

УДК 681.3.06(075.8)

Составители:
И.Т. Артемьев
Л.А. Ильина
Д.В. Ильин

Программирование на алгоритмических языках:
Лабораторный практикум/ Сост. И.Т. Артемьев, Л. А. Ильина,
Д.В. Ильин; Чуваш. ун-т. Чебоксары, 2005. 104 с.

Содержит варианты заданий, примеры их выполнения на языках Pascal, C++ и справочник. Помогут при проведении лабораторных занятий по курсам «Компьютерные науки», «Программирование» и «Информатика».

Для студентов I-II курсов математического факультета и факультета информатики, изучающих языки программирования.

Ответственный редактор: д-р физ.-мат. наук И. Т. Артемьев
Утверждено Методическим советом университета.

Общие указания

Лабораторный практикум по программированию на алгоритмических языках рассчитан на студентов математического и факультета информатики и предназначен для выработки навыков программирования алгоритмов различных структур на алгоритмических языках.

Лабораторные работы с 1-й по 3-ю содержат задания на разработку и программирование алгоритмов линейной, разветвляющейся, циклической структур и алгоритмов со структурой вложенных циклов.

В лабораторной работе 4 предложены задания для обработки символьных данных.

Лабораторные работы 5 и 6 содержат задания для использования подпрограмм.

Лабораторная работа 7 содержит варианты задач на построение графиков различных функций и нахождение точек их пересечения на ЭВМ.

В лабораторной работе 8 предложены задания для обработки текстовых файлов.

Задания *лабораторной работы 9* предназначены для работы с записями.

Лабораторная работа 10 – для обучения обработке типизированных файлов.

В лабораторной работе 11 предложены задания с перечисляемыми и ограниченными типами и оператором выбора.

Лабораторная работа 12 содержит задания для работы с множественными типами в Паскале. На других языках программирования таких типов нет.

Лабораторная работа 13 содержит задания на создание движущихся графических изображений.

Лабораторная работа 14 содержит задания для работы со списками.

Методические указания содержат примеры написания программ, с теоретическими сведениями можно ознакомиться в [5; 3].

Лабораторная работа 1. Программирование алгоритмов линейной структуры

Пример. Вычислить площадь прямоугольника по известной длине и ширине.

Вариант программы решения задачи на языке Pascal:

```
program lab1;
var a,b,s:real;
begin
  writeln('Введите          длину          и          ширину
прямоугольника');
  readln(a,b);
  s:=a*b;
  writeln('Площадь прямоугольника =' ,S);
end.
```

Вариант программы решения задачи на языке C++:

```
#include <iostream.h>
void main()
{
  float x, y;
  cout<<" введите длину \n"<<"x=";
  cin>>x;
  cout<<" введите ширину \n"<<"y=";
  cin>>y;
  cout<<"Площадь прямоугольника ="<<x*y;
}
```

Варианты заданий

1. Даны стороны треугольника. Найдите его площадь.
2. Дана сторона треугольника и высота, опущенная на эту сторону. Найдите площадь треугольника.
3. Даны две стороны треугольника и угол между ними. Найдите площадь.
4. Даны две стороны параллелограмма и угол между ними. Найдите площадь параллелограмма.
5. Найдите площадь параллелограмма по диагоналям и углу между ними.

6. Найдите площадь треугольника, зная его стороны и радиус описанной около него окружности.
7. Найдите площадь треугольника, зная его стороны и радиус вписанной в него окружности.
8. Дан диаметр круга, найдите его площадь и длину окружности.
9. Дана длина окружности, найдите площадь круга.
10. Найдите площадь трапеции по ее основаниям и высоте.
11. Найдите объем шара, зная его радиус.
12. Найдите боковую поверхность цилиндра по радиусу основания и высоте.
13. Найдите полную поверхность конуса по радиусу основания и образующей.
14. Найдите объем цилиндра по его радиусу основания r и высоте.
15. Даны скорость и время равномерного движения тела. Найдите пройденный им путь.
16. Найдите перемещение тела, зная ускорение, начальную и конечную скорости движения.
17. Найдите конечную скорость при равноускоренном движении с ускорением a , если тело начало движение из состояния покоя и совершило перемещение S .
18. Тело движется равномерно по окружности радиуса R со скоростью v , найдите его центростремительное ускорение.
19. Найдите кинетическую энергию тела массой m , движущегося со скоростью v .
20. Найдите потенциальную энергию тела массой m , находящегося на высоте h .
21. Найдите давление идеального газа, зная концентрацию его молекул и температуру.
22. Найдите площадь ромба по его диагоналям.
23. Найдите значение выражения $\frac{ax^2 + bx + c}{(a-x)(b-x)}$ для заданных a, b, c, x .
24. Найдите расстояние между двумя точками на плоскости, зная их координаты.
25. Найдите расстояние между двумя точками в пространстве, зная их координаты.

26. Найдите координаты середины отрезка на плоскости, зная координаты его концов.
27. Найдите радиус окружности, зная три точки на ней.
28. Найдите вершину параболы $ax^2 + bx + c$ для заданных a, b, c .
29. Найдите объем куба, зная длину его ребра.
30. Найдите объем усеченного конуса по радиусам его оснований и длине образующей.

Лабораторная работа 2. Программирование алгоритмов разветвляющейся и циклической структуры

Пример. Найти сумму положительных элементов массива $C(6)$.

Вариант программы решения задачи на языке Pascal:

```

program lab2;
var c:array[1..6] of real;
    i: integer; s:real;
begin
  writeln('Введите элементы массива');
  for i:=1 to 6 do
    readln(c[i]);
  s:=0;
  for i:=1 to 6 do
    if c[i]>0 then S:=S+c[i];
  writeln('Сумма положительных элементов=', S);
end.

```

Вариант программы решения задачи на языке C++:

```

#include <stdio.h>
void main()
{ int i;
  float c[6];
  float s=0;
  printf("Введите элементы массива C(6)\n");
  for (i=0;i<6;i++)
    scanf("%f",&c[i]);
  for (i=0;i<6;i++)
    if (c[i]>0)

```

```
s=s+c[i];  
printf("Сумма= %7.2f",s);  
}
```

Варианты заданий

1. Дан массив $A(7)$, замените в нем все целые элементы обратными, а дробные округлите.
2. Обнулите в массиве $B(14)$ все элементы из интервала (a,b) .
3. Все отрицательные элементы массива $A(13)$ занесите в массив B и упорядочите его по возрастанию элементов.
4. Все положительные элементы массива $A(9)$ занесите в массив B и упорядочите его по убыванию элементов.
5. Все четные элементы массива $A(10)$ занесите в массив C и найдите сумму его элементов.
6. Все дробные элементы массива $A(17)$ округлите и найдите сумму элементов полученного массива.
7. В массиве $C(14)$ все четные элементы разделите на 2, к нечетным прибавьте по 1.
8. В массиве $C(13)$ все элементы, кратные 9, разделите на 3, остальные замените на противоположные.
9. В массиве $C(16)$ все целые элементы разделите на 3, дробные округлите.
10. В массиве $G(12)$ все элементы, кратные 10, разделите на 5, остальные умножьте на 3.
11. В массиве $N(9)$ все элементы, кратные 6, разделите на 3, остальные уменьшите на 5.
12. В массиве $C(15)$ все элементы, кратные 8, умножьте на 3, остальные замените на противоположные.
13. В массиве $B(11)$ все элементы из интервала (a, b) занесите в массив C и найдите сумму элементов полученного массива.
14. В массиве $B(17)$ все целые элементы из интервала (a,b) занесите в массив C и найдите максимальный элемент массива C .
15. В массиве $B(12)$ все нецелые элементы из отрезка $[a,b]$ занесите в массив C и найдите минимальный элемент массива C .
16. Дан массив вещественных чисел $C(16)$. Все элементы этого массива, модуль которых больше 10, замените на 1, остальные округлите до ближайшего целого.

17. В массиве $D(15)$ элементы, кратные 5, разделите на 5, а кратные 3 – на 3.
18. Дан массив $F(14)$. Все элементы, имеющие четные индексы, обнулите, остальные разделите на 2.
19. Дан массив $G(10)$. Все элементы, индексы которых нечетны, разделите на 10, остальные замените на противоположные.
20. Дан массив $A(11)$. Все элементы этого массива из интервала (c, d) обнулите, найдите сумму остальных.
21. Дан массив $B(16)$. Элементы, не входящие в $[a, b]$, замените противоположными. Найдите произведение остальных.
22. Упорядочьте массив $B(10)$ по убыванию элементов, найдите минимальный элемент.
23. Упорядочьте массив $C(11)$ по возрастанию элементов, найдите максимальный элемент.
24. Даны 3 точки в пространстве. Найдите периметр пространственного треугольника, составленного из этих точек.
25. Даны 3 точки в пространстве A, B, C . Является ли треугольник ABC прямоугольным?
26. Даны два вектора в пространстве. Найдите их сумму, разность и модули.
27. Даны три вектора в пространстве. Найдите их сумму и модули.
28. Упорядочьте массив $S(13)$ в порядке убывания модулей его членов.
29. Упорядочьте отрицательные члены массива $S(10)$ в порядке убывания, а положительные – по возрастанию.
30. Даны 4 точки в пространстве, найдите периметр четырехугольника, образованного этими точками.

Лабораторная работа 3. Программирование алгоритмов со структурой вложенных циклов

Пример. Найти сумму элементов каждой строки матрицы $C(4,4)$.

Вариант программы решения задачи на языке Pascal:

```
program lab3;  
var c:array[1..4,1..4] of real;  
    i,j:integer; s:real;
```

```

begin
  writeln('Введите элементы матрицы');
  for i:=1 to 4 do
    for j:=1 to 4 do
      read(c[i,j]);
    writeln;
  {Выведем элементы матрицы красиво}
  for i:=1 to 4 do
    begin
      for j:=1 to 4 do
        write(c[i,j]:6:2);
      writeln;
    end;
  for i:=1 to 4 do
    begin s:=0;
      for j:=1 to 4 do
        s:=s+c[i,j];
      writeln('Сумма элементов', i, ' строки матрицы
    =', s);
    end;
  end.

```

Вариант программы решения задачи на языке C++:

```

#include <stdio.h>
void main()
{ int i,j;
  float c[4][4];
  float s;
  printf("Введите элементы матрицы C(4,4)\n");
  for (i=0;i<4;i++)
    for (j=0;j<4;j++)
      scanf("%f",&c[i][j]);
  for (i=0;i<4;i++)
    { s=0;
      for (j=0;j<4;j++)
        s=s+c[i][j];
      printf("Сумма элементов %d строки= %7.2f\n",
        i+1,s);
    }
}

```


Варианты заданий

1. Даны матрицы $A(5,3)$, $B(5,3)$. Вывести на печать матрицу $A+B$.
2. Даны матрицы $A(4,4)$, $B(4,4)$. Вывести на печать матрицу $A-B$.
3. Даны матрицы $A(5,2)$, $B(4,3)$. Найти сумму элементов матрицы A , сумму элементов матрицы B .
4. Даны матрицы $A(3,3)$, $B(3,3)$. Найти сумму элементов, стоящих на главных диагоналях этих матриц.
5. Дана матрица $X(4,4)$. Найти транспонированную матрицу.
6. Дана матрица $A(3,4)$. Найти максимальный и минимальный элементы.
7. Дана матрица $A(4,5)$. Вывести на печать все положительные элементы этой матрицы. Найти их сумму.
8. Дана матрица $A(3,5)$. Вывести на печать все отрицательные элементы этой матрицы. Найти их произведение.
9. Дана матрица $A(4,3)$. Вывести на печать все положительные элементы этой матрицы. Найти максимальный по модулю элемент.
10. Дана матрица $A(3,4)$. Упорядочить массив $C(12)$, состоящий из элементов A , по возрастанию, найти минимальный по модулю элемент.
11. Дана матрица $A(3,5)$. Составить массив, содержащий максимальные элементы каждого столбца.
12. В матрице $A(5,4)$ положительные элементы заменить на противоположные, а отрицательные обнулить. Вывести на печать полученную матрицу.
13. В матрице $A(4,3)$ поменять 2-ю и 4-ю строки местами.
14. Найти сумму элементов, стоящих выше главной диагонали матрицы $A(4,4)$.
15. Найти сумму элементов, стоящих ниже главной диагонали матрицы $A(5,5)$.
16. Дана матрица $B(3,5)$. Составить массив, содержащий произведения элементов каждой строки матрицы.
17. Дана матрица $B(5,4)$. Составить массив, содержащий суммы элементов каждой строки матрицы.
18. Вывести элементы главной диагонали матрицы $A(4,4)$, найти их сумму и произведение.
19. Проверить, симметрична ли матрица $C(4,4)$. Вычислить произведение ее элементов.

20. Даны матрицы $A(6,6)$, $B(6,6)$. Вывести на печать матрицу $C=A+B$ и транспонированную матрицу для C .
21. Даны матрицы $A(5,5)$, $B(5,5)$. Вывести на печать матрицу $2 \times A - 3 \times B$.
22. Дана матрица $A(4,3)$. Вывести на печать все положительные элементы этой матрицы. Найти их произведение.
23. Дана матрица $A(6,3)$. Вывести на печать все отрицательные элементы этой матрицы. Найти их сумму.
24. Упорядочить матрицу $A(3,4)$ по возрастанию элементов по строкам, найти минимальный по модулю элемент.
25. Упорядочить матрицу $B(3,5)$ по убыванию элементов по столбцам, найти минимальный элемент.
26. Дана матрица $A(5,6)$. Составить массив, содержащий максимальные элементы каждой строки.
27. Дана матрица $A(3,5)$. Составить массив, содержащий минимальные элементы каждого столбца.
28. В матрице $A(3,5)$ положительные элементы заменить на единичные, а отрицательные на противоположные. Вывести на печать полученную матрицу.
29. В матрице $A(4,5)$ проверить, есть ли совпадающие элементы в 1-й и 4-й строках этой матрицы, вывести их.
30. Вывести совпадающие элементы матриц $A(3,4)$ и $B(5,2)$.

Лабораторная работа 4. Обработка литерных величин

Пример 1. Переставьте буквы введенного слова в обратном порядке.

Вариант программы решения задачи на языке Pascal:

```

program lab4_1;
var
i, l, j: integer;
a: string;
c: char;
begin
writeln('введите слово');
readln(a);
l:=length(a);
i:=1; j:=l;
repeat

```

```

    c:=a[i];
    a[i]:=a[j];
    a[j]:=c;
    i:=i+1;
    j:=j-1;
until i>l/2;
writeln(a);
readln;
end.

```

Вариант программы решения задачи на языке C++:

```

// переставляет буквы слова в обратном порядке
#include <stdio.h>
#include <string.h>
void main()
{
    char a[20];
    char s;
    int i,j,m;
    // m номер позиции символа \0 в строке
    printf("введите слово\n");
    scanf("%s",&a);
    for (m=0;a[m]!='\0';m++);
    for (i=0,j=m-1;i<j;i++,j--)
    {
        s=a[i];
        a[i]=a[j];
        a[j]=s;
    }
    printf("%s\n",a);
}

```

Пример 2. Между одинаковыми буквами в слове вставить пробел.

Вариант программы решения задачи на языке Pascal:

```

program lab4_2;
var
    i,d:integer;
    a,c:string;
begin

```

```
writeln('введите слово');
readln(a);
d:=length(a);
for i:=1 to d do
  for j:=1 to d do
    if a[i]=a[j] and (i<>j) then c:=c+a[i]+' '
      else c:=c+a[i];
writeln('преобразованное слово');
writeln(c);
readln;
end.
```

Вариант программы решения задачи на языке C++:

```
// между одинаковыми буквами вставляет пробел
#include <stdio.h>
#include <string.h>
void main()
{ int i,j;
  char a[20],b[40];
  printf("введите слово\n");
  scanf("%s",&a);
  int l=strlen(a);
  b[0]=a[0];
  j=1;
  for (i=1;i<=l;i++)
    if (a[i]==a[i-1])
      {b[j]=' ';
       b[j+1]=a[i];
       j=j+2;}
    else
      {
       b[j]=a[i];
       j++;
      }
  printf("%s",b);
}
```

Пример 3. Удалить из строки символов, заканчивающейся точкой, повторяющиеся пробелы.

Вариант программы решения задачи на языке Pascal:

```

program lab4_3;
var
i,l:integer; a:string;
begin
writeln(' введите строку в конце точка');
readln(a);
l:=length(a);i:=1;
repeat
  if (a[i]=' ') and (a[i+1]=' ') then begin
    delete(a,i,1);
    i:=i-1 ;
  end;

  i:=i+1;
until i>l;
writeln(a);
readln;
end.

```

Вариант программы решения задачи на языке C++:

```

//удаление повторяющихся пробелов
#include <stdio.h>
void main()
{
  char z,s;
  printf("введите предложение с точкой в
конце:\n");
  for (z=s=' ';z!='. ';s=z)
  { //s- предыдущий символ
  scanf("%c",&z);
  if (z==' ' && s==' ')continue;
  printf("%c",z);
  }
}

```

Варианты заданий

Вариант 1

Определите, совпадают ли первая и последняя буквы слова.

Определите, является ли слово перевертышем.

Вычеркните из слова все буквы “а” и “б”, стоящие на нечетных местах.

Вариант 2

Определите, совпадают ли буквы на нечетных местах в слове.

После каждой буквы “а” в слове вставьте букву “к”.

Вычеркните из слова все сочетания “ку”.

Вариант 3

Определите, количество повторяющихся букв в слове.

После каждой буквы “о” в слове вставьте сочетание “ля”.

Вычеркните из слова все буквы “я”, стоящие на четных местах.

Вариант 4

Проверьте, все ли буквы в слове различные.

Вставьте между одинаковыми буквами в слове пробел.

Вычеркните из слова все буквы, встречающиеся более одного раза.

Вариант 5

Проверьте, совпадают ли последние буквы двух слов.

В слове после каждой буквы “и” вставьте “р”, если она встречается более одного раза.

Вычеркните из слова буквы, встречающиеся в другом слове.

Вариант 6

Проверьте, можно ли из букв слова X составить слово Y.

Вставьте перед каждой буквой “а” пробел, а после каждой буквы “б” букву “у”.

Вычеркните из слова все буквы, встречающиеся трижды.

Вариант 7

Посчитайте, сколько раз в слове встречается сочетание “ко”.

Вставьте в слове после каждой буквы, стоящей на нечетном месте, пробел.

Вычеркните из слова все буквы “ы”, стоящие после “ц”.

Вариант 8

Проверьте, является ли “кас” корнем слова без приставки.

