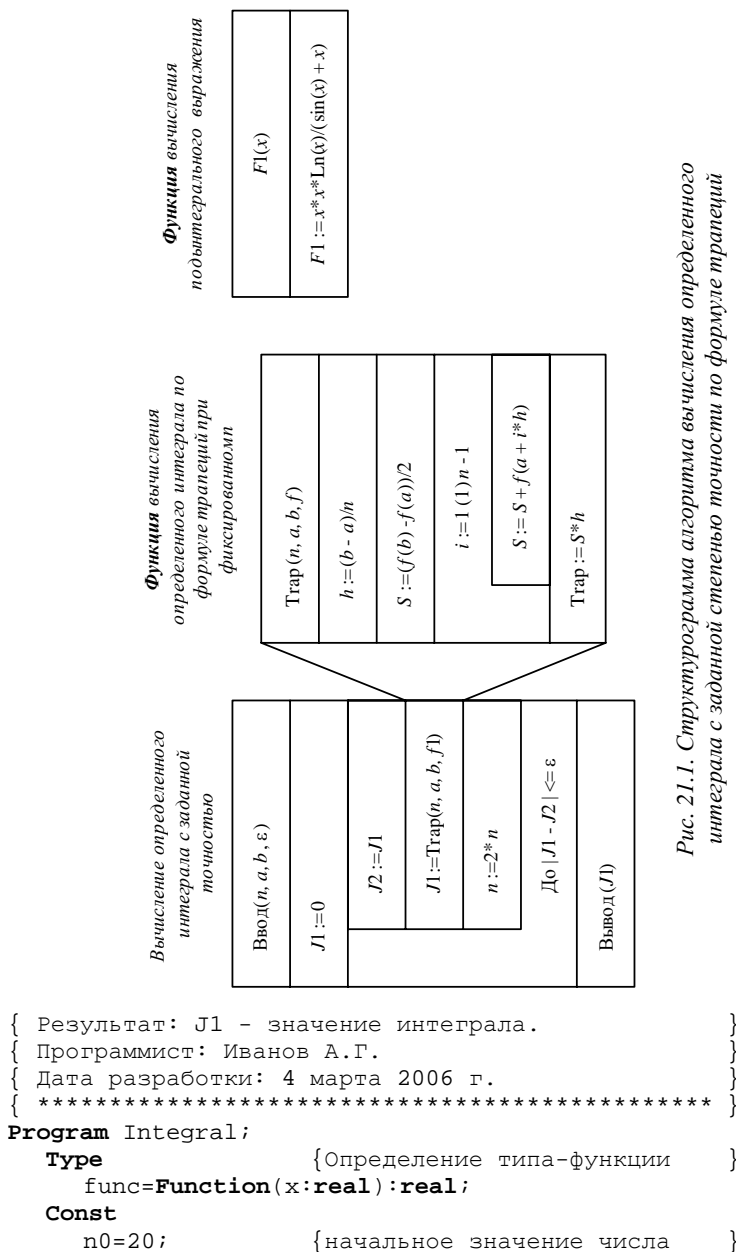


Рис. 21.1. Структурограмма алгоритма вычисления определенного интеграла с заданной степенью точности по формуле трапеций

```

Var                                {интервалов разбиения}
    nt:integer;                     {текущее значение числа}
                                {интервалов разбиения}
    a, b:real;                      {пределы интегрирования}
    J1,J2:real;                    {значения интеграла при 2*nt и nt}
    eps:real;                      {погрешность вычислений}
{Подынтегральная функция}
Function f1(x:real):real; far;
    Begin {f1}
        f1:=Sqr(x)*ln(x)/(sin(x)+x)
    End; {f1}
{Функция вычисления интеграла по формуле трапеций}
{ Параметры: n - число интервалов разбиения; }
{          a, b - пределы интегрирования; }
{          f - подынтегральная функция }
Function Trap(n:integer; a, b:real; f:func):real;
    Var
        i:integer;
        h, s:real;
    Begin {Trap}
        h:=(b-a)/n;
        s:=(f(a)+f(b))*0.5;
        for i:=1 to n-1 do
            s:=s+f(a+i*h);
        Trap:=h*s
    End; {Trap}
{Основная программа}
Begin{Integral}
    WriteLn('Введите a и b - пределы интегрирования,');
    WriteLn(' a также погрешность вычислений eps ');
    ReadLn(a, b, eps);
    WriteLn('Исходные данные:');
    WriteLn('a=',a:10:3,' b=',b:10:3,' eps=',eps:5:3);
    J1:= 0;
    nt:= n0;
    repeat
        J2:=J1;
        J1:=Trap(nt,a,b,f1); {вызов функции вычисления }
                                {интеграла по формуле }
                                {трапеций при фиксированном }
                                {числе разбиений }
        nt:=2*nt;
    until Abs(J1-J2)<=eps;
    WriteLn(' Значение интеграла: ', J1)
End. {Integral}

```

## ЗАДАНИЯ

В каждом варианте задания одну или несколько функций передавать в качестве параметров других подпрограмм, используя с этой целью процедурный тип. При выполнении заданий значения исходных данных выберите самостоятельно.

Вариант 1. Составить подпрограмму вычисления длины дуги

$$L = \sum_{i=0}^{n-1} \sqrt{[f(x_{i+1}) - f(x_i)]^2 + (x_{i+1} - x_i)^2},$$

образованной функцией  $f(x)$  в интервале  $(a, b)$ . Используя подпрограмму, определить самую длинную из дуг, образованных функциями  $f_1(x) = x^2 + 2 \cdot \ln(1 + x^2)$ ,  $f_2(x) = x \cdot \ln|x^2 - 2x|$ ,  $f_3(x) = (x^2 + 2x - 3)e^{-x}$  на интервале  $(a, b)$ .

Вариант 2. Составить подпрограмму определения минимума функции  $y = f(x)$ , заданной в дискретных точках отрезка  $[a, b]$  с постоянным шагом  $h$ . Используя ее, найти минимумы функций  $y_1 = \sin^2 x \cdot \cos x - \ln|x|$ ,  $y_2 = x^2 - 2x + 8$ .

Вариант 3. Составить подпрограмму определения максимума функции  $y = f(x)$ , заданной в дискретных точках отрезка  $[a, b]$  с постоянным шагом  $h$ . Используя ее, найти максимумы функций  $y_1 = e^{-x} \cdot \cos^2 x - 1$ ;  $y_2 = x \cdot \ln|x + 1|$ .

Вариант 4. Составить подпрограмму определения минимума функции  $z = f(x, y)$ , заданной в дискретных точках  $x_i = x_0 + i \cdot h_x$ ,  $y_j = y_0 + j \cdot h_y$  ( $i = \overline{1, n_x}$ ,  $j = \overline{1, n_y}$ ). Используя ее, найти минимумы функций  $z_1 = 3x^2 - 2y^2 + 4xy - 8x$ ,  $z_2 = 6x^2 + y^2 - 4x - 6y + 1$ .

Вариант 5. Составить подпрограмму определения максимума функции  $z = f(x, y)$ , заданной в дискретных точках  $x_i = x_0 + i \cdot h_x$ ,  $y_j = y_0 + j \cdot h_y$  ( $i = \overline{1, n_x}$ ,  $j = \overline{1, n_y}$ ). Используя ее, найти максимумы функций  $z_1 = 6x^2 + y^2 - 2xy + 2$ ,  $z_2 = 2x^2 + 3y^2 + 2x + 3y - 3xy$ .

Вариант 6. Составить подпрограмму вычисления определенного интеграла по формуле прямоугольников

$$\int_a^b f(x) dx \approx \frac{b-a}{n} \sum_{i=0}^{n-1} f\left(a + i \frac{b-a}{n}\right)$$

и, используя ее, вычислить

$$\int_0^{2\pi} (3x - \sin 2x) dx - \int_0^{\pi} (e^{-2x} - 2x + 1) dx.$$

Вариант 7. Составить подпрограмму вычисления площади треугольника, заданного координатами своих вершин, по формуле

$$S = \sqrt{p(p-a)(p-b)(p-c)},$$

где  $p = (a + b + c)/2$ ;  $a, b, c$  – длины сторон треугольника. Вычислять с использованием подпрограммы-функции как расстояние между вершинами треугольника (по формуле, приведенной в варианте 1). Найти суммарную площадь двух заданных треугольников.

Вариант 8. Составить подпрограмму вычисления коэффициента корреляции двух случайных величин  $x$  и  $y$  на основании выборок  $x = (x_1, x_2, \dots, x_n)$ ,  $y = (y_1, y_2, \dots, y_n)$  по формуле

$$R = \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) / \left( \sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2 \right).$$

Для вычисления  $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ ,  $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$  использовать подпрограмму-функцию. Найти  $R$  для произвольных выборок двух случайных величин  $x, y$ .

Вариант 9. Составить подпрограмму вычисления высот треугольника со сторонами  $a, b, c$  по формулам

$$h_a = \frac{2s}{a}, \quad h_b = \frac{2s}{b}, \quad h_c = \frac{2s}{c},$$

где  $S = \sqrt{p(p-a)(p-b)(p-c)}$ ,  $p = (a + b + c)/2$ , если заданы координаты вершин треугольника. Для определения длин сторон  $a, b, c$  использовать подпрограмму-функцию вычисления длины отрезка между двумя точками (по формуле, приведенной в варианте 1). Найти наименьшую из высот заданного треугольника.

Вариант 10. Составить подпрограмму определения координат точки пересечения двух прямых  $y = k_1x + b_1$  и  $y = k_2x + b_2$ , проходящих через заданные точки, по формулам

$$x_0 = \frac{b_2 - b_1}{k_1 - k_2}; \quad y_0 = \frac{k_1 b_2 - b_1 k_2}{k_1 - k_2}.$$

Коэффициенты  $k$  и  $b$  прямой  $y = kx + b$ , проходящей через точки  $(x_1, y_1)$ ,  $(x_2, y_2)$ , вычислить исходя из уравнения

$$\frac{y - y_1}{y_2 - y_1} = \frac{x - x_1}{x_2 - x_1}$$

с использованием подпрограммы-функции. Найти точку пересечения двух заданных прямых.

Вариант 11. Составить подпрограмму вычисления компонент вектора градиента функции  $F(x_1, x_2, \dots, x_n)$   $n$  переменных в точке  $X = (x_1, x_2, \dots, x_n)$  по конечно-разностным формулам

$$\nabla F(X) = (\nabla F_1(X), \nabla F_2(X), \dots, \nabla F_n(X)),$$

$$\text{где } \nabla F_i(X) = \frac{F(x_1, \dots, x_{i-1}, x_i + \Delta x_i, x_{i+1}, \dots, x_n) - F(x_1, x_2, \dots, x_n)}{\Delta x_i}.$$

Найти вектор градиента для функций  $z_1 = 2x^2 - 4y^2 + 8xy - 2x + 1$ ,  $z_2 = 8x^2y - 2xy$  в заданной точке  $(x, y)$  при  $\Delta x = \Delta y = \varepsilon$ .

Вариант 12. Составить подпрограмму вычисления  $k$ -й степени квадратной матрицы из  $n \times n$  элементов, используя подпрограмму умножения двух матриц. Найти  $A^3, B^2$ , где  $A, B$  – произвольные квадратные матрицы.

Вариант 13. Составить подпрограмму упорядочения элементов одномерного массива по убыванию их значений методом выбора максимального элемента, используя подпрограмму нахождения номера максимального элемента в последовательности чисел. Упорядочить по убыванию массивы  $X$  и  $Y$ .

Вариант 14. Составить функцию определения косинуса угла между двумя векторами по формуле

$$\cos \varphi = \frac{(\overline{a} \cdot \overline{b})}{|\overline{a}| \cdot |\overline{b}|},$$

$$\text{где } (\overline{ab}) = \sum_{i=1}^n a_i b_i; \quad |\overline{x}| = \sqrt{\sum_{i=1}^n x_i^2}.$$

Для вычисления скалярного произведения  $(\overline{ab})$  и модуля  $|\overline{x}|$  использовать функции. Найти косинусы углов между заданными векторами.

Вариант 15. Составить подпрограмму определения номера строки матрицы с максимальной характеристикой. Для вычисления характери-



стики использовать подпрограмму-функцию. В качестве характеристики строки рассмотреть следующие величины:

- а) сумму положительных элементов;
- б) среднее арифметическое всех элементов.

Найти строки с соответствующими максимальными характеристиками для произвольных матриц  $X$  и  $Y$ .

Вариант 16. Составить подпрограмму определения минимальной характеристики в прямоугольной матрице. Для вычисления характеристики использовать подпрограмму. В качестве характеристики рассмотреть следующие величины:

- а) произведение элементов каждого столбца матрицы;
- б) максимальный элемент столбца матрицы.

Найти соответствующие минимальные характеристики для заданных матриц  $A$  и  $B$ .

Вариант 17. Составить подпрограмму определения номеров строк матрицы с минимальной и максимальной характеристиками. Для вычисления характеристики использовать подпрограмму-функцию. В качестве характеристики рассмотреть следующие величины:

- а) среднее арифметическое элементов каждой строки матрицы;
- б) число положительных элементов в строке.

Поменять местами строки с максимальной и минимальной характеристиками типа а) в заданной матрице  $A$  и б) в заданной матрице  $B$ .

Вариант 18. Составить подпрограмму табулирования произвольной функции: вычисления множества значений  $z = \{z_i\}$ ,  $i = \overline{1, N}$ , функции  $z = f(x)$  для заданного множества значений аргумента  $X = \{x_i\}$ ,  $i = \overline{1, N}$ . Используя ее, найти множества  $z_1, z_2$  значений двух функций

$$f_1(x) = e^{-x} \cos x, \quad f_2(x) = 3 \sin^2 x$$

для двух заданных множеств значений аргумента  $x$ :  $X_1 = \{x_i\}$ ,  $i = \overline{1, N_1}$ ;  $X_2 = \{x_i\}$ ,  $i = \overline{1, N_2}$ .

Вариант 19. Назовем характеристикой строки двумерного символьного массива число элементов, относящихся к гласным русским буквам. Составить подпрограмму замены символов в строке с максимальной характеристикой на символ '\*'. Произвести указанную замену символов в произвольном исходном массиве. Характеристику определить с помощью функции.

Вариант 20. Составить подпрограмму, которая присваивает элементам одномерного массива  $Z$  значения функции  $f(x)$  в точках  $(x_1, x_2, \dots, x_n)$ . Сформировать одномерные массивы из значений функций  $f_1 = e^{-x} \cdot \cos x$ ,  $f_2 = 3 \sin^2 x$  в заданных точках.

Вариант 21. Найти полную поверхность треугольной пирамиды, заданной координатами своих вершин, используя подпрограмму-функцию вычисления длины отрезка между двумя точками (по формуле, приведенной в варианте 1).

Вариант 22. Найти объем правильной треугольной пирамиды, заданной длиной своего ребра.

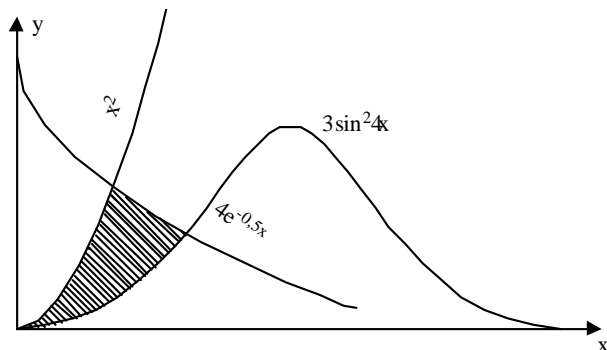
Вариант 23. Даны координаты вершин треугольника и координаты точки внутри его. Найти расстояние от данной точки до ближайшей стороны треугольника.

Вариант 24. Вычислить с точностью до  $\varepsilon$  интеграл

$$Z = \int_a^b \left\{ \int_1^{1+x} e^{-y} y^{x-1} dy \right\} dx$$

по формуле прямоугольников (см. вариант 6).

Вариант 25. С заданной погрешностью  $\varepsilon$  вычислить площадь заштрихованной фигуры, показанной на рисунке.



