

ВВЕДЕНИЕ

В качестве основного языка программирования выбран Паскаль, т. к. не требует чрезмерных усилий для усвоения его основных синтаксических конструкций, легко усваивается в процессе практической работы на любом персональном компьютере. Вместе с тем Паскаль обладает большими возможностями для решения широкого круга практических задач и овладение этим языком можно рассматривать как ценный навык алгоритмизации и программирования.

Данное пособие содержит следующие разделы дисциплины "Основы программирования":

Массивы. Линейные массивы

Двумерные массивы

Обработка символьных и строковых величин

Записи и файлы данных

Программирование математических задач

В разделе «Массивы. Линейные массивы» рассматриваются способы описания и формирования линейных массивов, описаны стандартные алгоритмы обработки линейных массивов.

Раздел «Двумерные массивы» включает методы формирования двумерных массивов, решение наиболее типичных задач обработки двумерных массивов.

В разделе «Обработка символьных и строковых величин» рассматриваются типы CHAR и STRING, процедуры и функции обработки строковых данных, приведены примеры решения наиболее типичных задач.

В разделе «Записи и файлы данных» рассматривается тип данных RECORD, описание файловых переменных, процедур и функций для работы с файлами данных, решение задач на запись данных в файлы и чтение данных из файлов.

Раздел «Программирование математических задач» ориентирован на использование знаний, полученных в результате изучения курса «Основы программирования», для решения задач с математическим содержанием.

Основная цель пособия - помочь студентам в проведении самостоятельной работы, которая может быть реализована в форме типового расчета.

К каждому разделу приводятся несколько типовых задач с решениями, что позволяет студентам самостоятельно выполнить типовой расчет.

По этой причине предлагаемое методическое пособие может быть использовано студентами младших курсов физического, математического факультетов для приобретения практических навыков алгоритмизации и программирования, может быть использовано студентами заочниками для самостоятельного овладения языком программирования.

ВАРИАНТЫ ЗАДАНИЙ

№ варианта	Разделы											
	I			II		III			IV		V	
1	1	26	51	1	26	1	26	51	1	26	1	26
2	2	27	52	2	27	2	27	52	2	27	2	27
3	3	28	53	3	28	3	28	53	3	28	3	28
4	4	29	54	4	29	4	29	54	4	29	4	29
5	5	30	55	5	30	5	30	55	5	30	5	30
6	6	31	56	6	31	6	31	56	6	31	6	31
7	7	32	57	7	32	7	32	57	7	32	7	32
8	8	33	58	8	33	8	33	58	8	33	8	33
9	9	34	59	9	34	9	34	59	9	34	9	34
10	10	35	60	10	35	10	35	60	10	35	10	35
11	11	36	61	11	36	11	36	61	11	36	11	36
12	12	37	62	12	37	12	37	62	12	37	12	37
13	13	38	63	13	38	13	38	63	13	38	13	38
14	14	39	64	14	39	14	39	64	14	39	14	39
15	15	40	65	15	40	15	40	65	15	40	15	40
16	16	41	66	16	41	16	41	66	16	41	16	41
17	17	42	67	17	42a	17	42	67	17	42	17	42
18	18	43	68	18	42b	18	43	68	18	43	18	43
19	19	44	69	19	42c	19	44	69	19	44	19	44
20	20	45	70	20	42d	20	45	70	20	45	20	45
21	21	46	71	21	42e	21	46	71	21	46	21	46
22	22	47	72	22	42f	22	47	72	22	47	22	47
23	23	48	73	23	42g	23	48	73	23	48	23	48
24	24	49	74	24	42h	24	49	74	24	49	24	49
25	25	50	75	25	42i	25	50	75	25	50	25	50

I. МАССИВЫ. ЛИНЕЙНЫЕ МАССИВЫ

Массив – это структурированный тип данных (структурированный тип данных определяет упорядоченную совокупность скалярных переменных и характеризуется типом своих компонентов), состоящий из фиксированного числа элементов, имеющих один и тот же тип. Тип элементов массива называется базовым. Число элементов массива фиксируется при описании и в процессе выполнения программы не меняется. Доступ к элементам массива осуществляется путем индексирования элементов массива. Тип индекса определяет границы изменения индекса. Для описания массива предназначено словосочетание *array of* (массив из).

Описание типа

Type <имя типа>=array [<тип индекса>] of <тип компонента>;

Var <идентификатор, ...> : <имя типа>;

Описание может быть и без представления типа в разделе type:

Var <идентификатор>:array[<тип индекса>] of <тип компонента>;

Примеры описаний массивов:

Type

Vektor =array [1..7] of integer;

Massiv =array [1..20] of real;

Var

A,B:vector; X: massiv;

То же в другом формате:

Var A, B : array [1..7] of integer;

X: array [1..20] of real;

Над всем массивом можно выполнять действия $A=B$, $A<>B$, $A:=B$.

Присваивать можно только массивы одинаковых типов. Обращение к отдельному элементу массива производится при помощи указания имени всего массива и в квадратных скобках – индекса конкретного элемента. Например: X[10] - элемент массива X с индексом 10.

Пример1.

Сформировать массив из 20 целых чисел, вводимых с клавиатуры, и вывести элементы массива в обратном порядке.

Program PR1;

Var

A : Array [1..20] Of Integer; I : Integer;

BEGIN

```
For I:=1 To 20 Do      {в цикле перебираем индексы элементов массива}
  Readln(A[I]);      {вводим очередной элемент массива с клавиатуры }
For I:=20 Downto 1 Do  {Распечатываем массив в обратном порядке}
  Write(A[I], ' ')
```

END.

Пример 2.

Сформировать массив из 50 элементов, заполнив его значениями квадратов индексов.

```
Program PR2;
Const N=50;           {Константа N содержит количество элементов массива}
Var
  A : Array [1..N] Of Integer; I : Integer;
BEGIN
  For I:=1 To N Do    {в цикле перебираем индексы элементов массива}
  begin
    A[I]:=I*I;       {формирование очередного элемента массива}
    Write(A[I]:5)    {вывод элементов массива}
  end;
END.
```

Для учебных целей удобнее использовать массивы, сформированные с помощью генератора случайных чисел. В языке Паскаль случайные числа, равномерно расположенными в интервале от 0 до 1, формирует функция **Random**. Для получения целых случайных чисел от 0 до N следует использовать функцию **Random(N+1)**. Чтобы выбрать целое случайное число из интервала **[a, b]** можно применить выражение **Random(b-a+1)+a**.

Пример 3.

Сформировать и распечатать массив из 20 целых случайных чисел от 10 до 100. Найти сумму элементов массива.

```
Program pr3;
Var
  A : Array [1..20] Of Integer; I, S : Integer;
BEGIN
  S:=0; {обнуляем значение суммы}
  For I:=1 To N Do begin
    A[I]:= Random(101)-50; {формирование элементов массива}
    Write(A[I], ' ');     {вывод массива}
    S:=S+A[I];           {суммирование элементов массива}
  end
END.
```

Пример 4.

