

**Паскаль –
язык структурного
программирования**

СОДЕРЖАНИЕ

Структура программы

Алфавит языка

Типы данных

**Операторы
присваивания,
ввода и вывода**

**Арифметические
операции и операции
отношения**

Бинарные операции

**Стандартные
математические функции**

**ПРАКТИЧЕСКАЯ РАБОТА
1**

Логические операции

**Программирование
ветвлений**

**ПРАКТИЧЕСКАЯ РАБОТА
2**

Оператор выбора Case

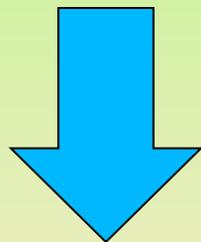
**ПРАКТИЧЕСКАЯ РАБОТА
3**

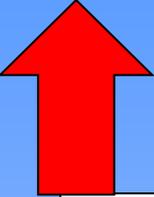
Программирование цикла

**ПРАКТИЧЕСКАЯ РАБОТА
4**

Вложенные циклы

**ПРАКТИЧЕСКАЯ РАБОТА
5**





Итерационные циклы

**ПРАКТИЧЕСКАЯ РАБОТА
6**

Подпрограммы

Массивы

**ПРАКТИЧЕСКАЯ РАБОТА
7**

**Организация ввода и
вывода данных с
использованием файлов**

**ПРАКТИЧЕСКАЯ РАБОТА
8**

Символьный тип данных

ПРАКТИЧЕСКАЯ РАБОТА 9

Строки символов

**ПРАКТИЧЕСКАЯ РАБОТА
10**

**Комбинированный тип
данных**

**ПРАКТИЧЕСКАЯ РАБОТА
11**

**ИТОГОВАЯ
ПРАКТИЧЕСКАЯ РАБОТА**

Структура программы

```
Program < имя программы >;  
Uses <раздел библиотек >;  
Label <раздел меток >;  
Const <раздел констант >;  
Type <раздел типов >;  
Var <раздел переменных >;  
Procedure (Function) <раздел подпрограмм >;  
Begin  
    <раздел операторов >;  
End.
```



Алфавит языка

Латинские буквы: от A до Z и от a до z

Цифры: 0,1,2,3,4,5,6,7,8,9

Специальные символы: + - * / = <> . , { } @ \$ #

Пробелы

Служебные слова: Program, Var, if и другие

Идентификаторы – Любая последовательность букв и цифр, начинающаяся с буквы.

Комментарии

{любой текст, не содержащий символы *фигурная скобка* }

(любой текст, не содержащий символы *звездочка и круглая скобка*)

Замечание

Для комментариев можно применять русский алфавит



Система типов данных Паскаля



Типы данных

Имя типа	Длина в байтах	Диапазон значений	Десятичных цифр в мантиссе
Целочисленные типы			
Integer	2	-32768..32767	
Byte	1	0..255	
Word	2	0..65535	
Shorting	1	-128..127	
Longint	4	-2147483648..2147483647	

Типы данных

Имя типа	Длина в байтах	Диапазон значений	Десятичных цифр в мантиссе
Вещественные типы			
Real	6	$2,9 \cdot 10^{-39}.. 1,7 \cdot 10^{38}$	11-12
Single	4	$1,5 \cdot 10^{-45}... 3,4 \cdot 10^{38}$	7-8
Double	8	$5 \cdot 10^{-324}.. 1,7 \cdot 10^{308}$	15-16
extended	10	$3,4 \cdot 10^{-4932}... 1,1 \cdot 10^{4932}$	19-20
Логический тип			
Boolean	2	True, False	
Символьный тип			
Char	1	Все символы 8 разрядной кодировки	

Перечисленный тип

Задается непосредственно перечислением всех значений, которые может принимать переменная данного типа

```
Type < имя типа > = (<список значений >)
```

пример

```
Type Gas = (C, O, N, F);  
      Metal = ( Fe, Co, Na, Gu, Zn);  
Var   G1, G2, G3: Gas;  
      Met1, Met2 : Metal;  
      Day: (Sun, Mon, Tue, Wed, Thu, Fri, Sat);
```

```
if Day=Sun Then Write ('Сегодня выходной!')
```

Ограниченный тип

Задается как упорядоченное ограниченное подмножество некоторого порядкового типа

```
<константа1>..<константа2>
```

пример

```
Type Numbers = 1..31  
      Alf = 'A'..'Z';  
Var   Data: Numbers;  
      Bukva: Alf;
```



Ввод данных

Read (<список ввода>)

Readln (<список ввода>)

Вывод данных

Write(<вывод данных>)

Writeln (<вывод данных>)

Оператор присваивания

<Название переменной> := <значение>



Арифметические операции

- унитарный минус
- + сложение
- вычитание
- * умножение
- / деление

Операции отношения

- = равно
- <> неравно
- > больше
- < меньше
- >= больше или равно
- <= меньше или равно



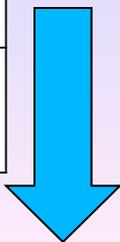
Бинарные операции Паскаля

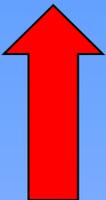
Знак	Выражение	Типы операндов	Тип результата	Операции
+	$A+B$	R,R I,I I,R; R,I	R I R	Сложение
-	$A-B$	R,R I,I I,R; R,I	R I R	Вычитание
*	$A*B$	R,R I,I I,R; R,I	R I R	Умножение
/	A/B	R,R I,I I,R; R,I	R I R	Вещественное деление
div	$A \text{ div } B$	I,I	I	Целочисленное деление



Стандартные математические функции Паскаля

Обращение	Тип аргумента	Тип результата	Функция
PI		R	Число $\pi=3,1415926536E+00$
abs(x)	I, R	I, R	Модуль аргумента
arctan(x)	I, R	R	Арктангенс (в радианах)
cos(x)	I, R	R	Косинус (в радианах)
exp(x)	I, R	R	e^x - экспонента
frac(x)	I, R	R	Дробная часть x
int(x)	I, R	R	Целая часть x
ln(x)	I, R	R	Натуральный логарифм





Стандартные математические функции Паскаля

Обращение	Тип аргумента	Тип результата	Функция
<code>random</code>		R	Псевдослучайное число в интервале [0, 1)
<code>random(x)</code>	I	I	Псевдослучайное число в интервале [0, x)
<code>round(x)</code>	R	I	Округление до ближайшего целого
<code>sin(x)</code>	I,R	R	Синус (в радианах)
<code>sqr(x)</code>	I, R	R	квадрат x
<code>sqrt(x)</code>	I, R	R	Корень квадратный x
<code>trunc(x)</code>	R	I	Ближайшее целое, не превышающее x по модулю



ЗАДАЧА

Найти радиусы вписанной и описанной окружностей в правильный треугольник, если известна сторона треугольника