Отделите в слове повторяющиеся более двух раз буквы запятыми.

Вычеркните из слова букву, предшествующую сочетанию “ча”.

Вариант 9

Проверьте, совпадают ли два слова.

Вставьте после буквы “т”, стоящей на нечетном месте, букву “у”.

Вычеркните из слова все буквы, встречающиеся в двух других словах.

Вариант 10

Проверьте, есть ли в слове приставка “при” или “пре”.

Вставьте в слове после “кас” букву “а”, если ее там нет.

Удалите из слова все буквы, совпадающие с последней буквой.

Вариант 11

Определите, есть ли в двух словах одинаковые символы.

Удвойте каждую букву слова.

Подсчитайте, сколько раз первая буква одного слова встречается в другом слове, и вычеркните их.

Вариант 12

Определите символ, чаще всего встречающийся в слове.

Напишите программу обращения слова.

Вычеркните из слова буквы, предшествующие “а” и следующие за “д”.

Вариант 13

Уясните, какая из букв слова, первая или последняя, встречается в слове чаще.

Вставьте в слове перед каждой буквой “к”, которой предшествует “с”, букву “о”.

Вычеркните из слова все буквы “ш” и “щ”, стоящие на четных местах.

Вариант 14

Определите, все ли буквы слова различны.

Замените в слове сочетание “му” на “а”, а букву “ы” на “ца”.

Вычеркните из одного слова все буквы, встречающиеся в другом слове.

Вариант 15

Определите, совпадает ли количество букв “а” в двух словах.

Вставьте после буквы “я” в слове букву “к” и перед ней букву “с”.

Вычеркните из слова все буквы, повторяющиеся трижды.

Вариант 16

Определите, сколько гласных букв в слове.

Между двумя одинаковыми буквами в слове вставьте “-”.

Вычеркните из слова сочетание “при”.

Вариант 17

Определите, можно ли в слове буквы переставить так, чтобы получилось слово - перевертыш.

Утройте каждую неповторяющуюся букву слова.

Вычеркните из слова все гласные буквы.

Вариант 18

Определите, совпадают ли первые и последние буквы двух слов.

Получите из одного слова другое, вставив последние две буквы в начало слова.

Вычеркните из слова согласные буквы.

Вариант 19

Определите, совпадают ли последние буквы трех слов.

После приставки “под” вставьте “ь”.

Вычеркните из трех слов одинаковые символы.

Вариант 20

Подсчитайте количество шипящих букв в слове.

Удалите из слова буквы, стоящие на четных местах.

Удвойте гласные в слове.

Вариант 21

Определите количество согласных в слове.

Замените буквы “а” и “о” в слове на “А” и “О”.

Вычеркните из слова неповторяющиеся буквы.

Вариант 22

Совпадают ли буквы в двух словах.

Вычеркните из двух слов совпадающие гласные буквы.

Вставьте между буквами в слове “ _ ”.

Вариант 23

Посчитайте количество букв “А” и “а”, стоящих на нечетных местах в словах *x* и *y*.

Удвойте все гласные буквы слова.

Вычеркните из слова все буквы “б”, за которыми следуют “а” или “о”.

Вариант 24

Посчитайте количество знаков препинания в предложении, заканчивающемся точкой.

Утройте каждую букву слова.

Вычеркните из слова все буквы, встречающиеся два раза.

Вариант 25

Посчитайте количество слов в предложении, считая, что слова разделены пробелами.

Удвойте первые и последние буквы слов *x* и *y*.

Вычеркните из слова все буквы, встречающиеся более одного раза.

Вариант 26

Посчитайте, сколько раз в слове встречается сочетание букв “*ту*”.

Вычеркните из слова все шипящие согласные.

Вставьте после каждого слова в предложении запятую, считая, что слова разделены пробелами, а в конце предложения стоит точка.

Вариант 27

Посчитайте количество различных букв в слове.

Вычеркните из слова все гласные буквы.

Запишите буквы слова в обратном порядке.

Вариант 28

Посчитайте количество гласных в слове.

Вычеркните из предложения все знаки препинания.

Удвойте все одиночные пробелы в предложении.

Вариант 29

Посчитайте количество букв, встречающихся в слове более одного раза.

В предложении разделите буквы в словах длиной более десяти символов пробелами.

Удалите из предложения все строчные латинские буквы.

Вариант 30

Упорядочите буквы слова по алфавиту.

Удалите из предложения все одинаковые слова, идущие друг за другом.

Удвойте буквы в слове, отличные от последней и первой.

Лабораторная работа 5. Программирование с использованием функций

Пример. Найдите значение $\frac{x^n x^m}{x^{n+m} + 1}$, используя функцию

вычисления степени числа, где m, n – целые числа.

Вариант программы решения задачи на языке

Pascal:

```
program lab5;
var
  f, x: real;
  i, n, m: integer;
function step(x: real; n: integer): real;
begin
  step:=exp(n*ln(x));
end;
begin
  writeln('x, n, m');
  readln(x, n, m);
  f:=step(x, n)*step(x, m)/step(x, n+m);
  writeln('f=', f);
  readln;
end.
```

Вариант программы решения задачи на языке C++:

```
#include<stdio.h>
#include <math.h>
float step(float x, int n)//определение функции
{
  return exp(n*log(x));
}
void main()
{int m, n;
float y, z, s, x;
printf("\n введите m=");
scanf("%d", &m);
printf("\n введите n=");
scanf("%d", &n);
printf("\n введите x=");
scanf("%f", &x);
y=step(x, n); z=step(x, m); s=step(x, m+n);
```

```
float w=y*z/(s+1);
printf("\n значение выражения = %f",w);
}
```

Варианты заданий

В заданиях на эту тему x, y, z - переменные, a, b, c - константы, вводимые с клавиатуры.

Используя функцию, выполните следующие задания:

1) вычислить $\frac{f(a) + g(b)}{f(a)f(b)}$,

где $f(x) = x^2 - 4x + 3$, $g(x) = x^4 - 5x - 4$;

2) вычислить

$$(s(a) + s(b))s(a + b), \text{ где } s(x) = \begin{cases} -1, x < 0 \\ x^4 - \cos x, x \in [0, 5]; \\ -1, x > 5 \end{cases}$$

3) вычислить

$$\operatorname{sh}(a) \operatorname{tg}(a + 1) - \operatorname{tg}^2(2 + \operatorname{sh}(a - 1))$$

4) вычислить $\frac{f(a^2 + b^2)}{f(a^2 - b^2)}$,

где $f(x) = \begin{cases} 0, x = 2m, m \in \mathbb{Z} \\ x^2 - \sqrt{x - 1}, x = 2m + 1, m \in \mathbb{Z} \end{cases}$;

5) найти значение $f(a^2 - b^2) + f(a)$, $f(t) = e^{\sin t + \cos t}$;

6) используя функцию для вычисления логарифма, найти значение выражения $(\log_a b)^x + (\log_b a)^{\frac{1}{x}}$;

7) составить программу для вычисления выражения $\frac{\log_2 a + \log_b c}{\log_{b+2}(a + c)}$;

8) найти $\frac{\operatorname{th} a - \operatorname{th}^2(a - b)}{\sqrt{\operatorname{th}(a^2 - b^2)}}$, где $\operatorname{th} x = \frac{e^x - e^{-x}}{e^x + e^{-x}}$;

9) $f(x, y) = \frac{x^2 + 4xy + y^2}{xy}$. Найти $\frac{f(a, b)}{f(a - b, a + b)}$;

10) задана функция $f(x, y, z) = x^3 + y^3 + z^3$

Найти значение $\frac{f(a, b, c)}{f(a-b, a+b, a+c)}$;

11) $f(x) = \sin x \cos^2 x$. Найти $f''(a)f'(b)f(a)f(b)$;

12) $f(x) = \sin^2 x - \cos x$. Найти $\frac{f(a)f'(b)}{f'(a-b)f(b)}$;

13) $f(x, y, z) = (x-y)^2 + (z-x)^2 + (z-x-y)^2$.

Найти $\frac{f(a, b, c)f(a+b, b+c, c+a)}{f(a-b, b-c, c-a)}$;

14) $f(x) = \sin x \cos x$, $\cos x - \sin x$. Найти

$f(a)g(a)f(a+b)g^2(a-b)$;

15) $f(x) = \sqrt{x^2 + 1}$, $g(x) = \sqrt[3]{x-1}$. Найти $\frac{f'(a)f'(b)}{g'(a+b)g'(a-b)}$;

16) даны отрезки a, b, c, d . Для каждой тройки этих отрезков, из которых можно построить треугольник, напечатать площадь треугольника;

17) из данных 10 комплексных чисел найти число с максимальным модулем;

18) найти производную функции $f(x) = x^3 - x - \cos x$ по определению, принимая $\Delta=0,01$, в точках a и b ;

19) найти производную функции $f(x) = \sin x \cos x$ по определению ($\Delta=0,001$) в точках из интервала $(0,5)$ с шагом 1;

20) найти производную функции $f(x) = \sin x(\cos x - x^3)$ по определению ($\Delta=0,1$) в точках из интервала $[0,5)$ с шагом 0,5;

21) найти вторую производную функции, зная $f(x) = x^4 - \cos^2 x$ по определению ($\Delta=0,1$) в точках c и d ;

22) данные 6 комплексных чисел упорядочить по возрастанию модулей;

23) найти площади треугольников, которые можно построить из четырех точек;

24) $f(x) = x^3 - \sin x \cos x - x$. Найти $f(a)f^2(a-b)$;

25) $g(x) = \frac{x^2}{\sqrt[3]{x-1}}$. Найти $g(a+b)g^3(b)$;

26) найти вторую производную функции $f(x) = \cos x - \sin^5 x$ по определению ($\Delta=0,1$) в точках a и c .

27) данные 8 комплексных чисел упорядочить по убыванию их модулей;

28) найти вторую производную функции, зная

$f'(x) = x^4 - x^3 - x$, по определению ($\Delta=0,01$) в точках интервала $(-2,3)$ с шагом 2;

29) найти значения функции $f(x) = x^3 \cos x - x \sin x$ в точках интервала $(-10,4)$ с шагом 3;

30) затабулировать функцию $f(x) = \frac{\sin x - \cos x}{x^3}$ на интервале $(-5, 5)$, шаг 1.

Лабораторная работа 6. Программирование с использованием процедур

Пример. Найти периметры и площади пяти прямоугольников по известным сторонам.

Вариант программы решения задачи на языке Pascal:

```
program lab6;
var s,p: array[1..5] of real;
{s,p массивы для площадей и периметров}
i,j:integer;
a:array[1..5,1..2] of real;
{матрица для сторон 5 прямоугольников}
procedure sp(x,y:real;var s,p:real);
begin
    s:=x*y;
    p:=(x+y)*2;
end;
begin
writeln('введите длину и ширину через
пробел');
for i:=1 to 5 do
```

```

begin
    writeln('длина    ширина');
    for j:=1 to 2 do
        read(a[i,j]);
    writeln;
end;
for i:=1 to 5 do
begin
sp(a[i,1],a[i,2],s[i],p[i]);
write('площадь ',i,' прямоугольника
',s[i]:8:3);
writeln('периметр      ',i,'      прямоугольника
',p[i]:8:3);
end;
readln;
end.

```

Вариант программы решения задачи на языке C++:

```

#include<stdio.h>
int pr(float a, float b,float *p, float*s)
{
if (a<0 || b<0)
return 0;
else
{
    *p=(a+b)*2;
    *s=a*b;
    return 1;
}
}
void main()
{
float a[5][2];
float s[5],p[5];
int i;
for (i=0;i<5;i++)
{printf("\n введите длину  %d  прямоугольника
",i+1);
scanf("%f",&a[i][0]);
printf("\n введите ширину  %d  прямоугольника
",i+1);
scanf("%f",&a[i][1]);
if(pr(a[i][0],a[i][1],&p[i],&s[i])==1)

```

```

    {printf("\n    периметр    прямоугольника    =
%f", p[i]);
    printf("\n    площадь    прямоугольника    =
%f", s[i]);
    }
    else printf("\n такого прямоугольника нет!");
}
}

```

Варианты заданий

1) Вычислите $z = \frac{x_1 + y_1}{x_2 y_2}$, где x_1, x_2 – корни уравнения

$x^2 - 4x + a = 0$, y_1, y_2 – корни уравнения $2y^2 + ay - a^2 = 0$
(мнимые корни обнулить).

2) Составьте программу для нахождения элементов целочисленных массивов $A(15)$, $B(20)$, $C(25)$, кратных трем.

3) Составьте программу для вычисления сумм элементов каждой строки матриц $A(5,6)$, $C(6,7)$.

4) Составьте программу для нахождения значения

$z = \frac{(a - x_1)(a - x_2)}{(b - y_1)(b - y_2)}$, где x_1, x_2 – корни уравнения

$ax^2 - x + 1 = 0$; y_1, y_2 – корни уравнения $2y^2 + y - b = 0$.
Мнимые корни считать равными нулю.

5) Составьте программу для нахождения корней квадратных уравнений: $ax^2 + bx - 4 = 0$, $y^2 - cy + d = 0$. Если корень комплексный, то считать его равным нулю.

6) Составьте программу для нахождения сумм положительных элементов строк каждой матрицы $A(n,m)$, $B(k,l)$, где n, m, k, l не превосходят 20. Результаты запомнить в массивах APL и BPL .

7) Составьте программу для вычисления средних значений положительных элементов каждой строки матриц $A(5,8)$, $B(4,8)$, $C(7,8)$ и записи полученных значений в 9-й столбец этих матриц.

8) Используя процедуру произведения двух матриц, найдите произведение трех матриц $A(3,4)$, $B(4,3)$, $C(3,3)$.

9) Даны два биквадратных уравнения $ax^4 + bx^2 + c = 0$ и $y^4 + dy^2 + g = 0$. Определите, есть ли у них совпадающие корни.

- 10) Используя подпрограмму вычисления транспонированной матрицы S^T , найдите для данных матриц A, B, C размерностью 3×3 значение выражения $(A + B)^T - (A + C)^T + (B + C)^T$.
- 11) Используя процедуру вычисления обратной матрицы размерностью 2×2 , найдите $D = (A^{-1} + B^{-1})^{-1}$, где A и B – матрицы второго порядка.
- 12) Даны три матрицы размерностью 3×4 . Найдите для каждой из них максимальный и минимальный элементы, используя процедуру. Выведите матрицы, у которой наибольший максимальный и наименьший минимальный элементы.
- 13) Найдите корни уравнений $ax^3 + bx^2 + cx = 0$ и $y^3 - ay^2 - b = 0$, используя процедуру. Есть ли у этих уравнений совпадающие корни?
- 14) Используя процедуру умножения двух комплексных чисел, найдите произведение четырех комплексных чисел.
- 15) Найдите значение выражения $z = \frac{ab}{c}$, где a, b, c – комплексные числа.
- 16) Упорядочьте данную матрицу по возрастанию элементов в каждой строке, каждом столбце. Выведите две полученные матрицы.
- 17) f и g – две таблично заданные монотонные функции. Используя подпрограмму, найдите промежутки возрастания и убывания этих функций. Проверьте, есть ли такие интервалы, где функции f и g возрастают (убывают) одновременно.
- 18) Даны 4 массива $A(8), B(10), C(7), D(11)$. Используя процедуру, упорядочьте их по убыванию и проверьте, совпадают ли у них минимальный или максимальный элементы.
- 19) Даны размеры 8 прямоугольников. Используя процедуру нахождения площади, периметра и радиуса описанной окружности, найдите эти величины для каждого из прямоугольников и проверьте, есть ли среди них квадраты.
- 20) Найдите все делители данных 10 чисел. Проверьте, у каких из них есть совпадающие делители.
- 21) Даны вещественные матрицы $A(4,3), B(3,4), C(4,5)$. Используя процедуру, которая в матрице обнуляет все дробные

отрицательные элементы, преобразуйте данные матрицы. В какой из полученных матриц больше нулевых элементов (сколько)?

22) Используя подпрограмму перестановки i и j столбцов матрицы, преобразуйте матрицы $A(4,5)$, $B(4,4)$, $C(5,4)$. Выведите полученные матрицы.

23) Используя подпрограмму вычисления корней уравнения $\sin x = a$ в интервале $[0, 2\pi]$, найдите корни уравнений $\sin bx = a^2$ и $\sin(b+y) = c$. Есть ли у этих уравнений совпадающие корни?

24) Используя подпрограмму вычисления корней уравнения $\cos x = a$ в интервале $[0, 2\pi]$, найдите корни уравнений $\cos bx = c$ и $\cos(b+y) = a$. Есть ли у этих уравнений совпадающие корни?

25) Используя подпрограмму вычисления корней уравнения $\operatorname{tg} x = a$ в интервале $(0, 2\pi)$, найдите корни уравнений $\operatorname{tg}(bx+a) = c$ и $\operatorname{tg}(b^2+y) = a$. Является ли 0 корнем этих уравнений? (Какого?)

26) Используя подпрограмму вычисления корней уравнения $\operatorname{ctg} x = a$ в интервале $(0, 2\pi)$, найдите корни уравнений $\operatorname{ctg}(bx-a) = c$ и $\operatorname{ctg}(b-y) = c^2$. Является ли 0 корнем этих уравнений? (Какого?)

27) Используя процедуру решения уравнения вида $|x - a| = b$, найдите корни уравнений $|x - a^2| = \sqrt{b}$ и $|y - b^2| = a$.

28) Используя процедуру решения неравенств вида $|x - a| \geq b$, найдите решения неравенств $|x - a^2| \geq b^2$ и $|y - b^{-1}| = a$.

29) Даны стороны трех треугольников. Найдите величины медиан, проведенных из всех вершин каждого из треугольников.

30) Даны 3 матрицы $A(4,3)$, $B(3,3)$, $C(5,3)$. Используя процедуру, упорядочьте все матрицы по возрастанию элементов по столбцам.

Лабораторная работа 7. Построение графиков функций

Пример. Построить графики функций $y = \frac{1}{x^2 + 1}$ и, $y = x - 1$

найти точки их пересечения.