В массиве хранятся оценки по математике студентов 102 группы. С помощью датчика случайных чисел заполнить массив целыми значениями, лежащими в диапазоне от 2 до 5 включительно. Найти среднюю оценку в группе.

```
Program pr5;
Const n=25;
Var  a:array[1..n] of integer;
     i,s:integer;
```

```

    sr:real;
BEGIN
    s:=0;                                {обнуляем значение суммы}
    for i:=1 to n do begin
        a[i]:=random(4)+2; write(a[i],' '); {формирование и вывод элементов массива}
        s:=s+a[i];                          {суммирование элементов массива}
    end;
    writeln;                               {переход на новую строку}
    sr:=s/n; write('sr=',sr:3:2)
END.

```

Пример 5.

В массиве хранятся цены на 10 видов мороженого. С помощью датчика случайных чисел заполнить массив целыми значениями, лежащими в диапазоне от 3 до 20 включительно. Определить порядковый номер самого дорогого мороженого.

```
Program pr5;
```

```
Var A:array[1..10] of integer;
```

```
    i, n, m: integer; {m- максимальный элемент, n- его порядковый номер}
```

```
BEGIN
```

```
    for i:=1 to 10 do begin {формирование и вывод элементов массива}
```

```
        a[i]:=random(18)+3; write(a[i],' ');
```

```
    end; writeln;
```

```
    m:=a[1]; n:=1; {выбираем 1-й элемент в качестве максимального }
```

```
    for i:=2 to 10 do {перебираем все элементы массива, начиная со 2-го}
```

```
        if a[i]>m then begin m:=a[i]; n:=i end; {запоминаем наибольший и его индекс}
```

```
    write('n=',n)
```

```
END.
```

Пример 6.

В массиве хранится возраст 15 человек. С помощью датчика случайных чисел заполнить массив целыми значениями, лежащими в диапазоне от 16 до 30 включительно. Найти количество человек моложе 25 лет.

```
Program pr6;
```

```
Var A:array[1..15] of integer;
```

```
    i, k : integer; {k - количество человек моложе 25 лет }
```

```
BEGIN
```

```
    for i:=1 to 15 do begin {формирование и вывод элементов массива}
```

```
        a[i]:=random(15)+16; write(a[i],' ');
```

```
    end;
```

```
    writeln; k:=0;
```

```
    for i:=1 to 15 do
```

```
        if a[i]<25 then k:=k+1; {подсчет количества человек моложе 25 лет }
```

```
    write('k=',k)
```

```
END.
```

Пример 7.

Массив A[1..9] заполняется числами, вводимыми с клавиатуры. Найти среднее арифметическое элементов, которые больше 12 и меньше 25.

Program pr7;

Var A: array[1..9] of integer;

i, k, s: integer;

BEGIN

for i:=1 to 9 do begin *{формирование массива}*

write('a['*i*,']='); readln(a[i])

end;

writeln; k:=0; s:=0;

for i:=1 to 9 do

if (a[i]<25) and (a[i]>12) then begin k:=k+1; s:=s+ a[i]; end;

if k>0 then writeln('среднее=',s/k:4:2) else writeln('Искомых элементов нет');

readln;

END.

Пример 8.

Сформировать массивы A и B, содержащие по 10 целых случайных чисел. Найти сумму элементов каждого массива (создать процедуру для формирования массива и определения суммы его элементов).

Program pr8;

Type massiv=array[1..10] of integer;

{процедура, формирующая массив и определяющая сумму его элементов}

Procedure SumMas(Var z: massiv; Var Sz: integer);

Var

i: integer;

Begin

Sz:=0; for i:=1 to 4 do begin

z[i]:=random(20); write(z[i]:4);

Sz:=Sz+z[i];

end;

End;

Var

a, b :massiv;

Sa, Sb: integer;*{Sa- сумма элементов массива a, Sb – сумма элементов массива b}*

BEGIN *{основная программа}*

Write('Массив A:'); SumMas(a,Sa); writeln;

Writeln('Сумма элементов массива A =', Sa);

Write(('Массив B:'); SumMas(b,Sb); writeln;

Writeln(' Сумма элементов массива B =',Sb);

Readln;

END.

Пример 9.

Задан массив $A[1..10]$. Сформировать два массива, включая в первый четные элементы исходного массива, а во второй - нечетные.

Program Pr9;

Const n=10;

Var a, x, y: array[1..n] of integer;

i, k, m: integer; {*k*- количество элементов массива *X*, *m*- количество элементов массива *Y*}

BEGIN

randomize;

writeln('Массив A:');

for i:=1 to n do begin {*формирование и вывод элементов массива A*}
a[i] := random(51); Write(a[i]:4);

end;

writeln; k:=0; m:=0;

for i:=1 to n do

{*если очередной элемент массива A четный, то заносим его в массив X, иначе - в массив Y*}

if a[i] mod 2=0 then begin k:=k+1; x[k]:=a[i]; end

else begin m:=m+1; y[m]:=a[i]; end;

writeln('Массив X'); for i:=1 to k do write(x[i]:4); {*вывод массива X*}

writeln; writeln('Массив Y'); for i:=1 to m do write(y[i]:4); {*вывод массива Y*}

END.

Задания для самостоятельной работы

1. Сформировать массив из 15 целых чисел, выбранных случайным образом из интервала $[-10, 30]$. Найти среднее арифметическое положительных элементов.
2. Сформировать массив из 10 целых чисел, выбранных случайным образом из интервала $[10, 50]$. Найти максимальный среди элементов с четными индексами.
3. Сформировать массив из 15 целых чисел, выбранных случайным образом из интервала $[10, 90]$. Поменять местами первый и минимальный элементы.
4. Задан одномерный массив $A[1..20]$. Найти минимальный элемент среди элементов массива с n -го по k -й (n и k вводятся с клавиатуры)
5. Массив содержит $2n$ чисел. Из суммы первых n его элементов вычесть сумму последних n элементов.
6. Заменить отрицательные элементы в числовом массиве из n чисел ($n > 10$) их квадратами, оставив остальные без изменения.
7. В заданном массиве найти среднее арифметическое положительных чисел, среднее арифметическое отрицательных чисел и число нулей.
8. В массиве из $2n$ чисел найти сумму квадратов элементов с четными индексами и сумму кубов элементов с нечетными индексами.
9. В заданном массиве найти максимальный элемент. Элементы, стоящие после максимального элемента заменить нулями.

10. В заданном массиве поменять местами наибольший и наименьший элементы.

11. Задан массив положительных вещественных чисел. Вычислить значения функции $y=0,5x$ при значениях аргумента, заданных в исходном массиве, и поместить их в другой массив. Вывести на экран дисплея оба массива в виде двух столбцов.

12. В одномерном массиве целых чисел a_1, a_2, \dots, a_n найти номер первого четного числа. Если четных чисел нет, то ответом может быть число 0.

13. Заданы два массива A и B. Написать программу нахождения элементов, общих для A и B.