## Тема 22. РЕКУРСИИ

### ПРИМЕРЫ СОСТАВЛЕНИЯ ПРОГРАММ

**Пример 22.1.** Вычислить число сочетаний из  $n$  по  $m$ , определяемое формулой

$$C_n^m = \frac{n!}{m!(n-m)!}.$$

Здесь трижды вычисляется значение факториала, поэтому алгоритм получения факториала целесообразно оформить в виде подпрограммы-функции. Функция вычисления факториала может быть реализована и обычным итерационным способом с помощью оператора цикла, однако более компактное решение дает рекурсивное описание функции в соответствии с рекуррентной формулой  $n! = n(n-1)!$  при этом  $0! = 1$ .

Схема алгоритма решения данной задачи представлена на рис. 22.1.

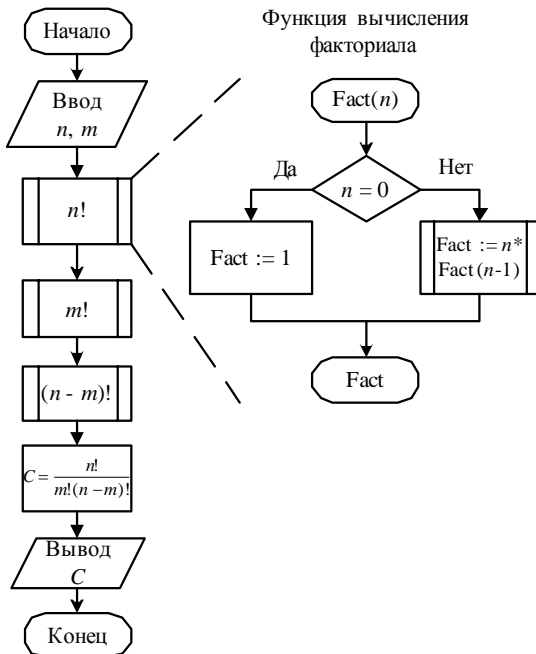


Рис. 22.1. Схема алгоритма вычисления числа сочетаний

Текст же программы имеет следующий вид:

```

Program Combination;
Var
  m, n:byte;
  C:real;
Function Fact(n:byte):LongInt;
Begin {Fact}
  if n=0
  then
    Fact:=1 {Терминальная ветвь }
  else
    Fact:=n*Fact(n-1) {Рекурсивная ветвь }
End; {Fact}
Begin {Combination}
  Write(' Введите n и m (n>=m)');
  ReadLn(n, m);
  WriteLn('Число сочетаний из ',n,' по ',m,' равно');
  C:=Fact(n)/(Fact(m)*Fact(n-m));
  WriteLn(C:10:0)
End. {Combination}

```

Здесь реализована так называемая *нисходящая рекурсия*. То есть вызов, например, вида  $Fact(5)$  означает, что функция  $Fact$  вызывает себя раз за разом:  $Fact(4)$ ,  $Fact(3)$ ,... до тех пор, пока не будет достигнута терминальная ситуация. При каждом вызове текущие вычисления «откладываются», локальные переменные и адрес возврата сохраняются в стеке. Терминальная ситуация  $Fact:=1$  достигается при  $n = 0$ . По достижении терминальной ситуации рекурсивный спуск заканчивается, начинается рекурсивный возврат из всех вызванных на данный момент копий функции: начинает строиться ответ:  $n * Fact(n-1)$ ; сохраненные локальные параметры выбираются из стека в обратной последовательности, а получаемые промежуточные результаты:  $1 \cdot 1$ ,  $2 \cdot 1$ ,  $3 \cdot 2 \cdot 1$ ,  $4 \cdot 3 \cdot 2 \cdot 1$ ,  $5 \cdot 4 \cdot 3 \cdot 2 \cdot 1$  – передаются вызывающим функциям.

Функцию вычисления факториала можно реализовать и с помощью *восходящей рекурсии*, когда ответ строится на каждой стадии рекурсивного вызова, получаемые промежуточные результаты вычисляются *перед* рекурсивным вызовом и передаются в виде дополнительного рабочего параметра подпрограммы до тех пор, пока не будет достигнута терминальная ситуация. К этому моменту ответ уже готов и нужно только передать его вызывающей функции верхнего уровня. Тогда описание функции будет иметь следующий вид:

```

Function Fact(n:byte; w:LongInt):LongInt;
  Begin {Fact}
    if n=0
    then
      Fact:=w {Терминальная ветвь }
    else
      Fact :=Fact(n-1,n*w) {Рекурсивная ветвь }
  End; {Fact}

```

Здесь  $w$  – рабочий параметр, применяемый для формирования результата. При первом вызове функции этот параметр надо инициализировать (придать ему начальное значение – 1), далее при каждом рекурсивном вызове, например при вычислении  $5!$ , он принимает последовательно значения:  $5 \cdot 1$ ,  $4 \cdot 5 \cdot 1$ ,  $3 \cdot 4 \cdot 5 \cdot 1$ ,  $2 \cdot 3 \cdot 4 \cdot 5 \cdot 1$ ,  $1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 1$ .

**Пример 22.2.** Нахождение наибольшего общего делителя (НОД) двух натуральных чисел  $n$  и  $m$  по алгоритму Евклида.

Алгоритм заключается в следующем: если  $m$  является точным делителем  $n$ , то  $\text{НОД} = m$ , в противном случае нужно брать функцию НОД от  $m$  и от остатка деления  $n$  на  $m$ .

В этом случае рекурсивная функция будет содержать две рекурсивные ветви:

```

Function NOD(n, m:byte):byte;
  Begin {NOD}
    if m>n
    then
      NOD:=NOD(m, n) {Рекурсивная ветвь }
    else
      if m=0
      then
        NOD:=n {Терминальная ветвь }
      else
        NOD:=NOD(m, n mod m) {Рекурсивная ветвь }
  End; {NOD}

```

Первая рекурсивная ветвь в описании функции позволяет писать аргументы в любом порядке. В линейной рекурсии каждый рекурсивный вызов приводит непосредственно к одному дальнейшему рекурсивному вызову. Возникает простая линейная последовательность рекурсивных вызовов.

**Пример 22.3.** Вычислить  $n$ -й член ряда Фибоначчи.

Числа Фибоначчи получаются на основании рекуррентной формулы

$$F(0) = 0, \quad F(1) = 1, \quad F(N) = F(N-1) + F(N-2).$$

Каждый элемент ряда Фибоначчи является суммой двух предшествующих элементов: 0 1 1 2 3 5 8 13 21 34 55...

Тогда описание соответствующей функции будет иметь следующий вид:

```
Function fib(n:integer):integer;
Begin {fib}
  if n=0
  then
    fib:=0 {Терминальная ветвь}
  else
    if n=1
    then
      fib:=1 {Терминальная ветвь}
    else
      fib:=fib(n-1)+fib(n-2) {Рекурсивная ветвь}
End; {fib}
```

Для определения текущего значения  $F(N)$  функция `fib` вызывает себя дважды в одной и той же рекурсивной ветви – «параллельно». Заметим, что параллельность является лишь текстуальной, но никак не временной: вычисление ветвей в стеке производится последовательно.

**Пример 22.4. Задача о ханойских башнях.** Даны три столбика –  $A$ ,  $B$ ,  $C$ . На столбике  $A$  один на другом находятся 4 диска разного диаметра и каждый меньший диск находится на большем. Требуется переместить эти четыре диска на столбик  $C$ , сохранив их взаиморасположение. Столбик  $B$  разрешается использовать как вспомогательный. За один шаг допускается перемещать только один из верхних дисков какого-либо столбика и больший диск не разрешается класть на диск меньшего диаметра.

Для определения подхода к решению поставленной задачи рассмотрим более общий случай с  $n$  дисками. Если мы сформулируем решение для  $n$  дисков в терминах решения для  $n - 1$  дисков, то поставленная проблема будет решена, поскольку задачу для  $n - 1$  дисков можно будет, в свою очередь, решить в терминах  $n - 2$  дисков и так далее до тривиального случая одного диска. А для случая одного диска решение элементарно: нужно переместить единственный диск со столбика  $A$  на столбик  $C$ . Таким образом, мы получим рекурсивное решение задачи. Рассмотрим словесное описание алгоритма.

1. Если  $n = 1$ , переместить единственный диск со столбика  $A$  на столбик  $C$  и остановиться, иначе перейти к пункту 2.

2. Переместить верхние  $n - 1$  дисков со столбика  $A$  на столбик  $B$ , используя столбик  $C$  как вспомогательный.

3. Переместить оставшийся диск со столбика  $A$  на столбик  $C$ .

4. Переместить  $n - 1$  дисков со столбика  $B$  на столбик  $C$ , используя столбик  $A$  как вспомогательный.

Приведем программу *Hanoi\_Towers*, которая решает поставленную задачу с помощью рекурсивной процедуры *Move\_Disks*.

```
{Ханойские башни}
Program Hanoi_Towers;
Var
    n:integer;
{Рекурсивная процедура}
{ n - число дисков на столбике Source}
{ Source - исходный столбик}
{ Dest - столбик, на который нужно переставить диски}
{ Tmp - вспомогательный столбик}
Procedure Move_Disks(n:byte; Source, Dest, Tmp:Char);
Begin {Move_Disks}
    if n=1
    then
        WriteLn('Переставить диск1 со столбика ',
            Source,' на столбик ',Dest)
    else
        begin
{Переставляем n-1 верхних дисков с исходного столбика на}
{вспомогательный, используя целевой диск как промежуточный}
            Move_Disks(n-1, Source, Tmp, Dest);
            WriteLn('Переставить диск ',n,' со столбика ',
                Source,' на столбик ',Dest);
{Переставляем n-1 дисков, расположенных на вспомогательном}
{столбике, на целевой, используя исходный диск как}
{промежуточный}
            Move_Disks(n-1, Tmp, Dest, Source);
        end
    End; {Move_Disks}
{Основная программа}
Begin {Hanoi_Towers}
    Write('Введите число дисков: ');
    ReadLn(n);
    Move_Disks(n, 'A', 'C', 'B');
End. {Hanoi_Towers}
```

**Пример 22.5.** Рассмотрим *метод быстрой сортировки* массива как пример поиска и построения рекурсивного алгоритма. Этот метод был предложен профессором Оксфордского университета К. Хоаром.

**Принцип метода**

Выбираем центральный элемент массива  $A$  и записываем его в переменную  $B$ . Затем элементы массива просматриваем поочередно слева направо и справа налево. При движении слева направо ищем элемент  $A[i]$ , который будет больше или равен  $B$ , и запоминаем его позицию. При движении справа налево ищем элемент  $A[j]$ , который будет меньше или равен  $B$ , и также запоминаем его позицию. Найденные элементы меняем местами и продолжаем встречный поиск до тех пор, пока встречные индексы  $i$  и  $j$  не пересекутся. После этого первый этап считается законченным, а элементы исходного массива окажутся разделенными на две части относительно значения  $B$  – все элементы, которые меньше или равны  $B$ , будут располагаться слева от границы пересечения индексов  $i$  и  $j$ , а все элементы, которые больше или равны  $B$ , будут располагаться справа.

На втором этапе повторяем действия первого этапа для левой и правой частей массива в отдельности. В результате массив окажется разбитым уже на 4 части, которые можно упорядочивать по отдельности. На третьем этапе повторяются действия первого этапа в отдельности для каждой из четырех частей и так далее, пока длина сортируемых частей не станет равной одному элементу, тогда все элементы массива будут упорядочены.

На каждом этапе повторяются одни и те же действия, но в разных индексных рамках массива; оформим их в виде рекурсивной процедуры (схема алгоритма представлена на рис. 22.1):

```
{ ***** }
{ Программа: Быстрая сортировка. }
{ Назначение: сортировка одномерного массива. }
{ Переменные: A – сортируемый массив; }
{ n – размер массива; }
{ i – параметр цикла. }
{ Подпрограммы: }
{ QSort – рекурсивная процедура быстрой сортировки. }
{ Программист: Иванов А.Г. }
{ Дата разработки: 15 марта 2006 г. }
{ ***** }
```

**Program Quick\_sort;**

**Const**

n=10;

**Var**

A:array[1..n] of integer;

i:integer;

{ Рекурсивная процедура }

{ L,R – индексы сортируемого массива }



```
Procedure QSort (L, R:word);
  Var
    B, Tmp, i, j:integer;
  Begin {QSort}
    B:=A[(L+R) div 2];
    i:=L;
    j:=R;
    while i<=j do
      begin
        while A[i]<B do
          i:=i+1;
        while A[j]>B do
          j:=j-1;
        if i<=j
        then
          begin
            Tmp:=A[i];
            A[i]:=A[j];
            A[j]:=Tmp;
            i:=i+1;
            j:=j-1;
          end
        end;
      if L<j
      then
        QSort(L, j);      {Рекурсивный вызов}
      if i<R
      then
        QSort(i, R);      {Рекурсивный вызов}
    End; {Qsort}
  {Основная программа}
  Begin {Quick_sort}
    WriteLn('Введите элементы массива:');
    for i:=1 to n do
      Read(A[i]);
    ReadLn;
    QSort(1, n);
    WriteLn('Отсортированный массив:');
    for i:=1 to n do
      Write( A[i]:4 );
    WriteLn;
  End. {Quick_sort}
```

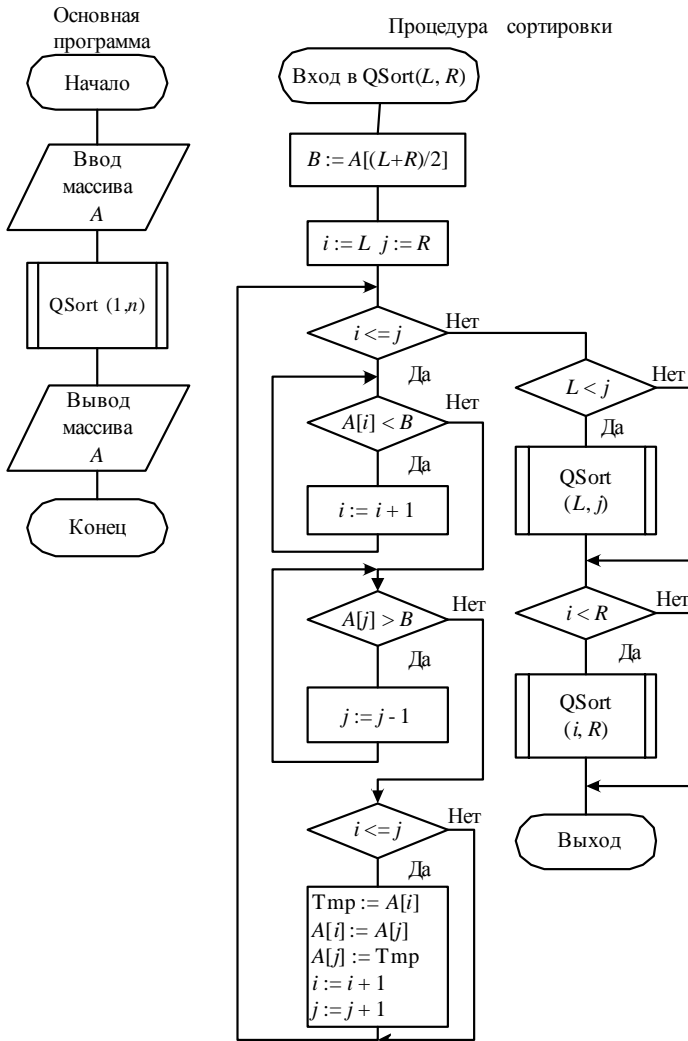


Рис. 22.2. Схема алгоритма быстрой сортировки одномерного массива

Результат работы программы *Quick\_sort*:

Введите элементы массива:

9 4 12 1 7 6 8 4 2 10

Отсортированный массив

1 2 4 4 6 7 8 9 10 12

## ЗАДАНИЯ

Для приведенных ниже заданий составить два варианта программы с использованием рекурсии и цикла и сравнить их.

Вариант 1. Вычислить сумму 12 членов рекуррентной последовательности

$$x_0 = 1; x_1 = 1; x_k = 0,7x_{k-1} + 1,1x_{k-2}, k = 2, 3, \dots$$

Вариант 2. Найти в упорядоченном массиве заданный элемент методом деления массива пополам (бинарный поиск).

Вариант 3. Определить в массиве максимальный и минимальный элементы.