Вариант программы решения задачи на языке Pascal:

```
uses graph; {подключаем модуль для работы с графикой}
const a=-1;b=3; {отрезок построения функции}
      m=80; {масштаб}
      number=300; {число точек для построения графика}
var
xm,x0,ym,y0,x,y,h,xt1,yt1,gt1:integer;
  xt,yt,gt,gx,e,k:real;
  grDriver, grMode, ErrCode: Integer;

begin
  grDriver := Detect;
  InitGraph(grDriver, grMode, 'egavga.bgi');
  ErrCode := GraphResult;
  if ErrCode = grOk then
begin
  setbkcolor(1); {установка синего фона}
  setcolor(14); {установка желтого цвета линий}
  xm:=getmaxx; {максимальное значение x}
  ym:=getmaxy; {максимальное значение y}
  x0:=round(xm/2); {вычисление начала координат}
  y0:=round(ym/2);
  line(0,y0,xm,y0); {ось Ox}
  line(xm,y0,xm-5,y0-5); {стрелки по оси x}
  line(xm,y0,xm-5,y0+5);
  {деления по оси x}
  for x:=0 to round(xm/m) do line(m*x,y0-3,m*x,y0+3);
  line(x0,0,x0,xm); {ось y}
  line(x0,0,x0-5,5); {стрелки по оси y}
  line(x0,0,x0+5,5);
```

```

    {деления оси y}
    for y:=0 to round(ym/m) do line(x0-3,m*y,
x0+3, m*y);
    {подписи системы координат}
    outtextxy(x0-10,10,'y');
    outtextxy(x0+10,y0+10,'0');
    outtextxy(xm-10,y0+10,'x');
    outtextxy(440,120,'y=x-1');
    outtextxy(500,200,'y=1/(x^2+1)');
    outtextxy(x0+m,y0+10,'1');
    outtextxy(x0-15,y0-m-10,'1');
    h:=round(m*(b-a)/number);{шаг}
    xt:=a;
    repeat
        yt:=1/(xt*xt+1);{текущее значение y для 1
графика}
        gt:=xt-1; {текущее значение y для 2
графика}
        gt1:=round(gt*m);
        yt1:=round(yt*m);
        xt1:=round(xt*m);
        putpixel(x0+xt1,y0-gt1,7);{точка 1
графика}
        putpixel(x0+xt1,y0-yt1,14); {точка 2
графика}
        xt:=xt+h/m;
        gx:=gx+h/m;

        {вычисление точки пересечения графиков}
        e:=xt*xt*xt-xt*xt+xt-2;
        if e<=0.01 then k:=xt;
    until xt>b;
    writeln('точка пересечения(',k:7:3,k-
1:5:2,')');
    readln;
    CloseGraph;
end
else
writeln('Graphics error:',
GraphErrorMsg(ErrCode));
end.

```

Вариант программы решения задачи на языке C++:

```
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
float a=-1, b=3,
      m=80,
      number=300;
int xm,x0,ym,y0,x,y,h,xt1,yt1,gt1;
float xt,yt,gt,gx,e,k;

int grDriver, grMode, ErrCode;

int main()
{
    grDriver=DETECT;
    initgraph(&grDriver, &grMode, "egavga.bgi");
    ErrCode=graphresult();
    if (ErrCode==grOk)
    {
        setbkcolor(1);
        setcolor(14);
        xm=getmaxx();
        ym=getmaxy();
        x0=xm/2;
        y0=ym/2;
        line(0,y0,xm,y0);
        line(xm,y0,xm-5,y0-5);
        line(xm,y0,xm-5,y0+5);
        for (x=0; x<=xm/m; x++) line(m*x,y0-
3,m*x,y0+3);
        line(x0,0,x0,xm);
        line(x0,0,x0-5,5);
        line(x0,0,x0+5,5);
        for (y=0; y<=ym/m; y++) line(x0-
3,m*y,x0+3,m*y);
        outtextxy(x0-10,10,"y");
        outtextxy(xm+10,y0+10,"0");
        outtextxy(xm-10,y0+10,"x");
        outtextxy(440,120,"y=x-1");
        outtextxy(500,200,"y=1/(x^2+1)");
        h=m*(b-a)/number;
        xt=a;
```

```

do
{
yt=-xt*xt;
gt=xt-1;
gt1=gt*m;
yt1=yt*m;
xt1=xt*m;
putpixel(x0+xt1,y0-gt1,7);
putpixel(x0+xt1,y0-yt1,14);
xt=xt+h/m;
gx=gx+h/m;
e=xt*xt*xt-xt*xt+xt-2;
if (e<=0.01) k=xt;
}
while (xt<=b);
printf("koren=%7.3f",k);
getch();
closegraph();
}
else
{
printf("Graphics error: %s",
grapherrormsg(ErrCode));
getch();
}
return 0;
}

```

Варианты заданий

Постройте графики функций и найдите точки их пересечения.

1. $y = x^2 - 5x + 6, y = x - 3$
2. $y = x^3 - 1, y = 3x - 3$
3. $y = x^4, y = x + 16$
4. $y = \sin x, y = \cos x$
5. $y = x^2 + 4, y = -2x^2 - 6x + 1$
6. $y = x^2 - 2, y = x - 1$
7. $y = x - 8, y = -x^3 + x$
16. $y = \sqrt{x - 2}, y = x - 4$
17. $y = \sin^2 x, y = 0,5 - \cos^2 x$
18. $y = e^x, y = 4x + 1$
19. $y = \ln x, y = \ln(x^2 - 4x)$
20. $y = \cos^4 x, y = 2\cos^2 x - 1$
21. $y = x^2 - 9, y = 5 - x^2$
22. $y = \sqrt{15 - x}, y = 6 - \sqrt{3 - x}$

8. $y = \operatorname{tg} x, y = 1$

9. $y = 4x, y = x^2 + 4$

10. $y = x^3 - 5, y = 2x - 3$

11. $y = \sqrt{x-1}, y = x - 10$

12. $y = \sqrt{2x-7}, y = x - 8$

13. $y = \cos x, y = x - 10$

14. $y = |4 - x|, y = 5x - 4$

15. $y = x^3 - x^2, y = x^2 - 27$

23. $y = x, y = x^2 - 25$

24. $y = 5 - x, y = \frac{6}{x}$

25. $y = 2 + x, y = \frac{2}{x-1}$

26. $y = x^4, y = 2x - 16$

27. $y = \sqrt[3]{x}, y = x - 7$

28. $y = x\sqrt{x-1}, y = x - 3$

29. $y = 1 - x, y = 2x|x-1|$

30. $y = \sqrt{x-1}, y = x^2 - 10$

Лабораторная работа 8. Обработка текстовых файлов

Пример. Дана матрица $C(4,4)$. Возведите в квадрат все ее элементы. Данные считывать из файла, результат вывести в файл.

Вариант программы решения задачи на языке Pascal:

1. Сначала напишем программу, которая записывает введенную с клавиатуры матрицу в файл "dat":

```

program form;
var
i, j: integer;
c: array[1..4, 1..4] of real;
f: text;
begin
assign(f, 'dat'); {переменная f связывается с файлом}
rewrite(f); {файл открывается для записи}
for i:=1 to 4 do
begin
for j:=1 to 4 do
begin
read(c[i, j]);
write(f, c[i, j]:6:2); {данные записываются в файл}
end;
end;
end;

```

```
writeln(f);
end;
  close(f);
  readln;
end.
```

2. Следующая программа считывает данные из файла 'dat', обрабатывает матрицу (возводит элементы в квадрат) и результат записывает в файл 'rez':

```
program rez;
var
i, j: integer;
c: array[1..4, 1..4] of real;
f, g: text;
begin
assign(f, 'dat'); {связывает f с файлом 'dat'}
assign(g, 'rez'); {связывает g с файлом 'rez'}
reset(f); {открывает файл для чтения}
while not Eof(f) do {пока не достигнут конец
файла}
for i:=1 to 4 do
for j:=1 to 4 do
begin
  Read(f, c[i, j]); {считывает данные из файла}
  end;
rewrite(g); {открывает файл 'rez' для записи}
for i:=1 to 4 do
begin
for j:=1 to 4 do
write(g, c[i, j]*c[i, j]:6:2); {записывает
данные в файл}
writeln(g);
end;
close(f); {закрывает файл}
close(g);
readln;
end.
```

Вариант программы решения задачи на языке C++:

1. Сначала напишем программу, которая записывает введенную с клавиатуры матрицу в файл "dat":

```
#include <stdio.h>
```

```

int main()
{ int i,j;
  int a[4][4];
  FILE *fp; //указатель на поток
  if ((fp=fopen("dat", "w"))==NULL)
  {perror("dat");
   return 1;
  }
  printf("введите элементы\n");
  for (i=0;i<4;i++)
  {for (j=0;j<4;j++)
  {
   scanf("%d",&a[i][j]);
   fprintf(fp,"%d ",a[i][j]);
  }
   fprintf(fp, "\n");
  }
  fclose(fp);
  return 0;
}

```

2. Следующая программа считывает данные из файла "dat", обрабатывает матрицу (возводит элементы в квадрат) и результат записывает в файл "rez"

```

#include <stdio.h>
int main()
{ int i,j;
  int a[4][4];
  FILE *fp,*fp1; //указатель на поток
  if ((fp=fopen("dat", "r"))==NULL)
  {perror("dat");
   return 1;
  }
  fp1=fopen("rez", "w");
  for (i=0;i<4;i++)
  {for (j=0;j<4;j++)
  {fscanf(fp,"%d",&a[i][j]);
   fprintf(fp1,"%d ",a[i][j]*a[i][j]);
  }
   fprintf(fp1, "\n");
  }
}

```



```
fclose(fp);  
fclose(fp1);  
return 0;  
}
```

Варианты заданий

В следующих заданиях данные считываются из файлов и результат выводится в виде файла:

1. Даны матрицы $A(5,6)$, $B(5,6)$. Найти матрицу $A + B$.
2. Даны матрицы $A(4,7)$, $B(4,7)$. Найти матрицу $A - B$.
3. Найти транспонированную матрицу для $X(5,6)$.
4. Дана матрица $A(6,4)$. Вывести минимальный и максимальный элементы каждой строки этой матрицы.
5. Дана матрица $A(6,5)$. Вывести все положительные элементы этой матрицы. Найти их сумму.
6. Дана матрица $A(5,5)$. Вывести все отрицательные элементы этой матрицы. Найти их произведение.
7. Дана матрица $A(4,8)$. Вывести все положительные элементы этой матрицы. Найти максимальный по модулю элемент.
8. Дана матрица $A(7,4)$. Вывести все отрицательные элементы этой матрицы. Найти минимальный по модулю элемент.
9. Дана матрица $A(5,6)$. Составить массив, содержащий минимальные элементы каждой строки.
10. Дана матрица $A(6,5)$. Составить массив, содержащий максимальные элементы каждого столбца.
11. В матрице $A(5,9)$ положительные элементы заменить на противоположные, а отрицательные обнулить. Вывести полученную матрицу.
12. В матрице $A(8,7)$ поменять 2-ю и 4-ю строки местами.
13. Дана матрица $B(8,5)$. Составить массив, содержащий произведения элементов каждой строки матрицы.
14. Дана матрица $B(5,8)$. Составить массив, содержащий суммы элементов каждой строки матрицы.
15. В матрице $A(9,4)$ положительные элементы заменить на обратные, а отрицательные возвести в квадрат. Вывести полученную матрицу.
16. Даны матрицы $A(5,4)$, $B(4,5)$. Вывести матрицу $A \times B$.

17. Упорядочить матрицу $A(3,4)$ по возрастанию элементов по строкам. Вывести полученную матрицу.
18. Упорядочить матрицу $B(3,5)$ по убыванию элементов по столбцам. Вывести полученную матрицу.
19. В матрице $A(6,6)$ четные элементы разделить на 4, а к нечетным прибавить 10. Вывести полученную матрицу.
20. Даны матрицы $A(4,4)$ и $B(4,4)$. Найти $C=A+B$ и транспонированную матрицу для C .
21. В матрице $A(6,7)$ положительные элементы заменить на единичные, а отрицательные обнулить. Вывести полученную матрицу.
22. В матрице $A(5,5)$ поменять 1-й и 5-й столбцы местами.
23. Дана матрица $B(7,5)$. Составить массив, содержащий произведения элементов каждого столбца матрицы.
24. Даны три матрицы размерностью 4×4 . Найти сумму матриц.
25. Вывести совпадающие элементы матриц $A(7,4)$ и $B(5,6)$.
26. Вывести различные элементы матриц $A(9,4)$ и $C(10,3)$.
27. Для данных матриц A, B, C найти $A-B-C$ и $C-A-B$.
28. Найти произведение трех матриц размерностью 3×3 .
29. В матрице размерностью 5×7 дробные элементы округлить, а целые заменить их модулями.
30. В матрице размерностью 6×5 , состоящей из элементов, являющихся квадратом какого-либо числа, извлечь квадратный корень, а остальные возвести в квадрат.

Лабораторная работа 9. Обработка записей

Пример. Имеется список студентов из 10 человек (фамилия, год рождения, пол). Выбрать из данного списка студентов мужского пола старше 20 лет.

Вариант программы решения задачи на языке Pascal:

```

program zap;
const n=10;
type student=record
    fio:string[20];
    god:1970..1990;
    pol:string[3];
end;

```

```

var gr:array[1..n] of student;
i:integer;
begin
{ввод данных}
for i:=1 to n do
  with gr[i] do
    begin
      write('фамилия ');readln(fio);
      write('год рождения ');readln(god);
      write('пол (муж /жен) ');readln(pol);
    end;
  writeln('список мужчин старше 20 лет:');
{выбор мужчин, родившихся до 1983 года}
  for i:=1 to n do
    with gr[i] do
      if (god<1983) and (pol='муж') then
        writeln(fio);
    end.

```

Вариант программы решения задачи на языке C++:

```

#include <stdio.h>
#include <conio.h>
#include <string.h>
const n=10;
struct student
{
  char fio[20];
  int god;
  char pol[3];
};
student gr[n];
int i;
int main()
{
  for (i=0; i<n; i++)
  {
    printf("фамилия "); scanf("%s",
&gr[i].fio);
    printf("год рождения "); scanf("%d",
&gr[i].god);
    printf("пол "); scanf("%s",
&gr[i].pol);

```

```

    }
    printf("Студенты старше 20 лет мужского
    пола:\n");
    for (i=0; i<n; i++)
        if ((gr[i].god<1983) &&
        !strcmp(gr[i].pol, "м")) printf("%s\n",
        gr[i].fio);
    getch();
    return 0;
}

```

Варианты заданий

Составить программу обработки сведений о студентах своей группы. Сведения представляются в виде таблицы, содержащей фамилии студентов и определенные характеристики.

| Характеристики | Обработка |
|--|---|
| 1. ФИО, дата рождения, пол | Выбор старшего студента |
| 2. ФИО, средний балл аттестата, оценки по 4 предметам последней сессии | Выбор имеющих средний балл по итогам сессии ниже среднего балла аттестата |
| 3. ФИО, оценки по 4 предметам последней сессии | Сдавшие сессию на «4» и «5» |
| 4. ФИО, оценки по 4 предметам последней сессии, размер стипендии | Получающие стипендию и имеющие тройки в сессии |
| 5. ФИО, средний балл аттестата, | Упорядочивание по среднему баллу аттестата |
| 6. ФИО, размер стипендии, место жительства | Получающие стипендию и проживающие дома |
| 7. ФИО, оценки по 4 предметам последней сессии, место жительства | Отличники, живущие в общежитии |
| 8. ФИО, дата рождения | Родившиеся в заданном году |
| 9. ФИО, средний балл аттестата, | Студенты с максимальным баллом аттестата |
| 10. ФИО, пол, оценки по 4 предметам последней сессии | Выбор студентов заданного пола, сдавших сессию на «4» и «5» |

Продолжение

| Характеристики | Обработка |
|---|---|
| 11. ФИО, дата рождения | Выбор родившихся в заданном месяце |
| 12. ФИО, пол, размер стипендии | Девушки, получающие повышенную стипендию |
| 13. ФИО, размер стипендии, место жительства | Выбор не получающих стипендию и проживающих в общежитии |
| 14. ФИО, оценки по 4 предметам последней сессии, размер стипендии | Выбор имеющих худшую успеваемость из получающих стипендию |
| 15. ФИО, пол, место жительства | Выбор студентов заданного пола, проживающих в общежитии |
| 16. ФИО, дата рождения | Упорядочивание по возрасту |
| 17. ФИО, оценки по 4 предметам последней сессии | Определение для всех среднего балла по результатам прошлой сессии |
| 18. ФИО, оценки по 4 предметам последней сессии | Выбор студента, имеющего максимальный средний балл по итогам сессии |
| 19. ФИО, дата рождения | Выбор родившихся в заданное время года |
| 20. ФИО, размер стипендии, место жительства | Выбор проживающих в общежитии и получающих стипендию большую, чем минимальная |
| 21. ФИО, размер стипендии, семейное положение | Выбор семейных студентов, не получающих стипендию |
| 22. ФИО, семейное положение, место жительства | Выбор семейных студентов, проживающих в общежитии |
| 23. ФИО, пол, место жительства | Выбор студентов заданного пола, проживающих в общежитии |

| Характеристики | Обработка |
|---|--|
| 24. ФИО, дата рождения, размер стипендии | Выбор младшего из студентов, получающих стипендию |
| 25. ФИО, дата рождения, пол | Выбор младшего из студентов заданного пола |
| 26. ФИО, пол, размер стипендии | Выбор студентов заданного пола, не получающих стипендию |
| 27. ФИО, оценки по 4 предметам последней сессии, семейное положение | Выбор семейных студентов, не сдавших сессию |
| 28. ФИО, дата рождения, семейное положение | Холостые студенты, родившиеся в заданном году |
| 29. ФИО, размер стипендии, место жительства | Получающие максимальную стипендию из проживающих вне общежития |
| 30. ФИО, средний балл аттестата | Упорядочивание по среднему баллу аттестата |

Лабораторная работа 10. Обработка файлов из записей

Пример. Имеется список студентов из 10 человек (фамилия, год рождения, пол). Выбрать из данного списка студентов мужского пола старше 20 лет.

Вариант программы решения задачи на языке Pascal:

1. Сформируем файл из записей и сохраним его под именем 'aaa' в текущем каталоге.

```

program fileform;
const n=10; {количество студентов}
type student=record
    fio:string[20];
    god:1970..1990;
    pol:string[3];
end;
ff=file of student;

```

```

var gr:array[1..n] of student;
i:integer;f:ff;
begin
{связываем файловую переменную с файлом «aaa»}
assign(f, 'aaa');
rewrite(f);{открываем файл для записи}
for i:=1 to n do {цикл ввода данных}
  with gr[i] do
    begin
      write('fio  ');readln(fio);
      write('god  ');readln(god);
      write('pol  ');readln(pol);
    end;
  for i:=1 to n do
    write(f,gr[i]);{записываем данные в файл}
    close(f);{закрываем файл}
end.