14. Определить, какие элементы массива A и сколько раз встречаются в массиве B.

15. Дан массив из n чисел как положительных, так и отрицательных. Нужно сначала записать положительные числа, а затем отрицательные в том же порядке, как они были записаны в исходном массиве.

16. Даны два одномерных массива A и B: a_1, a_2, \dots, a_n и b_1, b_2, \dots, b_n . Преобразовать массив B по правилу: если $a_i \leq 0$, то b_i увеличить в 10 раз, иначе b_i заменить 0 ($i=1 \div n$).

17. Заполнить массив A[1..8] числами, вводимыми с клавиатуры. Найти среднее арифметическое положительных элементов.

18. Задан одномерный массив A[1..20]. Просуммировать все отрицательные элементы, стоящие на нечетных местах.

19. Задан одномерный массив A[1..17]. Определить среднее значение нечетных положительных элементов массива.

20. Задан одномерный массив A[1..15]. Определить сумму четных положительных элементов массива с n-го по k-й.

21. Найти среднее значение чётных элементов заданного массива X[1..15].

22. Задан одномерный массив A[1..15]. Определить сумму чётных положительных элементов массива.

23. Задан одномерный массив A[1..15]. Определить количество четных положительных элементов массива.

24. Задан одномерный массив A[1..20]. Просуммировать все неотрицательные элементы, стоящие на четных местах.

25. Задан массив D[1..10]. Сформировать два массива, включая в первый элементы исходного массива с четными индексами, а во второй - с нечетными индексами.

26. Задан массив размером K[1..10]. Сформировать два массива, включая в первый четные элементы исходного массива с нечетными индексами, а во второй нечетные элементы с четными индексами.

27. Задан одномерный массив A[1..20]. Найти максимальный элемент среди элементов, стоящих на четных позициях.

28. Задан одномерный массив A[1..20]. Найти максимальный элемент среди элементов с n-го по k-й и его индекс.

29. Задан одномерный массив A[1..20]. Найти минимальный элемент среди элементов, стоящих на четных позициях и его номер.

30. Задан одномерный массив $A[1..20]$. Найти минимальный элемент среди элементов массива с n -го по k -й.

31. Задан одномерный массив $A[1..20]$. Найти минимальный элемент среди положительных элементов.

32. Задан одномерный массив $A[1..20]$. Найти максимальный элемент среди отрицательных элементов.

33. Задан одномерный массив $A[1..20]$. Найти сумму максимального и минимального элементов.

34. Заполнить массив $A[1..6]$ числами, вводимыми с клавиатуры. Найти количество положительных элементов.

35. Заполнить массив $A[1..8]$ числами, вводимыми с клавиатуры. Найти среднее арифметическое элементов с четными индексами.

36. Заполнить массив $A[1..8]$ числами, вводимыми с клавиатуры. Найти среднее арифметическое элементов с нечетными индексами.

37. Заполнить массив $A[1..6]$ числами, вводимыми с клавиатуры. Найти сумму элементов с нечетными индексами.

38. Заполнить массив $A[1..8]$ числами, вводимыми с клавиатуры. Найти среднее арифметическое элементов, не превышающих 50.

39. Заполнить массив $A[1..8]$ числами, вводимыми с клавиатуры. Найти сумму элементов с четными индексами.

40. В массиве хранятся цены на 15 видов товаров. С помощью датчика случайных чисел заполнить массив целыми значениями, лежащими в диапазоне от 20 до 300 включительно. Определить цену самого дешевого товара и его порядковый номер.

41. В массиве хранится информация о количестве осадков, выпавших за каждый день ноября. Определить, в какие числа каждого месяца осадков не было.

42. В массиве хранятся цены на 20 видов товаров. С помощью датчика случайных чисел заполнить массив целыми значениями, лежащими в диапазоне от 50 до 700 включительно. Определить цену самого дешевого и самого дорогого товаров.

43. В массиве хранятся сведения об общей стоимости товаров, проданных фирмой за каждый день марта. Определить дни, в которые стоимость проданных товаров превысила среднюю ежедневную сумму продаж.

44. В массиве хранятся сведения об общей стоимости товаров, проданных фирмой за каждый день марта. Определить дни, в которые стоимость проданных товаров превысила значение S .

45. В массиве хранится информация о количестве осадков, выпавших за каждый день июля. Определить самый жаркий день.

46. В массиве хранятся значения роста двенадцати человек. С помощью датчика случайных чисел заполнить массив целыми значениями, лежащими в

диапазоне от 163 до 190 включительно. Найти значение максимального элемента массива и его порядковый номер.

47. В массиве хранится информация о стоимости каждой из 50 марок легковых автомобилей. Определить, сколько стоит самый дешевый автомобиль.

48. В массиве хранится информация о количестве осадков, выпавших за каждый день октября. С помощью датчика случайных чисел заполнить массив целыми значениями, лежащими в диапазоне от 0 до 80 включительно. Определить количество дней, когда количество осадков не превышало 40.

49. В массиве хранится информация о годе рождения каждого из 30 человек. С помощью датчика случайных чисел заполнить массив целыми значениями, лежащими в диапазоне от 1965 до 1985 включительно. Определить количество людей, родившихся до 1975 года.

50. В массиве хранится информация о численности учеников в каждом из 32 классов школы. С помощью датчика случайных чисел заполнить массив целыми значениями, лежащими в диапазоне от 25 до 36 включительно. Определить максимальное количество учеников в классе.

51. В массиве хранится информация о стоимости каждой из 60 книг. Определить порядковый номер самой дорогой книги.

52. В массиве хранится информация о среднесуточной температуре за каждый день июля. Определить дни, когда температура превышала среднюю за месяц.

53. В массиве записаны результаты 20 игр футбольной команды (если игра окончилась выигрышем данной команды, то записано число 3, вничью — 2, если игра закончилась проигрышем — 1). С помощью датчика случайных чисел заполнить массив целыми значениями, лежащими в диапазоне от 1 до 3 включительно. Определить номера тех команд, которые проиграли.

54. В массиве хранится информация о количестве побед, одержанных 20 футбольными командами. С помощью датчика случайных чисел заполнить массив целыми значениями, лежащими в диапазоне от 0 до 5 включительно. Определить номера команд, одержавших более 3-х побед.

55. Рост 25 учеников класса представлен в виде массива. Определить порядковые номера самых высоких учеников.

56. В одномерном массиве хранится информация о зарплате 15 человек, работающих в отделе. Составить программу для определения: а) итоговой суммы по всему отделу; б) порядкового номера человека, получившего наименьшую зарплату; в) средней зарплаты по отделу.

57. В одномерном массиве хранятся сведения о стоимости товаров, проданных фирмой за каждый месяц года. Определить: а) общую сумму продаж за год; б) номера месяцев, когда сумма продаж превысила среднюю за год; в) в каком полугодии работа фирмы была более эффективна.