Вариант 4. Вычислить функцию Бесселя 8-го порядка с аргументом  $x$ :

$$J(0, x) = x, J(1, x) = 2x, J(n, x) = \frac{2(n-1)}{x} J(n-1, x) - J(n-2, x).$$

Вариант 5. Вычислить биномиальные коэффициенты  $C_n^m$  для  $n = 0 \div 7; m = 0 \div 7$ , если

$$C_n^0 = 1 \text{ для } n \geq 0;$$

$$C_n^m = 0 \text{ для } m > n \geq 0;$$

$$C_n^m = C_{n-1}^{m-1} + C_{n-1}^m \text{ для } n \geq m > 0.$$

Вариант 6. Вычислить  $\int_0^2 \frac{\sin 2x}{x+3} dx$  с погрешностью  $10^{-7}$ .

Вариант 7. Определить 14-й член рекуррентной последовательности:  $V_1 = a_1, V_2 = a_2 + a_1, V_k = a_k V_{k-1} + a_{k-1} / V_{k-2}, a(N)$  – массив вещественных чисел.

Вариант 8. Дана функция  $f(x) = 2x^2 - 5x + 1$ . Вычислить корень уравнения  $f(x) = 0$  на отрезке  $(1, 3)$  методом деления отрезка пополам с погрешностью  $\varepsilon = 10^{-6}$ .

Вариант 9. Дана последовательность  $x_1 = 4/3, x_k = x_{k-1} \frac{4k^2}{4k^2 - 1}, k = 2, 3, \dots$

Найти первое  $x_n$ , такое, что  $(x_n - x_{n-1}) < 10^{-6}$ .

Вариант 10. Определить сумму элементов данного массива.

Вариант 11. Вывести элементы массива в обратном порядке.

Вариант 12. Установить, является ли последовательность чисел упорядоченной.

Вариант 13. Слить две упорядоченные последовательности чисел в одну.

Вариант 14. Последовательность полиномов Лагеррра  $L_0(x), \dots, L_n(x)$  определяется следующим образом:

$$L_0(x) = 1, L_1(x) = x-1, L_k(x) = (x-2k+1)L_{k-1}(x) + (k-1)^2 L_{k-2}(x).$$

Вычислить  $L_6(10)$ .

Вариант 15. Установить, является ли строка символов идентификатором.

Вариант 16. Определить, принадлежит ли заданный элемент массиву.

Вариант 17. Определить корень уравнения  $2x + \lg(2x+3) = 1$  методом Ньютона с погрешностью  $10^{-4}$  на отрезке  $[0, 0.5]$ .

Вариант 18. Вычислить  $S1 - S2$ , где  $S1$  – сумма нечетных целых чисел от 2 до 22,  $S2$  – сумма четных чисел от 5 до 17.

Вариант 19. Удалить из массива заданный элемент.

Вариант 20. Вычислить  $\int_2^3 \frac{ax^2+bx+c}{x^3-1} dx$  с погрешностью  $10^{-4}$  методом

трапеции.

Вариант 21. Проверить, удовлетворяет ли вводимый текст следующему определению:

$$\langle \text{текст} \rangle ::= \langle \text{элемент} \rangle | \langle \text{элемент} \rangle \langle \text{текст} \rangle$$

$$\langle \text{элемент} \rangle ::= a|b|(\langle \text{текст} \rangle)|[\langle \text{текст} \rangle]|\{\langle \text{текст} \rangle\}$$

где символ ':' означает "это есть", а символ '|' – "или".

Вариант 22. Вводится формула, удовлетворяющая следующему определению:

$$\langle \text{формула} \rangle ::= \langle \text{цифра} \rangle | (\langle \text{формула} \rangle \langle \text{знак} \rangle \langle \text{формула} \rangle)$$

$$\langle \text{знак} \rangle ::= + | - | *$$

$$\langle \text{цифра} \rangle ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$$

Вычислить значение вводимой формулы. (Например,  $5 \rightarrow 5$ ,  $((2-4)*6) \rightarrow -12$ .)

Вариант 23. Вычислить значения полиномов Эрмита:

$$H_0(x) = 1,$$

$$H_1(x) = 2x,$$

$$H_n(x) = 2xH_{n-1}(x) - 2(n-1)H_{n-2}(x)$$

для заданного  $n > 1$ .

Вариант 24. Вычислить значения функции Аккермана для заданных  $m$  и  $n$ :

$$Ack(0, n) = n + 1,$$

$$Ack(m, 0) = Ack(m - 1, 1),$$

$$Ack(m, n) = Ack(m - 1, Ack(m, n - 1)).$$

Вариант 25. Вывести вначале все отрицательные элементы массива, а затем положительные.

## Тема 23. МОДУЛИ

### ПРИМЕР СОСТАВЛЕНИЯ ПРОГРАММЫ

Пусть, например, необходимо составить программу суммирования максимальных элементов двух одномерных массивов, поместив все подпрограммы в отдельный программный модуль.

Схема алгоритма в этом случае строится таким же образом, как и при использовании процедур и функций в одном программном модуле вместе с основной программой, т. е. приводится схема основной программы и набор схем подпрограмм, применяемых в ней (рис. 23.1).

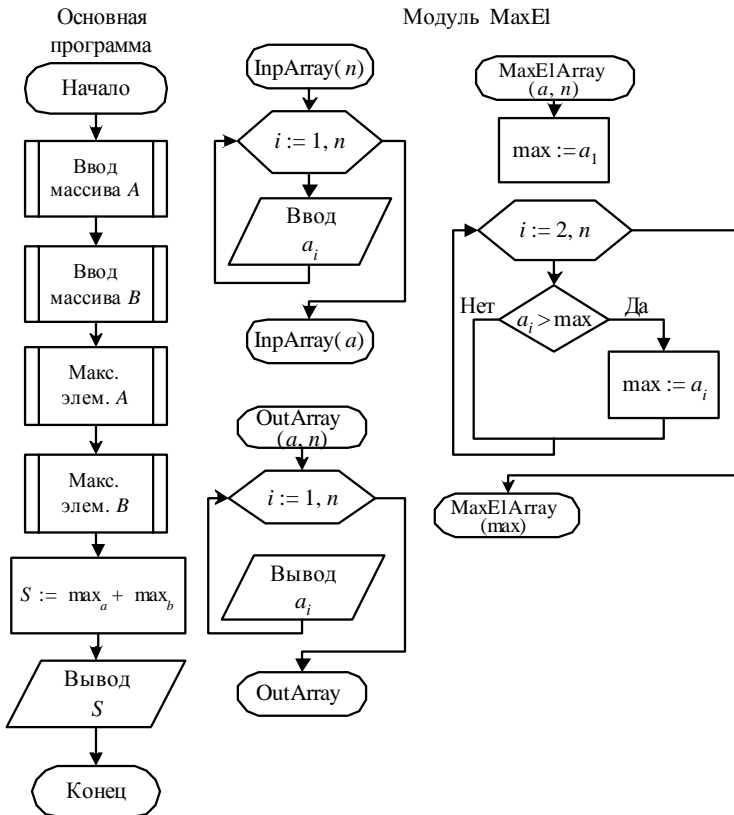


Рис. 23.1. Структура основной программы и модуля

Вначале создаем модуль, реализующий ввод-вывод одномерного массива и поиск в нем максимального элемента

```
{Модуль MaxEl, реализующий ввод-вывод одномерного      }
{ массива и поиск в нем максимального элемента,        }
{ размещается в отдельном текстовом файле MaxEl.pas   }
Unit MaxEl;
Interface          {Интерфейсная секция}
  Const
    Nmax=100;
  Type
    MyArray=array[1..Nmax] of integer;
  Procedure InpArray(Var a:MyArray; name:char;
                    n:integer);
  Procedure OutArray(a:MyArray; name:char; n:integer);
  Procedure MaxElArray(a:MyArray; n:integer;
                      Var max:integer);
Implementation {Исполняемая часть}
  {Процедура InpArray ввода одномерного массива}
  Procedure InpArray;
    Var
      i:integer;
    Begin {InpArray}
      WriteLn('Введите',n:4,' элементов массива ',
              name);
      for i:=1 to n do
        Read(a[i])
      End; {InpArray}
  {Процедура OutArray вывода одномерного массива}
  Procedure OutArray;
    Var
      i:integer;
    Begin {OutArray}
      WriteLn('Массив ',name,':');
      for i:=1 to n do
        Write(a[i]:8);
      WriteLn
    End; {OutArray}
  {Процедура MaxElArray определения максимального }
  {элемента одномерного массива                  }
  Procedure MaxElArray;
    Var
      i:integer;
    Begin {MaxElArray}
      max:=a[1];
      for i:=2 to n do
```

```

        if max<a[i]
        then
            max:=a[i]
        End {MaxElArray}
    End. {MaxEl}

```

Текст модуля помещается в файл *MaxEl.pas*, компилируется отдельно от основной программы, в результате создается *TPU*-файл с именем *MaxEl.tpu* и нашей программе могут стать доступными процедуры из новой библиотеки.

```

{Основная программа, в которой выполняются суммирование}
{максимальных элементов двух одномерных массивов}

```

```

Program MyProgram;
Uses MaxEl;
Var
    a, b:MyArray;
    maxa, maxb, na, nb, s:integer;
Begin {MyProgram}
    WriteLn('Введите количество элементов массива A');
    Read(na);
    InpArray(a, 'A', na);
    WriteLn('Введите количество элементов массива B');
    Read(nb);
    InpArray(b, 'B', nb);
    {Эхо-печать исходных массивов}
    OutArray(a, 'A', na);
    OutArray(b, 'B', nb);
    {Поиск максимальных элементов массивов}
    MaxElArray(a, na, maxa);
    MaxElArray(b, nb, maxb);
    s := maxa + maxb;
    WriteLn('Сумма максимальных элементов равна',s:8)
End. {MyProgram}

```

## ЗАДАНИЯ

Для выполнения задания необходимо взять соответствующий вариант из контрольных заданий темы 20, оформить подпрограммы и необходимые данные в библиотечный модуль, затем проверить его работу с помощью отдельной программы.



## Тема 24. СТРОКИ

### ПРИМЕРЫ СОСТАВЛЕНИЯ ПРОГРАММ

**Пример 24.1.** Рассмотрим вначале задачу проверки принадлежности букв некоторой введенной строки к латинскому алфавиту.

Для решения данной задачи необходимо во введенной строке выделять каждый символ и проверять его на принадлежность к латинскому алфавиту. Схема алгоритма представлена на рис. 24.1, а ниже приведен текст программы.

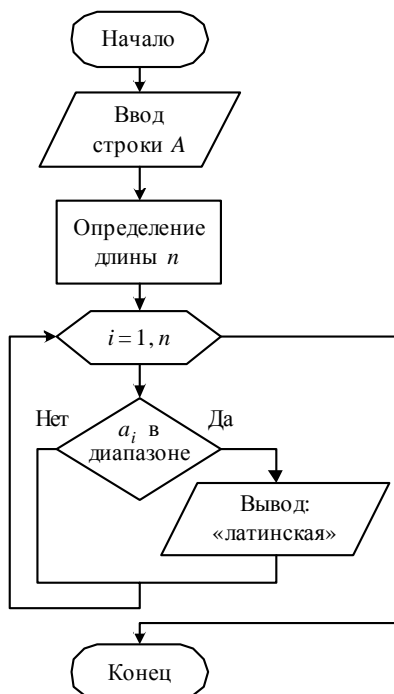


Рис. 24.1. Алгоритм определения принадлежности буквы к латинскому алфавиту

```
{Программа проверки символов на принадлежность }  
{ к латинскому алфавиту. }  
{Переменные: а – исходная строка; a[i] – символ; }  
{ i – параметр цикла; n – длина строки. }
```

```

{Программист: Нестеров В.К.}
{Дата: 29 марта 2006 г.}
Program Check;
  Var
    a:string;
    i, n:byte;
  Begin {Check}
    WriteLn('Введите строку');
    ReadLn(a);
    WriteLn('В строке');
    WriteLn(a);
    n:=Length(a);
    for i:= 1 to n do
      if (a[i]>='A') and (a[i]<='Z')
        or (a[i]>='a') and (a[i]<='z')
      then
        WriteLn('буква ',a[i], ' в позиции ',i,
          ' - латинская');
  End. {Check}

```

**Пример 24.2.** Пусть необходимо составить программу определения частоты появления строчных букв латинского алфавита в тексте, вводимом с клавиатуры.

Для решения поставленной задачи организуем прежде всего массив, который будет содержать число повторений каждой буквы. При вводе текста содержимое элемента массива, соответствующего вводимой букве, будет увеличиваться на единицу. Признаком конца вводимого текста можно считать одновременное нажатие комбинации клавиш <Ctrl>+<Z>.

Текст программы будет иметь следующий вид:

```

{Распределение частот появления букв}
Program LetterFreq;
  Var
    Letters:array['a'..'z'] of integer;
    Symbol:char;
    Index:char;
    n:integer;
  Begin {LetterFreq}
    n:=0;
    {Чтение текста}
    while not Eof do
      begin
        Read(Symbol);
        Inc(n);

```

```

    if (Symbol>='a') and (Symbol<='z')
    then
        Inc(Letters[Symbol])
    end;
{Печать результата}
    WriteLn('Литера Частота');
    for Index:='a' to 'z' do
        WriteLn(Index:3,Letters[Index]/n:10:2))
    end.{LetterFreq}

```

**Пример 24.3.** Пусть необходимо удалить из строки *St1* первое вхождение подстроки *St2*, независимо от регистра, в котором представлены строки. Это можно сделать с помощью следующего фрагмента программы:

```

{приведение строки St1 к верхнему регистру}
for i:=1 to Length(St1) do
    St1[i]:=UpCase(St1[i]);
{приведение строки St2 к верхнему регистру}
for i:=1 to Length(St2) do
    St2[i]:=UpCase(St2[i]);
{удаление подстроки St2 из строки St1}
Delete(St1,Pos(St2,St1), Length(St2));

```

**Пример 24.4.** Пусть дана строка символов. Группы символов в ней между одним или несколькими пробелами считаются словами. Подсчитать, сколько слов содержит данная строка и вывести каждое из них на отдельной строке.

При решении данной задачи следует обратить внимание на следующую важную особенность. Если бы слово от слова отделялось только одним пробелом, то мы могли бы посчитать количество пробелов и вычислить количество слов. В нашем случае необходимо перед каждым определением слова в строке удалять из нее первые пробелы и выделять слово до следующего пробела. После выделения слово удаляется из строки. Для выделения последнего слова строки, если в конце строки нет пробелов, за слово будет считаться оставшаяся часть строки.

Схема алгоритма решения данной задачи приведена на рис. 24.2. При этом алгоритм удаления пробелов из начала строки оформим в виде процедуры, а выделения слова – в виде функции.