Формализация

Воспользуемся формулами

$$R_O = \frac{a\sqrt{3}}{3}$$

$$R_V = \frac{a\sqrt{3}}{6}$$

```
Program radius;
```

```
Uses crt;
```

```
Var a, ro, rv: real;
```

```
Begin
```

```
  writeln ('введите сторону треугольника');
```

```
  read (a);
```

```
  ro:= a*sqrt(a)/3;
```

```
  rv:= a*sqrt(a)/6;
```

```
  writeln( ' радиус описанной окружности равен ', ro);
```

```
  writeln( ' радиус вписанной окружности равен ', rv);
```

```
  readkey
```

```
end.
```



Задачи для практической работы № 1

Вычислить значение выражения

$$z = \frac{|x| - |y|}{1 + |xy|}$$

$$z = \ln|(y - \sqrt{|x|}) \left(x - \frac{y}{2 + x^2 - 5x} \right)|$$

```
Program Praktik11;
```

```
Var x, y, z: real;
```

```
Begin
```

```
  read (x, y);
```

```
  z:=( abs(x) - abs(y)) /(1+
```

```
  abs(x*y));
```

```
  write (z)
```

```
end.
```

```
Program Praktik12;
```

```
Var x, y, z: real;
```

```
Begin
```

```
  read (x, y);
```

```
  z:=ln(abs((y-sqrt(abs(x)))*(x-y/ (2 +
```

```
  sqr(x)-5*x)))));
```

```
  abs(x*y));
```

```
  write (z)
```

```
end.
```



Логические операции

Конъюнкция (и) & или \wedge AND

Дизъюнкция (или) \vee OR

Отрицание (не) \neg NOT

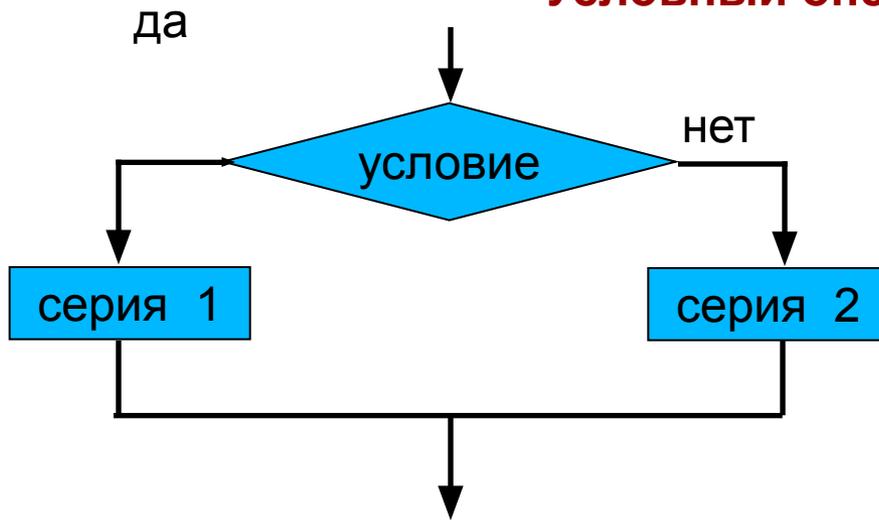
Исключающее или XOR

A	B	Not A	A and B	A or B	A xor B
T	T	F	T	T	F
T	F	F	F	T	T
F	T	T	F	T	T
F	F	T	F	F	F



Программирование ветвлений

Условный оператор IF

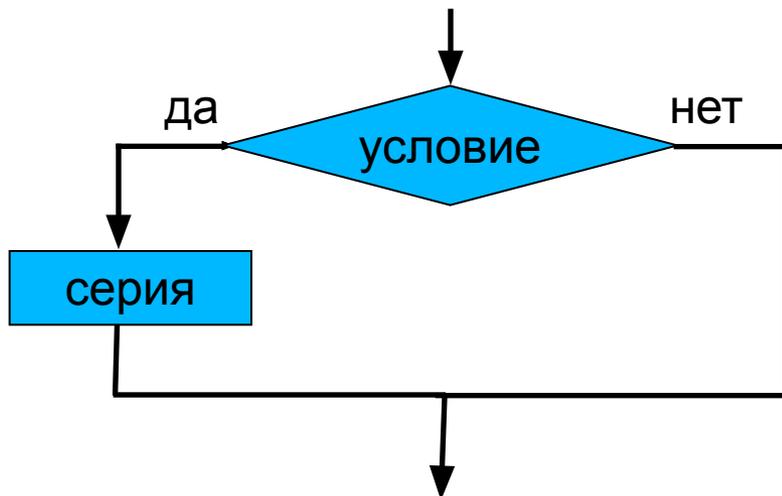


Полное ветвление

If <логическое условие>

Then <оператор 1>

Else < оператор 2 >



Неполное ветвление

If <логическое условие>

Then <оператор >

Задача

Решить квадратное уравнение

Формализация

$$Ax^2+Bx+C=0$$

$$D=B^2 - 4AC$$

если $D>0$

$$x_1 = \frac{-B + \sqrt{D}}{2A} \quad x_2 = \frac{-B - \sqrt{D}}{2A}$$

$D=0$

$$x = \frac{-B}{2A}$$

$D<0$ корней нет

Program kvuravn;

Uses crt;

Var a, b, c, d, x1, x2: real;

Begin

Writeln('Введите коэффициенты квадратного уравнения');

Read (a, b, c);

d:=sqr(b) – 4*a*c;

if d > 0 then begin

x1:=(-b+sqrt(d))/(2*a);

x2:=(-b-sqrt(d))/(2*a);

writeln ('2 корня', x1, x2);

end

else if d = 0 then begin

x1:=-b/(2*a);

writeln ('корень один', x1);

end

else write (' корней нет');

readkey;

end.



Задача для практической работы № 2

Найти значение следующей функции

$$y = \left\{ \begin{array}{ll} 2, & \text{если } x < 0 \\ x^2 + 2 & \text{если } 0 \leq x < 4 \\ \sqrt{x} & \text{если } x \geq 4 \end{array} \right\}$$

```
Program Praktik2;  
Uses crt;  
Var x,y: real;  
Begin  
  Write ('Введите x ');  
  Read(x);  
  If x<0 then y:=2 else  
  If x<4 then y:=sqr(x)+2 else y:=sqrt(x);  
  Write('y=',y:5:3);  
  Readkey  
End.
```



Программирование ветвлений

Оператор выбора Case

```
Case <селектор> of  
  <список констант 1>:<оператор1>;  
  ...  
  <список констант N>:<оператор N>;  
Else <оператор>  
End
```

Задача

Составить программу которая сообщает, сдал студент экзамен или не сдал.