58. В одномерном массиве хранится информация о прибыли каждого из 8 магазинов фирмы «Луч» за месяц. Определить: а) общую прибыль фирмы;

б) номера магазинов с максимальной и минимальной прибылью; в) среднее значение прибыли.

59. В одномерном массиве хранится информация о коммунальных платежах каждой из семей 20-квартирного дома за месяц. Определить: а) общую сумму платежей; б) номера квартир, которые не оплатили коммунальные услуги; в) номера квартир, платежи которых превысили заданное значение.

60. В одномерном массиве хранится информация о отчислениях на развитие предприятия за каждый месяц года. Определить: а) общую сумму отчислений; б) номера месяцев, когда отчислений не было; в) номера месяцев, когда сумма отчислений была максимальной.

61. В одномерном массиве хранится информация об отчислениях на благотворительность каждой из 15 фирм. Определить: а) общую сумму отчислений; б) номера фирм, которые перечислили сумму выше средней; в) номера фирм, перечисливших минимальную сумму.

62. В одномерном массиве хранится информация о количестве осадков, выпавших за каждый день сентября. Определить: а) в какие дни осадков не было; б) дни, когда количество осадков превысило среднее значение; в) общее количество осадков за месяц.

63. В одномерном массиве хранится информация о ценах на 20 видов товаров. Определить: а) цену самого дешевого товара и его порядковый номер; б) цену самого дорогого товара и его порядковый номер; в) номера товаров, цена которых превышает среднее значение;

64. Задан одномерный массив $A[1..18]$. Сформировать новый массив $B[1..17]$, элементы которого определяются по формуле $B[i]=abs(a[i]-a[i+1])$. Найти сумму четных элементов массива B .

65. Сформировать массивы A, B и C , содержащие по 10 целых случайных чисел. Найти среднее арифметическое элементов каждого массива. Создать процедуру для формирования массива и определения среднего арифметического его элементов.

66. Сформировать массивы A, B и C , содержащие по 15 целых случайных чисел. Найти сумму элементов каждого массива. Создать процедуру для формирования массива и определения суммы его элементов.

67. Заданы массивы $a[1..20]$, $b[1..10]$ и $c[1..15]$. Найти среднее арифметическое минимальных элементов массива (использовать процедуру формирования массива и процедуру определения минимального элемента).

68. Заданы массивы $a[1..n]$, $b[1..n]$ и $c[1..n]$. Определить наименьший среди максимальных элементов массива (использовать процедуру формирования массива и процедуру определения максимального элемента).

69. Заданы массивы $a[1..10]$ и $b[1..15]$. Определить индексы максимальных элементов каждого массива (использовать процедуру формирования массива и процедуру определения индекса минимального элемента)..

70. Заданы массивы $a[1..14]$ и $b[1..14]$. Найти сумму минимальных элементов массивов (использовать процедуру формирования массивов и процедуру определения минимального элемента)..

71. Заданы массивы $a[1..n]$, $b[1..m]$, $c[1..k]$. Определить, какой из массивов содержит больше положительных чисел (использовать процедуру формирования массива и процедуру определения количества положительных элементов).

72. Заданы массивы $a[1..n]$ и $b[1..m]$. В обоих массивах все члены, следующие за максимальным, заменить на -1 (использовать процедуру формирования массива и процедуру замены элементов массива).

73. Заданы массивы $a[1..10]$ и $b[1..15]$. Определить среднее арифметическое нечетных элементов каждого массива (использовать процедуру формирования массива и процедуру определения среднего арифметического нечетных элементов).

74. Заданы массивы $a[1..15]$ и $b[1..20]$. Найти количество отрицательных элементов в каждом массиве (использовать процедуру формирования массива и процедуру определения количества отрицательных элементов).

75. Заданы массивы $a[1..10]$ и $b[1..10]$. Найти количество четных элементов в каждом массиве (использовать процедуру формирования массива и процедуру определения количества четных элементов).

II. ДВУМЕРНЫЕ МАССИВЫ

Структура данных в языке Паскаль для хранения табличных данных называется двумерным массивом. Описать такой массив можно двумя способами:

Var

A: Array[1..20] of Array [1..30] of Integer;

или

Var

A : Array [1..20,1..30] Of Integer;

В обоих случаях описан двумерный массив, соответствующий таблице, состоящей из 20 строк и 30 столбцов. Отдельный элемент двумерного массива адресуется двумя индексами. Например, элемент, находящаяся в 5-й строке и 6-м столбце будет обозначаться $A[5,6]$.

Пример 1.

Найти сумму элементов целочисленной матрицы размером $N*M$.

Program pr1;

Const n=2; m=5;

Var Mt: array [1..n,1..m] of integer;

J, S, I: integer;

BEGIN *{ввод элементов матрицы}*

Writeln ('введи элементы матрицы по строкам');

For I:=1 to n do

```

    For j:=1 to m do begin
        Write ('Mt [' , I, ', ' , j, '= '); Readln (Mt[i,j]);
    End;
{нахождение суммы элементов матрицы}
S:=0; For I:=1 to n do
    For j:=1 to m do
        S:=S+Mt[i,j];
    Writeln ('сумма = ', S);
END.

```

Пример 2.

Сформировать и вывести на экран в виде таблицы массив A[1..4,1..6], заполнив его целыми случайными числами из интервала [25,80].

```

Program pr2;
Var a: array[1..4,1..6] of integer;
    i, j: integer;
BEGIN
    randomize;           {формирование и вывод построчно элементов массива}
    for i:=1 to 4 do begin           {I – номер строки}
        for j:=1 to 6 do begin           {j – номер столбца}
            a[i,j]:=random(56)+25; write(a[i,j], ' ');
        end;
        writeln;           {переход на новую строку}
    end;
END.

```

Пример 3.

Дана целочисленная матрица размером N*M. Все отрицательные элементы заменить нулями, а положительные – единицами. Полученную матрицу вывести в виде таблицы.

```

Program pr3;
Const n=3; m=3;
Type massiv=array [1..n,1..m] of integer;
Var Mt: massiv;
    i, j: integer;
BEGIN           {ввод элемента матрицы}
    Writeln ('введи элемент матрицы по строкам');
    For I:=1 to n do
        For j:=1 to m do begin
            Write ('Mt [' , I, j, '= '); readln (Mt [i, j])
        End;
    For I:=1 to n do           {перебираем построчно элементы массива}
        For j:=1 to m do
            If Mt [i, j] <=0 then Mt [i, j] :=0 Else Mt[i, j]:=1;
        {вывод в виде таблицы}

```

```

for I:=1 to n do begin
  for j:=1 to m do write (Mt [I, j]);
  writeln      {переходим на новую строку}
end;
END.

```

Пример 4.