{Программа: Подсчета и вывод слов строки.	}
{Переменные: StText – исходный текст;	}
{                  k – число слов текста.	}
{Подпрограммы:	}
{  Trim – процедура удаления пробелов;	}

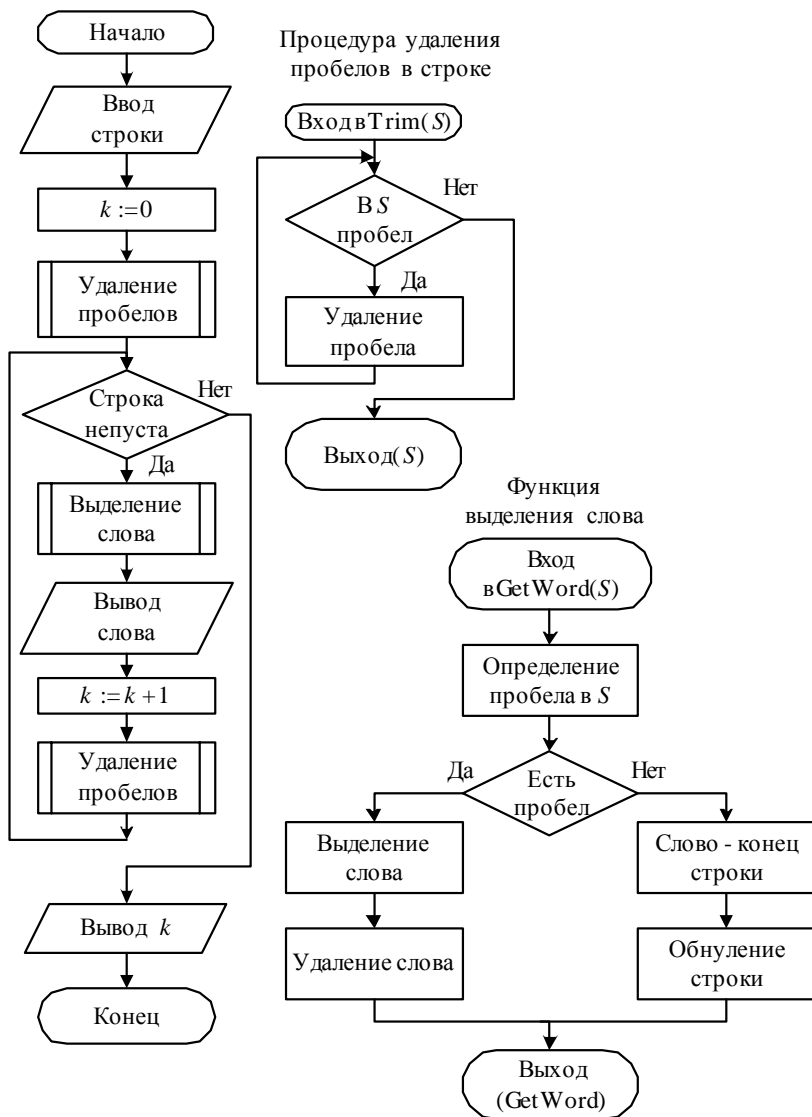


Рис. 24.2. Алгоритм подсчета количества слов в строке и их вывод

```

{ GetWord - функция выделения и удаления слова. }
{ Программист: Иванов А.Г., }
{ Дата разработки: 31 марта 2006 г. }

```

```

program Words;
  Var
    StText:string;           {Строка текста      }
    k:integer;               {Число слов в тексте }
  {Усечение пробелов в начале строки}
  procedure Trim(var S:string);
  Begin {Trim}
    while (S[1]=' ') and (Length(S)>0) do
      Delete(S,1,1);
  End; {Trim}
  {Возвращение очередного слова строки и }
  {удаление его из строки                }
  function GetWord(var S:string):string;
  Var
    i:byte;                  {Позиция пробела в строке }
  Begin {GetWord}
    i:=Pos(' ',S);
    if i>0
    then
      begin
        GetWord:=Copy(S,1,i-1);
        Delete(S,1,i);
      end
    else
      begin
        GetWord:=S;
        S:='';
      end;
    end; {GetWord}
  Begin {Words}
    WriteLn('Введите строку текста:');
    ReadLn(StText);
    WriteLn('В тексте:');
    WriteLn(StText);
    WriteLn('Содержатся слова:');
    k:=0;
    Trim(StText);
    while StText<>' ' do
      begin
        WriteLn(GetWord(StText));
        Inc(k);
        Trim(StText);
      end;
    WriteLn('Число слов в тексте - ',k)
  End. {Words}

```

## ЗАДАНИЯ

В каждом задании реализовать предложенный алгоритм в виде процедуры или функции, используя в случае необходимости библиотечные функции. В основной программе организовать ввод и вывод информации и обращение к спроектированной подпрограмме. По желанию ввод и вывод также могут быть представлены в виде процедуры или функций.

Вариант 1. Определить, является ли вводимая последовательность символов идентификатором.

Вариант 2. Подсчитать количество сдвоенных символов 'сс', 'нн', 'лл' во введенном тексте.

Вариант 3. Разбить произвольный текст на строки определенной длины. При переносе слова предусмотреть вывод дефиса.

Вариант 4. Дана символьная строка. Посчитать, сколько раз в ней встречается подслово АВВА.

Вариант 5. Найти во введенном тексте самое длинное и самое короткое слово.

Вариант 6. Из заданной строки исключить все символы, входящие в нее более одного раза.

Вариант 7. Проверить, правильно ли в заданном тексте расставлены круглые скобки.

Вариант 8. В заданной последовательности символов подсчитать общее количество символов '+', '-', '\*' и исключить их из текста.

Вариант 9. Вводится последовательность ключевых слов. Отсортировать их по алфавиту.

Вариант 10. В предложении, содержащем не менее двух слов, поменять местами первое и последнее слова.

Вариант 11. Сформировать строку, состоящую из символов, входящих одновременно в обе заданные строки.

Вариант 12. Откорректировать заданный текст, заменив в нем все вхождения одной буквы на другую.

Вариант 13. В заданном тексте перевернуть каждое слово.

Вариант 14. Дана символьная строка. Определить длину самой длинной подстроки из подряд стоящих букв «а».

Вариант 15. Дана строка символов. Определить, сколько в ней слов, начинающихся и кончающихся на одну и ту же букву.

Вариант 16. В заданной строке  $x$  заменить все вхождения подстроки  $y$  на подстроку  $z$ .

Вариант 17. Для заданного символа определить, сколько раз он встречается во введенном тексте.

Вариант 18. Из произвольной последовательности символов исключить группы символов, расположенных между круглыми скобками.

Вариант 19. Из строки символов исключить однобуквенные слова.

Вариант 20. Из заданной последовательности символов удалить лишние пробелы, разделяющие слова.

Вариант 21. Выяснить, верно ли, что среди символов строки произвольной длины имеются все символы, входящие в слово ДЕНЬ.

Вариант 22. Для каждого из слов заданного предложения указать, сколько раз оно встречается в предложении.

Вариант 23. В заданной строке символов исключить все группы символов вида  $ABC$ .

Вариант 24. Определить, можно ли из символов заданной строки составить вашу фамилию.

Вариант 25. В заданной строке символов исключить нелитерные символы.

## **КОНТРОЛЬНЫЕ ЗАДАНИЯ**

Во всех вариантах заданий исходные данные вводятся в виде текста с клавиатуры. При этом текст может содержать несколько строк. Результаты работы программы должны выводиться вместе с исходными данными.

Вариант 1. В заданной последовательности цифр найти самую длинную последовательность, которая является арифметической прогрессией.

Вариант 2. В заданном тексте найти максимальную по длине последовательность цифр.

Вариант 3. В заданном тексте определить, сколько раз встречается каждое различное двухбуквенное сочетание.

Вариант 4. Расположить все слова заданного текста на русском языке по алфавиту.

Вариант 5. В заданной последовательности латинских букв определить для каждой буквы число раз ее использования и удалить из текста слова, начинающиеся с наиболее часто встречающейся буквы.

Вариант 6. Отредактировать заданный текст, удалив из него слова, которые уже встречались в предложении ранее.

Вариант 7. Расположить все слова заданного текста в порядке убывания их длины.

Вариант 8. Составить программу поиска заданного слова в тексте и вывести число вхождений этого слова в текст.

Вариант 9. Заданы два текста. Вывести совпадающие в этих текстах слова.

Вариант 10. В заданном тексте найти все пары слов, из которых одно является обращением другого.

Вариант 11. Для каждого из слов заданного текста указать, сколько раз оно встречается в тексте.

Вариант 12. Из заданного текста вывести только те слова, которые встречаются в тексте один раз.

Вариант 13. Заданы два текста. Вывести слова первого текста, которые не встречаются во втором.

Вариант 14. Отредактировать заданный текст, удалив из него слова, которые встречаются заданное число раз.

Вариант 15. Заменить окончание «ый» каждого слова в заданном тексте на «ого». Вывести измененный текст и количество проведенных замен.

Вариант 16. В заданной последовательности символов заменить каждую последовательность из одинаковых символов длиной более трех на  $(k)s$ , где  $k$  – число повторений;  $s$  – повторяющийся символ.

Вариант 17. Отредактировать заданную последовательность целых чисел, удалив из нее числа, лежащие в заданном диапазоне.

Вариант 18. Вывести все слова заданного текста, состоящие из тех же букв, из которых состоит и первое слово.

Вариант 19. Заданы два текста. Вывести 10 самых длинных слов из двух текстов.

Вариант 20. Выполнить слияние двух текстов, расположив слова этих текстов через два пробела.

Вариант 21. Отредактировать заданный текст, удалив из него все символы, не являющиеся идентификаторами, и расположив слова по два в строке.



Вариант 22. Отредактировать заданный текст следующим образом: первое слово должно быть последним, второе – предпоследним и т. д.

Вариант 23. Из заданного текста на русском языке вывести слова с максимальным количеством гласных букв.

Вариант 24. Расположить все слова данного текста в порядке, обратном их первоначальному следованию в тексте. Вывести 10 наиболее длинных слов.

Вариант 25. Вывести данные о том, сколько раз каждая буква русского алфавита встречается в данном тексте.

## Тема 25. ЗАПИСИ

### ПРИМЕР СОСТАВЛЕНИЯ ПРОГРАММЫ

В памяти ЭВМ требуется хранить анкетные данные, представленные в виде таблицы.

АНКЕТА							
№ n/n	Ф. И. О.			ДАТА РОЖДЕНИЯ			П О Л
	ФАМИЛИЯ	ИМЯ	ОТЧЕСТВО	ГОД	МЕСЯЦ	ЧИСЛО	

Обозначим запись через идентификатор `anketa`, который определим в разделе описания переменных следующим образом:

**Var**

```
anketa: record
    npp: integer;
    fio: record
        fam, im, ot: string[15];
    end;
    dr: record
        god: 1900..2100;
        mes: string[8];
        den: 1..31;
    end;
    pol: string[3];
end;
```

Элементами записи `anketa` являются также записи `fio` и `dr`. В результате получается сложная запись.

При обращении к компонентам сложных записей можно использовать оператор **with**, имеющий вложенную структуру с любой степенью вложенности. Например, обращение к полю ДАТА РОЖДЕНИЯ – ГОД происходит с помощью составного имени `anketa.dr.god`, а с использованием оператора присоединения – с помощью конструкции

```
with anketa do
    with dr do
        god:=1965;
```

Или несколько проще:

```
with anketa, dr do
  god := 1965 ;
```

Использование оператора присоединения существенно облегчает написание программ и делает более эффективным многократное обращение к полям одной записи.

Оформив ввод и вывод записи в виде соответствующих процедур, программу формирования записи предложенного вида можно записать следующим образом:

```
{Цель: формирование и вывод записи.}
{Переменные: MyType - тип записи;}
{
  anketa - запись;
}
{
  npp - поле 'номер по порядку';
}
{
  fio - вложенная запись 'Ф.И.О.';
}
{
  fam - поле 'ФАМИЛИЯ';
}
{
  im - поле 'ИМЯ';
}
{
  ot - поле 'ОТЧЕСТВО';
}
{
  dr - вложенная запись 'ДАТА РОЖДЕНИЯ';
}
{
  god - поле 'ГОД';
}
{
  mes - поле 'МЕСЯЦ';
}
{
  den - поле 'ДЕНЬ';
}
{
  pol - поле 'ПОЛ'.
}
{Процедуры: InRec - ввод записи;
  OutRec - вывод записи.}
{Программист: Розанов В.С.}
{Дата: 7 апреля 2006 г.}
```

```
Program InOutRecord;
```

```
  Type
```

```
    MyType= record
      npp:integer;
      fio: record
        fam, im, ot:string[15];
      end;
      dr: record
        god:1900..2100;
        mes:string[8];
        den:1..31;
      end;
      pol:string[3];
    end;
```

```
  Var
```

```
    Anketa:MyType;
```

```

{Процедура ввода записи                                     }
{rec - формальный параметр типа записи }
Procedure InRec(Var rec:MyType);
  Begin {InRec}
    with rec do
      begin
        Write('Введите порядковый номер:');
        ReadLn(npp);
        with fio do
          begin
            Write('фамилия:');
            ReadLn(fam);
            Write('имя:');
            ReadLn(im);
            Write('отчество:');
            ReadLn(ot)
          end;
        with dr do
          begin
            Write('год рождения:');
            ReadLn(god);
            Write('месяц:');
            ReadLn(mes);
            Write('день:');
            ReadLn(den)
          end;
        Write('пол:');
        ReadLn(pol)
      end
    end; {InRec}
{Процедура вывода записи                                     }
{rec - формальный параметр типа записи }
Procedure OutRec(rec:MyType);
  Begin {OutRec}
    with rec do
      WriteLn('|',npp:3,'|',
        fio.fam:15-length(fio.fam),'|',
        fio.im:15-length(fio.im),'|',
        fio.ot:15-length(fio.ot),'|',dr.god,'|',
        dr.mes:8-length(dr.mes),'|',dr.den:6,'|',
        pol,'|')
    End; {OutRec}
Begin {InOutRecord}
  {Ввод записи}
    WriteLn('Введите запись');
    InRec(anketa);

```

```

{Вывод записи}
{Вывод заголовка таблицы}
  WriteLn(' _____',
    ' _____');
  WriteLn(' | _____ АНКЕТА',
    ' | _____');
  WriteLn(' | _____',
    ' | _____');
  WriteLn(' | № | _____ Ф.И.О. _____',
    ' | _____ ДАТА РОЖДЕНИЯ | ПОЛ | _____');
  WriteLn(' | _____',
    ' | _____');
  WriteLn(' | п/п | _____ ФАМИЛИЯ _____ ИМЯ _____',
    ' | _____');
  WriteLn(' | ОТЧЕСТВО | _____ ГОД | МЕСЯЦ | ЧИСЛО | _____',
    ' | _____');
  WriteLn(' | _____',
    ' | _____');
{Вывод строки записи}
  OutRec(anketa);
{Вывод конца таблицы}
  WriteLn(' _____',
    ' _____');
End. {InOutRecord}

```

Следует обратить внимание на то, что при вводе записи, содержащей как числовые так и строковые элементы, каждое поле записи целесообразно вводить с помощью оператора `ReadLn`, так как в противном случае в потоке ввода непрочитанным остается символ конца строки, который при вводе дальнейших символьных или строковых данных будет воспринят как пустая строка, что приведет к ошибочному вводу данных.

Запись `anketa` содержит анкетные данные одного лица. Если программист намеревается хранить анкетные данные о 100 человеках, то либо запись `anketa` должна иметь тип массива с числом элементов 100, либо каждый элемент второго уровня (`npp`, `fio`, `dr`, `pol`) должен быть массивом с числом элементов 100.

## ЗАДАНИЯ

Вариант 1. Сформировать запись библиотечной картотеки: шифр, автор, наименование книги, количество страниц, год издания, издательство. Вывести запись в виде библиотечной карточки.

Вариант 2. Сформировать запись, содержащую анкетные данные: Ф. И. О., дату рождения, пол, место рождения. Вывести в виде таблицы.

Вариант 3. Сформировать запись, содержащую информацию о детали: наименование, масса, габаритные размеры, материал, количество на складе.

Вариант 4. Сформировать запись, содержащую сведения о вашем месте жительства.

Вариант 5. Сформировать запись по следующей форме:

Дата			Номер заказа	Телефон		Вызываемый город
Число	Месяц	Год		вызываемый	заказчика	

Вариант 6. Сформировать запись по следующей форме:

№ п/п	Наименование товара	Количество, шт.	Стоимость, руб.	
			за штуку	Всего

Вариант 7. Сформировать запись по следующей форме:

Ф. И. О.	Профессия	Разряд	Стаж
----------	-----------	--------	------

Вариант 8. Сформировать запись о результатах спортивных соревнований: Ф. И. О. спортсмена, время на 100 м, время на 1 км, прыжок в высоту, прыжок в длину. Вывести в виде таблицы.

Вариант 9. Сформировать запись по следующей форме: Ф. И. О., оценки по пяти экзаменам. Вывести в виде таблицы.

Вариант 10. Сформировать запись по следующей форме:

№ п/п	Наименование	Единица измерения	План выпуска	
			по кварталам	Всего

Вариант 11. Сформировать запись по форме:

№ п/п	Наименование изделия	Количество, шт.	Выпуск	
			План	Факт.

Вариант 12. Сформировать запись, содержащую сведения о составе вашей семьи и возрасте ее членов.

Вариант 13. Сформировать запись, содержащую сведения о количестве, стоимости и сроке хранения молочного продукта.

Вариант 14. Сформировать запись, содержащую сведения о характеристиках микропроцессора вашего компьютера.

Вариант 15. Сформировать запись, содержащую сведения о погоде на завтра.

Вариант 16. Сформировать запись, содержащую сведения о детали: название, материал, из которого сделана деталь, ее вес, габаритные размеры.

Вариант 17. Сформировать запись, содержащую сведения об автомобиле: марка, госномер, номер двигателя, номер шасси, цвет, год производства.

Вариант 18. Сформировать запись, содержащую сведения о растении: название, класс, вид, возраст, требования по уходу.