Формализация

Если оценка одна из следующих 3, 4, 5, то экзамен сдан; если 2, то не сдан.

```
Program stydent;  
Uses crt;  
Var n:integer;  
Begin  
  WriteLn ('введите оценку');  
  Read (n);  
  Case n of  
    3, 4, 5: WriteLn ('Экзамен сдан');  
    2: WriteLn ('Экзамен не сдан');  
  Else WriteLn ('Нет такой оценки')  
  End;  
Readkey  
End.
```



Задача для практической работы № 3

По номеру года определить его название в старом японском календаре

Формализация

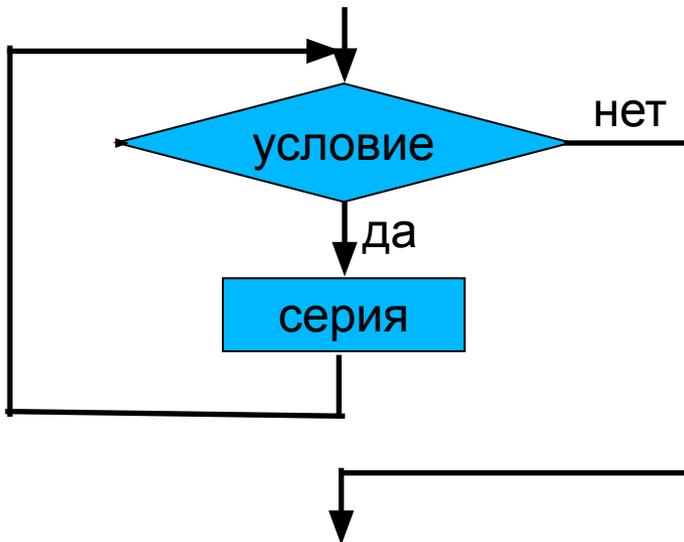
Началом цикла будем считать 1996 год — год крысы. Чтобы вывести название нужно анализировать остаток от деления номера года на 12. Название годов - крыса, корова, тигр, заяц, дракон, змея, лошадь, овца, обезьяна, петух, собака и свинья.

```
Program praktik3;
Uses crt;
Var Year : Integer;
Begin
  WriteLn('Введите номер года ');
  ReadLn(Year);
  Case (Year mod 12) Of
    0: WriteLn('Год Обезьяны');
    1: WriteLn('Год Петуха');
    2: WriteLn('Год Собаки');
    3: WriteLn('Год Свиньи');
    4: WriteLn('Год Крысы');
    5: WriteLn('Год Коровы');
    6: WriteLn('Год Тигра');
    7: WriteLn('Год Зайца');
    8: WriteLn('Год Дракона');
    9: WriteLn('Год Змеи');
    10: WriteLn('Год Лошади');
    11: WriteLn('Год Овцы')
  End;
  Readkey
End.
```



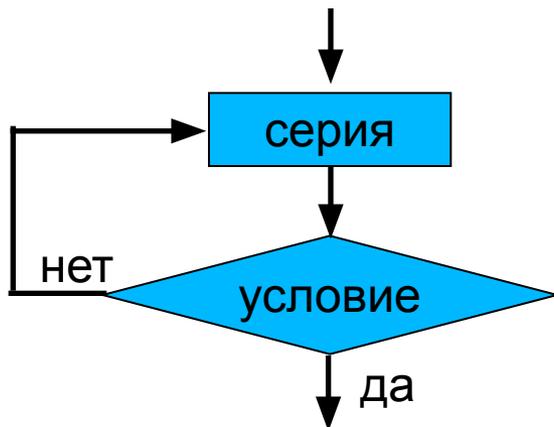
Программирование цикла

Цикл с предусловием (цикл- пока)



While <логическое выражение>
Do <оператор>;

Цикл с постусловием (цикл- до)



Repeat <оператор>
Until <логическое выражение>;

Программирование цикла

Цикл с параметром

For <параметр цикла>:<выражение 1> **To** <выражение 2> **Do**
<оператор>;

For <параметр цикла>:<выражение 1> **Downto** <выражение 2> **Do**
<оператор>;

Задача

Составить программу проверки знаний учеником таблицы умножения.

Формализация

Компьютер задает ученику 10 вопросов на умножение чисел от 2 до 9. На каждое задание ученик вводит свой ответ, компьютер сообщает верный ответ.

Подсказка: Нужно использовать функцию **random(x)**

Program test;

Var x,y,i,z: integer;

Begin

for i:=1 to 10 **do**

begin

 x:= random(8) +2;

 y:= random(8) +2;

writeln ('Сколько будет',x,'*',y,'?');

read (z);

if z=x*y

then **writeln** ('Правильно!')

else **writeln** ('Неправильно! ', x,'*' ,y ,'=' ,x*y);

end;

end.



Задача для практической работы № 4

1 вариант

Составить на Паскале программу для вычисления суммы ряда:

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

```
Program Praktik41;  
Uses crt;  
Var n,i:integer;s:real;  
Begin  
  Write('введите значение N');  
  Read (n);  
  s:=1;  
  For i:=2 to n do  
    S:=s+(1/sqr(n));  
  Write('сумма равна 's:8:2);  
  Readkey  
End.
```

2 вариант

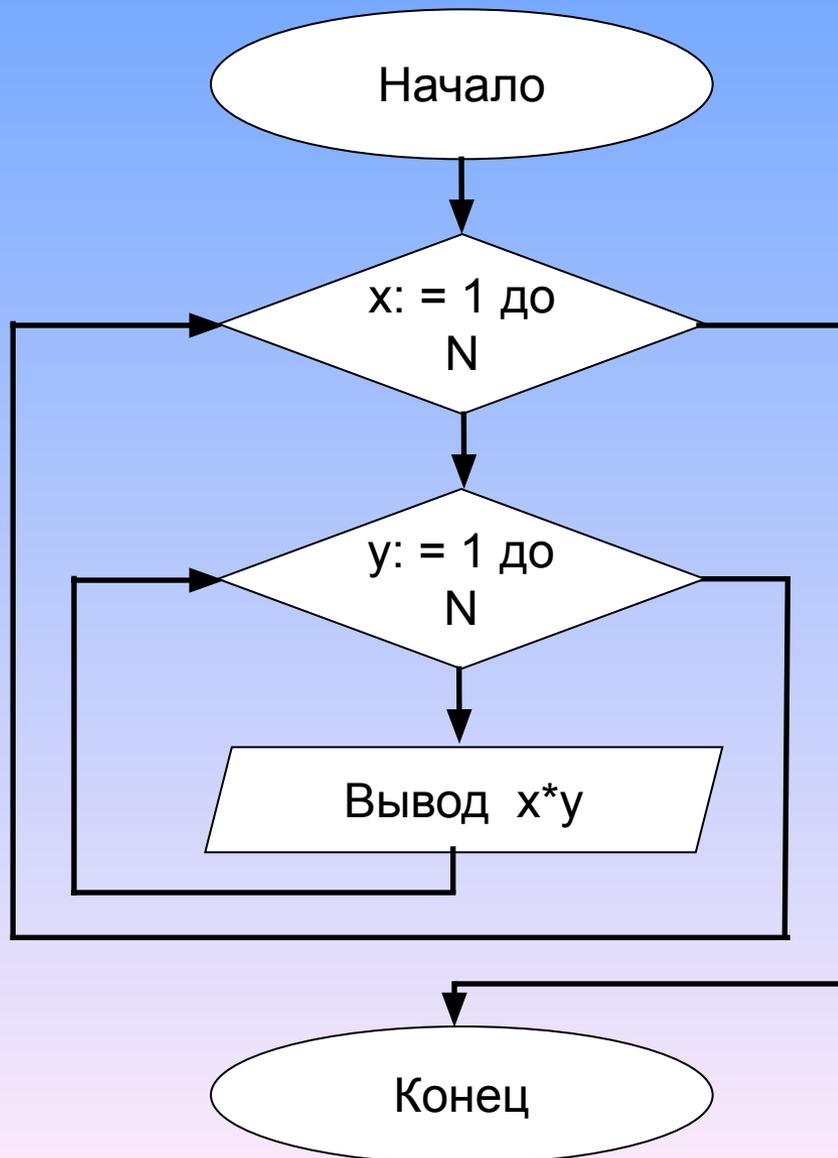
Составить на Паскале программу для вычисления суммы ряда