```

2. Следующая программа считывает данные из существующего файла «aaa», созданного предыдущей программой, выбирает студентов мужского пола старше 20 лет и заносит их в файл «ссс».

```

program filerez;
const n=10;{количество студентов}
type student=record
  fio:string[20];
  god:1970..1990;
  pol:string[3];
end;
ff=file of student;
var gr:array[1..n] of student;
i:integer;f:ff; g:text;
begin
writeln;
assign(f, 'aaa');{связывает файловую переменную
и файл}
reset(f);{открывает файл 'aaa' для чтения}
for i:=1 to n do
  read(f,gr[i]);{читает записи из файла}
assign(g, 'ссс');{связывает файловую переменную
и файл}

```

```

rewrite(g);{открывает файл «sss» для записи}
  writeln(g,' список:');
  for i:=1 to n do
  with gr[i] do
  begin
    if (god<1983) and (pol='m') then
writeln(g,fio);
    end;

    close(f);{закрывает файл f}
    close(g); {закрывает файл g}
end.

```

Вариант программы решения задачи на языке C++:

1. Сформируем файл из записей и сохраним его под именем 'dat' в текущем каталоге.

```

#include <stdio.h>
#include <conio.h>
#include <string.h>
const n=10;
struct student
{
  char fio[20];
  int god;
  char pol[3];
};
student gr[n];
FILE *OutFile;
int i;
int main()
{
  OutFile=fopen("dat", "wb");
  for (i=0; i<n; i++)
  {printf("Фамилия "); scanf("%s",
&gr[i].fio);
  printf("год рождения "); scanf("%d",
&gr[i].god);
  printf("пол "); scanf("%s", &gr[i].pol);
  }
  for (i=0; i<n; i++)

```



```

        fwrite(&gr[i], sizeof(student), 1,
OutFile);
        fclose(OutFile);
        getch();
        return 0;
}

```

2. Следующая программа считывает данные из существующего файла «dat», созданного предыдущей программой, выбирает студентов мужского пола старше 20 лет и заносит их в файл «rez».

```

#include <stdio.h>
#include <conio.h>
#include <string.h>
const n=10;
struct student
{
    char fio[20];
    int god;
    char pol[3];
};
student gr[n];
int i;
FILE *InFile, *OutFile;
int main()
{
    InFile=fopen("dat", "rb");
    OutFile=fopen("rez", "wt");
    for (i=0; i<n; i++)
        fread(&gr[i], sizeof(student), 1,
InFile);
    fprintf(OutFile, "Студенты старше 20 лет,
мужчины:\n");
    for (i=0; i<n; i++)
        if ((gr[i].god<1983) &&
!strcmp(gr[i].pol, "M")) fprintf(OutFile,
"%s\n", gr[i].fio);
    getch();
    return 0;
}

```

Варианты заданий

Выполните задания лабораторной работы 9, считывая данные из файла и записывая результат в файл.

Лабораторная работа 11. Перечисляемые и ограниченные типы. Оператор варианта

Пример. Для введенного символа определить, является он русской, латинской буквой или цифрой.

Вариант программы решения задачи на языке Pascal:

```
program lab11;
var ch:char;
begin
write('Введите символ ');readln(ch);
  case ch of
    'A'..'Z', 'a'..'z': writeln('латинская
буква');
    '0'..'9':          writeln('цифра');
    'а'..'я':          writeln('русская
буква');
  else
    writeln('Не буква, не цифра, не знак
препинания');
  end;
end.
```

Вариант программы решения задачи на языке C++:

В языке C++ нет ограниченных типов. Есть перечисляемые константы и переключатель, который используется не совсем так, как оператор Case в Паскале. Предложенный пример с его помощью решить нельзя, поэтому используем условный оператор

```
#include <stdio.h>
void main()
{ char c,s;
  printf("введите символ ");
  scanf("%c",&c);
  for (s='0';s<='9';s++)
    if (c==s) printf("\n %c-цифра",c);
```

```

        for (s='a';s<'z';s++)
            if (c==s) printf("\n %c-латинская
буква", c);
        for (s='a';s<'я';s++)
            if (c==s) printf("\n %c-русская
буква", c);
    }

```

Варианты заданий

При выполнении задания на языке C++, где возможно используйте перечисляемые константы и переключатель. На языке Pascal оператор Case.

1. По заданным четвертям определите знаки x и y .
2. Решите задачу 1 для пространства.
3. По заданному целому числу определите, является ли оно положительным, отрицательным или нулем.
4. По заданным четвертям определите знаки функции $\sin x$.
5. По заданным четвертям определите знаки функции $\cos x$.
6. По заданным четвертям определите знаки функции $\operatorname{tg} x$.
7. По заданным четвертям определите знаки функции $\operatorname{ctg} x$.
8. По заданному месяцу выведите время года (считайте январь - 1, февраль - 2, ..., декабрь - 12).
9. По заданному месяцу выведите квартал (считайте январь - 1, февраль - 2, ..., декабрь - 12).
10. Определите, чем является введенный символ: буквой, цифрой или оператором.
11. Определите, чем является введенная буква: латинской или русской, заглавной или строчной.
12. По заданному номеру выведите название дня недели и определите, является ли он выходным или рабочим днем.
13. Используя оператор выбора, определите функцию

$$f(x) = \begin{cases} 0, & x \in -100..0, \\ 1, & x \in 1..10, \\ -x, & x \in 11..100 \\ x, & x \in 101..1000 \end{cases}$$

Найдите значения $f(x)$ для заданного числа.

14. Используя оператор выбора, определите функцию

$$f(x) = \begin{cases} -1, & x \in -10..0, \\ 1, & x \in 1..10, \\ -x, & x \in 11..20, \\ x^2, & x \in 21..100. \end{cases}$$

Найдите значения $f(x)$ для заданных x .

15. По заданным коэффициентам квадратного уравнения $ax^2 + bx + c$ определите количество его корней, используя оператор выбора.

16. По заданному номеру выведите цвет, используемый в Паскале.

17. Определите функцию $f(x)$ на интервале $-20..200$, которая для положительного x вычисляет его корень квадратный, для отрицательного $x > -10$ вычисляет его квадрат, для остальных печатает $|x|$.

18. По заданному семестру выведите список предметов, по которым студенты сдают экзамены.

19. По заданному семестру выведите список предметов, по которым студенты сдают зачеты.

20. По заданному году рождения учащегося определите класс, в котором он может учиться.

21. По введенному классу, в котором учится ученик, определите год окончания средней школы.

22. По возрасту ребенка определите, какую группу детского сада он посещает.

23. По номеру городского троллейбуса определите конечные остановки его следования.

24. По номеру городского автобуса определите конечные остановки его следования.

25. Число из интервала от 0 до 10 запишите прописью.

26. Определите, является введенная буква гласной, согласной или латинской.

27. По номеру пары напечатайте время ее начала и окончания.

28. Напишите расписание занятий на какой-либо день недели, в зависимости от пары.

29. Опишите алгоритм перехода улицы пешеходом (действия на определенный свет светофора).
30. По номеру курса выведите год поступления студентов.

Лабораторная работа 12. Множественные типы¹

Пример. Дана непустая последовательность символов. Постройте и напечатайте два множества, элементами первого являются встречающиеся в последовательности буквы от *a* до *z*, второго – знаки препинания.

Вариант программы решения задачи на языке Pascal:

```
program mn;
var c,i,j: char;
lb:set of 'a'..'z';pr:set of char;
k:integer;
begin
  write('=>');
  lb:=[];pr:=[];
  repeat
  read(c);
  if c in ['a'..'z'] then lb:=lb+[c]
  else
  if c in [',','.',':',';',',','?','!'] then
pr:=pr+[c];
  until eoln;
  writeln('lat:');
  for i:='a' to 'z' do
  if i in lb then write(i);
  writeln;
  writeln('');
  for j:='!' to '?' do
  if j in pr then write(j);
  writeln;
  readln;
end.
```

Варианты заданий

1. Имеются 3 множества символьного типа:

¹ Данные типы определены только в языке Pascal

$$Y1 = \{ 'A', 'B', 'D', 'R', 'M' \};$$

$$Y2 = \{ 'A', 'H', 'D', 'R' \};$$

$$Y3 = \{ 'A', 'R' \};$$

Постройте множество $X = (Y1 \cap Y2) \cup (Y1 \setminus Y2)$. Выведите его на печать и проверьте, включено ли множество $Y3$ в множество X .

2. Даны 2 множества символьного типа:

$$X1 = \{ 'A', 'C', 'E', 'F', 'N' \};$$

$$Y2 = \{ 'N', 'A', 'H', 'E' \};$$

Постройте множество $X = (X1 \cup X2) \setminus (X1 \setminus X2)$ и выведите его на печать.

3. Дано множество целых чисел от 5 до 30. Выберите из него подмножества $X1$ – чисел, кратных 5, и $X2$ – чисел, кратных 3. Найдите $X = X1 / X2$.

4. Дано множество целых чисел от 10 до 100. Найдите подмножество A – целых чисел, кратных 5, B – чисел, кратных 6, C – чисел, кратных 10. Найдите $A \cup B \cup C$ и $A \cap B \cap C$.

5. Задано множество целых чисел от 30 до 150. Выделите подмножества чисел, кратных 8, 15 и 45. Найдите объединение и пересечение этих подмножеств.

6. Задано множество целых чисел от 15 до 99. Выделите подмножества чисел, кратных 2, 3, 6, 9. Найдите их объединение и пересечение.

7. Дано множество чисел A : [1..25] и B : [5..30]. Найдите подмножество A_1 – чисел из A , кратных 6, и B_1 – чисел из B , кратных 3. Найдите $A_1 \cup B_1$, $A_1 \cap B_1$, $A_1 \setminus B_1$ и проверьте, является ли $A_1 \subset B$ и $A_1 \subset B_1$.

8. Дана непустая последовательность символов. Постройте и напечатайте множество, элементами которого являются встречающиеся в последовательности цифры от 0 до 9.

9. Дана непустая последовательность символов. Постройте и напечатайте множество, элементами которого являются встречающиеся в последовательности буквы от A до F и от X до Z.

10. Дана непустая последовательность символов. Постройте и напечатайте множество, элементами которого являются

встречающиеся в последовательности буквы от G до N и цифры от 0 до 9.

11. Дана непустая последовательность символов. Постройте и напечатайте множество, элементами которого являются встречающиеся в последовательности знаки препинания.

12. Дана непустая последовательность символов. Постройте и напечатайте множество, элементами которого являются встречающиеся в последовательности буквы от A до Z и цифры от 0 до 5.

13. Дана непустая последовательность символов. Постройте и напечатайте множество, элементами которого являются встречающиеся в последовательности буквы от T до X и знаки препинания.

14. Дана непустая последовательность символов. Постройте и напечатайте множество, элементами которого являются встречающиеся в последовательности цифры от 5 до 9 и знаки арифметических операций.

15. Дана непустая последовательность символов. Постройте и напечатайте множество, элементами которого являются встречающиеся в последовательности знаки арифметических операций и знаки препинания.

16. Дана непустая последовательность символов. Постройте и напечатайте множество, элементами которого являются встречающиеся в последовательности цифры и знаки арифметических операций.

17. Дана непустая последовательность символов. Постройте и напечатайте множество, элементами которого являются встречающиеся в последовательности знаки препинания и буквы от E до N .

18. Дана непустая последовательность символов. Постройте и напечатайте множество, элементами которого являются встречающиеся в последовательности знаки операций отношения.

19. Дана непустая последовательность символов. Постройте и напечатайте множество, элементами которого являются встречающиеся в последовательности цифры от 3 до 9, буквы от A до F и знаки препинания.

20. Дана непустая последовательность символов. Постройте и напечатайте множество, элементами которого являются встречающиеся в последовательности знаки арифметических операций и операций отношения.
21. Дана непустая последовательность символов. Постройте и напечатайте множество, элементами которого являются встречающиеся в последовательности буквы от F до M и знаки арифметических операций.
22. Дана непустая последовательность символов. Постройте и напечатайте множество, элементами которого являются встречающиеся в последовательности знаки препинания и операций отношения.
23. Вычислите значение выражения:
 $[2..13] \times [3, 13..60] + [4..10] - [5..15] \times [6]$.
24. Вычислите значение выражения:
 $[2..10] - [4, 6] - [2..12] \times [8..15]$.
25. Вычислите значение выражения:
 $(['0'..'7'] + ['2'..'9']) \times (['a'] + ['z'])$.
26. Упростите (A и B – множества)
 $A \times B - A$.
27. Упростите (A и B – множества)
 $A - (A - B)$.
28. Упростите (A и B – множества)
 $(A + B) - (A - B) - (B - A)$.
29. Упростите (A и B – множества)
 $(A - B) + (B - A) + A \times B$.
30. Задано множество целых чисел от 5 до 95. Выделите подмножества чисел, кратных 12, 3, 9. Найдите их объединение и пересечение.

Лабораторная работа 13. Создание движущихся графических изображений

Пример. Написать программу движения ракеты, основанного на операциях поворота и масштабирования. Ракета совершает сначала один полный оборот по круговой траектории по часовой стрелке, а затем движется прямолинейно слева направо, уменьшаясь в размерах. За центр вращения принята центральная точка экрана, за центр масштабирования – точка, имеющая

максимальную абсциссу и ординату, отстоящую на расстояние радиуса орбиты от центра.

В программе приведенной ниже, приняты следующие обозначения: l -высота первой ступени ракеты, r -радиус орбиты, d -диаметр ракеты, ls -высота второй ступени ракеты x, y -массивы исходных координат вершин ракеты, $x1, y1$ – массивы преобразованных координат ракеты, xn, yn – координаты начальной точки (правая верхняя вершина ракеты), xc, yc – координаты центра орбиты, xt, yt – координаты центра масштабирования, xa, ya – коэффициенты, учитывающие разные разрешающие способности вдоль координатных осей, t – текущее значение угла поворота, k – текущее значение коэффициента масштабирования.

Вариант программы решения задачи на языке

Pascal:

```
program raketa;
uses graph;crt;
const l=60;r=120;d=20;ls=15;
type mas=array[1..15] of integer;
var x,y,x1,y1:mas;
gd,gm,xn,yn,xc,yc,xm,ym,i,j:integer;
xa,ya:word;
t,k:real;  ErrCode:integer;
{процедура определения координат повернутой точки}
procedure pow(xc,yc,x,y:integer;var
x1,y1:integer;t:real);
begin
x1:=xc+round((x-xc)*cos(t)+(y-yc)*sin(t)*ya/xa);
y1:=yc+round((y-yc)*cos(t)-(x-xc)*sin(t)*xa/ya);
end;
{процедура определения координат
промасштабированной точки}
procedure massch(xc,yc,x,y:integer;k:real;var
x1,y1:integer);
var r:real;
begin
r:=1-k;
x1:=round(x*k+r*xc);
y1:=round(y*k+r+yc);
end;

{процедура высвечивания ракеты}
procedure raket(x,y:mas);
var i:integer;
```

```

begin
line(x[1],y[1],x[2],y[2]);
line(x[2],y[2],x[3],y[3]);
line(x[3],y[3],x[4],y[4]);
line(x[4],y[4],x[1],y[1]);
line(x[4],y[4],x[5],y[5]);
line(x[1],y[1],x[5],y[5]);
line(x[5],y[5],x[4],y[4]);
for i:=1 to 2 do
begin
line(x[6+(i-1)*5],y[6+(i-1)*5],x[7+(i-1)*5],y[7+(i-1)*5]);
line(x[7+(i-1)*5],y[7+(i-1)*5],x[8+(i-1)*5],y[8+(i-1)*5]);
line(x[8+(i-1)*5],y[8+(i-1)*5],x[9+(i-1)*5],y[9+(i-1)*5]);
line(x[9+(i-1)*5],y[9+(i-1)*5],x[10+(i-1)*5],y[10+(i-1)*5]);
end;
end;

begin
gd:=detect;
  InitGraph(gd, gm,'egavga.bgi');
  ErrCode := GraphResult;
  if ErrCode = grOk then
begin setcolor(12);
{определение поправочных коэффициентов,
учитывающих}
{разные разрешающие способности вдоль осей X и Y}
GetAspectRatio(xa,ya);
xn:=getmaxx div 2;
yn:=getmaxy div 2-r;
xm:=getmaxx; {координаты центра}
ym:=yn;      {масштабирования}
xc:=xn;      {координаты центра}
yc:=yn+r;    {вращения}
{задание начальных координат точек ракеты}
x[1]:=xn;y[1]:=yn;
x[2]:=xn-1;y[2]:=yn;
x[3]:=xn-1;y[3]:=yn+d;
x[4]:=xn;y[4]:=yn+d;
x[5]:=xn+1s;y[5]:=yn+d div 2;
x[6]:=xn-round(3*1/4);y[6]:=yn;
x[7]:=xn-1;y[7]:=yn-round(d/4);
x[8]:=xn-1-1 div 2;y[8]:=yn-round(d/4);
x[9]:=xn-round(5*1/4);y[9]:=yn+round(d/8);

```

```

x[10]:=xn-1;y[10]:=yn+round(d/8);
x[11]:=xn-round(3*l/4);y[11]:=yn+d;
x[12]:=xn-1;y[12]:=yn+d+round(d/4);
x[13]:=xn-1-1 div 2;y[13]:=yn+round(5*d/4);
x[14]:=xn-1-round(l/4);y[14]:=yn+round(7*d/8);
x[15]:=xn-1;y[15]:=yn+round(7*d/8);
{движение ракеты по круговой траектории вокруг}
{центра вращения по часовой стрелке один оборот}
for i:=1 to 37 do
begin
t:=(i-1)*10*pi/180;{угол поворота}
for j:=1 to 15 do {10 градусов}
pow(xc,yc,x[j],y[j],x1[j],y1[j],-t);
raket(x1,y1);{рисование ракеты}
delay(100);{задержка перед стиранием}
cleardevice;{стирание изображения}
end;
raket(x1,y1);
{движение ракеты по прямой с уменьшением размеров}
for i:=1 to 20 do
begin
k:=1-0.05*i;{коэффициент масштабирования}
for j:=1 to 15 do
massch(xm,ym,x[j],y[j],k,x1[j],y1[j]);
raket(x1,y1);{рисование промасштабированной ракеты}
delay(100);{задержка перед стиранием}
cleardevice;{стирание изображения}
end;
closegraph;
end else writeln('error');
end.