Вариант 19. Сформировать запись, содержащую сведения о составе разборной игрушки: название игрушки, какие детали и в каком количестве присутствуют.

Вариант 20. Сформировать запись, содержащую сведения о человеке.

Вариант 21. Сформировать запись, содержащую сведения о географическом объекте.

Вариант 22. Сформировать запись, содержащую сведения о некотором животном: вид, класс, порода, цвет, рост, возраст, кличка.

Вариант 23. Сформировать запись, содержащую сведения о водоеме: географическое положение, название, глубина, площадь, наличие промысловой рыбы, состав воды.

Вариант 24. Сформировать запись, содержащую сведения об электрическом приборе: название, мощность, потребляемый ток, входное напряжение.

Вариант 25. Сформировать запись о содержимом вашего портфеля (сумочки).

## Тема 26. МАССИВЫ ЗАПИСЕЙ

### ПРИМЕР СОСТАВЛЕНИЯ ПРОГРАММЫ

Пусть, например, имеются некоторые данные о заводах города, сведенные в следующий документ:

Наименование завода	Основные сведения					Примечание
	Занимаемая площадь	Объем выпускаемой продукции		Количество обслуживающего персонала		
		по плану	фактически	с высшим образованием	со средним образованием	
АЗЛК	800	484,9	484,9	282	204	
ВАЗ	396	348,5	348,7	130	669	
ЗИЛ	203	384,3	399,4	448	125	
ИЖ	544	667,3	701,3	396	157	
ИТОГО						

Необходимо описать массив записей, содержащий данную информацию, произвести расчет и заполнить итоговую строку, а результат отпечатать.

Для решения данной задачи следует обеспечить ввод массива записей, эхо-печать введенных данных, накопления суммы по каждому полю записи для формирования итоговой строки таблицы и вывод полученного результата.

Схема алгоритма в общем виде представлена на рис. 26.1.

Ниже приведен текст программы.

```

{Цель: обработка массива записей.      }
{Переменные:                            }
{ Summary - массив записей;              }
{ plant - строка записи;                  }
{ name - наименование завода;            }
{ information - основные сведения;        }
{ area - площадь завода;                 }
{ production - объем продукции;           }
{ plan - по плану; fact - фактически;     }
{ person - обслуживающий персонал;       }
{ superior - с высшим образованием;      }
{ second - со средним образованием;      }
{ note - примечание;                     }

```



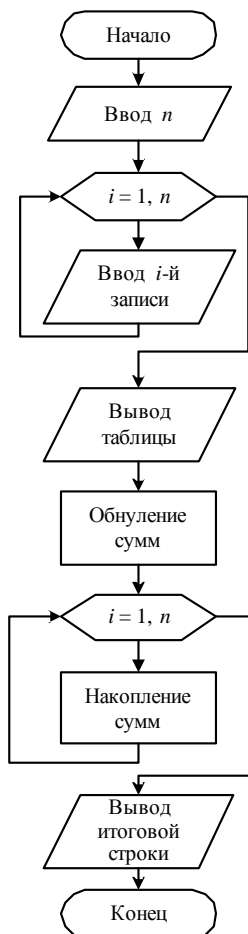


Рис. 26.1. Схема алгоритма обработки записей

```

{ Nmax - максимальное число записей; }
{ n - вводимое число записей; }
{ i - номер текущей записи; }
{ k - параметр цикла; }
{ s1,s2,s3,s4,s5 - суммы колонок. }
{Программист: Голубев А.Т. }
{Дата: 14 апреля 06 г. }

```

**Program** Example;

**Const**

Nmax=10;

```

Type
  plant= record
    name:string[5];
    information: record
      area:integer;
      production: record
        plan:real;
        fact:real
      end;
      person: record
        superior:integer;
        second:integer
      end
    end;
    note:string[8]
  end;

Var
  Summary:array[1..Nmax] of plant;
  i, k, n:integer;
  s1, s2, s3:real;
  s4, s5:integer;
Begin {Example}
  WriteLn('Введите количество заводов <= 10');
  ReadLn(n);
  {Ввод записей}
  for i:=1 to n do
    with Summary[i] do
      begin
        WriteLn('Введите данные о ',i,'-м заводе:');
        Write('наименование:');
        ReadLn(Name);
        with information do
          begin
            Write('занимаемая площадь:');
            Read(area);
            with production do
              begin
                WriteLn('производство продукции');
                Write(' по плану:');
                Read(plan);
                Write(' фактически:');
                Read(fact)
              end;
            with person do
              begin

```

```

        WriteLn('численность персонала');
        Write(' с высшем образованием:');
        Read(superior);
        Write(' со средним образованием:');
        ReadLn(second)
    end
end;
    Write('примечание:');
    ReadLn(note)
end;
{Вывод заголовка таблицы}
for k=1 to 80 do
    Write('_');
    WriteLn;
    WriteLn('|','|':15,'Основные сведения':33,'|':18,'|':13);
    Write('|','|':15);
    for k=1 to 50 do
        Write('_');
        Write('|','|':13);
        WriteLn('|','Наименование |':15,'Площадь |':10,
            'Объем продукции':18,'|':4,'Персонал':13,'|':6,
            'Примечание |':13);
        Write('|','|':15,'|':10);
    for k=1 to 40 do
        Write('_');
        Write('|','|':13);
        WriteLn('|','|':15,'|':10,'по плану |':11,'факт':7,'|':4,
            'высшее |':9,'среднее |':10,'|':13);
    for k=1 to 80 do
        Write('_');
        WriteLn;
    {Вывод строк таблицы}
    for i=1 to n do
        with Summary[i] do
            begin
                WriteLn('| ',name,' ':12-Length(name),'|':2,
                    information.area:8,'|':2,
                    information.production.plan:7:1,'|':2,
                    information.production.fact:7:1,'|':2,
                    information.person.superior:7,'|':2,
                    information.person.second:8,'|':2,
                    note,' ':12-Length(note),'|');
                for k=1 to 80 do
                    Write('_');
                    WriteLn;
                {Формирование и вывод итоговой строки}
            end
        end
    end
end

```

```

s1:=0;
s2:=0;
s3:=0;
s4:=0;
s5:=0;
for i=1 to n do
    with Summary[i].information do
        begin
            s1:=s1+area;
            s2:=s2+production.plan;
            s3:=s3+production.fact;
            s4:=s4+person.superior;
            s5:=s5+person.second
        end;
with Summary[n+1] do
    begin
        name:='ИТОГО';
        information.area:=s1;
        information.production.plan:=s2;
        information.production.fact:=s3;
        information.person.superior:=s4;
        information.person.second:=s5;
        WriteLn(' ',name,' ':12-Length(name),'|':2,s1:8,'|':2,
            s2:7:1,'|':2,s3:7:1,'|':2,s4:7,'|':2,s5:8,'|':2,'|':13)
    end;
for k=1 to 80 do
    Write(' ');
    WriteLn;
End. {Example}

```

## ЗАДАНИЯ

Вариант 1. Описать массив записей и поместить в него сведения о нескольких книгах. Предусмотреть возможность выдачи наименования книги по фамилии автора.

Вариант 2. Организовать массив записей, содержащий информацию о фамилии, имени, отчестве и номере телефона пяти ваших товарищей. Поместить в массив сведения о районе проживания этих товарищей, определив его по первым 2–3 цифрам телефона.

Вариант 3. Организовать массив записей, содержащий информацию о результатах сдачи последней сессии вашими товарищами. Определить и дополнить данные средним баллом.

Вариант 4. Описать массив записей, содержащий анкетные данные (год, месяц и день рождения; пол; место рождения; национальность) нескольких человек. Ввести данные и результат отпечатать. Предусмотреть возможность выдачи данных по введенной фамилии.

Вариант 5. Описать массив записей, который содержит информацию о нескольких деталях (наименование, масса, габаритные размеры: длина, ширина, высота; материал). Определить массу всех деталей.

Вариант 6. Описать массив записей, содержащий информацию об итогах сдачи вами экзаменационных сессий в институте. Определить средний балл.

Вариант 7. Организовать массив записей, содержащий информацию о месте жительства нескольких ваших товарищей. Предусмотреть возможность выдачи адреса по введенной фамилии.

Вариант 8. Описать массив записей и поместить в него сведения о месте работы и занимаемой должности ваших родителей. Организовать выдачу данных об одном из родителей.

Вариант 9. Описать массив записей и поместить в него следующие анкетные данные: фамилию, имя, отчество, пол, адрес (город, улица, номер дома, номер квартиры) нескольких жильцов. Предусмотреть возможность выдачи сведений о жилье по введенному адресу.

Вариант 10. Описать массив записей, содержащий информацию о нескольких деталях: наименование, материал, габариты (длина, высота, ширина), масса. Определить среднюю массу детали.

Вариант 11. Описать запись приведенной ниже таблицы, заполненной данными для нескольких человек.

Номер цеха	Ф. И. О.	Сведения о рабочих		
		Профессия	Разряд	Стаж

Предусмотреть возможность выдачи данных по введенной фамилии.

Вариант 12. Имеется документ в виде справки для 10 человек. Описать его в виде записи по следующей форме:

Сотрудник		Должность	Месячный оклад
Фамилия	Табельный номер		

Предусмотреть возможность выдачи данных по введенной фамилии.

Вариант 13. Сводка выполнения плана содержит сведения: наименование изделия, шифр, единица измерения, план выпуска, фактически выпущено, отклонение от плана (перевыполнение, недовыполнение). Описать и заполнить запись для пяти позиций сводки. Предусмотреть возможность выдачи данных по введенному наименованию изделия.

Вариант 14. Описать и заполнить запись для сводки о выполнении плана выпуска продукции для нескольких наименований по следующей форме:

№ n/n	Наимено- вание	Единица измерения	Шифр	План выпуска				
				Всего	по кварталам			
					I	II	III	IV

Предусмотреть возможность выдачи данных по введенному наименованию изделия.

Вариант 15. Описать и заполнить запись для описания следующего документа:

Инвентарная ведомость							
№ п/п	Инвен- тарный номер	Дата			Приход- ный номер	Коли- чество	Единица измерения
		Число	Месяц	Год			

Предусмотреть возможность выдачи данных по введенному инвентарному номеру изделия.

Вариант 16. Ведомость сдачи экзамена содержит следующие графы: номер по порядку, фамилия студента, номер зачетной книжки, оценка (Неуд., Удовл., Хор., Отл.). Описать и заполнить запись для студентов группы. Подсчитать процент успеваемости.

Вариант 17. Таблица содержит результаты спортивных соревнований: Ф. И. О., время на 100 м, время на 1 км, прыжок в высоту, прыжок в длину. Описать и заполнить запись для шести спортсменов. Определить лучшего спортсмена по каждому виду спорта.

Вариант 18. Пусть имеется таблица футбольного чемпионата, содержащая результаты игр между  $n$  командами. Описать таблицу в виде массива записей и составить программу подсчета количества очков, набранных каждой командой.

Вариант 19. Табель успеваемости группы студентов содержит следующие сведения: номер по порядку, фамилию, имя, отчество студента, оценки по каждому экзамену. Описать его в виде массива записей и составить программу определения количества отличников в группе.

Вариант 20. Для задания 19 составить программу выдачи оценок студентов группы по любому экзамену сессии.

Вариант 21. Для задания 19 составить программу выдачи фамилий студентов, имеющих задолженность хотя бы по одному предмету.

Вариант 22. Для задания 19 составить программу определения «качества» успеваемости, т. е. процент студентов, сдавших все экзамены на 5 и 4.

Вариант 23. Для задания 19 составить программу определения названия предмета, который был сдан лучше всего.

Вариант 24. Описать массив записей, содержащий информацию о внешности людей, такую, как пол, рост, вес, цвет волос и глаз. Написать программу, выдающую фамилию человека с указанными приметами.

Вариант 25. Для задания 24 написать программу, выдающую список людей с указанной приметой.

## Тема 27. ФАЙЛЫ

### ПОСЛЕДОВАТЕЛЬНОГО ДОСТУПА

#### ПРИМЕР СОСТАВЛЕНИЯ ПРОГРАММЫ

В качестве примера рассмотрим задачу создания и вывода файла.

Для этого необходимо создать файл, внести в него данные с клавиатуры и затем вывести содержимое файла через принтер на бумажный носитель. Схема алгоритма представлена на рис. 27.1.

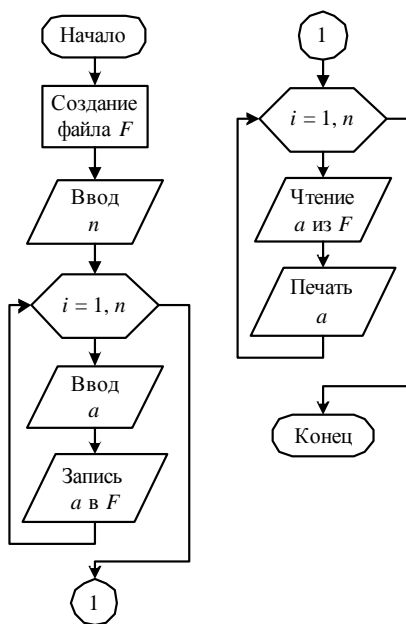


Рис. 27.1. Схема алгоритма формирования и печати файла

Программа же будет иметь следующий вид:

```
{Цель: формирование файла и вывод его }
{   на печать.                         }
{Программист: Филин С.Т.              }
{Дата написания: 21 апреля 2006 г.    }
{*****}
```

```
Program Pr;
Uses printer;
```



```
Var
  a:real;           {промежуточная переменная }
  f:file of real;   {создаваемый файл           }
  i:integer;        {счетчик элементов           }
  n:integer;        {число элементов файла       }
Begin {pr}
{создание файла}
  Assign(f,'li.lab'); {присвоить имя li.lab   }
  Rewrite(f);         {создать файл на диске   }
  WriteLn('Введите длину файла n');
  ReadLn(n);
  for i:=1 to n do
    begin
      WriteLn('Введите',i:3,'-й элемент файла');
      ReadLn(a);   {ввести элемент           }
      Write(f,a);  {записать элемент в файл   }
    end;
  Close(f);        {закрыть файл             }
{чтение и печать созданного файла}
  WriteLn(lst,'созданный файл');
  Reset (f);       {установить указатель в начало файла}
  for i:=1 to n do
    begin
      Read(f,a);
      WriteLn(lst,i:3,'-й элемент = ',a);
    end
End. {pr}
```

## ЗАДАНИЯ

Вариант 1. Сформировать файл последовательности 15 чисел, в которой каждый  $i$ -й компонент определяется по формуле

$$y = \begin{cases} \sin(i\pi/8), & \text{если } i \leq 8; \\ 4\cos(i(\pi+1)/5), & \text{если } i > 8. \end{cases}$$

Определить количество положительных значений, содержащихся в сформированном файле.

Вариант 2. Сформировать файл последовательности 20 чисел, в которой каждый  $i$ -й компонент определяется по формуле

$$y = \begin{cases} \sin(i\pi/8), & \text{если } i \leq 8; \\ 4\cos(i(\pi+1)/5), & \text{если } i > 8. \end{cases}$$

Определить количество отрицательных значений, содержащихся в сформированном файле.

Вариант 3. Сформировать файл из значений случайных величин:

0.324, 0.524, 0.789, 0.556, 0.761, 0.248, 0.345, 0.911, 0.216.

Определить для данной последовательности среднее арифметическое компонентов, значения которых меньше 0.5.

Вариант 4. Сформировать файл из значений случайных величин:

0.324, 0.524, 0.789, 0.556, 0.761, 0.248, 0.345, 0.911, 0.216.

Определить для данной последовательности сумму компонентов, значения которых больше 0.5.

Вариант 5. Сформировать файл, содержащий фамилии нескольких студентов. Добавить к полученному файлу фамилии еще 2–3 студентов.

Вариант 6. Записать в файл оценки (в баллах), полученные некоторым студентом на экзаменах в течение всех сессий. Добавить в начало файла оценки, полученные на вступительных экзаменах.

Вариант 7. Записать в файл оценки (в баллах), полученные некоторым студентом на экзаменах в течение всех сессий, и определить средний балл.

Вариант 8. Сформировать два файла. В один из них поместить фамилии пяти ваших знакомых, а в другой – номера их телефонов. Составить программу, которая по фамилии вашего знакомого определяет номер его телефона.