В массиве A[1..5,1..5] найти сумму элементов главной диагонали, кратных 5.

```

Program p2;
Var a: array[1..5,1..5] of integer;
    i, j, s: integer;
BEGIN
  randomize;  for i:=1 to 5 do begin {формируем и выводим таблицу}
    for j:=1 to 5 do begin
      a[i,j]:=random(80); write(a[i,j], ' ');
    end;
    writeln;
  end;
  s:=0;      for i:=1 to 5 do {перебираем построчно элементы массива}
    for j:=1 to 5 do
      {если элемент находится на главной диагонали и кратен 5, то суммируем }
      if (i=j) and (a[i,j] mod 5=0) then s:=s+a[i,j];
    writeln('s=',s)
  end;
END.

```

Пример 5.

В двумерном массиве A[1..5,1..5], найти произведение отрицательных элементов побочной диагонали.

```

Program pr5;
Var a: array[1..5,1..5] of integer;
    i, j, p: integer;
BEGIN
  randomize;  for i:=1 to 5 do begin {формируем и выводим таблицу}
    for j:=1 to 5 do begin
      a[i,j]:=random(80); write(a[i,j], ' ');
    end;
    writeln;
  end;
  p:=1;      for i:=1 to 5 do
    for j:=1 to 5 do
      {если элемент находится на побочной диагонали и он отрицательный, то
      находим произведение }
      if (i+j=6) and (a[i,j]<0) then p:=p*a[i,j];
    writeln('p=',p)
  end;
END.

```

Пример 6

В двумерном массиве хранится информация о количестве студентов в каждой из трех групп каждого курса с первого по пятый (в 1-ой строке – информация о первом курсе, во 2-ой строке – о втором курсе и т. д.). Найти численность самой большой группы на 3 курсе. На каком курсе больше студентов, на первом или на пятом?

Program pr6;

Var a: array[1..5,1..3] of integer;

i, j, m, s1, s5: integer;

BEGIN

randomize; *{формируем и выводим таблицу}*

for i:=1 to 5 do

for j:=1 to 3 do begin

a[i,j]:=random(10)+20; write(a[i,j], ' ');

end;

writeln

end;

m:=a[3,1]; *{находим наибольший элемент в 3-ей строке}*

for i:=1 to 3 do

if a[3, i]>m then m:=a[3,i];

writeln('численность самой большой группы на третьем курсе ', m);

{находим сумму элементов 1-й строки и сумму элементов 5-й строки}

for j:=1 to 3 do begin

s1:=s1+a[1,i];

s5:=s5+a[5,i]

end;

{сравниваем количество студентов на 1-м и на 5-м курсах}

if s1>s5 then writeln('студентов больше на 1 курсе')

else if s1<s5 then writeln('студентов больше на 5 курсе')

else writeln('количество студентов на 5-м и на 1-м курсах

равно')

END.

Задания для самостоятельной работы

1. Заменить положительные элементы двумерного массива на 1, а отрицательные на 0.

2. В двумерном массиве числа, кратные n, заменить частными от деления на n. Если таких чисел нет, то вывести на экран сообщение об этом.

3. Следом квадратной матрицы называют число, равное сумме элементов главной диагонали. Составить программу нахождения следа квадратной матрицы порядка n.

4. В двумерном массиве найти сумму всех тех элементов, сумма индексов которых равна n .
5. Из данной прямоугольной таблицы вывести на экран строки, содержащие хотя бы один нулевой элемент.
6. Составить программу для определения номера строки и номера столбца прямоугольной матрицы, на пересечении которых находится наибольший по абсолютной величине элемент этой матрицы.
7. Составить программу нахождения максимального элемента в каждом столбце и минимального в каждой строке квадратной матрицы.
8. Составить программу обмена местами максимального и минимального элементов главной диагонали матрицы.
9. Дана матрица $N \times N$. Вывести на экран дисплея элементы той строки, сумма элементов которой максимальна.
10. Дана квадратная матрица порядка n . Составить программу вычисления количества положительных элементов в нижнем левом треугольнике, включая диагональные элементы.
11. Сформировать и вывести на экран в виде таблицы массив $A[1..4,1..6]$, заполнив его целыми случайными числами из интервала $[30,75]$. Найти среднее арифметическое каждого столбца.
12. В массиве $B[1..4,1..4]$ найти сумму элементов главной диагонали.
13. В двумерном массиве хранится информация о количестве учеников в каждом классе каждой параллели школы с первой по одиннадцатую (в первой строке – информация о классах первой параллели, во второй – второй параллели и т.д.). В каждой параллели школы имеются 4 класса. Определить: а) общее число учеников в параллели 5-х классов; б) самый большой по наполняемости класс в параллели 9-х классов и его порядковый номер.
14. Найти наибольший и наименьший элементы данного двумерного массива. Указать их индексы.
15. В двумерном массиве числа, кратные 2, заменить частными от деления на 2. Если таких чисел нет, то вывести на экран сообщение об этом.
16. Составить программу для построения массива, элементы которого вычисляются по формуле $a_{i,j} = \min(i,j)$.
17. Составить программу для определения номера строки и номера столбца прямоугольной матрицы, на пересечении которых находится наименьший по абсолютной величине элемент этой матрицы.
18. Составить программу нахождения минимального элемента в каждом столбце и максимального в каждой строке квадратной матрицы.
19. Составить программу обмена местами максимального и минимального элементов главной диагонали матрицы.

20. Дана матрица $N \times N$. Вывести на экран дисплея элементы той строки, сумма элементов которой максимальна.

21. Дана квадратная матрица порядка n . Составить программу вычисления количества четных элементов в нижнем левом треугольнике, включая диагональные элементы.

22. Вывести на экран матрицу 5×5 . Определить сумму минимальных элементов столбцов матрицы.

23. Вывести на экран матрицу 4×5 . Определить номера столбцов, содержащих более половины положительных элементов.

24. Вывести на экран матрицу 6×6 , элементами которой являются целые случайные числа из интервала $[20,50]$. Определить сумму минимальных элементов строк матрицы.

25. Вывести на экран матрицу 5×5 , элементами которой являются целые случайные числа из интервала $[19,49]$. Найти сумму нечетных элементов каждого столбца.

26. Вывести на экран матрицу 4×6 , элементами которой являются целые случайные числа из интервала $[25,50]$. Определить произведение максимальных элементов столбцов матрицы.

27. Вывести на экран матрицу 8×5 , элементами которой являются целые случайные числа из интервала $[20,40]$. Определить номер строки, содержащей не менее 2-х четных чисел.

28. Вывести на экран матрицу 3×6 , элементами которой являются целые случайные числа из интервала $[10,90]$. Определить номер минимального элемента в каждой строке.

29. Вывести на экран матрицу 5×5 , элементами которой являются целые случайные числа из интервала $[-19,49]$. Найти сумму четных элементов каждого столбца.

30. Вывести на экран матрицу 8×8 , элементами которой являются целые случайные числа из интервала $[-9,9]$. Определить максимальный элемент в каждой строке и поменять его местами с последним элементом строки.

31. Вывести на экран матрицу 3×4 , элементами которой являются целые случайные числа из интервала $[19,99]$. Определить минимальный элемент в каждой строке и поменять его местами с первым элементом строки.