```

Вариант программы решения задачи на языке C++:

```

#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <math.h>
#include <graphics.h>
float l=60, r=120, d=20, ls=15;
typedef int mas[16];

mas x,y,x1,y1;
int gd,gm,xn,yn,xc,yc,xm,ym,i,j;
int xa,ya;

float t,k;
int ErrCode;

```

```

void powerot(int xc, int yc, int x, int y, int &x1,
int &y1, float t)
{
    x1=xc+(x-xc)*cos(t)+(y-yc)*sin(t)*ya/xa;
    y1=yc+(y-yc)*cos(t)-(x-xc)*sin(t)*xa/ya;
}
void massch(int xc, int yc, int x, int y, float k,
int &x1, int &y1)
{
    float r;
    r=1-k;
    x1=x*k+r*xc;
    y1=y*k+r+yc;
}
void raket(mas x, mas y)
{
    int i;
    line(x[1],y[1],x[2],y[2]);
    line(x[2],y[2],x[3],y[3]);
    line(x[3],y[3],x[4],y[4]);
    line(x[4],y[4],x[1],y[1]);
    line(x[4],y[4],x[5],y[5]);
    line(x[1],y[1],x[5],y[5]);
    line(x[5],y[5],x[4],y[4]);
    for (i=1; i<=2; i++)
    {
        line(x[6+(i-1)*5],y[6+(i-1)*5],x[7+(i-1)*5],y[7+(i-1)*5]);
        line(x[7+(i-1)*5],y[7+(i-1)*5],x[8+(i-1)*5],y[8+(i-1)*5]);
        line(x[8+(i-1)*5],y[8+(i-1)*5],x[9+(i-1)*5],y[9+(i-1)*5]);
        line(x[9+(i-1)*5],y[9+(i-1)*5],x[10+(i-1)*5],y[10+(i-1)*5]);
    }
}
int main()
{
    gd=DETECT;
    initgraph(&gd, &gm,"egavga.bgi");
    ErrCode=graphresult();
    if (ErrCode==grOk)
    {
        setcolor(12);
        getaspectratio(&xa,&ya);
        xn=getmaxx()/2;
        yn=getmaxy()/2-r;
    }
}

```

```

xm=getmaxx();
ym=yn;
xc=xn;
yc=yn+r;
x[1]=xn;          y[1]=yn;
x[2]=xn-1;       y[2]=yn;
x[3]=xn-1;       y[3]=yn+d;
x[4]=xn;         y[4]=yn+d;
x[5]=xn+1s;      y[5]=yn+d/2;
x[6]=xn-3*1/4;   y[6]=yn;
x[7]=xn-1;       y[7]=yn-d/4;
x[8]=xn-1-1/2;   y[8]=yn-d/4;
x[9]=xn-5*1/4;   y[9]=yn+d/8;
x[10]=xn-1;      y[10]=yn+d/8;
x[11]=xn-3*1/4;  y[11]=yn+d;
x[12]=xn-1;      y[12]=yn+d+d/4;
x[13]=xn-1-1/2;  y[13]=yn+5*d/4;
x[14]=xn-1-1/4;  y[14]=yn+7*d/8;
x[15]=xn-1;      y[15]=yn+7*d/8;
for (i=1; i<=37; i++)
{
t=(i-1)*10*M_PI/180;
for (j=1; j<=15; j++)
    poworot(xc,yc,x[j],y[j],x1[j],y1[j],-t);
raket(x1,y1);
getch();
delay(100);
cleardevice();
}
raket(x1,y1);
for (i=1; i<=20; i++)
{
k=1-0.05*i;
for (j=1; j<=15; j++)
    massch(xm,ym,x[j],y[j],k,x1[j],y[j]);
raket(x1,y1);
getch();
delay(100);
cleardevice();
}
closegraph();
}
else printf("error");
getch();
return 0;
}

```

Варианты заданий

Написать следующие программы:

1. Движение машины по круговой траектории по часовой стрелке.
2. Движение машины по эллипсоидной траектории по часовой стрелке.
3. Движение машины слева направо по прямой дороге.
4. Движение машины справа налево вдоль прямой.
5. Движение машины сверху вниз вдоль прямой.
6. Движение машины снизу вверх вдоль прямой.
7. Движение машины вдоль синусоиды.
8. Движение машины по прямоугольной траектории.
9. Движение самолета по прямой траектории слева направо.
10. Движение самолета по прямой траектории справа налево.
11. Движение самолета по прямой траектории сверху вниз.
12. Движение самолета по прямой траектории снизу вверх.
13. Движение самолета по круговой траектории по часовой стрелке.
14. Движение самолета по круговой траектории против часовой стрелки.
15. Движение самолета по диагонали экрана снизу вверх.
16. Движение самолета по диагонали экрана сверху вниз.
17. Движение самолета, управляемого стрелками вправо и влево.
18. Движение самолета, управляемого стрелками вверх и вниз.
19. Движение автобуса, управляемого стрелками вправо и влево.
20. Движение автобуса, управляемого стрелками вверх и вниз.
21. Движение прыгающего на высоту h мяча, высота задается случайным образом.
22. Движение удаляющегося самолета (снизу вверх по центру экрана).

23. Движение приближающегося самолета (сверху вниз по центру экрана).
24. Движение приближающегося автобуса.
25. Движение удаляющегося по диагонали (из правого нижнего в левый верхний угол) автобуса.
26. Движение удаляющегося по диагонали (из левого нижнего в правый верхний угол) автобуса.
27. Движение падающей снежинки.
28. Написать программу Движение облака по небу.
29. Написать программу изображения циферблата часов с двигающейся минутной стрелкой.
30. Написать программу Движение качающихся качелей.

Лабораторная работа 14. Списки

Пример . Написать программу сортировки однонаправленного списка записей со следующей структурой: фамилия, год рождения, номер группы, время забега. Сортировка должна проводиться по фамилии (по алфавиту). Данные вводятся с клавиатуры, количество элементов списка не ограничено и задается в процессе работы программы.

Процедурой `formspisok` формируется однонаправленный список записей, процедурой `rechspisok` выводится исходный список. Сортировка осуществляется процедурой `sort` (сортировка методом «пузырька», но в отличие от сортировки массива здесь изменяются связи между элементами списка, если два соседних элемента связаны не в правильной последовательности), результат выводится процедурой `rechspisok`.

Вариант программы решения задачи на языке Pascal:

```
program spisok;  
type  
ptr=^student;  
student=record
```

```

    fam:string;
    year:integer;
    groop:string[9];
    time:real;
    next:ptr;
end;
var headptr:ptr;

procedure vvod(var stud:student);
{чтение значений полей записи}
begin
with stud do
    begin
    write('введите фамилию ');readln(fam);
    write('год рождения ');readln(year);
    write('номер группы ');readln(groop);
    write('время забега ');readln(time);
    end;
end;{vvod}

procedure formspisok;
{построение списка}
var
p:ptr;
let:char;
begin headptr:=nil;
    repeat
write('Продолжить? (y-да, n-нет) ');readln(let);
    if let='n' then exit;
    if headptr=nil then
        {формирование первого элемента списка}
        begin new(headptr);
            p:=headptr;
        end
    else
        {формирование очередного элемента списка}
        begin
            new(p^.next);
            p:=p^.next;
        end;
    vvod(p^);
    p^.next:=nil;
until let='n';
end;{formspisok}

procedure pech(stud:student);
{печать значений полей записи}

```



```

begin
with stud do
writeln (fam:14,year:6,groop:11,time:9:2);
end;{pech}

procedure pechspisok;
{печать списка}
var p:ptr;
begin
p:=headptr;
while p<>nil do
begin
pech(p^);
p:=p^.next;
end;
end;{pechspisok}

procedure sort;
{сортировка списка}
var p1,p2,q1,q2,temp{вспомог. указатель}:ptr;
begin
if headptr=nil then exit {если списка нет}
else if headptr^.next=nil then exit
{если список из одного элемента}
else
begin
p1:=headptr;{поиск с начала}
p2:=nil;
while p1^.next<>nil do
begin
q1:=p1^.next;{адрес очередного эл.}
q2:=p1;
while q1<>nil do {пока не конец}
begin
if p1^.fam>q1^.fam then
begin
if p1=q2
then
{выбираем следующий элемент списка}
begin
p1^.next:=q1^.next;
q1^.next:=p1;
p1:=q1;
q1:=q2;
q2:=q1;
end
else{изменяем связи в списке}

```

```

begin
temp:=p1^.next;
p1^.next:=q1^.next;
q1^.next:=temp;
temp:=p1;
p1:=q1;
q1:=temp;
q2^.next:=q1;
end;
if p2=nil then headptr:=p1
else p2^.next:=p1;
end;
q2:=q1;
q1:=q1^.next;
end;
p2:=p1;
p1:=p1^.next;
end;
end;
end;{sort}

begin
formspisok;
writeln('Исходный список:');
pechspisok;
sort;
writeln;
writeln('Список после сортировки:');
pechspisok;
end.

```

Вариант программы решения задачи на языке C++:

```

#include <stdio.h>
#include <conio.h>
#include <string.h>

struct student {
    char fam[30];
    int year;
    char groop[9];
    int time;
    student* next;
};

typedef student* ptr;
ptr headptr;

```

```

void vvod(student &stud)
{
    printf(" "); scanf("%s", &stud.fam);
    printf(" "); scanf("%d", &stud.year);
    printf(" "); scanf("%s", &stud.groop);
    printf(" "); scanf("%d", &stud.time);
}

int formspisok()
{
    ptr p;
    char let;
    headptr=NULL;
    do
    {
        printf("y-İa@ª@<|ËBM, n-Ş €@-зËBM\n");
        let=getch();
        if (let=='n') return 1;
        if (headptr==NULL)
        { headptr=new student;
          p=headptr;
        }
        else
        {
            p->next=new student;
            p=p->next;
        };
        vvod(*p);
        p->next=NULL;
    } while (1);
}

void pech(student stud)
{
    printf("%14s %6d %11s %9d\n", stud.fam, stud.year,
    stud.groop, stud.time);
}

void pechspisok()
{
    ptr p;
    p=headptr;
    while (p!=NULL)
    {
        pech(*p);
        p=p->next;
    }
}

```

```

}

int sort()
{
ptr p1,p2,q1,q2,temp;
if (headptr==NULL) return 1;

if (headptr->next==NULL) return 1;

p1=headptr;
p2=NULL;
    while (p1->next!=NULL)
    {
        q1=p1->next;
        q2=p1;
        while (q1!=NULL)
        {
if (strcmp(p1->fam, q1->fam)>0)
        {
            if (p1==q2)
            {
                p1->next=q1->next;
                q1->next=p1;
                p1=q1;
                q1=q2;
                q2=q1;
            }
            else
            {
                temp=p1->next;
                p1->next=q1->next;
                q1->next=temp;
                temp=p1;
                p1=q1;
                q1=temp;
                q2->next=q1;
            }
            if (p2==NULL) headptr=p1;
            else p2->next=p1;
        }
        q2=q1;
        q1=q1->next;
    };
    p2=p1;
    p1=p1->next;
};

return 0;

```

```

}

int main()
{
    clrscr();
    formspisok();
    printf("spisok\n");
    pechspisok();
    sort();
    printf("\nsort\n");
    pechspisok();
    getch();
    return 0;
}

```

Варианты заданий

Написать программу сортировки однонаправленного списка записей со следующей структурой:

| Структура | Сортировка |
|--|---|
| 1. ФИО, год рождения, пол | По фамилии (по алфавиту) |
| 2. ФИО, группа, средний балл аттестата, | По среднему баллу аттестата в порядке возрастания |
| 3. ФИО, средний балл аттестата, место жительства | По среднему баллу аттестата в порядке убывания |
| 4. ФИО, оценки по 4 предметам последней сессии, размер стипендии | По размеру стипендии в порядке возрастания |
| 5. ФИО, средний балл аттестата, номер группы | По номерам групп |
| 6. ФИО, размер стипендии, место жительства | По месту жительства |
| 7. ФИО, место жительства, место учебы | По месту учебы |
| 8. ФИО, дата рождения, место рождения | По месту рождения |
| 9. ФИО, средний балл аттестата, адрес | По адресу |
| 10. ФИО, пол, факультет | По факультетам |
| 11. ФИО, дата рождения, год поступления | По году поступления |
| 12. ФИО, пол, размер стипендии | По полу (сначала девушек, затем юношей) |

| Структура | Сортировка |
|--|---|
| 13. ФИО, размер стипендии, место жительства | По размеру стипендии в порядке убывания |
| 14. ФИО, год рождения, размер стипендии | По фамилии (от я до а) |
| 15. ФИО, адрес, телефон | По номеру телефона |
| 16. ФИО, дата рождения | По возрасту |
| 17. Автор, название книги, год издания | По фамилии автора |
| 18. Автор, название книги, год издания | по названию книги |
| 19. Автор, название книги, год издания | по году издания |
| 20. Автор, название книги, издательство | По издательству |
| 21. Автор, название книги, количество страниц | По количеству страниц |
| 22. Автор, название книги, жанр | По жанру |
| 23. ФИО, место жительства, марка автомобиля | По фамилии |
| 24. ФИО, адрес, марка автомобиля | По адресу |
| 25. ФИО, место жительства, марка автомобиля | По марке автомобиля |
| 26. ФИО, место жительства, марка автомобиля, год выпуска | По году выпуска |
| 27. ФИО, место жительства, марка автомобиля, производитель | По производителю |
| 28. ФИО, дата рождения, семейное положение | По семейному положению |
| 29. ФИО, семейное положение, количество детей | По количеству детей |
| 30. ФИО, место рождения, год рождения | По месту рождения |

Справочник

Язык Паскаль

Простые типы языка Паскаль

| Название | | Идентификатор | Множество допустимых значений |
|--------------|------------------------|---------------|---|
| Порядковый | Короткий целый | Shortint | -128..127 |
| | Байтовый | Byte | 0..255 |
| | Слово | Word | 0..65535 |
| | Целый | Integer | -32768..32767 |
| | Длинный целый | Longint | -2147483648..2147483647 |
| | Символьный | Char | Символы из расширенного набора символов кода ASCII |
| | Булевый | Boolean | True, False |
| Вещественный | Вещественный | Real | $-1,7 \cdot 10^{38} \dots 2,9 \cdot 10^{-39}$, $2,9 \cdot 10^{-39} \dots 1,7 \cdot 10^{38}$ |
| | С одинарной точностью | Single | $-3,4 \cdot 10^{38} \dots 1,5 \cdot 10^{-45}$, $1,5 \cdot 10^{-45} \dots 3,4 \cdot 10^{38}$ |
| | С двойной точностью | Double | $-1,7 \cdot 10^{308} \dots 5,0 \cdot 10^{-324}$, $5,0 \cdot 10^{-324} \dots 1,7 \cdot 10^{308}$ |
| | С повышенной точностью | Extended | $-1,1 \cdot 10^{4932} \dots 1,9 \cdot 10^{-4951}$, $1,9 \cdot 10^{-4951} \dots 1,1 \cdot 10^{4932}$ |
| | Сложный | Comp | $-2^{63} + 1 \dots 2^{63} - 1$ |

Базовые операции

| | Название | Тип операндов | Тип результата | Вид |
|---------------------------|------------------|------------------------|------------------------|------------------|
| Первый (высший) приоритет | | | | |
| @ | Взятие адреса | Любой | Pointer | Унарные операции |
| - | Унарный минус | Целый или вещественный | Целый или вещественный | |
| + | Унарный плюс | Целый или вещественный | Целый или вещественный | |
| not | Логическое «не» | Логический | Логический | |
| not | Поразрядное «не» | Целый | Целый | |

Продолжение

| | Название | Тип операндов | Тип результата | Вид |
|------------------|---------------------------|---|--|-------------------|
| Второй приоритет | | | | |
| * | Операция умножения | Целый * Целый Целый*Вещественный Вещественный*Целый Вещественный*Вещественный | Целый Вещественный Вещественный Вещественный | |
| / | Операция деления | Целый/Целый Целый/Вещественный Вещественный/Целый Вещественный/Вещественный | Вещественный Вещественный Вещественный Вещественный | |
| div | Целочисленное деление | Целый div целый | Целый | |
| mod | Остаток от деления нацело | Целый mod целый | Целый | |
| and | Логическое «и» | Логический | Логический | |
| and | Поразрядное «и» | Целый | Целый | |
| shl | Циклический сдвиг влево | Целый | Целый | |
| shr | Циклический сдвиг вправо | Целый | Целый | |
| Третий приоритет | | | | |
| + | Операция сложения | Целый +Целый Целый +Вещественный Вещественный+ Целый Вещественный+Вещественный | Целый Вещественный Вещественный Вещественный | Бинарные операции |
| - | Операция вычитания | Целый-Целый Целый-Вещественный Вещественный-Целый Вещественный-Вещественный | Целый Вещественный Вещественный Вещественный | |
| or | Логическое «или» | Логический | Логический | |
| or | Поразрядное «или» | Целый | Целый | |

Окончание

| | Название | Тип операндов | Тип результата | Вид |
|-------------------------------------|-------------------------------------|--|----------------|----------------------|
| xor | Логическое исключающее «или» | Логический | Логический | |
| xor | Поразрядное исключающее «или» | Целый | Целый | |
| Четвертый (низший) приоритет | | | | |
| = <> < > <= >= | Операции отношения | Число и число Строка и число Строка и литеры Pointer и pointer Множества | Логический | Бинарные операции |
| in | Вхождение в множество | Элементарный и множество | Логический | |

Функции над символами

| Название | Описание |
|----------------------|---|
| Pred (c: char):char | Нахождение значения предшествующего <i>c</i> элемента |
| Succ (c: char): char | Нахождение значения последующего элемента |
| Ord (c:char):byte | Возвращает код символа <i>c</i> |
| Chr (x:byte):char | Возвращает символ по коду <i>x</i> |