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

```
Program Praktik42;  
Uses crt;  
Var n,i:integer;s:real;  
Begin  
  Write('введите значение N');  
  Read (n);  
  s:=1;  
  For i:=2 to n do  
    S:=s+(1/sqrt(n));  
  Write('сумма равна 's:8:2);  
  Readkey  
End.
```



Вложенные циклы - если в теле одного цикла имеется другой цикл



x=1; y=1,2,3, ...,N
x=2; y=1,2,3, ...,N
...
x=N; y=1,2,3, ...,N

Задача

Составить таблицу умножения

```
Program tablumn;  
Uses crt;  
Var x, y: 1..9;  
Begin  
  For x:=1 to 9 do  
    Begin  
      Writeln;  
      For y:=1 to 9 do  
        Write (x*y:3)  
      End;  
    End;  
  Readkey;  
End.
```



Задача для практической работы № 5

Сколько можно купить быков, коров и телят на 100 руб., если в сумме должно быть куплено 100 голов скота, а цена быка – 10 руб., цена коровы – 5 руб., цена теленка – 0,5 руб.?

формализация

A – количество быков

B – количество коров

C- количество телят

$$A + B + C = 100$$

$$10*A + 5*B + 0,5 C = 100$$

```
Program Praktik5;
```

```
Uses crt;
```

```
Var a, b, c: integer;
```

```
Begin
```

```
  for a:= 1 to 10 do
```

```
    for b:=1 to 100 do
```

```
      for c:=1 to 100 do
```

```
        if (a+b+c = 100) and
```

```
          (10*a + 5*b + 0.5*c) =100 then
```

```
            writeln (a,' быков ', b,' коров ',c,'телят');
```

```
Readkey;
```

```
end.
```



Итерационный цикл – это цикл, для которого число повторений тела цикла заранее неизвестно

Задача

Составить программу которая считает значение e^x по формуле

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!} + \dots$$

В сумму нужно включить только слагаемые, значение которых больше некоторой малой величины ε (эпсилон).

```
Program exponenta;  
Var exp, x, an, eps: real; i: integer;  
Begin  
  Write ('Введите x='); Read (x);  
  Write ('eps='); Read (eps);  
  exp:=0; i:=0; an:=1;  
  While an>eps do  
  Begin  
    Exp:=exp+an;  
    i:=i+1;  
    an:=(an*x)/i;  
  End;  
  Writeln('exp=',exp)  
End.
```



Задача для практической работы № 6

Составить программу определения количества четных и нечетных цифр в записи данного натурального числа?

Подсказка.

Сначала нужно отделить последнюю цифру. Потом проверить четная она или нечетная (использовать операцию **mod**).

Затем отбросить последнюю цифру (использовать операцию **div**). Процесс повторить.

```
Program Praktik6;
Uses crt;
var x,y: Longint; chet,nechet:word;
begin
  Write ('Введите целое число ');
  Readln (x);
  Chet:=0;
  Nechet:=0;
  While (x>0) do
    begin
      Y:= x mod 10;
      if Y mod 2=0 then chet:=chet+1 else
nechet:=nechet+1;
      x:=x div 10
    end;
  Writeln ('число четных чисел равно ',chet);
  writeln ('число нечетных чисел равно ',nechet);
  Readkey
End.
```



Подпрограммы

Процедуры

Результат – *любое число величин*

Описание :

Procedure <имя программы>[(список формальных параметров)];
<блок>

Обращение – *оператор процедуры*

<имя процедуры>{(список фактических параметров)}

Функции

Результат- *одна величина*

Описание:

Function<имя функции> [(список формальных параметров)]: <тип функции>; <блок>

Обращение - *операнд выражения*

<имя функции>(<список фактических параметров>)

Параметры подпрограмм

Параметры - *переменные*

Описание:

Var <список переменных>:<тип>

Фактические переменные:
переменные

Параметры - *значения*

Описание :

<список переменных>:<тип>

Фактические переменные:
выражения

Задача

По координатам вершин треугольника вычислить его периметр, используя подпрограмму вычисления длины отрезка между двумя точками.

Формализация

Если точка $A(x_1; x_2)$ и $B(y_1; y_2)$ то

$$AB = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

```

Program perimetr1;
Uses crt;
Var a1,a2,b1,b2,c1,c2,p:real;
Procedure dlina (x1,x2,y1,y2: real;var
p:real);
begin
  p:=p+sqrt(sqr(x1-y1)+sqr(x2-y2));
end;
begin
  Write('Введите координаты точки A ');
  Read (a1,a2);
  Write(' Введите координаты точки B ');
  Read (b1,b2);
  Write(' Введите координаты точки C ');
  Read (c1,c2);
  dlina (a1,a2,b1,b2,p);
  dlina (a1,a2,c1,c2,p);
  dlina (b1,b2,c1,c2,p);
  Write ('Периметр равен ',p:10:3);
  Readkey
End.

```

```

Program perimetr2;
Uses crt;
Var a1,a2,b1,b2,c1,c2,p:real;
Function pr (x1,x2,y1,y2: real):real;
begin
  pr:=pr+sqrt(sqr(x1-y1)+sqr(x2-y2));
end;
begin
  Write('Введите координаты точки A ');
  Read (a1,a2);
  Write(' Введите координаты точки B ');
  Read (b1,b2);
  Write(' Введите координаты точки C ');
  Read (c1,c2);
  pr (a1,a2,b1,b2);
  pr (a1,a2,c1,c2);
  p:=pr(b1,b2,c1,c2);
  Write ('Периметр равен ',p:10:3);
  Readkey
End.