Вариант 9. Сформировать два файла. В один из них поместить фамилии пяти ваших знакомых, а в другой – номера их телефонов. Составить программу, которая по номеру телефона вашего знакомого определяет его фамилию.

Вариант 10. Сформировать файл, компоненты которого являются записями, содержащими информацию о фамилии и дате рождения 10 ваших товарищей. Составить программу определения даты рождения по фамилии вашего товарища.

Вариант 11. Сформировать файл, компоненты которого являются записями, содержащими информацию о фамилии и дате рождения 10 ваших товарищей. Составить программу определения фамилии вашего товарища по дате его рождения.

Вариант 12. Сформировать файл, состоящий из пяти записей, каждая из которых имеет вид: марка автомобиля, госномер. По госномеру автомобиля определить его марку.

Вариант 13. Сформировать файл, состоящий из пяти записей, каждая из которых содержит фамилию любимого вами актера и название фильма, в котором он снимался. Составить программу определения названия фильма по фамилии актера, который в нем снимался.

Вариант 14. Сформировать файл, состоящий из пяти записей, каждая из которых содержит фамилию любимого вами актера и название фильма, в котором он снимался. Составить программу определения фамилии актера по названию фильма, в котором он снимался.

Вариант 15. Сформировать файл, компонентами которого являются действительные значения, вычисляемые по формуле

$$a_i = (i + 1)^2 \sin \frac{i\pi}{10},$$

где  $i$  – номер компонента файла.

Определить, сколько в полученном файле содержится положительных значений.

Вариант 16. Сформировать файл, компонентами которого являются действительные значения, вычисляемые по формуле

$$a_i = (i + 1)^2 \sin \frac{i\pi}{10},$$

где  $i$  – номер компонента файла.

Определить, сколько в полученном файле содержится отрицательных значений.

Вариант 17. Сформировать файл целых чисел. Вывести на экран только те компоненты файла, значения которых лежат в интервале от 0 до 25.

Вариант 18. Сформировать файл целых чисел. Вывести на экран только четные значения компонентов файла.

Вариант 19. Сформировать файл, компонентами которого являются названия нескольких троллейбусных остановок по некоторому маршруту. Добавить в конец файла названия еще нескольких остановок данного маршрута.

Вариант 20. Сформировать файл, элементами которого являются значения функции  $y = \sin(x_i) + 2\cos(x_i)$  в точках  $X = (0.1, 0.2, 0.25, 0.33, 1.78, 2.05, 2.23)$ . Определить компонент файла, имеющий минимальное значение.

Вариант 21. Сформировать файл, элементами которого являются значения функции  $y = \sin(x_i) + 2\cos(x_i)$  в точках  $X = (0.1, 0.2, 0.25, 0.33, 1.78, 2.05, 2.23)$ . Определить компонент файла, имеющий максимальное значение.

Вариант 22. Сформировать файл целых чисел. Вывести на экран только нечетные значения компонентов файла.

Вариант 23. Записать в файл оценки (в баллах), полученные студентами группы по некоторому предмету, и определить средний балл.

Вариант 24. Записать в файл оценки (в баллах), полученные студентами группы по некоторому предмету, и определить процент успеваемости.

Вариант 25. Записать в файл оценки (в баллах), полученные студентами группы по некоторому предмету, и выдать список студентов, получивших оценку «отлично».

## Тема 28. ТЕКСТОВЫЕ ФАЙЛЫ

### ПРИМЕР СОСТАВЛЕНИЯ ПРОГРАММЫ

Их применение рассмотрим на примере программы, которая в интерактивном (диалоговом) режиме обеспечивает выполнение действий по созданию, расширению и выводу текстового файла.

Алгоритм основной программы приведен на рис. 28.1.

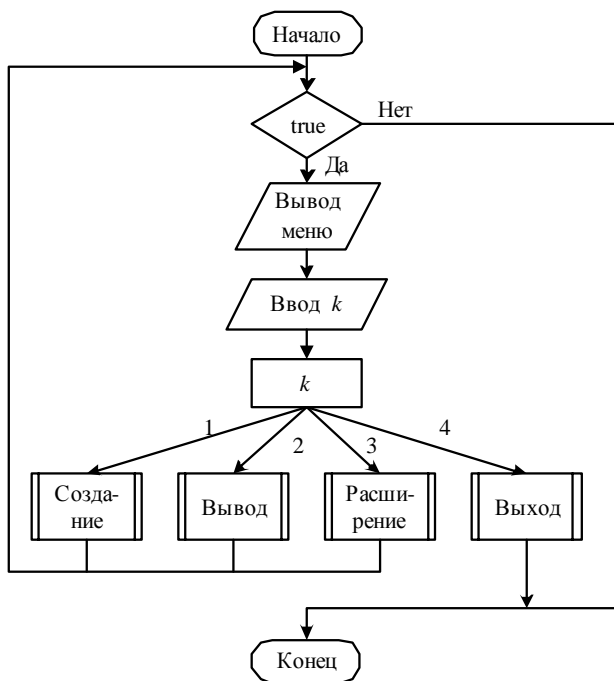


Рис. 28.1. Схема алгоритма основной программы обработки текстовых файлов

Для того чтобы иметь возможность многократно выполнять различные действия по обработке файлов, в программе организован *бесконечный цикл*, выход из которого обеспечивается процедурой прерывания программы. В выводимом в начале каждого шага цикла небольшом меню предлагается выбрать режим работы с файлом и в зависимости от значе-

ния введенной переменной  $k$  осуществляется создание, вывод или расширение файла. Схемы алгоритмов указанных процедур даны на рис. 28.2.

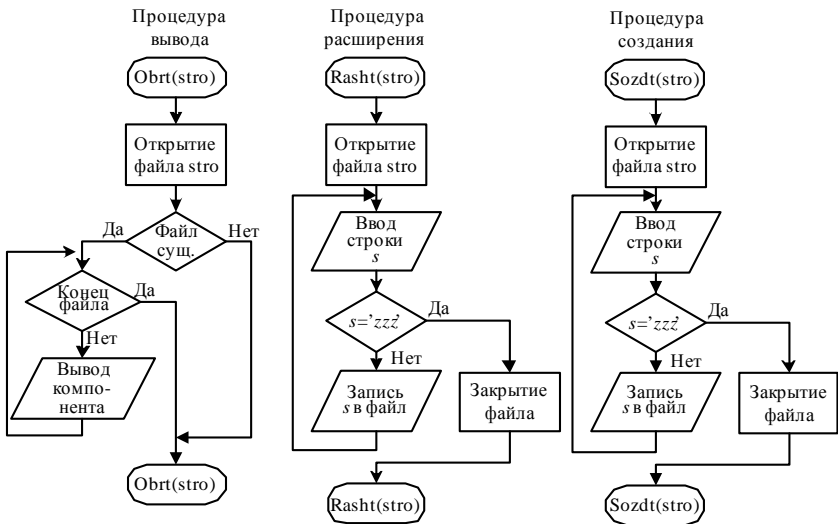


Рис. 28.2. Схемы алгоритмов процедур обработки файлов

Программа имеет следующий вид:

```

{*****}
{Программа: TextFiles.}
{Цель: создание, расширение и вывод}
{текстовых файлов.}
{Описание параметров и переменных:}
{приведены в разделе описания}
{переменных.}
{Требуемые подпрограммы:}
{obr - процедура вывода файла;}
{rasht - процедура расширения файла;}
{sozdt - процедура создания файла.}
{Метод: обработка текстовых файлов}
{Программист: Старкова Т.В.}
{Дата написания: 28 апреля 2006 г.}
{*****}
  
```

```

Program TextFiles;
Uses
  Crt;
Type
  Dlna=string[60];
  
```

```
Var
  Ft:text;           {файловая переменная      }
  S:Dlina;           {промежуточная переменная  }
  Rej:char;          {селектор режима            }
  FileName:string[14]; {имя файла                }
{процедура чтения и вывода файла}
Procedure obrt(Var stro:text);
Begin {obrt}
  Write('имя выводимого файла:');
  ReadLn(FileName);
  Assign(stro,FileName);
  {$I-}
  Reset(stro);
  if IOResult <> 0
  then
    begin
      WriteLn('Файл ',FileName,' не существует');
      Exit
    end;
  {$I+}
  while not SeekEof(stro) do
    begin
      ReadLn(stro,s);
      WriteLn(s)
    end;
End; {obrt}
{процедура расширения файла}
Procedure rasht(Var stro:text);
Begin {rasht}
  repeat
    WriteLn('имя расширяемого файла');
    ReadLn(FileName);
    Assign(stro,FileName);
    {$I-}
    Append(stro)
    {$I+}
  until IOResult=0;
  while true do
    begin
      WriteLn('введите строку');
      ReadLn(s);
      if s='zzz';
      then
        begin
          Close(stro);
```

```

        Exit
    end;
    WriteLn(stro,s)
end
End; {rasht}
{процедура создания файла}
Procedure sozdt(Var stro:text);
Begin {sozdt}
    WriteLn('имя создаваемого файла');
    ReadLn(FileName);
    Assign(stro,FileName);
    Rewrite(stro);
    while true do
        begin
            WriteLn('Введите строку (zzz - окончание ввода)');
            ReadLn(s);
            if s='zzz'
            then
                begin
                    Close(stro);
                    Exit
                end;
            WriteLn(stro,s)
        end
    end; {sozdt}
{основная программа}
Begin {TextFiles}
    while true do
        begin
            ClrScr;
            WriteLn('укажите режим');
            WriteLn('1: Создание текстового файла');
            WriteLn('2: Вывод текстового файла');
            WriteLn('3: Расширение текстового файла');
            WriteLn('4: Выход из программы');
            ReadLn(rej);
            WriteLn;
            case rej of
                '1' : sozdt(Ft);
                '2' : obrt(Ft);
                '3' : rasht(Ft);
                '4' : halt;
            else
                WriteLn ('повторите ввод номера');
            end;
        end;
    end;
end;

```



```
WriteLn('для продолжения нажмите "Enter"');  
ReadLn  
end;  
End. {TextFiles}.
```

В программе использованы стандартные средства обработки текстовых файлов. Например, процедура *Append(f<sub>v</sub>)* открывает и устанавливает указатель на маркер конца файла. Используется только для добавления элементов в текстовый файл.

Встроенная функция *IOResult* типа *Integer* позволяет определить попытку инициировать чтение из несуществующего файла или логического устройства *prn*. В программе функция *IOResult* позволяет определить, существует или не существует файл.

Логическая функция *SeekEof(f<sub>v</sub>)* пропускает все пробелы, знаки табуляции и маркеры конца строки до маркера конца файла или до первого значащего символа и возвращает значение *True*, если маркер обнаружен. Аналогично логическая функция *SeekEoln(f<sub>v</sub>)* возвращает значение *True*, если обнаружен маркер конца строки. Если аргумент *f<sub>v</sub>* не задан, то функция проверяет стандартный файл *Input*.

Процедура *Exit* осуществляет немедленный выход из исполняемой процедуры или функции в вызывающую программу. Вызов процедуры *Exit* в основной программе приводит к останову программы.

Процедура *Halt* также прекращает исполнение программы, но в некоторых случаях позволяет использовать код завершения в дальнейшем.

Процедура *ClrScr* входит в состав внешнего модуля *Crt* и становится доступной только после объявления в предложении *Uses*, которое открывает текст программы. Процедура *ClrScr* в рассматриваемом примере используется для очистки экрана дисплея перед выводом меню режимов.

## ЗАДАНИЯ

Вариант 1. Определить, является ли последовательность символов, находящаяся в текстовом файле, идентификатором.

Вариант 2. Подсчитать количество двоек символов 'сс', 'нн', 'лл' в тексте, расположенном в текстовом файле.

Вариант 3. Разбить произвольный текст, находящийся в файле, на строки определенной длины. При переносе слова предусмотреть вывод дефиса.

Вариант 4. Подсчитать число слов в предложении, записанном в текстовом файле.

Вариант 5. Найти в текстовом файле самое длинное и самое короткое слово.

Вариант 6. Из строки, расположенной в текстовом файле, исключить все символы, входящие в нее более одного раза.

Вариант 7. Проверить, правильно ли расставлены в тексте, расположенном в текстовом файле, круглые скобки.

Вариант 8. В последовательности символов, заданной в текстовом файле, подсчитать общее количество символов '+', '-', '\*'.

Вариант 9. Слова текста, расположенного в текстовом файле, вывести на экран в виде строки и в виде столбика.

Вариант 10. В текстовом файле, в предложении, содержащем не менее двух слов, поменять местами первое и последнее слово.

Вариант 11. В текстовом файле две строки текста. Необходимо сформировать третью строку, состоящую из символов, входящих одновременно в обе исходные строки, и дописать ее в текстовый файл.

Вариант 12. Откорректировать текст, расположенный в текстовом файле, заменив в нем все вхождения одной буквы на другую.

Вариант 13. Переписать текстовый файл таким образом, чтобы все слова исходного текста были перевернуты.

Вариант 14. В произвольный текстовый файл добавить в конец первое и третье слово из исходного текста.

Вариант 15. В произвольный текстовый файл добавить в конец свою фамилию.

Вариант 16. В исходном текстовом файле  $X$  заменить все вхождения подстроки  $P$  на подстроку  $Q$ .

Вариант 17. Для заданного символа определить, сколько раз он встречается во введенном тексте файла.

Вариант 18. Из текста, расположенного в файле, исключить группы символов, расположенных между круглыми скобками.

Вариант 19. Из текста, расположенного в файле, исключить однобуквенные слова. Результат записать в другой файл.

Вариант 20. Из текста, расположенного в файле, удалить лишние пробелы, разделяющие слова.

Вариант 21. Выяснить, верно ли, что среди символов строки произвольной длины, расположенной в файле, имеются все символы, входящие в слово ДЕНЬ.

Вариант 22. Для каждого из слов предложения, расположенного в одном текстовом файле, указать, сколько раз оно встречается в предложении, расположенном в другом текстовом файле.

Вариант 23. В произвольном текстовом файле исключить первое и последнее слово.

Вариант 24. Определить, можно ли из символов заданной в файле строки составить вашу фамилию.

Вариант 25. Из произвольной последовательности символов, расположенной в текстовом файле, исключить специальные символы.

## Тема 29. ФАЙЛЫ ПРОИЗВОЛЬНОГО ДОСТУПА

### ПРИМЕР СОСТАВЛЕНИЯ ПРОГРАММЫ

Пусть необходимо составить программу, используя файл записей. Задание формулируется следующим образом: для каждого студента указаны фамилия и оценки (в баллах), полученные на экзаменах по пяти дисциплинам. Требуется вычислить средний балл для каждого студента и упорядочить список студентов по убыванию среднего балла.

Для решения данной задачи необходимо вначале создать файл, компонентами которого являются записи, содержащие такие сведения о каждом студенте, как фамилия, оценки, полученные на экзаменах, и средний балл сессии (рис. 29.1). Затем в цикле производится ввод каждого компонента записи с клавиатуры и запись в файл, вычисление среднего балла сессии для каждого студента путем чтения соответствующей записи из файла, суммированием баллов на каждом экзамене и делением полученной суммы на число экзаменов. Вновь полученная запись с заполненным полем среднего балла должна быть помещена на прежнее место в файле. Поскольку при чтении или записи компонентов файла происходит сдвиг указателя файла на следующий компонент, то перед записью сформированного компонента в файл указатель файла необходимо вернуть на предыдущую позицию с помощью процедуры

`Seek(<файловая переменная>,  
<номер компонента>)`

Сортировка файла производится точно так же, как и сортировка одномерного массива (см. примеры 17.2 и 20.2). Однако при сортировке компонентов файла перед обращением к каждому компоненту необходимо устанавливать указатель файла с помощью процедуры `Seek` в позицию, соответствующую номеру компонента. Процедура вывода компонентов файла принципиально не отличается от соответствующей процедуры предыдущего примера, за исключением того, что в ней не

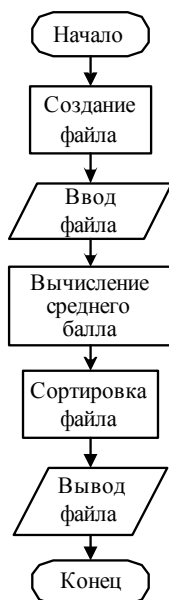


Рис. 29.1. Обобщенная схема алгоритма

проверяется существование файла, так как его вывод всегда производится только после его создания.