Математические функции и процедуры

| | Вызов | Тип аргумента | Тип значения | Назначение |
|---------|------------|---------------------------|-----------------|---|
| ФУНКЦИИ | abs (x) | Целый или вещественный | Как у аргумента | Абсолютное значение (модуль) <i>x</i> |
| | pi | - | Вещественный | Значение числа π |
| | sin (x) | Вещественный | Вещественный | Синус <i>x</i> радиан |
| | cos (x) | Вещественный | Вещественный | Косинус <i>x</i> радиан |
| | arctan (x) | Вещественный | Вещественный | Арктангенс <i>x</i> радиан |

| | | | | |
|-----------|---|------------------------|--|---|
| | Вызов | Тип аргумента | Тип значения | Назначение |
| | sqrt (<i>x</i>) | Целый или вещественный | Как у аргумента | Корень квадратный из <i>x</i> , $x > 0$ |
| | sqg (<i>x</i>) | Целый или вещественный | Как у аргумента | x^2 |
| | exp (<i>x</i>) | Вещественный | Вещественный | e^x |
| | ln (<i>x</i>) | Вещественный | Вещественный | Натуральный логарифм <i>x</i> , $x > 0$ |
| | trunc(<i>x</i>) | | | Целая часть <i>x</i> |
| | frac (<i>x</i>) | Вещественный | Вещественный | Дробная часть <i>x</i> |
| | int (<i>x</i>) | Вещественный | Вещественный | Целая часть числа <i>x</i> |
| | round (<i>x</i>) | Вещественный | longint | Округление числа |
| | Random(<i>x</i>) | Word | Word | Случайное число 0.. <i>x</i> |
| | Odd(<i>x</i>) | Целый | Логический | Возвращает True, если <i>x</i> нечетное |
| ПРОЦЕДУРЫ | Описание | | Назначение | |
| | Randomize | | Гарантирует несовпадение последовательностей случайных чисел, вызываемых функцией random | |
| | Inc(var <i>x</i> : целое) | | Увеличивает значение <i>x</i> на 1 | |
| | Dec(var <i>x</i> : целое) | | Уменьшает значение <i>x</i> на 1 | |
| | Inc(var <i>x</i> : целое; <i>n</i> : целое) | | Увеличивает значение <i>x</i> на <i>n</i> | |
| | Dec(var <i>x</i> : целое; <i>n</i> : целое) | | Уменьшает значение <i>x</i> на <i>n</i> | |

Функции и процедуры для обработки строк

| Название | Назначение |
|--|---|
| РЕДАКТИРОВАНИЕ СТРОК | |
| Length (<i>s</i> :string) :byte | Длина строки <i>s</i> |
| Concat (<i>s</i> ₁ , <i>s</i> ₂ , ..., <i>s</i> _{<i>n</i>}) :string | Конкатенация(слияние) строк <i>s</i> ₁ , <i>s</i> ₂ , ..., <i>s</i> _{<i>n</i>} |

| Название | Назначение |
|--|--|
| Copy(s:string; start, len:integer) | Возвращает подстроку длиной len начиная с позиции start строки s |
| Delete(var s: string; start, len:integer) | Удаляет из строки s начиная с символа start len символов |
| Insert(var s: string; sub: string; start: integer) | Вставляет в строку s подстроку sub начиная с позиции start |
| Pos(sub, s: string):byte | Возвращает номер первого символа вхождения строки sub в строку s или 0, если s не содержит sub |
| ПРОЦЕДУРЫ ПРЕОБРАЗОВАНИЯ | |
| Str(x; var s: string) | Преобразование числового значения x в строковое s. Возможно задание формата для x |
| Val(s: string; var x; var errcode:integer) | Преобразование строкового значения s (строки цифр) в число x |

Процедуры и функции для работы с файлами

| Функция или процедура | Описание |
|--|---|
| procedure Append (var f: text) | Открывает существующий файл, связанный с файловой переменной f, для добавления в него новых записей |
| procedure Assign (var f: String) | Связывает внешний файл, имя которого указано в строковой константе String, с файловой переменной f |
| procedure BlockRead (var F: File; var Buf; Count: Word) | Считывает из нетипизированного файла, связанного с файловой переменной f, одну или несколько записей (их количество задается целочисленным выражением Count) в переменную Buf |
| procedure BlockWrite (var F: File; var Buf; Count: Word) | Записывает в нетипизированный файл, связанный с файловой переменной f, одну или несколько записей (их количество задается целым выражением Count) из переменной Buf |

| Функция или процедура | Описание |
|--|--|
| procedure ChDir (S: String) | Выполняет смену текущего каталога на каталог, маршрут к которому указан в строковой переменной <i>S</i> |
| procedure Close (var F:file) | Закрывает открытый файл, связанный с файловой переменной <i>F</i> |
| procedure Erase (var F:file) | Стирает внешний файл, связанный с файловой переменной <i>F</i> |
| function Eof (var F:file): Boolean | Возвращает для файла, связанного с файловой переменной <i>F</i> , состояние End-of-file (конец файла): True — если текущее положение указателя находится в конце файла или файл пустой; False — во всех остальных случаях |
| function FilePos (var F:file): Longint | Возвращает текущую позицию для файла, связанного с файловой переменной <i>F</i> . При положении текущего указателя в начале файла возвращает нулевое значение. Для текстовых файлов не используется |
| function FileSize (var F:file): Longint | Возвращает текущий размер файла, связанного с файловой переменной <i>F</i> . Если файл пустой, возвращает нулевое значение. Для текстовых файлов не используется |
| procedure Flush (var F: text) | Сбрасывает буфер текстового файла, связанного с файловой переменной <i>F</i> и открытого для вывода процедурами Rewrite или Append. Это дает гарантию того, что вся информация, записываемая в файл, будет сохранена во внешнем файле. Не влияет на файлы, открытые для ввода |
| procedure GetDir (D: Byte; var S: String): | Возвращает в строке <i>S</i> текущий каталог на заданном диске. Имя каталога находится в строковой переменной <i>S</i> , а диск задается значением параметра <i>D</i> : 0 — текущий диск; 1 — диск А; 2 — диск В; 3 — диск С и т. д. Если значение, заданное в параметре <i>D</i> , неверное, возвращается результат «X:\» |

Продолжение

| Функция или процедура | Описание |
|---|--|
| function IOResult:Integer | Возвращает целое значение, являющееся состоянием последней выполненной операции ввода/вывода. Нулевое значение соответствует нормальному завершению операции |
| procedure Mkdir (S:String) | Создает подкаталог, имя для которого задается строковой переменной <i>S</i> |
| procedure Read (var F: text; v1, ...) | Считывает одно или несколько значений из файла, связанного с файловой переменной <i>F</i> , в одну или несколько переменных <i>v1,...</i> |
| procedure ReadLn (var F: text; v1, ...) | То же, что и Read, но выполняет пропуск до начала следующей строки текстового файла |
| procedure Rename (var F, S) | Переименовывает внешний файл, связанный с файловой переменной <i>F</i> , присваивая ему имя, содержащееся в строковой переменной <i>S</i> |
| procedure Reset (var F) | Открывает существующий файл, связанный с файловой переменной <i>F</i> . Указатель устанавливается в начало файла. Текстовый файл открывается только для чтения |
| procedure Rewrite (var F) | Создает и открывает новый файл, связанный с файловой переменной <i>F</i> . Если файл с указанным именем уже существует, старый файл будет стерт, а на его месте создан новый пустой файл. Текущий указатель устанавливается в начало файла |
| procedure Rmdir (S: String) | Удаляет пустой подкаталог, маршрут которого указан в строковой переменной <i>S</i> |
| procedure Seek (var F; N: Longint) | Перемещает текущую позицию в файле, связанном с файловой переменной <i>F</i> , на заданный элемент. Началу файла соответствует нулевое значение <i>N</i> . Для текстовых файлов не используется |
| function SeekEof (var F:text): Boolean | Возвращает для текстового файла, связанного с файловой переменной <i>F</i> , состояние «конец файла» |

| Функция или процедура | Описание |
|---|--|
| function SeekEoln (var F: text): Boolean | Возвращает для текстового файла, связанного с файловой переменной <i>F</i> , состояние «конец строки» |
| procedure SetTextBuf (var F: Text; var Buf) | Назначает для текстового файла, связанного с файловой переменной <i>F</i> , буфер ввода/вывода. Никогда не применяется к открытым файлам, так как в этом случае возможны потери данных |
| procedure Truncate (var F:file) | Усекает размер файла, связанного с файловой переменной <i>F</i> , до текущей позиции. Вся информация после текущего положения указателя теряется. Для текстовых файлов не используется |
| procedure Write (var F: text, v1, ...) | Записывает в файл, связанный с файловой переменной <i>F</i> , одно или несколько значений, хранящихся в переменных <i>v1, ..., vn</i> . Файловая переменная не обязательно связана с текстовым файлом. В случае текстового файла значения могут иметь целый, вещественный, строковый, символьный или булевый тип. В случае типизированного файла тип значений должен совпадать с типом компонентов |
| Procedure WriteLn (var F: text, v1, ...) | Делает то же, что Write, но затем записывает в текстовый файл признак конца строки |

Графические режимы

| Константа | Значение | Описание графического режима |
|------------|----------|--------------------------------|
| EGALo | 0 | 640x200, 16 цветов, 4 страницы |
| EGANi | 1 | 640x350, 16 цветов, 2 страницы |
| EGA64Lo | 0 | 640x200, 16 цветов, 1 страница |
| EGA64Ni | 1 | 640x350, 4 цвета, 1 страница |
| HercMonoHi | 0 | 720x348, 2 страницы |
| VGALo | 0 | 640x200, 16 цветов, 4 страницы |
| VGAMed | 1 | 640x350, 16 цветов, 2 страницы |
| VGANi | 2 | 640x480, 16 цветов, 1 страница |
| IBM8514Lo | 0 | 640x480 точек, 256 цветов |
| IBM8514Hi | 1 | 1024x768 точек, 256 цветов |

Процедуры модуля Graph для вывода основных графических элементов

| Процедура | Описание |
|--|--|
| procedure PutPixel(x, y: Integer; color : Word) | Выводит графическую точку (пиксел) в положении, заданном графическими координатами x и y , и с указанным цветом color |
| procedure Line(xstart, ystart, xend, yend : Integer) | Выводит отрезок прямой линии из точки (xstart, ystart) в точку (xend, yend) |
| procedure LineRel(xshift, yshift: Integer) | Выводит отрезок прямой линии из текущего положения указателя в точку, заданную сдвигами по x - координате (xshift) и y - координате (yshift) |
| procedure LineTo(xend, yend: Integer) | Выводит отрезок прямой линии из текущего положения указателя в точку, заданную координатами (xend, yend) |
| procedure Circle(x, y: Integer; radius: Word) | Выводит окружность с центром в точке (x, y) и радиусом radius |
| procedure Rectangle (x1, y1, x2, y2: Integer) | Выводит прямоугольник, у которого координаты левого верхнего и правого нижнего углов составляют соответственно (x1, y1) и (x2, y2) |
| procedure OutTextXY(x, y: Integer; Text_String : String) | Выводит строку текста, начиная с точки с координатами (x, y) |
| procedure OutText (Text_String:String) | Выводит строку текста начиная с текущего положения указателя |
| procedure DrawPoly (NumVertices: Word; var coordinates) | Выводит ломаную линию, проходящую через NumVertices вершин, координаты которых хранятся в переменной coordinates. Эта переменная представляет собой массив записей, содержащих два поля — координаты x и y . Здесь удобно использовать встроенный тип модуля Graph: <pre> PointType = record X, Y : integer; end;</pre> |

| Процедура | Описание |
|--|---|
| procedure Ellipse(x, y: Integer; Start_Angle, End_Angle, xradius, yradius: Word) | Выводит эллиптическую дугу с центром в точке (x,y), начальным и конечным углами Start_Angle и End_Angle и радиусами вдоль осей координат xradius и yradius. Углы отсчитываются от положительного направления оси x в градусах |
| procedure Arc(x, y : Integer; Start_Angle, End_Angle, radius: Word) | Выводит дугу окружности с центром в точке (x,y), начальным и конечным углами Start_Angle и End_Angle и радиусом radius. Углы отсчитываются от положительного направления оси x в градусах |
| procedure FloodFill(x, y: Integer; border: Word) | Закрашивает замкнутую область, в которой содержится точка (x, y) согласно заданным параметрам. Если точка находится вне области, закрашивается внешняя часть рисунка. Параметр border задает цвет границы |

Процедуры и функции модуля Graph для определения параметров графического режима и графических элементов

| Процедура (функция) | Описание |
|---|--|
| function GetMaxX: Integer | Возвращает разрешение по горизонтали для текущего графического режима |
| function GetMaxY: Integer | Возвращает разрешение по вертикали для текущего графического режима |
| procedure DetectGraph (Graph_Driver, Graph_Mode: Integer) | Определяет, какой графический драйвер (Graph_Driver) и графический режим (Graph_Mode) нужно использовать |
| function GetX: Integer | Возвращает текущую x-координату указателя |
| function GetY : Integer | Возвращает текущую y-координату указателя |

| Процедура (функция) | Описание |
|--|--|
| function GetColor: Word | Возвращает текущий цвет рисунка |
| function GetBkColor: Word | Возвращает текущий цвет фона |
| function GetMaxColor: Word | Возвращает максимальный номер цвета, который можно использовать в процедуре SetColor |
| function GetPixel(x, y: Integer): Word | Возвращает цвет графической точки с указанными координатами |

Процедуры и функции модуля Graph для установки параметров графического режима и графических элементов

| Процедура (функция) | Описание |
|--|--|
| procedure MoveTo(x, y: Integer) | Устанавливает указатель в точку с графическими координатами (x, y) |
| procedure MoveRel(xshift, yshift: Integer) | Смещает указатель в точку согласно заданным смещениям относительно текущего положения по горизонтали и по вертикали |
| procedure SetGraphMode (mode: Integer) | Переключает экран в заданный режим и очищает его |
| procedure SetColor(color: Word) | Устанавливает цвет рисунка |
| procedure SetBkColor (color: Word) | Устанавливает цвет фона |
| procedure SetFillStyle (pattern, color: Word) | Устанавливает тип закрашки (pattern) и ее цвет (color) |
| procedure SetFillPattern (pattern: FillPatternType; color: Word) | Устанавливает тип закрашки, определенный пользователем, и ее цвет. Для описания закрашки используется встроенный тип FillPatternType = array [1..8] of Byte; |

| Процедура (функция) | Описание |
|--|--|
| <pre>procedure SetLineStyle (Line_Style, pattern, thickness: Word)</pre> | <p>Устанавливает параметры рисования линий – тип (Line_Style), задаваемый пользователем тип(pattern) и толщину (thickness). Используются следующие типы линий:</p> <p>solidln — сплошная; dotted — точечная; centerln — штрихпунктирная; dashedln — пунктирная; userbitln — пользовательский тип</p> <p>Пользователь может задать тип линии с помощью 16-битного шаблона. Толщина линии — normwidth (нормальная) или thickwidth(толстая)</p> |
| <pre>procedure SetTextStyle(font, direction, charsize: Word)</pre> | <p>Устанавливает параметры вывода текста в графическом режиме:</p> <p>font - шрифт; direction - направление (HorizDir, VertDir); charsize — размер символов</p> |
| <pre>procedure SetVisualPage (page: _Word)</pre> | <p>Устанавливает номер отображаемой страницы</p> |
| <pre>procedure SetPalette(Color_Num: Word; color: ShortInt)</pre> | <p>Заменяет цвет палитры, заданный параметром Color_Num, значением color</p> |

**Встроенные константы Турбо Паскаля, обозначающие цвета
и соответствующие им числовые коды**

| Цвет | Константа | Код |
|-----------|-----------|-----|
| Черный | Black | 0 |
| Синий | Blue | 1 |
| Зеленый | Green | 2 |
| Бирюзовый | Cyan | 3 |
| Красный | Red | 4 |

Окончание

| Цвет | Константа | Код |
|------------------|--------------|-----|
| Розовый | Magenta | 5 |
| Коричневый | Brown | 6 |
| Светло-серый | LightGray | 7 |
| Темно-серый | DarkGray | 8 |
| Светло-синий | LightBlue | 9 |
| Светло-зеленый | LightGreen | 10 |
| Светло-бирюзовый | LightCyan | 11 |
| Светло-красный | LightRed | 12 |
| Светло-розовый | LightMagenta | 13 |
| Желтый | Yellow | 14 |
| Белый | White | 15 |

Стили заполнения геометрических фигур

| Константа | Код | Описание |
|----------------|-----|--|
| EmptyFill | 0 | Сплошное заполнение цветом фона |
| SolidFill | 1 | Сплошное заполнение заданным цветом |
| LineFill | 2 | Заполнение горизонтальными линиями |
| LtSlashFill | 3 | Диагональное заполнение (//) |
| SlashFill | 4 | Диагональное заполнение толстыми линиями (///) |
| BkSlashFill | 5 | Обратное диагональное заполнение толстыми линиями (\\) |
| LtBkSlashFill | 6 | Обратное диагональное заполнение (\\) |
| HatchFill | 7 | Клетчатое заполнение |
| XhatchFill | 8 | Косое клетчатое заполнение |
| InterleaveFill | 9 | Чередующееся линейное заполнение |
| WideDotFill | 10 | Редко расположенные точки |
| CloseDotFill | 11 | Часто расположенные точки |
| UserFill | 12 | Стиль определен пользователем |

Язык Си

Основные типы данных

| Тип данных | Размер, бит | Диапазон значений |
|-------------------|-------------|-------------------|
| unsigned char | 8 | 0...255 |
| char | 8 | -128...127 |
| enum | 16 | -32768...32767 |
| unsigned int | 16 | 0...65535 |
| short int (short) | 16 | -32768...32767 |

| Тип данных | Размер, бит | Диапазон значений |
|----------------|-------------|--------------------------|
| unsigned short | 16 | 0...65535 |
| int | 16 | -32768...32767 |
| unsigned long | 32 | 0...4294967295 |
| long | 32 | -2147483648...2147483647 |
| float | 32 | -3,4e-38...3,4e+38 |
| double | 64 | -1,7e308...1,7e+308 |
| long double | 80 | -3,4e-4932...1,1e+4932 |

Математические функции (math.h)

| Функция | Прототип и краткое описание действий |
|---------|---|
| abs | int abs(int i); Возвращает абсолютное значение целого аргумента |
| acos | double acos(double x); Функция арккосинуса. Значение аргумента должно находиться в диапазоне от -1 до +1 |
| asin | double asin(double x); Функция арксинуса. Значение аргумента должно находиться в диапазоне от -1 до +1 |
| atan | double atan (double x); Функция арктангенса |
| atan2 | double atan2(double y, double x); Функция арктангенса от значения y/x |
| cabs | double cabs(struct complex znum); Вычисляет абсолютное значение комплексного числа znum. Определение структуры (типа) complex - в файле math.h |
| ceil | double ceil (double x); Вычисляет ближайшее целое, не меньшее, чем аргумент x |
| cos | double cos(double x); Функция косинуса. Угол (аргумент) задается в радианах |
| exp | double exp(double x); Вычисляет значение e^x (экспоненциальная функция) |
| fabs | double fabs (double x); Возвращает абсолютное значение вещественного аргумента x двойной точности |
| floor | double floor(double x); Находит наибольшее целое, не превышающее значение x. Возвращает его в форме double |
| fmod | double fmod(double x, double y); Возвращает остаток от деления нацело x на y |

| Функция | Прототип и краткое описание действий |
|---------|--|
| frexp | double frexp(double value, int *exp); Разбивает число с плавающей точкой value на нормализованную мантиссу и целую часть как степень числа 2. Целочисленная степень записывается в область памяти, на которую указывает exp, а мантисса используется как значение, которое возвращает функция |
| hypot | double hypot(double x, double y); Вычисляет гипотенузу z прямоугольного треугольника по значениям катетов x, y |
| labs | long labs(long x); Возвращает абсолютное значение целого аргумента long x |
| ldexp | double ldexp(double v, int e) Возвращает значение выражения $v \cdot 2^e$ |
| log | double log(double x); Возвращает значение натурального логарифма (ln x) |
| Log10 | double log10(double x); Возвращает значение десятичного логарифма x |
| modf | double modf(double value, double *iptr); Разделяет число с плавающей точкой value на целую и дробную части. Целая часть записывается в область памяти, на которую указывает iptr, дробная часть является значением, возвращаемым функцией |
| poly | double poly(double x, int n, double c[]); Вычисляет значение полинома: $c[n]x^n + c[n-1]x^{n-1} + \dots + c[1]x + c[0]$ |
| pow | double pow(double x, double y); Возвращает значение x в степени y |
| pow10 | double pow10(int p); Возвращает значение 10 в степени p |
| sin | double sin(double x); Функция синуса. Угол (аргумент) задается в радианах |
| sinh | double sinh(double x); Возвращает значение гиперболического синуса |
| sqrt | double sqrt(double x); Возвращает положительное значение квадратного корня из x |
| tan | double tan(double x); Функция тангенса. Угол (аргумент) задается в радианах |
| tanh | double tanh(double x); Возвращает значение гиперболического тангенса для x |