32. Вывести на экран матрицу 5×4 , элементами которой являются целые случайные числа из интервала $[30,70]$. Определить номер минимального элемента в каждом четном столбце и номер максимального элемента в каждом нечетном столбце.

33. Вывести на экран матрицу 4×6 . Найти сумму элементов каждой строки. Вывести результат в виде одномерного массива и найти его максимальный элемент.

34. Вывести на экран матрицу 10×8 , элементами которой являются целые случайные числа из интервала $[19,49]$. Определить минимальный элемент в каждом столбце и выбрать из них максимальный.

35. Вывести на экран матрицу 5×8 . Определить номера столбцов, содержащих хотя бы один отрицательный элемент.

36. Вывести на экран матрицу 4×8 , элементами которой являются целые случайные числа из интервала $[-20,30]$. Определить номер максимального элемента в каждой строке и номер минимального в каждом столбце.

37. Вывести на экран матрицу 8×5 , элементами которой являются целые случайные числа из интервала $[30,70]$. Определить среднее арифметическое минимального и максимального элемента в каждом столбце.

38. Вывести на экран матрицу 5×5 . Определить среднее арифметическое минимальных элементов столбцов матрицы.

39. Вывести на экран матрицу 4×5 . Определить номера столбцов, содержащих не более половины положительных элементов.

40. Вывести на экран матрицу 5×5 , элементами которой являются целые случайные числа из интервала $[19,49]$. Найти среднее арифметическое нечетных элементов каждого столбца.

41. Вывести на экран матрицу 8×5 , элементами которой являются целые случайные числа из интервала $[20,40]$. Определить номер строки, содержащей не более 3-х чётных чисел.

42. Сформировать матрицу вида:

a) $\begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 & 0 \end{pmatrix}$

b) $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 4 & 3 & 4 & 1 \\ 1 & 2 & 3 & 4 & 5 \\ 5 & 4 & 3 & 2 & 1 \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix}$

c) $\begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix}$

d) $\begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix}$

e) $\begin{pmatrix} 1 & 0 & 0 & 0 & 5 \\ 0 & 2 & 0 & 4 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 2 & 0 & 4 & 0 \\ 1 & 0 & 0 & 0 & 5 \end{pmatrix}$

f) $\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 3 & 3 & 3 & 3 & 3 \\ 0 & 0 & 0 & 0 & 0 \\ 5 & 5 & 5 & 5 & 5 \end{pmatrix}$

g) $\begin{pmatrix} 1 & 0 & 0 & 0 & 5 \\ 2 & 0 & 0 & 0 & 4 \\ 3 & 0 & 0 & 0 & 3 \\ 4 & 0 & 0 & 0 & 2 \\ 5 & 0 & 0 & 0 & 1 \end{pmatrix}$

h) $\begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix}$

i) $\begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 2 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 4 & 0 & 4 & 0 \\ 5 & 0 & 0 & 0 & 5 \end{pmatrix}$

III. ОБРАБОТКА СИМВОЛЬНЫХ И СТРОКОВЫХ ВЕЛИЧИН

В стандарте языка Паскаль описаны два типа переменных для литерных величин. Это - **String** и **Char**. Переменная типа Char может содержать в себе только один единственный символ, тип String предназначен для хранения строковых величин до 255 символов длиной.

При описании переменной этого типа можно указать максимальное число символов, которое можно занести в нее (это число не должно превышать 255). Например:

```
Var S : String[30]; {максимальная длина строки S – 30 символов}
    St: String; ]; {максимальная длина строки St – 255 символов}
```

При использовании строковой переменной, к каждому ее символу можно обратиться отдельно, указав его порядковый номер.

Пример 1.

Вывести посимвольно строку 'primer'

```
Program pr1;
Var st: string[6]; i: integer;
BEGIN
    st:='primer';
    for i:=1 to 6 do write(st[i], ' ');
END.
```

В результате работы данной программы на экран будут распечатаны следующие значения: *p r i m e r*

Таким образом, первым символом $st[1]$ в переменной st является буква p , вторым символом $st[2]$ является r , третьим – i и т.д.

Две строковые величины можно объединять. Эта операция называется **конкатенацией** и обозначается знаком "+".

Например, результатом выполнения следующих команд:

```
R:='kadabra';
H:='abra';
S:=H+R;
```

в переменной S будет значение 'abrakadabra'.

Для конкатенации результат зависит от порядка операндов (в отличие от операции сложения). Следует помнить о том, какой максимальной длины может быть результирующая переменная, так как в случае превышения значением выражения числа, указанного после String в описании переменной, "лишние" символы в переменную не попадут.

Строковые величины можно сравнивать между собой. Сравнение строк происходит посимвольно. Большим из двух символов считается тот, код которого больше. Если равны первые символы, то сравнивается следующая пара до тех пор, пока не будет найдено различие. Если начало строк совпадает, а одна из них кончается раньше, то вторая автоматически называется большей.

Код символа в Паскале можно определить при помощи функции **Ord(C)**, где *C* - либо непосредственно указанный символ, либо переменная символьного типа, либо один символ строковой переменной.

Есть и обратная функция, которая возвращает символ по известному коду. Это функция **Chr(N)**, где *N* - выражение, приводящее к целому числу в интервале от 0 до 255 (возможные значения кода символа). Очевидно, что $\text{Chr}(\text{Ord}(C))=C$, $\text{Ord}(\text{Chr}(N))=N$.

Пример 2.

Вывести на экран кодовую таблицу:

```
Program Pr2;  
Var I : Byte;  
BEGIN  
  For I:=32 to 255 do Write(I:4, '-',Chr(I))  
END.
```

Цикл в программе начинается с 32 потому, что символы с кодами от 0 до 31 являются управляющими и не имеют соответствующего графического представления.

В Паскале используются следующие процедуры и функции для обработки строковых величин:

Функция **Concat(S1[,S2,...,SN]): string** – сцепляет (объединяет) строки *S1*, *S2*, ..., *SN*. Параметры, указанные в квадратных скобках, не являются обязательными.

Функция **Copy(S: string; Index, Count: integer): string** – копирует из строки *S* количество *Count* символов, начиная с символа с номером *Index*.

Функция **Length(S: string):byte** – определяет длину строки *S*.

Функция **Pos(SubS, S: string):byte** – отыскивает в строке *S* первое вхождение подстроки *SubS* и определяет номер позиции, с которой она начинается. Если подстрока не найдена, то выдается 0.

Функция **UpCase(C: char):char** – преобразует строчную латинскую букву в прописную. Любые другие символы возвращаются без преобразования.

Процедура **Delete(var S: string; Index, Count: integer)** – удаляет количество *Count* символов из строки *S*, начиная с символа с номером *Index*.

Процедура **Insert(SubS: string; var S: string; Index: integer)** – вставляет подстроку *SubS* в строку *S*, начиная с символа с номером *Index*.

Процедура **Str(X [:Width [:Decimals]]; Var S: string)** – преобразует число *X* в строку символов *S*. Параметры *Width* и *Decimals* задают формат преобразования (общую ширину поля и количество символов в дробной части соответственно).