```



Массивы

Переменная величина представляющая собой совокупность пронумерованных однотипных величин.

Описание массива

`Var <идентификатор>: array [<тип индекса>] of <тип компонентов>`

`Var T: array [1 ..10] of real`

Элемент массива идентифицируется в виде переменной с индексами

`T[5], T[k], T[i+j], T[p mod r]`

Тип индекса может быть любым скалярным порядковым типом, кроме *Integer*

`Var cod: array [Char] of 1..100`

`L: array [Boolean] of Char`

`cod['y']; L[False]; cod[chr(33)];`

Описание массива

В качестве индекса можно использовать перечисленный тип

```
Type Index = (A, B, C, D);
```

```
Var class 10: array[ Index ] of Byte;
```

Структурному типу присваивается имя в разделе типов, которое затем используется в разделе описания переменных.

```
Type Mas1 = array [1..100] of Integer;
```

```
    Mas2 = array [-10..10] of Char;
```

```
Var Num: Mas1; Sim: Mas2;
```

Многомерный массив

Двумерный массив представляет собой таблицу

1	2	3	4
2	3	3	3
3	4	3	4
4	5	6	5

Описание массива

Var <идентификатор>: array [<тип индекса>] of array [<тип индекса>] of <тип компонентов>

Var Tab1: array [1 ..4] of array [1 ..4] of real

Var Tab1: array [1 ..4,1 ..4] of real

Трёхмерный массив

Var Tab1: array [1 ..n,1 ..m, 1 ..k] of real

Замечание: Не допускается употребление динамических массивов, т. е. таких, размер которых определяется в процессе выполнения.

Действия над массивом как единым целым

- Присваивание значений одного массива другому
- Применение к массивам операции «равно» или «не равно»

Замечание : тип индексов и тип элементов должны быть одинаковы у массивов

```
Var X, Y: array[1..5, 1..4] of integer;  
X:=Y;
```

Элементы массива X станут равными соответствующим элементам массива Y

```
Var X, Y: array[1..5, 1..4] of integer;  
X[3]:=Y[4];
```

Элементы 3 строки массива X станут равными соответствующим элементам 4 строки массива Y

```
Var X, : array[1..4] of integer;  
      Y: array[1..5, 1..4] of integer;  
X:=Y[4];Y[4]:=Y[2]; Y[2]:=X
```

Элементы 4 строки массива Y поменяны местами с элементами 2 строки массива Y

Ввод и вывод массивов

Одномерный

```
For I:=1 To 12 Do  
  Read ( T[i] );
```

```
For I:=1 To 12 Do  
  Write ( T[i]:5:2);
```

Двумерный

```
For I:=1 To Imax Do  
  For J:=1 To Jmax Do  
    Readln( T[I,j]);
```

Значение вводится с
НОВОЙ СТОКИ

```
For I:=1 To Imax Do  
  Begin  
    For J:=1 To Jmax Do  
      Write ( T[I,j]:6:2);  
      Writeln  
    End;
```

Построчный вывод матрицы

Задача

Дан массив из N действительных чисел. Посчитать сколько в нем отрицательных, положительных и нулевых элементов.

```
Program Masiv1;  
Uses crt;  
Var t: array [1 ..10] of real; i,pol,otr,nol: integer;  
Begin  
  Writeln ('введите 10 элементов массива');  
  For l:=1 To 10 Do Read ( t[l] );  
  pol:=0; otr:=0;nol:=0;  
  for i:= 1 to 10 do  
    if t[i]>0 then pol:=pol+1 else  
      if t[i]<0 then otr:=otr+1 else nol:=nol+1;  
  Writeln ('число положительных элементов ',pol);  
  Writeln ('число отрицательных элементов ',otr);  
  Writeln ('число нулевых элементов ',nol);  
  Readkey  
end.
```



Задача для практической работы № 7

Матрицу размером 5×10 заполните случайными двоичными цифрами (0 и 1). Вывести матрицу. Определите номер строки с наибольшим количеством нулей и их количество.

Формализация

Считаем количество нулей в первой строке. Запоминаем это количество. Берем следующую строку и считаем количество нулей в ней. Если количество нулей больше, то запоминаем эту строку и количество нулей.

Матрицу размером 5x10 заполните случайными двоичными цифрами (0 и 1). Вывести матрицу. Определите номер строки с наибольшим количеством нулей и их количество.

```
Program Praktik7;  
Uses crt;  
Var matrica: array[1..5,1..10] of integer;  
Var i,j,nmax,k,kmax:integer;  
Begin  
  For i:=1 to 5 do  
    for j:=1 to 10 do  
      matrica[i,j]:=random(2);  
nmax:=1;kmax:=0;  
  For J:=1 to 10 do  
    if matrica[1,j]=0 then kmax:=kmax+1;  
  For i:=2 to 5 do  
    begin  
      k:=0;  
      for j:=1 to 10 do  
        if matrica[i,j]=0 then k:=k+1;
```

```
    if k>kmax then  
      begin  
        kmax:=k;  
        nmax:=i;  
      end;  
    end;  
  For i:=1 to 5 do  
    Begin  
      for j:=1 to 10 do  
        Write (matrica[i,j]:4);  
        Writeln;  
      End;  
    Writeln ('Номер строки ',nmax);  
    Write('Количество нулей ',kmax);  
    Readkey  
  End.
```



Организация ввода данных с использованием файлов

Read или Readln – чтение из файла

Close – закрытие файла

Assign - назначает файловой переменной имя внешнего файла.

Reset - открывает существующий файл.

Описание

Readln (<файловая переменная>, <имя файла>)

Close (<файловая переменная>)

Assign (<файловая переменная>, <имя файла>)

Reset (<файловая переменная>)

Организация вывода данных с использованием файлов

Write или **Writeln** – запись в файл

Close – закрытие файла

Assign - назначает файловой переменной имя внешнего файла.

Rewrite - открывает файл для записи.

Описание

Write (<файловая переменная>, <список вывода>)

Writeln (<файловая переменная>, <список вывода>)

Close (<файловая переменная>)

Assign (<файловая переменная>, <имя файла>)

Rewrite (<файловая переменная>)

Замечание: 1. При выводе, информация которая была в файле удаляется;
2. Русский алфавит не передается;

Задача

В текстовом файле с именем *matrica.txt* записана следующая числовая матрица. Создать массив по правилу: 1 элемент этого массива равен сумме элементов 1 строки матрицы, 2 элемент – сумме 2 строки и т.д. В файл с именем *matrica2.txt* вывести числовую матрицу и полученный массив.