Тогда программа будет иметь следующий вид:

```
{Цель: определение среднего балла и сортировка.}
{Описание параметров и переменных:}
{stud - тип записи;}
{fam - фамилия студента;}
{b1, b2, b3, b4, b5 - оценки на экзаменах;}
{sb - средний балл сессии;}
{Fl - файловый тип;}
{tbl - файл табеля успеваемости группы;}
{t, y - вспомогательные переменные типа}
{компонентов файла;}
{m - длина файла;}
{i, j - параметры циклов;}
{k, r - вспомогательные переменные.}
{Подпрограмма:}
{OutFile - процедура вывода файла.}
{Метод: обработка файлов методом прямого}
{доступа.}
{Программист: Черемухин В.Н.}
{Дата написания: 12 мая 2006 г.}
```

**Program** Sball;

    Type

        stud=record

            fam:string[15];

            b1,b2,b3,b4,b5:2..5;

            sb:real;

        end;

    Fl=file of stud;

**Var**

    tbl:Fl;

    t, y:stud;

    name:string[12];

    i, j, k, m:integer;

    x:real;

**procedure** OutFile(**Var** F:Fl);

**Var**

        y:stud;

        i, m:integer;

**Begin** {OutFile}

        Reset(F);

        m:=FileSize(F);

**for** i:=1 **to** m **do**

```

begin
    Read(F,y);
    with y do
        Writeln(i:3,fam:15,sb:6:3)
    end;
    Close(F)
End; {OutFile}
Begin {Sball}
    Writeln('Введите имя файла');
    ReadLn(name);
    Assign(tbl,name);
    Rewrite(tbl);
    {ввод исходных данных}
    Writeln('Введите компоненты записей');
    while not Eof do
        begin
            with y do
                begin
                    Write('Фамилия:');
                    ReadLn(fam);
                    Write('Оценки по пяти экзаменам:');
                    ReadLn(b1,b2,b3,b4,b5)
                end;
            Write(tbl,y)
        end;
    Close(tbl);
    {Эхо-печать}
    Writeln('Введенный список:');
    OutFile(tbl);
    {вычисление среднего балла}
    Reset(tbl);
    m:=FileSize(tbl);
    for i:=1 to m do
        begin
            Read(tbl,y);
            with y do
                sb:=(b1+b2+b3+b4+b5)/5;
            Seek(tbl,i-1);
            Write(tbl,y)
        end;
    Close(tbl);
    {сортировка списка группы студентов}
    Reset(tbl);
    for i:=0 to m-2 do
        begin
            Seek(tbl,i);

```

```
k:=i;
Read(tbl,y);
x:=y.sb;
for j:=i+1 to m-1 do
  begin
    Read(tbl,t);
    if t.sb>x
    then
      begin
        k:=j;
        x:=t.sb
      end
    end;
  Seek(tbl,k);
  Read(tbl,t);
  Seek(tbl,i);
  Write(tbl,t);
  Seek(tbl,k);
  Write(tbl,y)
end;
Close(tbl);
{печать результатов}
WriteLn('Отсортированный список:');
OutFile(tbl)
End. {Sball}
```

## ЗАДАНИЯ

При выполнении заданий все преобразования производить только над исходным файлом без использования промежуточных файлов. Предусмотреть вывод содержимого файла до и после преобразования.

Вариант 1. Составить программу, определяющую правильность следования скобок в строке символов, используя для этой цели стек на основе файла.

Вариант 2. На примере своей и старшей групп создать файл-базу данных, содержащий запись вида: номер группы, номер семестра, предметы, где поле Предметы представляет собой список произвольной длины с элементами вида: название предмета, кафедры, обеспечивающая курс, фамилия лектора.

Вариант 3. Используя файл задания 2, определить, какие предметы обеспечивает данная кафедра.

Вариант 4. Используя файл задания 2, определить название предмета, который читает данный лектор, а также номер группы и номер семестра.

Вариант 5. Используя файл задания 2, определить, в каких группах и в каких семестрах преподавался определенный предмет.

Вариант 6. Используя файл задания 2, определить, какие предметы читались в заданном семестре в одной из групп.

Вариант 7. Создать файл-список идентификаторов произвольной длины.

Вариант 8. Отсортировать файл, созданный в задании 7, по алфавиту методом пузырька, не используя дополнительных файлов.

Вариант 9. Преобразовать файл, содержащий текст программы, таким образом, чтобы каждый внутренний оператор был сдвинут на две позиции вправо по сравнению с внешним.

Вариант 10. Для программы, записанной в файле в виде непрерывного текста, преобразовать файл таким образом, чтобы каждый оператор располагался на отдельной строке.

Вариант 11. В столбцах матрицы произвольного размера, размещенной во внешнем файле, провести перестановку ее элементов таким образом, чтобы максимальный элемент каждого столбца оказался на главной диагонали.

Вариант 12. Перемножить два сверхдлинных целых числа, записанных в файле. Результат записать в тот же файл.

Вариант 13. Произвести сортировку файла целых чисел методом пузырька.

Вариант 14. В файле, полученном в предыдущем задании, методом бинарного поиска найти заданный элемент и удалить его.

Вариант 15. Во внешнем файле создать очередь произвольной длины. Удалять или дополнять ее произвольным количеством элементов.

Вариант 16. Во внешнем файле создать очередь длиной  $n$  элементов. При поступлении очередного элемента первый элемент удаляется.

Вариант 17. Файл целых чисел циклически сдвинуть влево или вправо на  $K$  элементов в зависимости от знака числа  $K$ .

Вариант 18. Во внешний файл записать два многочлена в виде последовательности пар чисел – коэффициента и показателя соответствующей степени. Сложить многочлены и результат поместить в исходный файл.

Вариант 19. Пусть некоторый файл содержит числа Фибоначчи  $U_0, U_1, \dots, U_n$ . Получить и дополнить этот файл очередным компонентом  $U_{n+1}$ .



Вариант 20. Дан символьный файл  $f$ . Записать в этот же файл компоненты файла в обратном порядке.

Вариант 21. В файле целых чисел  $f$  исключить повторное вхождение одних и тех же чисел.

Вариант 22. Удалить из символьного файла  $f$  все однобуквенные слова, не используя дополнительные файлы.

Вариант 23. Дан файл целых чисел. Преобразовать его таким образом, чтобы вначале шли все отрицательные числа, а затем – неотрицательные.

Вариант 24. Заданный файл целых чисел отсортировать методом максимального элемента, не используя дополнительный файл.

Вариант 25. В некотором файле содержится таблица футбольного чемпионата, в котором участвовало  $n$  команд. Перестроить эту таблицу таким образом, чтобы команды располагались в соответствии с занятыми ими местами.

## Тема 30. ОРГАНИЗАЦИЯ ДАННЫХ В МНОЖЕСТВА

### ПРИМЕР СОСТАВЛЕНИЯ ПРОГРАММЫ

Напечатаем в порядке возрастания все простые числа из диапазона 2...64. Для решения этой задачи используем известный метод, называемый *решето Эратосфена*. Введем два множества: *sv* – множество анализируемых чисел, *prim* – множество простых чисел. Начальное значение *sv* – это все числа от 2 до 64. Берем первое простое число – 2, включаем его в множество простых чисел *prim*, а затем из исходного множества *sv* удаляем число 2 и все числа, кратные ему. Затем берем из множества *sv* наименьшее число – оно будет простым, поэтому также включаем его в множество простых чисел *prim*, а из исходного множества *sv* удаляем это число и все числа, кратные ему. Поступаем так до тех пор, пока множество *sv* не станет пустым. Программа имеет следующий вид:

```
{Цель: определение множества простых чисел. }
{Переменные: N – диапазон чисел; }
{ sv – исходное множество всех чисел; }
{ prim – множество простых чисел; }
{ nt – простое число; }
{ next – число, кратное простому. }
{Метод: решето Эратосфена. }
{Программист: Хлебников С.И. }
{Дата написания: 19 мая 2006 г. }
Program Simple;
  Const
    N=64;
  Var
    sv, prim:set of 2..N;
    next, nt:byte;
  Begin {Simple}
    sv:=[2..N]; {исходное множество }
    prim:=[]; {множество простых чисел }
    nt:=2; {первое простое число }
    repeat
      while not(nt in sv) do
        nt:=nt+1; {поиск очередного простого числа}
      prim:=prim+[nt]; {включение в prim простого числа}
      next:=nt;
    while next<=N do
```

```

begin
    sv:=sv-[next]; {сокращение исходного}
                  {множества}
    next:=next+nt
end
until sv=[ ];
{Вывод элементов множества простых чисел}
writeln('Простые числа');
for nt:=2 to N do
    if nt in prim
    then
        write(nt:5)
End. {Simple}

```

Выводятся простые числа из диапазона 2...64:

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61

В Турбо Паскале максимальный исходный диапазон может быть 2...255. Для дальнейшего увеличения исходного диапазона можно ввести массив множеств.

## ЗАДАНИЯ

Вариант 1. Вычислить сумму тех элементов матрицы  $A$ , номера строк и столбцов которых принадлежат к заданным множествам целых чисел  $S1$  и  $S2$ .

Вариант 2. Дан текст из цифр и латинских букв. Определить, каких букв – гласных ( $a, e, i, o, u$ ) или согласных – больше в этом тексте.

Вариант 3. Вывести в возрастающем порядке все цифры, входящие в десятичную запись некоторого натурального числа  $n$ .

Вариант 4. Вывести в возрастающем порядке все цифры, не входящие в десятичную запись некоторого натурального числа  $n$ .

Вариант 5. Дан текст, состоящий из латинских букв. Вывести все буквы, входящие в текст не менее двух раз.

Вариант 6. Дан текст, состоящий из латинских букв. Вывести все буквы, входящие в текст по одному разу.

Вариант 7. Дано предложение, состоящее из латинских букв. Вывести все гласные буквы, входящие в это предложение.

Вариант 8. Вывести в возрастающем порядке все целые числа из диапазона 1...1000, представимые в виде  $n^2 + m^2$ , где  $n, m > 0$ .

Вариант 9. В некотором районе города находится 5 продовольственных магазинов. В каждый из них завезли некоторые продукты из тех, что есть на базе: *хлеб, масло, молоко, мясо, рыба, соль, сыр, сахар, чай, кофе, творог, мука*. Определить, какие продукты есть во всех магазинах, какие — хотя бы в одном; каких нет нигде.

Вариант 10. Дано предложение, состоящее из латинских букв. Вывести все согласные буквы, которые входят хотя бы в одно слово.

Вариант 11. Дано предложение, состоящее из латинских букв. Вывести все согласные буквы, которые входят только в одно слово.

Вариант 12. Несколько городов одного региона обслуживает одно автотранспортное предприятие, наладившее между ними автобусные сообщения. Каждый автобус за один рейс заходит в некоторые из этих городов. Всего 5–10 рейсов. Составить программу, которая отвечала бы на вопросы: в какие города можно добраться на автобусе за один рейс, какими рейсами можно добраться из города *A* в город *B*?

Вариант 13. Дано предложение, состоящее из латинских букв. Вывести все гласные буквы, которые не входят более чем в одно слово.

Вариант 14. Введите символьное множество. Разбейте его на три подмножества: цифры; буквы; специальные символы. Проверьте, есть ли среди них пустое множество.

Вариант 15. В тексте определить и вывести повторяющиеся гласные буквы.

Вариант 16. В группе есть студенты, увлекающиеся спортом, и студенты, увлекающиеся искусством. Определить множество студентов группы, увлекающихся и спортом и искусством.

Вариант 17. В группе есть студенты, увлекающиеся спортом, и студенты, увлекающиеся искусством. Определить множество студентов, которые не увлекаются ни спортом, ни искусством.

Вариант 18. В группе есть студенты, увлекающиеся спортом, и студенты, увлекающиеся искусством. Определить множество студентов, которые увлекаются либо спортом, либо искусством.

Вариант 19. Множество *S* состоит из букв, которые содержатся хотя бы в одном из слов *Москва, Ленинград, Рязань*. Определить, можно ли из множества *S* составить слово *Липецк*.

Вариант 20. В заданном тексте из букв латинского алфавита определить общее число вхождений в него букв *a, e, c, h*.

Вариант 21. Вывести в убывающем порядке все нечетные простые числа из максимально допустимого диапазона.

Вариант 22. Вывести в убывающем порядке все четные простые числа из максимально допустимого диапазона.

Вариант 23. Написать программу, формирующую случайным образом множество целых чисел и определяющую количество элементов в этом множестве.

Вариант 24. Написать программу, формирующую случайным образом множество целых чисел и осуществляющую вывод элементов этого множества.

Вариант 25. Написать программу, формирующую случайным образом два числовых множества и определяющую, в каком отношении находятся эти множества.

## Тема 31. ДИНАМИЧЕСКИЕ СТРУКТУРЫ ДАННЫХ

### ПРИМЕР СОСТАВЛЕНИЯ ПРОГРАММЫ

Пусть необходимо сформировать два списка действительных элементов: *List1* и *List2*. Минимальный элемент списка *List1* поместить первым в список *List2*, максимальный элемент второго списка *List2* поместить последним в список *List1*. Если количество элементов соответствующего списка четно, вставить в середину списка значение среднего арифметического его элементов, если нечетно – удалить центральный элемент и на его место поместить значение среднего арифметического.

```
{Программа: Lists.}
{Цель: обработка динамических списков.}
{Переменные: List1,List2 – списки действительных}
{элементов.}
{Подпрограммы:}
{ InPutList – процедура ввода списка;}
{ OutPutList – процедура вывода списка;}
{ MinElem – функция определения минимального}
{элемента списка;}
{ MaxElem – функция определения максимального}
{элемента списка;}
{ LenList – функция определения количества}
{элементов списка;}
{ SrElem – функция определения среднего арифметического}
{элементов списка;}
{ FindElem – функция поиска элемента списка;}
{ DelElem – процедура удаления элемента списка;}
{ InsElem – процедура вставки элемента списка;}
{ Chet – процедура вставки среднего арифметического}
{элементов в список.}
{Программист: Чепрасов Д.В.}
{Дата: 26 мая 2006 г.}
Program Lists;
Uses Crt;
Type
  TData=real;           {тип элемента списка}
  PLink=^TElement;     {тип указателя списка}
  TElement = record     {динамический объект}
    Data:TData;          {поле данных}
    Link:PLink;          {ссылка на следующий}
  end;                  {элемент списка}
end;
```

```

{процедура ввода}
Procedure InPutList(Name:string; Var List:PLink);
Var
    ElNew, ElPred:PLink;
    Value:TData;
    i,n:integer;
Begin {InPutList}
    ClrScr; {очистка экрана}
    Write('Введите колич. элементов списка ',Name,':');
    ReadLn(n);
    List:=Nil;
    Writeln('Ведите элементы списка ',Name,':');
    for i:=1 to n do
        begin
            Read(Value);
            New(ElNew);
            ElNew^.Data:=Value;
            ElNew^.Link:=Nil;
            if List=nil
            then
                List:=ElNew
            else
                ElPred^.Link:=ElNew;
                ElPred:=ElNew;
            end;
    End; {InPutList}
{процедура вывода}
Procedure OutPutList(Name:string;List:PLink);
Var
    i:integer;
Begin {OutPutList}
    Writeln('Элементы списка ',Name,':');
    while List<>nil do
        begin
            Write(List^.Data:4:2,' ');
            List:=List^.Link;
        end;
    Writeln;
End; {OutPutList}
{Функция поиска минимального элемента списка}
Function MinElem(List:PLink):TData;
Var
    Min:PLink;
Begin {MinElem}
    Min:=List;
    while List<>nil do

```

```

begin
    if List^.Data<Min^.Data
    then
        Min:=List;
        List:=List^.Link;
    end;
    MinElem:=Min^.Data;
End; {MinElem}
{Функция поиска максимального элемента списка}
Function MaxElem(List:PLink):TData;
Var
    Max:PLink;
Begin {MaxElem}
    Max:=List;
    while List<>nil do
        begin
            if List^.Data>Max^.Data
            then
                Max:=List;
                List:=List^.Link;
            end;
            MaxElem:=Max^.Data;
        End; {MaxElem}
    }
{Функция определения количества элементов списка}
Function LenList(List:PLink):integer;
Var
    i:integer;
Begin {LenList}
    i:=0;
    while List<>nil do
        begin
            i:=i+1;
            List:= List^.Link;
        end;
        LenList:=i;
    End; {LenList}
}
{Функция вычисления ср. арифметического элементов списка}
Function SrElem(List:PLink):TData;
Var
    i:integer;
    Sr:TData;
Begin {SrElem}
    Sr:=0;
    i:=0;
    while List<>nil do