Процедура **Val(S: String; var X; var Code: integer)** – преобразует строковое значение *S* в его численное представление *X* типа *Real* или *Integer*. Параметр *Code* содержит признак ошибки преобразования (0 – нет ошибки),

Пример 3.

Определить длину введенной пользователем строковой величины.

```

Program Pr3;
Var S : String;
BEGIN
    Writeln('Введите последовательность символов'); Readln(S);
    Writeln('Вы ввели строку из ',Length(S), ' символов')
END.

```

Пример 4.

Задан список из 6 слов. Определить, сколько слов списка начинается на букву «п».

```

Program Pr4;
Var s: string[20]; i,k:integer;
BEGIN
    k:=0; for i:=1 to 6 do begin
        writeln('Введите слово'); readln(s);
        if s[1]='п' then k:=k+1;
    end;
    writeln(k);
END.

```

Пример 5.

Определить, является ли введенная строка "перевертышем".

Перевертышем называется такая строка, которая одинаково читается с начала и с конца. Например, "казак" и "потоп" - перевертыши, "канат" - не перевертыш".

Поступим следующим образом: из введенной строки сформируем другую строку из символов первой, записанных в обратном порядке, затем сравним первую строку со второй; если они окажутся равны, то ответ положительный, иначе - отрицательный. Естественно, предложенный способ решения не является единственно возможным.

```

Program Pr5;
Var
    S,B : String;    I : Byte;
BEGIN
    Writeln('Введите строку'); Readln(S);
    B:= "";          {Переменной B присваиваем значение "пустая строка"}
    For I:=1 to Length(S) do          {перебираем символы строки}
        B:=S[I]+B; {Символы строки S присоединяются к переменной B слева}
    If B=S Then Writeln('Перевертыш') Else Writeln('Не перевертыш')
END.

```

Пример 6.

Найти сумму цифр введенного натурального числа.

```

Program Pr6;
Var S : String; I,X,A,C : Integer;

```

```

BEGIN
  Writeln('Введите натуральное число');
  Readln(S);           {Число вводится в строковую переменную}
  A:=0;
  For I:=1 To Length(S) Do Begin
    Val(S[I],X,C);     {Цифра преобразуется в число}
    A:=A+X             {Числа суммируются}
  End;
  Writeln('Сумма цифр равна ',A)
END.

```

Пример 7.

Во введенной строке заменить все вхождения подстроки 'ABC' на 'KLMNO'.

```

Program Pr7;
Var S : String; A : Byte;
BEGIN
  Writeln('Введите строку'); Readln(S);
  While Pos('ABC',S)<>0 Do {пока подстрока 'ABC' содержится в строке S }
  Begin
    A:= Pos('ABC',S); {определяем позицию вхождения подстроки 'ABC' в строку S}
    Delete(S,A,3); {удаляем из строки S 3 символа, начиная с позиции A }
    Insert('KLMNO',S,A) {вставляем в строку S на это место подстроку 'KLMNO'}
  End;
  Writeln(S)
END.

```

Пример 8.

Во введенной строке подсчитать процент вхождения символа 'о' и символа 'е' (создать функцию для определения процентного содержания заданного символа в строке)

```

Program Pr8;
{функция, определяющая процент вхождения символа c в строку st}
Function proc_char(st:string;c:char):real;
Var k, n, I:integer; {k – количество заданных символов, n – длина строки,
                    I - номер очередного символа строки}
Begin
  n:=length(st);           {вычисляем длину строки}
  k:=0; for I:=1 to n do   {перебираем символы строки}
    if st[I]=c then k:=k+1; {если очередной символ строки равен заданному символу, то увеличиваем счетчик k}
  proc_char:=k/n*100;     {определяем процент заданных символов}
End;

```

```

Var st: string;
    p1,p2: real;
BEGIN
    write('st=');readln(st);
    p1:=proc_char(st,'o');
    p2:=proc_char(st,'e');
    writeln('o-', p1:2:1,'%');
    writeln('e-', p2:2:1,'%');
    readln;
END.

```

{основная программа}
{вводим строку}
{определяем процентное содержание символа 'o'}
{определяем процентное содержание символа 'e'}
{вывод результата}

Пример 9.

Задан список из 8 слов. Найти самое короткое слово в списке. Если таких слов несколько, то распечатать их в один столбец.

Решение поставленной задачи сводится к нескольким этапам: ввести список слов в виде массива строковых переменных; подсчитать длину каждой строки; определить наименьшую из длин; распечатать те строки массива, длина которых совпадает с наименьшей.

Program Pr9;

```

Var s: array[1..8] of string[20];
    n: array[1..8] of integer;
    i, min: integer;
BEGIN
    for i:=1 to 8 do begin
        writeln('Введите слово'); readln(s[i]);
        n[i]:=length(s[i]);
    end;
    min:=n[1]; for i:=2 to 8 do if min>n[i] then min:=n[i];
    for i:=1 to 8 do if n[i]=min then writeln(s[i]);
END.

```

В данной программе для обозначения слов используется массив строковых переменных *s*, для обозначений соответствующих им длин слов – целочисленный массив *n*. Наименьший элемент массива *n* хранится в виде переменной *min*.

Пример 10.

Задана строка из двух слов, разделенных пробелом. Поменять в данной строке слова местами.

Program Pr10;

```

Var s: string[40];
    s1,s2: string[20];
    i,n: integer;
BEGIN
    writeln('Введите строку');
    readln(s);

```

```
n:=length(s); i:=pos(' ', s);  
s1:=copy(s, 1, i); s2:=copy(s, i+1, n-i); s:=s2 + ' ' + s1;  
writeln(s);
```

END.

В данной программе переменная *s* служит для обозначения строки из двух слов, переменные *s1* и *s2* – для обозначения первого и второго слова строки соответственно. Алгоритм решения заключается в том, что сначала с помощью функции *Pos* выясняется, каким по счету символом в строке *s* находится пробел. Далее в строке выделяются две подстроки: одна включает все символы левее пробела (первое слово), другая – правее пробела (второе слово). В конце программы слова сцепляются в обратном порядке с использованием разделяющего их пробела.

Задания для самостоятельной работы

1. Зашифровать введенную с клавиатуры строку, поменяв местами первый символ с третьим, второй с четвертым и т. д. Провести дешифровку.
2. В заданном тексте удалить часть текста, заключенную в скобки (вместе со скобками).
3. Напечатать самое длинное слово из заданного текста (слова разделены пробелами).
4. Определить, какой процент слов в тексте содержит букву О (слова разделены пробелами).
5. Сколько раз в тексте встречается заданное слово (слова разделены пробелами).
6. В тексте вставить между словами вместо одного пробела запятую и пробел.
7. Разделить заданный текст на строки. В качестве разделителя в тексте используется символ @.
8. Определить, какой процент слов в тексте начинается на букву К. Слова разделены пробелами.
9. Определить длину каждого слова из заданного текста (слова разделены пробелами).
10. В заданном наборе слов найти слово максимальной длины.
11. Написать программу, исключаящую из символьной строки все цифры.
12. Определить, содержит ли данный текст символы, отличные от букв и пробела.
13. Имеется ли в заданном наборе такая пара слов, из которых одно является обращением другого (например: нос-сон).
14. В заданном слове поменять местами равноотстоящие от концов слова буквы.
15. Составить программу, результатом работы которой является удаление из текста пробелов и знаков препинания.
16. Удалить из введенной строки все цифры, кратные 3.
17. Изменить введенную строку, уменьшив все ее цифры, кроме 0, на 1.
18. Определить количество "2" и "3" во введенной строке.