1	6	8	7	4
8	-5	0	9	8
2	4	3	4	7
4	-3	5	-5	2

Формализация

Ввести элементы матрицы в программу из текстового файла .
Посчитать сумму 1 строки и присвоить это значение первому элементу нового массива, затем также для второй строки и т.д.
Вывести его в текстовый файл персональную матрицу и новый массив.

```

Program Matrica;
Var M1: array [1..4,1..5] of integer;
    M2: array [1..4] of integer;
    i,j: byte; s:integer;
    F1: text; {файловая
                Переменная}
Begin
  Assign (F1,'matrica.txt'); {Связывание
                              F1 с файлом matrica.txt}
  Reset(F1); {Открытие файла
              для чтения}
  For i:=1 to 4 do
  Begin
    For j:=1 to 5 do
      Read (F1, M1[i,j]);
{Последовательное чтение из одной
строки}
      Readln(F1) {переход к
                  следующей строке}
    End;
  Close(F1); {Заккрытие файла}

```

```

For i:=1 to 4 do
  Begin
    s:=0;
    For j:=1 to 5 do
      s:=s+M1[i,j];{находим сумму
                    элементов i строки}
    M2[i]:=s
  End;
  Assign (F1,'matrica2.txt');
{Связывание F1 с файлом matrica2.txt}
  Rewrite (F1); {открытие файла
                 для записи}
  For i:=1 to 4 do
  Begin
    For j:=1 to 5 do
      Write (F1, M1[i,j]:4);
      Writeln(F1)
    End;
    Writeln(F1);
    For i:=1 to 4 do Write(F1,M2[i]:4);
    Close(F1); {Заккрытие файла}
  End.

```



Задача для практической работы № 8

Написать программу, по которой из текстового файла с именем *kvur.txt* будут прочитаны три числа a , b , c — коэффициенты квадратного уравнения, затем будут вычислены корни этого уравнения и выведены на экран в текстовый документ *korni.txt*.

```

Program Praktik8;
Uses crt;
Var a, b, c, d, x1, x2: real;
    G: text;
Begin
  Assign (G,'kvur.txt');
  Reset(G);
  Readln (G,a,b,c);
  Close(G);
  d:=sqr(b)-4*a*c;
  if d > 0 then
  begin
    x1:=(-b+sqrt(d))/(2*a);
    x2:=(-b-sqrt(d))/(2*a);
    Assign (G,'korni.txt ');
    Rewrite (G);
    Writeln (G,'2 корня ', x1:7:2, x2:7:2);
    Close(G);
    Writeln ('2 корня ', x1:7:2, x2:7:2)
  end
end

```

```

else if d = 0 then
  begin
    x1:=-b/(2*a);
    Assign (G,'korni.txt ');
    Rewrite (G);
    Writeln (G,'1 корень ', x1:7:2);
    Close(G);
    Writeln ('1 корень ', x1:7:2)
  end
else
  begin
    Assign (G,'korni.txt ');
    Rewrite (G);
    Writeln (G,'нет корней');
    Close(G);
    Write ('нет корней')
  end;
  Readkey
End.

```



Символьный тип данных

Символьная константа записывается между апострофами.
Например 'T', '-', '5', '{'.

Символьный тип - CHAR

Пример:

```
Var c1, c2 : Char;
```

- Символы - упорядоченное множество;
- У каждого символа в этом множестве есть свой порядковый номер;
- Между символами работает соотношение «следующий – предыдущий».

Функция $Ord(X)$ – функция от аргумента порядкового типа, которая возвращает порядковый номер значения x в этом типе данных.

Пример:

```
Ord('A')=65, Ord('5')= 53.
```

$Chr(x)$ – функция от целочисленного аргумента, результатом которой является символ с кодом x .

Пример:

```
Chr(65)='A', Chr(53) ='5'.
```

Задача

С помощью датчика случайных чисел заполнить массив *alfavit* [0..15] прописными латинскими буквами. Затем массив отсортировать в алфавитном порядке.

```
Program Simvol1;
Uses crt;
Var Alfavit: array[1..15] of char;
    s: char; i,j: integer;
Begin
  ClrScr;
  randomize; {Заполнение массива
случайными буквами}
  Writeln('Исходный массив');
  For i:=1 to 15 do
  begin
    Alfavit[i]:= chr(random(26)+ ord('A'));
    Write( Alfavit[i]:3)
  end;
```

```
Writeln;
{Сортировка метод пузырька}
For i:=1 to 15 do
  for j:=1 to 15-i do
    if Alfavit[j]>Alfavit[j+1] then
    begin
      S:=Alfavit[j];
      Alfavit[j]:=Alfavit[j+1];
      Alfavit[j+1]:=s
    end;
  Writeln('Отсортированный массив');
  For i:=1 to 15 do write (Alfavit[i]:3);
  Readkey
End.
```



Задача для практической работы № 9

Вывести на экран символы и их код, согласно таблицы кодировки *ASCII*, начиная с 33 и заканчивая 100. Результат вывести по 10 записей в строке.

```
Program Praktik9;  
Uses crt;  
Var i,k:integer;  
Begin  
  ClrScr;  
  k:=0;  
  For i:=33 to 100 do  
  begin  
    Write(Chr(i),'-',i:3,' ');  
    k:=k+1;  
    if k=10 then  
      begin  
        k:=0;  
        Writeln  
      end  
    end;  
  Readkey  
End.
```



Строки символов

Строка – это последовательность символов.
Каждый символ занимает 1 байт.

Длина строки – количество символов в строке.
Длина может находиться от 1 до 256.

Строковые величины могут быть константами и переменными

Пример

```
'Информатика – наука об информации'  
'23-15-58'
```

Описание `Var <Идентификатор>: String (<максимальная длина строки>)`

Пример

```
Var Name: String (20);  
Name1: String
```

Символ внутри строки индексируются (нумеруются), начиная с единицы

Пример

```
Name[5], Name[j], Name1[i+1]
```

Операции над строковыми данными

Операция сцепления (+)

Пример

'Фамилия ' + 'Имя ' + 'Отчество'
Фамилия Имя Отчество

Операция отношения

=, <, >, <=, >=, <>

Пример

'cost1' > 'cost2'
'pascal' > 'PASCAL'
'Алина' < 'Арина'
'Ключ ' <> 'Ключ'

Функции

Функция **Copy** (S, Poz, N) выделяет из строки S подстроку длиной N символов, начиная с позиции Poz . N и Poz – целочисленные выражения.

Пример

'ABCDEFGFG'

$Copy(S, 2, 3)$ - 'BCD'

$Copy(S, 4, 4)$ – 'DEFG'

Функция **Concat** ($S1, S2, \dots, SN$) выполняет сцепление (конкатенацию) строк $S1, S2, \dots, SN$ в одну строку.

Пример

$Concat('A', 'B', 'C',)$ - 'ABC'

Функция **Length** (S) определяет текущую длину строки S .
Результат – значение целочисленного типа

Пример

' Matrica '

$Length(S)$ - 7

Функции

Функция $Pos(S1, S2)$ обнаруживает первое появление в строке $S2$ подстроки $S1$. Результат – целое число, равное номеру позиции, где находится первый символ подстроки $S1$.