```



```

        begin
            Sr:=Sr+List^.Data;
            List:=List^.Link;
            i:=i+1;
        end;
        SrElem:=Sr/i;
    End; {SrElem}
{Функция поиска элемента по номеру в списке}
    Function FindElem(n:integer;List:PLink):PLink;
    Var
        i:integer;
    Begin {FindElem}
        for i:=1 to n-1 do
            List:=List^.Link;
            FindElem:=List
        End; {FindElem}
{Процедура удаления элемента по номеру в списке}
    Procedure DelElem(n:integer;Var List:PLink);
    Var
        El:PLink;
    Begin {DelElem}
        El:=FindElem(n,List)^.Link;
        FindElem(n-1,List)^.Link:=El;
    End; {DelElem}
{Процедура вставки элемента в список на заданную позицию}
    Procedure InsElem(n:integer;Elem:TData;Var List:PLink);
    Var
        ElNew:PLink;
    Begin {InsElem}
        New(ElNew);
        ElNew^.Data:=Elem;
        if n<>1
        then
            begin
                ElNew^.Link:=FindElem(n-1,List)^.Link;
                FindElem(n-1,List)^.Link:=ElNew;
            end
        else
            begin
                ElNew^.Link:=List;
                List:=ElNew;
            end
        End; {InsElem}
{Процедура вставки ср. арифметического элементов в список}
    Procedure Chet(n:integer;Var List:PLink);

```

```

Var
    S:TData;
Begin {Chet}
    S:=SrElem(List);
    if Odd(n)
    then
        begin
            DelElem(n div 2+1,List);
            InsElem(n div 2+1,S,List);
        end
    else
        InsElem(n div 2+1,S,List)
    End; {Chet}

Var
    List1, List2:PLink; {указатели на списки}
Begin {Lists} {Основная программа}
    InPutList('List1',List1);      {ввод исходных данных}
    InPutList('List2',List2);
    if (List1=nil) or (List2=nil)
    then
        begin
            Writeln('Один из списков пуст');
            Exit
        end;
        Writeln('Исходные данные:'); {эхо-печать}
        OutPutList('List1',List1);
        OutPutList('List2',List2);
        InsElem(1,MinElem(List1),List2);
        InsElem(LenList(List1)+1,MaxElem(List2),List1);
        Chet(LenList(List1),List1);
        OutPutList('List1',List1);
        Chet(LenList(List2),List2);
        OutPutList('List2',List2);
End. {Lists}

```

Результат работы программы:

Исходные данные:

Элементы списка List1:

4.50 -7.00 3.15 7.89 2.00 1.00

Элементы списка List2:

-9.00 5.00 2.00 6.40 1.00 8.00 -1.75

Элементы списка List1:

4.50 7.00 -3.15 2.79 2.00 1.00 8.00

Элементы списка List2:

-7.00 -9.00 5.00 2.00 0.58 6.40 1.00 8.00 -1.75

## ЗАДАНИЯ

При формировании списков значения элементов задавать произвольно, каждый этап алгоритма выполнять в виде отдельной процедуры. Пренебрегать вывод на печать всех промежуточных результатов работы.

Вариант 1. Составить процедуру нахождения среднего арифметического элементов непустого списка  $L$ . Используя данную процедуру, найти максимальное среднее арифметическое в списках  $K$ ,  $M$ ,  $N$ .

Вариант 2. Составить процедуру проверки упорядоченности символьных элементов списка  $L$  по алфавиту. Используя данную процедуру, проанализировать элементы списков  $M$ ,  $N$ ,  $K$ .

Вариант 3. Составить функцию, подсчитывающую количество слов списка, которые начинаются и оканчиваются одной и той же литерой. Используя данную функцию, найти сумму числа слов, начинающихся и оканчивающихся одной и той же литерой в списках  $M$ ,  $K$ ,  $L$ .

Вариант 4. Составить процедуру, которая помещает в начало списка  $L$  количество четных элементов, а в конец списка – количество нечетных элементов. С использованием данной процедуры преобразовать списки  $M$ ,  $N$  и  $K$ .

Вариант 5. Составить процедуру, проверяющую на равенство значения элементов списков  $L1$  и  $L2$  и подсчитывающую количество одинаковых элементов в них. Используя процедуру, проанализировать пары списков  $M1$  и  $M2$ ,  $N1$  и  $N2$ .

Вариант 6. Составить процедуру, определяющую вхождение списка  $L1$  в список  $L2$  и наоборот. Если один из списков длиннее, удалить лишние элементы из его начала. Используя процедуру, проанализировать пары списков  $M1$  и  $M2$ ,  $N1$  и  $N2$ .

Вариант 7. Составить процедуру, определяющую порядковый номер наибольшего элемента последовательности натуральных чисел. Используя данную процедуру, проанализировать последовательности натуральных чисел  $M$  и  $N$ .

Вариант 8. Составить процедуру вставки элемента  $E$  после каждого элемента списка, превышающего некоторое значение  $P$ . Подсчитать количество вставленных элементов.

Вариант 9. В списке натуральных чисел переставить элементы по следующему правилу: если текущий элемент больше некоторого числа  $P$ , то поместить следующий за ним элемент в конец цепочки; если те-

кущий элемент меньше или равен числу  $P$ , перенести в начало цепочки текущий элемент (первый оставить без изменения).

Вариант 10. Построить список  $L1$  – копию списка  $L$ , расположив элементы в обратном порядке (первый элемент списка  $L$  – последний элемент списка  $L1$ ). Заменить элементы списка  $L$ , имеющие четные значения, на элементы списка  $L1$ , имеющие нечетные значения.

Вариант 11. Построить список  $L$ , упорядочив его по возрастанию, из двух неупорядоченных списков  $L1$  и  $L2$ .

Вариант 12. Определить, входит ли элемент  $E$  в список  $L$ ; подсчитать количество вхождений данного элемента в список. Вставить первый элемент цепочки после каждого вхождения  $E$  в список.

Вариант 13. Построить список  $L$ , упорядочив его по убыванию, из четных элементов  $L1$  и нечетных элементов  $L2$ .

Вариант 14. Сформировать список  $L$  из элементов, которые входят одновременно в списки  $L1$  и  $L2$ . Дописать в начало элементы, которые входят в  $L1$ , но не входят в  $L2$ , а в конец – элементы, которые входят в  $L2$ , но не входят в  $L1$ .

Вариант 15. Сформировать список  $L$ , включив в него положительные элементы списка  $L1$  и отрицательные элементы списка  $L2$ . Список  $L$  отсортировать в порядке возрастания абсолютных значений элементов.

Вариант 16. Определить, является ли список  $L$  пустым; если список непуст, поменять местами первый и последний элементы списка и найти среднее арифметическое значений элементов.

Вариант 17. Сформировать списки  $L1$  и  $L2$  из списка  $L$  по следующему правилу: в  $L1$  поместить четные положительные элементы списка  $L$ , в  $L2$  – нечетные отрицательные элементы списка  $L$ . Подсчитать количество компонентов в списках  $L1$  и  $L2$ .

Вариант 18. Сформировать списки  $L1$  и  $L2$  из списка  $L$  по следующему правилу: в список  $L1$  занести порядковые номера положительных компонентов, а в список  $L2$  – отрицательных, считая от начала списка  $L$ . В начало списка  $L1$  и конец списка  $L2$  добавить порядковые номера нулевых компонентов списка  $L$ .

Вариант 19. Дописать в список  $L$  после первого вхождения элемента  $E$  список  $L1$  и удалить из списка  $L$  все оставшиеся элементы  $E$ , если таковые имеются.

Вариант 20. Дописать после каждого вхождения в список  $L$  компонента  $E$  элемент, который представляет собой среднее арифметическое положительных элементов списка  $L$ .

Вариант 21. Определить максимальные элементы списков  $L1$  и  $L2$ . Поменять местами найденные максимальные элементы в списках. Вывести порядковые номера найденных элементов, считая от начала списка.

Вариант 22. Дописать в конец списка  $L1$  максимальный четный элемент списка  $L2$ , а в начало списка –  $L2$  минимальный нечетный элемент списка  $L1$ .

Вариант 23. Вставить после максимального элемента цепочки  $L1$  компонент, равный среднему арифметическому компонентов списка  $L2$ ; перед минимальным элементом списка  $L1$  – компонент, равный произведению компонентов списка  $L2$ .

Вариант 24. Сформировать цепочку  $L$  из минимальных и максимальных элементов цепочек  $L1$  и  $L2$ , поместив минимальные элементы в начало, а максимальные в конец, а среднее арифметическое компонентов  $L1$  и  $L2$  – в середину цепочки  $L$ .

Вариант 25. Заменить минимальный и максимальный элемент списка  $L$  компонентом, равным среднему арифметическому компонентов списка  $L$ , минимальный элемент дописать в начало цепочки, максимальный – в конец.

## ЛИТЕРАТУРА

Епанешников А., Епанешников В. Программирование в среде Turbo Pascal 7.0. 4-е изд. М.: Диалог-МИФИ, 2002. 367 с.

Новичков В. С., Парфилова Н. И., Пылькин А. Н. Алгоритмизация и программирование на Турбо Паскале: Учеб. пособие. М.: Горячая линия – Телеком, 2005. 438 с.: ил.

Справочник по процедурам и функциям Borland Pascal with Objects 7.0. Киев: Диалектика, 1993. 272 с.

Фаронов В. В. Турбо Паскаль 7.0. Начальный курс: Учеб. пособие. 7-е изд., перераб. М.: Нолидж, 2000. 576 с.

# ОГЛАВЛЕНИЕ

Предисловие.....	3
Тема 1. ПРОГРАММИРОВАНИЕ ЛИНЕЙНЫХ АЛГОРИТМОВ .....	4
Пример составления программы.....	4
Задания .....	6
Тема 2. СОСТАВЛЕНИЕ И ПРИМЕНЕНИЕ ПРОСТЕЙШИХ ФУНКЦИЙ ПОЛЬЗОВАТЕЛЯ .....	10
Примеры составления программ.....	10
Задания .....	13
Тема 3. ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ РАЗВЕТВЛЯЮЩЕЙСЯ СТРУКТУРЫ С ИСПОЛЬЗОВАНИЕМ УСЛОВНОГО ОПЕРАТОРА.....	20
Примеры составления программ.....	20
Задания .....	29
Тема 4. ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ РАЗВЕТВЛЯЮЩЕЙСЯ СТРУКТУРЫ С ИСПОЛЬЗОВАНИЕМ ОПЕРАТОРА ВАРИАНТА.....	34
Пример составления программы.....	34
Задания .....	36
Контрольные задания.....	41
Тема 5. ТАБУЛИРОВАНИЕ ФУНКЦИЙ С ИСПОЛЬЗОВАНИЕМ ЦИКЛИЧЕСКИХ АЛГОРИТМОВ С ПОСТУСЛОВИЕМ.....	46
Пример составления программы.....	46
Задания .....	48
Тема 6. ПРОЕКТИРОВАНИЕ АЛГОРИТМОВ И ПРОГРАММ С ИСПОЛЬЗОВАНИЕМ ЦИКЛА С ПРЕДУСЛОВИЕМ. ОБРАБОТКА ФУНКЦИЙ .....	54
Пример составления программы.....	54
Задания .....	56

Тема 7. ОБРАБОТКА ЧИСЛОВЫХ ДАННЫХ .....	60
Пример составления программы.....	60
Задания .....	61
Тема 8. ВЫЧИСЛЕНИЕ ЧИСЛОВЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ С ПОМОЩЬЮ ЦИКЛА С ПАРАМЕТРОМ .....	63
Пример составления программы.....	63
Задания .....	64
Тема 9. ВЫЧИСЛЕНИЕ КОНЕЧНЫХ СУММ И ПРОИЗВЕДЕНИЙ .....	68
Пример составления программы.....	68
Задания .....	70
Тема 10. ВЫЧИСЛЕНИЕ СУММ ПРОСТЕЙШИХ ЧИСЛОВЫХ РЯДОВ .....	75
Примеры составления программ.....	75
Задания .....	77
Тема 11. ТАБУЛИРОВАНИЕ ФУНКЦИИ С ПОМОЩЬЮ ОПЕРАТОРА ЦИКЛА С ПАРАМЕТРОМ.....	80
Пример составления программы.....	80
Задание .....	82
Контрольные задания.....	82
Тема 12. ИТЕРАЦИОННЫЕ ЦИКЛЫ. УТОЧНЕНИЕ КОРНЕЙ УРАВНЕНИЙ МЕТОДОМ ПОЛОВИННОГО ДЕЛЕНИЯ.....	87
Пример составления программы.....	87
Задания .....	89
Тема 13. ИТЕРАЦИОННЫЕ ЦИКЛЫ. УТОЧНЕНИЕ КОРНЕЙ УРАВНЕНИЙ МЕТОДОМ ИТЕРАЦИЙ.....	91
Пример составления программы.....	91
Задания .....	92
Тема 14. ИТЕРАЦИОННЫЕ ЦИКЛЫ. ВЫЧИСЛЕНИЕ СУММЫ БЕСКОНЕЧНОГО РЯДА.....	93
Примеры составления программ.....	93
Задания .....	99
Контрольные задания.....	101



Тема 15. КОНСТРУИРОВАНИЕ ПРОГРАММ СО СТРУКТУРОЙ ВЛОЖЕННЫХ ЦИКЛОВ .....	107
Пример составления программы .....	107
Задания .....	109
Тема 16. ОРГАНИЗАЦИЯ ВВОДА-ВЫВОДА. ПРЕДСТАВЛЕНИЕ РЕЗУЛЬТАТОВ РЕШЕНИЯ .....	114
Примеры составления программ .....	114
Задания .....	124
Тема 17. ОБРАБОТКА ОДНОМЕРНЫХ МАССИВОВ ДАННЫХ .....	128
Примеры программирования задач .....	128
Задания .....	136
Тема 18. ОБРАБОТКА ДВУМЕРНЫХ МАССИВОВ ДАННЫХ. МАТРИЦЫ .....	139
Примеры составления программ .....	139
Задания .....	144
Контрольные задания .....	147
Тема 19. ПРОСТЕЙШИЕ ПРОЦЕДУРЫ .....	150
Пример составления программы .....	150
Задание .....	152
Тема 20. ИСПОЛЬЗОВАНИЕ ПРОЦЕДУР ДЛЯ ОБРАБОТКИ МАССИВОВ .....	153
Примеры составления программ .....	153
Задания .....	160
Контрольные задания .....	162
Тема 21. ПАРАМЕТРЫ ПРОЦЕДУР И ФУНКЦИЙ .....	166
Пример составления программы .....	166
Задания .....	169
Тема 22. РЕКУРСИИ .....	174
Примеры составления программ .....	174
Задания .....	182

---

Тема 23. МОДУЛИ .....	185
Пример составления программы .....	185
Задания .....	187
Тема 24. СТРОКИ .....	188
Примеры составления программ .....	188
Задания .....	193
Контрольные задания .....	194
Тема 25. ЗАПИСИ .....	197
Пример составления программы .....	197
Задания .....	200
Тема 26. МАССИВЫ ЗАПИСЕЙ .....	203
Пример составления программы .....	203
Задания .....	207
Тема 27. ФАЙЛЫ ПОСЛЕДОВАТЕЛЬНОГО ДОСТУПА .....	211
Пример составления программы .....	211
Задания .....	212
Тема 28. ТЕКСТОВЫЕ ФАЙЛЫ .....	216
Пример составления программы .....	216
Задания .....	220
Тема 29. ФАЙЛЫ ПРОИЗВОЛЬНОГО ДОСТУПА .....	223
Пример составления программы .....	223
Задания .....	226
Тема 30. ОРГАНИЗАЦИЯ ДАННЫХ В МНОЖЕСТВА .....	229
Пример составления программы .....	229
Задания .....	230
Тема 31. ДИНАМИЧЕСКИЕ СТРУКТУРЫ ДАННЫХ .....	233
Пример составления программы .....	233
Задания .....	238
ЛИТЕРАТУРА .....	241