Функции и макросы проверки и преобразования символов (ctype.h)

| Функция | Прототип и краткое описание действий |
|---------|--|
| isalnum | <code>int isalnum(int c);</code> Дает значение не нуль, если <i>c</i> - код буквы или цифры (<i>A - Z, a - z, 0-9</i>), и нуль - в противном случае |
| isalpha | <code>int isalpha(int c);</code> Дает значение не нуль, если <i>c</i> - код буквы (<i>A - Z, a + z</i>), и нуль - в противном случае |
| isascii | <code>int isascii(int c);</code> Дает значение не нуль, если <i>c</i> есть код ASCII, т.е. принимает значение от 0 до 127, в противном случае - нуль |
| iscntrl | <code>int iscntrl(int c);</code> Дает значение не нуль, если <i>c</i> - управляющий символ с кодами 0x00 - 0x01F или 0x7F, и нуль - в противном случае |
| isdigit | <code>int isdigit(int c);</code> Дает значение не нуль, если <i>c</i> - цифра (0 - 9) в коде ASCII, и нуль - в противном случае |
| isgraph | <code>int isgraph(int c);</code> Дает значение не нуль, если <i>c</i> - видимый (изображаемый) символ с кодом (0x21 + 0x7E), и нуль - в противном случае |
| islower | <code>int islower(int c);</code> Дает значение не нуль, если <i>c</i> - код буквы на нижнем регистре (<i>a-z</i>), и нуль - в противном случае |
| isprint | <code>int isprint(int c);</code> Дает значение не нуль, если <i>c</i> - печатный символ с кодом (0x20 - 0x7E), и нуль - в противном случае |
| ispunct | <code>int ispunct(int c);</code> Дает значение не нуль, если <i>c</i> - символ-разделитель (т.е. соответствует <code>iscntrl</code> или <code>isspace</code>) и нуль - в противном случае |
| isspace | <code>int isspace(int c);</code> Дает значение не нуль, если <i>c</i> - обобщенный пробел: пробел, символ табуляции, символ новой строки или новой страницы, символ возврата каретки (0x09 - 0x0D, 0x20), и нуль - в противном случае |
| isupper | <code>int isupper(int c);</code> Дает значение не нуль, если <i>c</i> -код буквы на верхнем регистре (<i>A - Z</i>), и нуль - в противном случае |

| Функция | Прототип и краткое описание действий |
|----------|--|
| isxdigit | <code>int isxdigit(int c);</code> Дает значение не нуль, если <i>c</i> - код шестнадцатеричной цифры (0-9, A - F, a - f), и нуль – в противном случае |
| toascii | <code>int toascii(int c);</code> Преобразует целое число <i>c</i> в символ кода ASCII, обнуляя все биты, кроме младших семи. Результат от 0 до 127 |
| tolower | <code>int tolower(int c);</code> Преобразует код буквы <i>c</i> к нижнему регистру, не буквенные коды не изменяются |
| toupper | <code>int toupper(int c);</code> Преобразует код буквы <i>c</i> к верхнему регистру, не буквенные коды не изменяются |

Функции ввода-вывода для стандартных потоков (stdio.h)

| Функция | Прототип и краткое описание действий |
|---------|---|
| getch | <code>int getch(void);</code> Считывает один символ с клавиатуры без отображения на экране |
| getchar | <code>int getchar(void);</code> Считывает очередной символ из стандартного входного потока (stdin) |
| gets | <code>char *gets(char *s);</code> Считывает строку <i>s</i> из стандартного входного потока (stdin) |
| printf | <code>int printf(const char *format [, argument, ...]);</code> Функция форматированного вывода в стандартный поток stdout |
| putchar | <code>int putchar(int c);</code> Записывает символ <i>c</i> в стандартный поток вывода (stdout) |
| puts | <code>int puts(const char *s);</code> Записывает строку <i>s</i> в стандартный поток вывода (stdout) |
| scanf | <code>int scanf(const char *format[, address...]);</code> Форматированный ввод из стандартного потока stdin |
| sprintf | <code>int sprintf(char *s, const char *format[, argument, ...]);</code> Функция форматированной записи в строку <i>s</i> |
| sscanf | <code>int sscanf(const char *s, const char *format[, address, ...]);</code> Функция форматированного чтения из строки <i>s</i> |

| Функция | Прототип и краткое описание действий |
|---------|---|
| ungetch | int ungetch(int c); Возвращает символ c в стандартный поток ввода stdin, заставляя его быть следующим считываемым символом |

Функции для работы со строками (string.h, stdlib.h)

| Функция | Прототип и краткое описание действий |
|---------|---|
| atof | double atof(const char *str); Преобразует строку str в вещественное число типа double |
| atoi | int atoi (const char *str); Преобразует строку str в целое число типа int |
| atol | long atol (const char *str); Преобразует строку str в целое число типа long |
| itoa | char *itoa(int v, char *str, int baz); Преобразует целое v в строку str. При изображении числа используется основание baz ($2 < baz < 36$). Для отрицательного числа и baz = 10 первый символ "минус" (-) |
| ltoa | char *ltoa(long v, char *str, int baz); Преобразует длинное целое v в строку str. При изображении числа используется основание baz ($2 < baz < 36$) |
| strcat | char *strcat (char *sp, const char*si); Приписывает строку si к строке sp (конкатенация строк) |
| strchr | char *strchr (const char *str, int c); Ищет в строке str первое вхождение символа c |
| strcmp | int strcmp(const char *str1, const char *str2); Сравнивает строки str1 и str2. Результат отрицателен, если str1 < str2; равен нулю, если str1==str2, и положителен, если str1 > str2 (сравнение беззнаковое) |
| strcpy | char *strcpy(char *sp, const char *si); Копирует байты строки si в строку sp. |
| strcspn | int strcspn(const char *str1, const char *str2); Определяет длину первого сегмента строки str1, содержащего символы, не входящие во множество символов строки str2 |
| strdup | char *strdup (const char *str); Выделяет память и переносит в нее копию строки str |

Продолжение

| Функция | Прототип и краткое описание действий |
|-----------------------|--|
| <code>strlen</code> | <code>unsigned strlen (const char *str);</code> Вычисляет длину строки <code>str</code> |
| <code>strlwr</code> | <code>char *strlwr(char *str);</code> Преобразует буквы верхнего регистра в строке в соответствующие буквы нижнего регистра |
| <code>strncat</code> | <code>char *strncat(char *sp, const char *si, int kol);</code> Приписывает <code>kol</code> символов строки <code>si</code> к строке <code>sp</code> (конкатенация) |
| <code>strncmp</code> | <code>int strncmp(const char *str1, const char *str2, int kol);</code> Сравнивает части строк <code>str1</code> и <code>str2</code> , причем рассматриваются первые <code>kol</code> символов. Результат отрицателен, если <code>str1 < str2</code> ; равен нулю, если <code>str1 == str2</code> , и положителен, если <code>str1 > str2</code> |
| <code>strncpy</code> | <code>char *strncpy (char *sp, const char *si, int kol);</code> Копирует <code>kol</code> символов строки <code>si</code> в строку <code>sp</code> ("хвост" отбрасывается или дополняется пробелами) |
| <code>strnicmp</code> | <code>int strnicmp(char *str1, const char *str2, int kol);</code> Сравнивает не более <code>kol</code> символов строки <code>str1</code> и строки <code>str2</code> , не делая различия регистров (см. функцию <code>strncmp</code> ()) |
| <code>strnset</code> | <code>char *strnset(char *str, int c, int kol);</code> Заменяет первые <code>kol</code> символов строки <code>str</code> символом <code>c</code> |
| <code>strpbrk</code> | <code>char *strpbrk(const char *str1, const char *str2);</code> Ищет в строке <code>str1</code> первое появление любого из множества символов, входящих в строку <code>str2</code> |
| <code>strrchr</code> | <code>char *strrchr (const char *str, int c);</code> Ищет в строке <code>str</code> последнее вхождение символа <code>c</code> |
| <code>strset</code> | <code>char *strset (char *str, int c);</code> Заполняет строку <code>str</code> заданным символом <code>c</code> |
| <code>strspn</code> | <code>int strspn(const char *str1, const char *str2);</code> Определяет длину первого сегмента строки <code>str1</code> , содержащего только символы, из множества символов строки <code>str2</code> |

| Функция | Прототип и краткое описание действий |
|---------------------|--|
| <code>strstr</code> | <code>char *strstr(const char *str1, const char *str2);</code> Ищет в строке <code>str1</code> подстроку <code>str2</code> . Возвращает указатель на тот элемент в строке <code>str1</code> , с которого начинается подстрока <code>str2</code> |
| <code>strtod</code> | <code>double strtod (const char *str, char **endptr);</code> Преобразует символьную строку <code>str</code> в число двойной точности. Если <code>endptr</code> не равен <code>NULL</code> , то <code>*endptr</code> возвращается как указатель на символ, при достижении которого прекращено чтение строки <code>str</code> |
| <code>strtok</code> | <code>char *strtok(char *str1, const char *str2);</code> Ищет в строке <code>str1</code> лексемы, выделенные символами, из второй строки |
| <code>strtol</code> | <code>long strtol (const char *str, char **endptr, int baz);</code> Преобразует символьную строку <code>str</code> к значению "длинное число" с основанием <code>baz</code> ($2 < baz < 36$). Если <code>endptr</code> не равен <code>NULL</code> , то <code>*endptr</code> возвращается как указатель на символ, при достижении которого прекращено чтение строки <code>str</code> |
| <code>strupr</code> | <code>char *strupr(char *str);</code> Преобразует буквы нижнего регистра в строке <code>str</code> в буквы верхнего регистра |
| <code>ultoa</code> | <code>char *ultoa(unsigned long v, char *str, int baz);</code> Преобразует беззнаковое длинное целое <code>v</code> в строку <code>str</code> |

Функции для выделения и освобождения памяти (`alloc.h, stdlib.h`)

| Функция | Прототип и краткое описание действий |
|---------------------|--|
| <code>calloc</code> | <code>void *calloc(unsigned n, unsigned m);</code> Возвращает указатель на начало области динамически распределенной памяти для размещения <code>n</code> элементов по <code>m</code> байт каждый. При неудачном завершении возвращает значение <code>NULL</code> |
| <code>free</code> | <code>void free (void *bl);</code> Освобождает ранее выделенный блок динамически распределяемой памяти с адресом первого байта <code>bl</code> |

| Функция | Прототип и краткое описание действий |
|---------|--|
| malloc | <code>void *malloc(unsigned s);</code> Возвращает указатель на блок динамически распределяемой памяти длиной в <i>s</i> байт. При неудачном завершении возвращает значение NULL |
| realloc | <code>void *realloc(void *bl, unsigned ns);</code> Изменяет размер участка динамической памяти с адресом начала <i>bl</i> , делая его равным <i>ns</i> (байт). Если <i>bl</i> равен NULL, то функция выполняется как <code>malloc()</code> . При неудачном завершении возвращает значение NULL. |

Функции для работы с терминалом в текстовом режиме (conio.h)

| Функция | Прототип и краткое описание действий |
|-----------|---|
| clreol | <code>void clreol(void);</code> Стирает символы от позиции курсора до конца строки в текстовом окне |
| clrscr | <code>void clrscr(void);</code> Очищает экран |
| cgets | <code>char *cgets(char *str);</code> Выводит на экран строку <i>str</i> |
| cprintf | <code>rintf(const char *format[, ...argument, ...]);</code> Выводит форматированную строку в текстовое окно, созданное функцией <code>window()</code> |
| cputs | <code>int cputs(char *str);</code> Помещает в символьный массив <i>str</i> строку с клавиатуры (консоли) |
| cscanf | <code>int cscanf(const char *format[, address, ...]);</code> Функция форматированного ввода, которая используется при работе с терминалом в текстовом режиме |
| delline | <code>void delline(void);</code> Удаляет строку в текстовом окне (где находится курсор) |
| gotoxy | <code>void gotoxy(int x, int y);</code> Перемещает курсор в позицию текстового окна с координатами (<i>x,y</i>) |
| highvideo | <code>void highvideo(void);</code> Повышает яркость символов, выводимых на экран после ее вызова |

Окончание

| Функция | Прототип и краткое описание действий |
|----------------|---|
| movetext | <code>int movetext(int x0, int y0, int x1, int y1, int x, int y);</code> Переносит текстовое окно в область экрана, правый верхний угол которой (x,y). Координаты угловых точек окна – (x0, y0),(x1, y1) |
| normvideo | <code>void normvideo(void);</code> Устанавливает нормальную яркость выводимых на экран символов |
| textattr | <code>void textattr(int newattr);</code> Устанавливает атрибуты (фон, цвет) символов, выводимых на экран |
| textbackground | <code>void textbackground(int c);</code> Устанавливает цвет фона по значению параметра c |
| textcolor | <code>void textcolor(int c);</code> Устанавливает цвет символов по значению параметра c |
| textmode | <code>void textmode(int m);</code> Переводит экран в текстовый режим по значению параметра m |
| wherex | <code>int wherex(void);</code> Возвращает значение горизонтальной координаты курсора |
| wherey | <code>int wherey(void);</code> Возвращает значение вертикальной координаты курсора |
| window | <code>void window(int x0, int y0, int x1, int y1);</code> Создает текстовое окно по координатам угловых точек (x0, y0) |

Специальные функции

| Функция | Прототип и краткое описание действий | Место нахождения |
|---------|--|------------------|
| delay | <code>void delay(unsigned x);</code> Приостанавливает выполнение программы на x мсек | dos.h |
| getenv | <code>char *getenv(const char *name);</code> Ищет в списке переменных среды имя, заданное параметром name. В случае нахождения возвращает указатель на строку, являющуюся значением этой переменной среды | stdlib.h |

Продолжение

| Функция | Прототип и краткое описание действий | Место нахождения |
|---------|---|------------------|
| kbhit | int kbhit(void); Возвращает ненулевое целое, если в буфере клавиатуры присутствуют коды нажатия клавиш, в противном случае – нулевое значение | conio.h |
| memcmp | int memcmp(const void *s1, const void *s2, unsigned n); Сравнивает посимвольно две области памяти s1 и s2 длиной n байт. Возвращает значение меньше нуля, если s1 < s2, нуль, если s1 == s2, и больше нуля, если s1 > s2 | mem.h |
| memcpy | int memcpy(void *p, const void *i, unsigned n); Копирует блок длиной n байт из области памяти i в область памяти p | mem.h |
| memicmp | int memicmp(const void *s1, const void *s2, unsigned n); Подобна memcpy, за тем исключением, что игнорируются различия между буквами верхнего и нижнего регистра | mem.h |
| memmove | void *memmove(void *dest, const void *src, int n); Копирует блок длиной n байтов из src в dest. Возвращает указатель dest | mem.h |
| memset | void *memset(void *s, int c, unsigned n); Записывает во все байты области памяти s значение c. Длина области s равна n байт | mem.h |
| nosound | void nosound(void); Прекращает подачу звукового сигнала, начатую функцией sound() | dos.h |
| peek | int peek(unsigned s, unsigned c); Возвращает целое значение (слово), записанное в сегменте s со смещением c | dos.h |
| peekb | char peekb(unsigned s, unsigned c); Возвращает один байт, записанный в сегменте s со смещением c, т.е. по адресу s: c | dos.h |
| poke | void poke(unsigned s, unsigned c, int v); Помещает значение v в слово сегмента s со смещением c, т.е. по адресу s: c | dos.h |

Окончание

| Функция | Прототип и краткое описание действий | Место нахождения |
|---------|--|------------------|
| pokeb | void pokeb(unsigned s, unsigned c, char v); То же, что и roke, но помещает один байт v по адресу s: c | dos.h |
| putenv | int putenv(const char *name); Добавляет строку в окружение программы | stdlib.h. |
| rand | int rand(void); Возвращает псевдослучайное целое число из диапазона 0 - (2 ¹⁵ - 1), может использовать функцию srand() | stdlib.h |
| signal | int signal(int sig); Вызывает программный сигнал с номером sig. Используется для обработки исключительных ситуаций в языке Си | signal.h |
| sound | void sound(unsigned f); Вызывает звуковой сигнал с частотой f Гц | dos.h |
| srand | void srand(unsigned seed); Функция инициализации генератора случайных чисел (rand); seed - любое беззнаковое целое число | stdlib.h |