Если в $S2$ не обнаружена подстрока $S1$, то результат равен 0

Пример

$S2 - 'abcdefg' \quad Poz('cd', S2) - 3$
 $Poz('k', S2) - 0$

Процедуры

Процедура $Delete(S, Poz, N)$ удаляет N символов из строки S начиная с позиции Poz .

Пример

$S - 'abcdefg' \quad Delete(S, 3, 2) - 'abefg'$

Процедура $Insert(S1, S2, Poz)$ выполняет вставку строки $S1$ в строку $S2$, начиная с позиции Poz .

Пример

$S2 - 'ЭВМ PC' \quad Insert('IBM-', S2, 5) \quad 'ЭВМ IBM-PC'$

Процедуры

Str(x, st). Процедура преобразует численное выражение **x** в его строковое представление и помещает результат в **st**.

x: арифметическое выражение; **st**: string

Пример

```
x:=1234; y:=5678;.
str(x,stroka1);
str(y,stroka2);
stroka:=stroka1+stroka2;
writeln(stroka);
```

12345678

Val(st, x, code) Процедура преобразует строковую запись числа, содержащуюся в **st**, в числовое представление, помещая результат в **x**, **x** - может быть как целой, так и действительной переменной.

st: string, **x**: числовая переменная; **code**: integer)

Пример

```
stroka1:='23.02.1965';
stroka2:=copy(stroka1,7,4);
val(stroka2,rog);
let:=2012-rog;
```

47

Если после выполнения процедуры **code** равно 0, то это свидетельствует об успешно произошедшем преобразовании.

Задача

Дана строка, заканчивающаяся точкой. Подсчитать, сколько в ней слов.

```
Program Slovo;  
uses crt;  
Var s: string;  
    i,k: integer;  
Begin  
    Writeln ('Введите строку');  
    Read (s);  
    i:=1; k:=k+1;  
    While S[i]<>'.' do  
        Begin  
            if s[i]=' ' then k:=k+1;  
            i:=i+1  
        End;  
    k:=k+1;  
    Write('Количество слов в строке равно ',k);  
    Readkey  
End.
```



Задача для практической работы № 10

В строке заменить все двоеточия (:)
точкой с запятой (;).
Посчитать количество замен.

```
Program Praktik10;  
Uses crt;  
Var s:string; i,k,n:integer;  
Begin  
  Write('Введите строку');  
  write;  
  Read(S);  
  n:=Length(S);k:=0;  
  Writeln('Новая строка');  
  for i:=1 to n do  
    begin  
      if s[i]=':' then  
        begin  
          s[i]=';'; k:=k+1  
        end;  
      write(s[i]);  
    end;  
  Writeln;  
  Write('Количество замен', k);  
  Readkey  
End.
```



Комбинированный тип данных

Это структурный тип, состоящий из фиксированного числа компонентов (полей) разных типов.

Описание

```
Type <имя> = record  
    <имя поля 1>; <тип>;  
    ...  
    <имя поля N>; <тип>  
End
```

Замечание:

Поля могут быть любого типа

Пример:

Экзаменационная
ведомость

```
Type results = record  
    Family: string[15]; {Оценка ученика}  
    Rus: 2..5;           {Оценка по русскому языку}  
    Alg: 2..5;           {Оценка по алгебре}  
    Geom: 2..5;         {Оценка по геометрии}  
End;  
Var exam: results;
```

Величина комбинированного типа называется **записью**. Элементы записи идентифицируются составными именами следующей структуры:

<имя переменной >, <имя поля>

exam.family, exam.rus

Задача

На экзаменационном листе содержатся сведения о результатах экзаменов, сданных 30 учениками класса. Ввести эти данные в компьютер и вывести список отличников.

Исходные данные организуются в массив следующей структуры:

```
Var list:array[1..30] of results;
```

Конечный фрагмент: вывод списка:

```
For i:=1 to 30  
If (list[i].rus=5) and (list[i].alg=5) and (list[i].geom=5)  
then Writeln (list[i].family);
```

Для ввода списка целесообразно использовать тестовой файл следующего вида:

Таблица успеваемости 11 класса

Фамилия	Русский язык	Алгебра	Геометрия
Арефьев	4	3	4
Довгалец	5	5	5
Дунаев	5	4	4
...			

```

Program exzam;
Uses crt;
Type results = record
    Family: string[15];
    Rus: 2..5;
    Alg: 2..5 ;
    Geom: 2..5;
End;
Var list:array[1..30] of results;
    i:integer; F1,F2: text;
Begin
    Assign(F1,'vedom.txt');
        {связывание F1 с файлом
        vedom.txt}
    Assign(F2,'res.txt'); ');{связывание
        F2 с файлом res.txt}
    Reset(F1); {Открытие файла
        F1 для чтения}
    Rewrite(F2); {Открытие файла
        F2 для записи}
    Readln(F1); Readln(F1); {Пропуск
        двух строк}

```

```

For i:=1 to 30 do
    Readln(F1,list[i]. Family,list[i].Rus,
        list[i].Alg, list[i].Geom); {Ввод
        таблицы}
    Writeln('Отличники');
    For i:=1 to 30 do
        If (list[i].rus=5) and (list[i].alg=5)
            and (list[i].geom=5) Then
            Begin
                Writeln (list[i].family);
                {Вывод фамилии на экран}
                Writeln (F2, list[i].family);
                {Запись фамилии в файл }
            End;
    Close(F1); Close(F2);
    Readkey
End.

```

Дополнение

Для упрощения работы с записями предназначен оператор *With*

Описание

With <Имя записи>do<оператор>

Один раз указав переменную типа запись в операторе *With*, можно работать с именами полей как с обычными переменными, т.е. без указания переменной типа запись

Пример:

```
For i:=1 to 30 do  
  Readln(F1,list[i].Family,list[i].Rus,  
    list[i].Alg, list[i].Geom);
```

```
For i:=1 to 30 do  
  With list[i] do  
    Readln(F1,Family,Rus,Alg,Geom);
```

Дополнение

Выше предложенную задачу можно решить без использования массива. А также выяснить число записей в текстовом редакторе

```
Var list:array[1..30] of results;
```

```
Var list: results;
```

```
For i:=1 to 30 do  
  Readln(F1,list[i].Family,  
    list[i].Rus,list[i].Alg,list[i].Geom);
```

```
i:=0;  
While not EOF(F1) do  
Begin  
  Readln(F1,list.Family,list.Rus, list.Alg,  
    list.GeoM);  
  I:=I+1; {Подсчет количества записей}  
End
```

Стандартная логическая функция EOF(end of file) принимает значение *True*, когда процесс чтения из файла дойдет до его конца.

Задача повышенной трудности

Сведения о результатах сессии содержат следующую информацию: Фамилия, имя студента, номер группы, его оценки по итогам 3 экзаменов. Получить отдельные списки отличников, хорошистов, троечников и двоечников в алфавитном порядке. (Фамилия, имя студента, номер группы)..

На вход программы подаются сведения о студентах.

В первой строке указывается количество студентов N , каждая из следующих N строк имеет следующий формат:

<Фамилия>пробел<Имя>пробел<Номер группы>пробел<Оценка за 1 экзамен > пробел <Оценка за 2 экзамен>пробел<Оценка за 3 экзамен>

```

PROGRAM Student;
Uses crt;
Type data=record
    family:string[20];
    name:string[15];
    nomer:integer;
    exz1,exz2,exz3:2..5
end;
var spisok:array[1..50] of data;
    spisok1: data;
    n,k,i,j,l:integer; v:integer;
    s,s1:string;
begin
    writeln('Введите количество
студентов');
    readln(n);
    writeln('Введите информацию о
студенте по форме');
    Writeln('Фамилия имя
номер_группы 1экзамен 2 экзамен 3
экзамен');

```

```

{Ввод данных}
For i:=1 to n do
begin
    readln(s);
    k:=pos(' ',s); {считаем длину 1
слова до пробела}

    spisok[i].family:=copy(s,1,k-1);{копируе
м из строки 1 слово}
    delete(s,1,k);{удаляем из строки 1
слово}
    k:=pos(' ',s);
    spisok[i].name:=copy(s,1,k-1);
    delete(s,1,k);
    k:=pos(' ',s);
    s1:=copy(s,1,k-1);
    val(s1,v);{переводим символ s1 в
число}
    spisok[i].nomer:=v;
    delete(s,1,k);
    k:=pos(' ',s);
    s1:=copy(s,1,k-1);
    val(s1,v);
    spisok[i].exz1:=v;
    delete(s,1,k);

```

```
k:=pos(' ',s);
  s1:=copy(s,1,k-1);
  val(s1,v);
  spisok[i].exz2:=v ;
  delete(s,1,k);
  val(s,v);
  spisok[i].exz3:=v;
end;
{Сортируем списки по возрастанию}
For i:=1 to n do
  for j:=1 to n-i do
    if spisok[j].family > spisok[j+1].family
then
  begin
    spisok1.family:=spisok[j].family;
    spisok1.name:=spisok[j].name;
    spisok1.nomer:=spisok[j].nomer;
    spisok1.exz1:=spisok[j].exz1;
    spisok1.exz2:=spisok[j].exz2;
    spisok1.exz3:=spisok[j].exz3;
    spisok[j].family:=spisok[j+1].family;
    spisok[j].name:=spisok[j+1].name;
    spisok[j].nomer:=spisok[j+1].nomer;
```

```
spisok[j].exz1:=spisok[j+1].exz1;
spisok[j].exz2:=spisok[j+1].exz2;
spisok[j].exz3:=spisok[j+1].exz3;
spisok[j+1].family:=spisok1.family;
spisok[j+1].name:=spisok1.name;
spisok[j+1].nomer:=spisok1.nomer;
  spisok[j+1].exz1:=spisok1.exz1;
  spisok[j+1].exz2:=spisok1.exz2;
  spisok[j+1].exz3:=spisok1.exz3;
end;
Clrscr;
{Выбираем отличников}
  Writeln('Отличники');l:=0;
  For i:=1 to n do
    With spisok[i] do
      if (exz1=5) and (exz2=5) and
(exz3=5) then
```

```
begin
    Writeln(family, ' ', name, ' ',
nomer);l:=l+1;
    end;
    if l=0 then writeln (Отличников
нет');
    Writeln;
    Writeln('Хорошисты');l:=0;
{Выбираем хорошистов}
    For i:=1 to n do
        With spisok[i] do
            if ((exz1=4) and (exz2>=4) and
(exz3>=4) or (exz1>=4) and (exz2=4)
and (exz3>=4) or (exz1>=4) and
(exz2>=4) and (exz3=4)) then
                begin
                    Writeln(family, ' ', name, ' ',
                    nomer);l:=l+1;
                end;
            if l=0 then writeln ('Хорошистов
нет');
```

```
Writeln;
    Writeln('Троечники');l:=0;
{Выбираем троечников}
    For i:=1 to n do
        With spisok[i] do
            if ((exz1=3) and (exz2>=3) and
(exz3>=3) or (exz1>=3) and (exz2=3)
and (exz3>=3) or (exz1>=3) and
(exz2>=3) and (exz3=3)) then
                begin
                    Writeln(family, ' ', name, ' ', nomer);
                    l:=l+1;
                end;
            if l=0 then writeln ('Троечников
нет');
                readkey
            end.
```

Задача для практической работы № 11

Вычислить средний балл учеников класса, если известны оценки каждого ученика по математике, русскому языку и физике.

Выведите на печать фамилию и средний балл лучшего ученика.

```
Program Practic11;
Uses crt;
Type Tabl = record
    Family: string[15];
    Mat: 2..5 ;
    Rus: 2..5;
    Phiz: 2..5;
    Point:real
End;
Var list:array[1..50] of Tabl;
    i,n:integer; F1,F2: text;
    Sr:real;
Begin
    Clrscr;
    Assign(F1,'vedomost.txt');
    Reset(F1);
    Readln(F1); Readln(F1);
    n:=1;
```

```
While not EOF(F1) do
Begin
    Readln(F1,list[n].Family,list[n].Mat, list[n].Rus,
        list[n].Phiz);
    list[n].Point:=(list[n].Mat+list[n].Rus+
        list[n].Phiz)/3;
    n:=n+1
End;
sr:=0;
For i:=1 to n-1 do
    With list[i] do
        If point>= sr then sr:=point;
For i:=1 to n-1 do
    With list[i] do
        If point=sr then
            Writeln(Family:15,'---',point:8:2);
    Readkey
End.
```



Задача для итоговой практической работы

**В массиве хранятся данные об учениках класса: школа, фамилия, класс.
Вывести список учеников, которые учатся в восьмом классе.**

```
Program itogpraktik;
Uses Crt;
Type
Uchenik=record
    Shkola : integer;
    Fam : string[15];
    Klass : integer;
end;
Var
    i,n,a,j : integer;
    Massiv : array[1..100] of Uchenik;
End;
Begin
    ClrScr;
    writeln('Введите число учеников
');
    read(n);
    for i:=1 to n do
        begin
            writeln('Введите через пробел
номер школы и фамилию ученика
');
            with massiv[i] do
```

```
begin
            readln(Shkola,Fam);
            write('Введите класс ученика ');
            read(Klass);
            end;
        end;
    writeln('Ученики 8-ых классов:');
    writeln('Школа   Фамилия       Класс');
    writeln('-----');
    for i:=1 to n do
        if massiv[i].klass=8
            then
                with massiv[i] do
                    writeln(Shkola:6,' ',Fam:15,' ',
                        klass);
                ReadKey;
            End.
```