**Функции для работы с файлами, связанными с потоками
(stdio.h)**

| Функция | Прототип |
|----------|---|
| clearerr | void clearerr (FILE *stream); Сбрасывает индикаторы достижения конца файла и обнаружения ошибки для файла, связанного с потоком stream |
| fclose | int fclose (FILE *stream); Закрывает файл, связанный с потоком stream |
| feof | int feof(FILE *stream); Проверяет, достигнут ли конец файла, связанного с потоком stream |
| ferror | int ferror(FILE *stream); Проверяет, не возникла ли ошибка записи или чтения при доступе к файлу, связанному с потоком stream |
| fgetc | int fgetc (FILE *stream); Считывает символ из файла, связанного с потоком stream |

| Функция | Прототип |
|---------|---|
| fgetpos | <code>int fgetpos(FILE *stream, fpos_t *pos);</code> Копирует значение указателя текущей позиции в файле, связанном с потоком <code>stream</code> , в объект, на который указывает <code>pos</code> . Тип принимающей переменной задан в заголовочном файле <code>stdio.h</code> |
| fgets | <code>char *fgets(char *string, int n, FILE *stream);</code> Читает не более $n-1$ символов из файла, связанного с потоком <code>stream</code> , в массив, адресуемый указателем <code>string</code> . За последним символом, помещенным в массив, записывается '\0' |
| fopen | <code>FILE *fopen(const char *filename, const char *mode);</code> Открывает файл, на имя которого указывает <code>filename</code> , в режиме, задаваемом <code>mode</code> , и связывает его с потоком. Указатель на поток является значением, возвращаемым функцией |
| fprintf | <code>int fprintf(FILE *stream, const char *format [, argument, ...]);</code> Функция форматного вывода в файл, связанный с потоком <code>stream</code> (см. <code>printf()</code>) |
| fputc | <code>int fputc(int c, FILE *stream);</code> Выводит символ <code>c</code> в файл, связанный с потоком <code>stream</code> |
| fputs | <code>int fputs(const char *string, FILE *stream);</code> Записывает символьную строку, на которую указывает <code>string</code> , в файл, связанный с потоком <code>stream</code> |
| fscanf | <code>int fscanf(FILE *stream, const char *format[, argument, ...]);</code> Функция форматного ввода из файла, связанного с потоком <code>stream</code> (см. <code>scanf()</code>) |
| fseek | <code>int fseek(FILE *stream, long offset, int point);</code> Перемещает указатель текущей позиции в файле, связанном с потоком <code>stream</code> , на <code>offset</code> байт относительно точки отсчета, определенной значением <code>point</code> (0 - от начала файла, 1 - от текущей позиции, 2 - от конца файла). |
| ftell | <code>long ftell(FILE *stream);</code> Возвращает значение указателя текущей позиции в файле, связанном с потоком <code>stream</code> |

Функции низкого уровня для работы с файлами (io.h, fcntl.h, unistd.h)

| Функция | Прототип и краткое описание действий |
|---------|---|
| close | <code>int close (int handle);</code> Закрывает файл, связанный с дескриптором handle |
| creat | <code>int creat (const char *path, int mode);</code> Создает новый файл в режиме, заданном в mode |
| eof | <code>int eof(int handle);</code> Проверяет, достигнут ли конец файла, связанного с дескриптором – handle |
| lseek | <code>long lseek(int handle, long offset, int point);</code> Перемещает указатель текущей позиции в файле, связанном с дескриптором handle на offset байтов относительно точки отсчета, определенной значением point (0 - от начала, 1 - от текущей позиции, 2 - от конца файла) |
| open | <code>int open (const char *path, int access [, unsigned mode]);</code> Открывает файл, на имя которого указывает path в режиме, заданном в access. Возвращает дескриптор файла. Mode определяет тип файла и права доступа |
| read | <code>int read(int handle, void *buf, unsigned len);</code> Читает len байтов из файла, связанного с дескриптором handle в буфер, на который указывает buf |
| sopen | <code>int sopen(const char *path, int access, int shflag, [, unsigned mode]);</code> Открывает файл, на имя которого указывает path, для совместного использования несколькими процессами. Access - режим доступа; shflag - режим разделения файла; mode определяет тип файла и права доступа. Возвращает дескриптор файла |
| tell | <code>long tell(int handle);</code> Возвращает значение указателя текущей позиции в файле, связанном с дескриптором handle |
| write | <code>int write (int handle, void *buf, unsigned nbyte);</code> Записывает nbyte байтов из буфера, на который указывает buf, в файл, связанный с дескриптором handle |

Функции для управления графической системой

Для работы с графической библиотекой необходимо включить в программу заголовочный файл **graphics.h**

| Функция | Прототип и краткое описание действий |
|-------------------|---|
| closegraph | <code>void far closegraph(void);</code> Перевод системы из графического режима в текстовый |
| graphdefaults | <code>void far graphdefaults(void);</code> Устанавливает по умолчанию все параметры графической системы |
| _graphfreemem | <code>void far _graphfreemem(void far *ptr, unsigned size);</code> Введение этой функции в программу позволяет отслеживать запросы на освобождение size байт памяти функциями графической библиотеки |
| _graphgetmem | <code>void far *far _graphgetmem(unsigned size);</code> Введение этой функции в программу позволяет отслеживать запросы на выделение size байт памяти функциями графической библиотеки |
| initgraph | <code>void far initgraph(int far *graphdriver, int far *graphmode, char far *pathdriver);</code> Перевод системы в графический режим (из текстового режима), инициализация графики |
| installuserdriver | <code>int far installuserdriver(char far *name, int huge *detect)(void);</code> Добавляет новый графический драйвер name (.BGI) в таблицу драйверов BGI |
| installuserfont | <code>int far installuserfont(char far *name);</code> Устанавливает шрифты, содержащиеся в файле name(.CHR) |
| registerbgidriver | <code>int registerbgidriver(void (*driver)(void));</code> Регистрирует драйвер driver, встроенный в текущую выполняемую программу |

Продолжение

| Функция | Прототип и краткое описание действий |
|--|---|
| registerbgiFont | int registerbgiFont (void (*font) (void)); Регистрирует шрифт font, встроенный в текущую выполняемую программу |
| restorecrtmode | void far restorecrtmode (void); Временный переход в текстовый режим, из которого была вызвана функция initgraph() |
| setgraphbufsize | unsigned far setgraphbufsize (unsigned bufsize); Устанавливает размер внутреннего буфера для графических функций |
| setactivepage | void far setactivepage(int page); Устанавливает активной для вывода графики страницу page |
| setallpalette | void far setallpalette(struct palettetype far *palette); Устанавливает все цвета палитры |
| setaspectratio | void far setaspectratio(int xasp, int yasp); Устанавливает коэффициент сжатия по координатам x и y |
| setbkcolor | void far setbkcolor(int color); Установка цвета фона |
| setcolor | void far setcolor(int color); Установка цвета точки (изображения) |
| _setcursortype (прототип в файле conio.h) | void _setcursortype(int cur_t); Устанавливает тип отображения курсора для текстовых режимов |
| setfillpattern | void far setfillpattern(char far *upattern, int color); Установка заданного пользователем шаблона закрашки экрана или области экрана |
| setgraphmode | void far setgraphmode (int mode); Переход к графическому режиму, отличному от установленного функцией initgraph () |
| setfillstyle | void far setfillstyle(int pattern, int color); Установка одного из стандартных шаблонов заполнения экрана или области экрана |

| Функция | Прототип и краткое описание действий |
|-----------------|--|
| setlinestyle | void far setlinestyle(int linestyle, unsigned upattern, int thickness); Установка толщины и типа изображаемой линии |
| setpalette | void far setpalette(int colornum, int color); Устанавливает один из цветов палитры |
| setrgbpalette | void far setrgbpalette(int colornum, int red, int green, int blue); Устанавливает цвета для графического адаптера |
| settextjustify | void far settextjustify(int horiz, int vert); Установка правил выравнивания текста при горизонтальном или вертикальном выводе функцией outtext |
| settextstyle | void far settextstyle(int font, int direction, int charsize); Установка стиля (шрифт, размеры символов) текста, вводимого функцией outtext () |
| setusercharsize | void far setusercharsize(int multx, int divx, int multy, int divy); Установка размеров символов, выводимых в графическом режиме |
| setviewport | void far setviewport(int left, int top, int right, int bottom, int clip); Установка размеров текущего окна экрана для вывода изображений или текста |
| setvisualpage | void far setvisualpage(int page); Делает видимой графическую страницу page |
| setwritemode | void far setwritemode(int mode); Устанавливает режим вывода линий в графическом режиме |

Функции для получения изображения на экране

| Функция | Прототип и краткое описание действий |
|---------------|---|
| arc | void far arc (int x, int y, int stangle, int endangle, int radius); Вычерчивание дуги окружности с центром (x, y) |
| bar | void far bar(int left, int top, int right, int bottom); Вычерчивание закрашенного прямоугольника |
| bar3d | void far bar3d (int left, int top, int right, int bottom, int depth, int topflag); Вычерчивание закрашенного параллелепипеда |
| circle | void far circle(int x, int y, int radius); Вычерчивание окружности с центром (x, y) |
| cleardevice | void far cleardevice(void); Очистка экрана цветом фона |
| clearviewport | void far clearviewport(void); Очистка ранее установленного окна графического экрана |
| drawpoly | void far drawpoly(int numpoints, int far *polypoints); Вычерчивание контура многоугольника с вершинами numpoints |
| ellipse | void far ellipse(int x, int y, int stangle, int endangle, int xradius, int yradius); Вычерчивание дуги эллипса с центром (x, y) |
| fillellipse | void far fillellipse(int x, int y, int xradius, int yradius); Вычерчивание эллипса с центром в точке (x, y) и заполнение его установленным ранее шаблоном закраски |
| fillpoly | void far fillpoly(int numpoints, int far *polypoints); Вычерчивание закрашенного многоугольника с numpoints вершинами и заполнение его установленным ранее шаблоном закраски |

Продолжение

| Функция | Прототип и краткое описание действий |
|-----------|---|
| floodfill | void far floodfill(int x, int y, int border); Заполнение установленным ранее шаблоном закраски ограниченной области экрана, в которую попадает точка с координатами (x, y) |
| line | void far line(int x1, int y1, int x2, int y2); Вычерчивание линии от (x1, y1) до (x2, y2) |
| linere1 | void far linere1(int dx, int dy); Вычерчивание линии из текущей точки в точку, отстоящую от нее на величину (dx, dy) |
| lineto | void far lineto(int x, int y); Проведение линии из текущей точки в точку с абсолютными координатами x, y |
| moverel | void far moverel(int dx, int dy); Перемещение указателя позиции из текущей точки в точку, отстоящую от нее на величину (dx, dy); |
| moveto | void far moveto(int x, int y); Перемещение указателя позиции из текущей точки в точку с абсолютными координатами x, y |
| outtext | void far outtext(char far *textstring); Вывод текстовой строки textstring начиная с текущей позиции |
| outtextxy | void far outtextxy(int x, int y, char far *textstring); Вывод текстовой строки textstring начиная с точки с координатами x, y |
| pieslice | void far pieslice (int x, int y, int stangle, int endangle, int radius); Вычерчивание закрашенного сектора круга с центром в точке (x, y) |
| putimage | void far putimage(int left, int top, void far *bitmap, int op); Вывод ранее сохраненного графического изображения в окно экрана с левым верхним углом (left, top) |
| putpixel | void far putpixel(int x, int y, int color); Вычерчивание точки с координатами x, y |

| Функция | Прототип и краткое описание действий |
|-----------|---|
| rectangle | void far rectangle(int left, int top, int right, int bottom); Вычерчивание прямоугольника с заданными вершинами |
| sector | void far sector(int x, int y, int stangle, int endangle, int xradius, int yradius); Вычерчивание сектора эллипса с центром в точке (x, y) и заполнение его установленным ранее шаблоном закрашки |

Функции для получения параметров изображения

| Функция | Прототип и краткое описание действий |
|-------------------|--|
| detectgraph | void far detectgraph(int far *graphdriver, int far *graphmode); Функция возвращает тип вашего графического адаптера graphdriver и режим graphmode с разрешением, максимально возможным для данного адаптера |
| getarccoords | void far getarccoords(struct arccoordstype far *arccoords); Возвращает в структуре arccoords значения координат дуги, построенной при последнем обращении к arc() |
| getaspectratio | void far getaspectratio(int far *xasp, int far *yasp); Возвращает коэффициент сжатия (yasp/xasp) по координатам x и y |
| getbkcolor | int far getbkcolor(void); Возвращает номер текущего цвета фона |
| getcolor | int far getcolor(void); Возвращает номер текущего цвета изображения |
| getdefaultpalette | struct palettetype *far getdefaultpalette(void); Возвращает указатель на структуру типа palettetype, содержащую информацию о палитре (наборе цветов), устанавливаемой по умолчанию функцией initgraph () |

Продолжение

| Функция | Прототип и краткое описание действий |
|-----------------|--|
| getdrivername | char *far getdrivername(void); Возвращает указатель на строку, содержащую название текущего графического драйвера |
| getfillpattern | void far getfillpattern(char far *pattern); Получение кодов, применяемых пользователем для задания шаблона заполнения экрана или его области |
| getfillsettings | void far getfillsettings(struct fillsettingstype far *fillinfo); Возвращает в структуре fillinfo значения параметров заполнения и цвета экрана |
| getgraphmode | int far getgraphmode(void); Возвращает номер графического режима |
| getimage | void far getimage(int left, int top, int right, int bottom, void far *bitmap); Получение и сохранение в области памяти, на которую указывает bitmap, окна экрана, заданного координатами вершин |
| getlinesettings | void far getlinesettings(struct linesettingstype far *lineinfo); Возвращает в структуре lineinfo значения параметров линии |
| getmaxcolor | int far getmaxcolor (void); Возвращает наибольший номер цвета, который возможно установить в текущем режиме графического драйвера помощью функции setcolor () |
| getmaxmode | int far getmaxmode(void); Возвращает наибольший номер режима, который возможно установить для текущего графического драйвера |
| getmodename | char *far getmodename(int mode_n); Возвращает указатель на строку с названием графического режима mode_n |
| getmoderange | void far getmoderange(int graphdriver, int far *lomode, int far *himode); Возвращает диапазон доступных режимов для графического драйвера graphdriver |

Продолжение

| Функция | Прототип и краткое описание действий |
|-----------------|---|
| getmaxx | int far getmaxx(void); Возвращает целое значение, равное размеру экрана по горизонтали (максимальное значение <i>x</i>) |
| getmaxy | int far getmaxy(void); Возвращает целое значение, равное размеру экрана по вертикали (максимальное значение <i>y</i>) |
| getpalette | void far getpalette(struct palettetype far *palette); Выдает указатель <i>palette</i> на структуру типа <i>palette</i> , содержащую информацию о текущей палитре (наборе цветов) |
| getpalettesize | int far getpalettesize(void); Возвращает количество цветов, доступных в текущем графическом режиме |
| getpixel | unsigned far getpixel(int <i>x</i> , int <i>y</i>); Возвращает цвет заданной (<i>x</i> , <i>y</i>) точки |
| gettextsettings | void far gettextsettings(struct textsettingstype far *texttypeinfo); Возвращает в структуре <i>texttypeinfo</i> значения параметров текста |
| getviewsettings | void far getviewsettings (struct viewporttype far *viewport); Возвращает в структуре <i>viewport</i> значения параметров окна экрана |
| getx | int far getx(void); Возвращает целое значение координаты <i>x</i> текущей позиции на экране |
| gety | int far gety(void); Возвращает целое значение координаты <i>y</i> текущей позиции на экране |
| graphresult | int far graphresult(void); Возвращает номер ошибки графической операции (целое число от -15 до -1); значение 0 говорит об отсутствии ошибок |
| grapherrormsg | char *far grapherrormsg(int errorcode); Возвращает указатель на строку, содержащую описание ошибки номер <i>errorcode</i> |

| Функция | Прототип и краткое описание действий |
|------------|--|
| imagesize | unsigned far imagesize (int left, int top, int right, int bottom); Возвращает объем буфера, нужного для сохранения графической информации в окне экрана с заданными вершинами |
| textheight | int far textheight(char far *textstring); Возвращает целое значение высоты в пикселах символов из строки textstring |
| textwidth | int far textwidth (char far *textstring); Возвращает в пикселах целое значение ширины строки символов textstring |

Список рекомендуемой литературы

1. Петров А.В., Алексеев В.Е., Ваулин А.С. и др. Вычислительная техника и программирование: Учеб. для технических вузов. М.: Высш. шк., 1990. 479 с.
2. Алексеев В.Е., Ваулин А.С., Петрова Г.Б. Вычислительная техника и программирование: Практ. пособие. М.: Высш. шк., 1991. 400 с.
3. Поляков Д.Б., Круглов И.Ю. Программирование в среде Турбо Паскаль (версия 5.5). М.: Изд-во МАИ, 1992. 576 с.
4. Агабеков Л.Е., Борисов С.В., Ваулин А.С. Программирование в среде Турбо Паскаль: Практ. пособие (Инструментальные средства персональных ЭВМ: В 10 кн.). Кн. 4. М.: Высш. шк., 1993. 142 с.
5. Артемьев И.Т., Новикова С.В. Программирование на языке TURBO PASCAL: Лаб. практикум/Чуваш.ун-т. Чебоксары, 2000. 160с.
6. Подбельский В.В., Фомин С.С. Программирование на языке Си: Учеб. пособие. М.: Финансы и статистика, 2002. 600 с.
7. Подбельский В.В. Язык Си++: Учеб. пособие. М.: Финансы и статистика, 2002. 560 с.

Оглавление

| | |
|---|----|
| Общие указания..... | 2 |
| Лабораторная работа 1. Программирование алгоритмов линейной структуры..... | 3 |
| Лабораторная работа 2. Программирование алгоритмов разветвляющейся и циклической структуры | 5 |
| Лабораторная работа 3. Программирование алгоритмов со структурой вложенных циклов | 7 |
| Лабораторная работа 4. Обработка литерных величин | 10 |
| Лабораторная работа 5. Программирование | |
| с использованием функций | 18 |
| Лабораторная работа 6. Программирование с использованием процедур..... | 21 |
| Лабораторная работа 7. Построение графиков функций | 26 |
| Лабораторная работа 8. Обработка текстовых файлов..... | 30 |
| Лабораторная работа 9. Обработка записей | 34 |
| Лабораторная работа 10. Обработка файлов из записей | 38 |
| Лабораторная работа 11. Перечисляемые и ограниченные типы. Оператор варианта..... | 42 |
| Лабораторная работа 12. Множественные типы | 45 |
| Лабораторная работа 13. Создание движущихся графических изображений | 48 |
| Лабораторная работа 14. Списки | 55 |
| Справочник | 63 |
| <i>Язык Паскаль</i> | 63 |
| <i>Язык Си</i> | 75 |
| Список рекомендуемой литературы..... | 97 |

ПРОГРАММИРОВАНИЕ НА АЛГОРИТМИЧЕСКИХ ЯЗЫКАХ

Лабораторный практикум

Отв. за выпуск М.А. Титова

Подписано в печать 16.09.05. Формат 60×84 /16. Бумага газетная.
Гарнитура Times. Печать офсетная. Усл. печ. л. 5,81. Уч.-изд. л.5,70
Тираж 300 экз. Заказ №553

Чувашский государственный университет
Типография университета
428015 Чебоксары, Московский просп., 15