19. Преобразовать введенную строку так, чтобы из нее были удалены буквы с ASCII - кодами от 70 до 75.
20. Определить, имеются ли во введенной строке следующие подряд две "4".
21. Введенную строку букв и цифр преобразовать так, чтобы после каждой цифры следовал пробел.
22. Изменить введенную строку так, чтобы каждая из цифр увеличилась на 1 (9 заменить 0).
23. Преобразовать введенную строку так, чтобы сначала были расположены цифры, потом буквы.
24. Введена строка маленьких латинских букв. Преобразовать ее, превратив маленькие буквы в большие.
25. Зашифровать введенное слово, сместив все буквы на 1 позицию (последняя становится первой).
26. Определить, каких символов во введенной строке больше: тех, ASCII-коды которых принадлежат [48,57] или тех, ASCII-коды которых принадлежат [97,122].
27. Преобразовать буквы (a-e) введенной строки в (A-E).
28. Подсчитать количество букв O и A во введенной строке.
29. Подсчитать количество цифр во введенной строке, кратных 3
30. Введенную строку цифр вывести, расположив в каждой подстроке по 5 цифр.
31. Преобразовать буквы введенной строки так, чтобы их ASCII-коды увеличились на 3.
32. Преобразовать введенную строку цифр, поставив перед каждой цифрой "5" цифру "6".
33. Определить, заключена ли в диапазоне [20,25] сумма цифр введенной строки цифр.
34. Определить наименьшую из цифр, занимающих во введенной строке четные позиции.
35. Определить, четна ли сумма цифр, занимающих во введенном числе нечетные позиции.
36. Преобразовать введенную строку так, чтобы выводимые части строк начинались с больших букв.
37. Найти сумму цифр введенного натурального числа .
38. Изменить нечетные цифры введенного числа, уменьшив их на 1.
39. Определить длину самой длинной группы цифр во введенной строке.
40. Найти сумму цифр введенной строки, содержащей цифры и буквы.
41. Обратить введенную строку: abcd -> dcba.
42. Составить программу ,заменяющую окончания строк: "ion" -> "ed"
43. Изменить введенную строку цифр, заменяя цифры по схеме: $N \rightarrow N \bmod 10$
44. Преобразовать цифры введенной строки по схеме: $N \rightarrow 9-N$
45. Вводится цена в центах. Вывести ее значение в виде долларов и центов.
46. Изменить введенную строку цифр, разместив сначала цифры занимающие нечетные места, потом четные.
47. Вывести введенную строку группами символов по 4 символа в каждой.

48. Преобразовать строку цифр, расположив цифры в порядке возрастания.
49. Определить, содержатся ли во введенной строке одинаковые символы.
50. Подсчитать количество больших и маленьких букв во введенной строке.
51. Создать функцию пользователя, определяющую наибольшую цифру введенного натурального числа.
52. Создать функцию пользователя, определяющую наименьшую цифру введенного натурального числа.
53. Создать функцию пользователя, определяющую количество четных цифр введенного числа.
54. Создать процедуру, определяющую количество букв и количество цифр во введенной строке.
55. Введены 3 строки. Подсчитать сумму цифр, кратных 3, в каждой из них (создать функцию, подсчитывающую сумму цифр, кратных 3, в строке).
56. Даны 2 строки. Определить сумму цифр в каждой из них (создать функцию, подсчитывающую сумму цифр)
57. Создать функцию пользователя, определяющую количество символов введенной строки, ASCII-коды которых ≥ 70 .
58. Создать процедуру, позволяющую изменить введенную строку, добавив справа заданное количество заданных символов.
59. Создать процедуру, позволяющую из заданной строки удалить пробелы.
60. Создать процедуру, позволяющую записывать введенное слово в зеркальном отображении.
61. Создать процедуру, которая позволяет введенный текст разбить на строки по k символов.
62. Создать процедуру, которая позволит во введенной строке через каждые n символов вставить k пробелов.
63. Создать процедуру, которая в заданном тексте заменяет слово A1 на слово A2 (длины слов не совпадают).
64. Создать процедуру, которая во введенном слове заменяет один символ другим.
65. Создать процедуру, которая в тексте убирает лишние пробелы между словами, оставив по одному.
66. Вводятся три строки. Зашифровать каждую из них, заменив все буквы "с" на "о"(создать процедуру, заменяющую в заданной строке один символ другим)
67. Преобразовать три введенные строки, чтобы после каждой цифры следовал символ '!' (создать процедуру, вставляющую пробел после каждой цифры в строке)
68. Дано предложение. Все пробелы в нем заменить на символ "_" (создать соответствующую процедуру).
69. Дано предложение. Удалить из него буквы «р» и «с»(создать процедуру, удаляющую из строки заданный символ).
70. Вводятся 2 строки. Заменить в каждой из них цифры символом «*» (создать соответствующую процедуру).
71. Вводятся 2 строки. Удалить из каждой строки все цифры, кратные 3(создать соответствующую процедуру).

72. Создать процедуру, позволяющую изменить введенную строку, добавив слева заданное количество заданных символов.

73. Преобразовать три введенные строки, чтобы перед каждой цифрой стоял символ '+' (создать соответствующую процедуру)

74. Вводятся 2 строки. Зашифровать 1-ю строку, заменив в ней все буквы "a" на "o" и 2-ю, заменив "e" на "y" (создать процедуру, заменяющую в заданной строке один символ другим)

75. Вводятся 2 строки. Удалить из первой строки все символы 'd', а из второй – 'w' (создать процедуру, позволяющую из заданной строки удалить заданный символ).

IV. ФАЙЛЫ

Файл представляет собой последовательность компонент одного типа, расположенных на внешнем устройстве (например, на диске). Элементы могут быть любого типа, за исключением самого типа-файла. Число элементов в файле при описании не объявляется. Работа с физическими файлами происходит через так называемые файловые переменные.

Для задания типа-файла следует использовать зарезервированные слова **File** и **Of**, после чего указать тип компонент файла.

Пример:

Type

N = File Of Integer; {Тип-файл целых чисел}

C = File Of Char; {Тип-файл символов}

Есть заранее определенный в Паскале тип файла с именем **Text**. Файлы этого типа называют текстовыми.

Введя файловый тип, можно определить и переменные файлового типа:

Var

F1 : N;

F2 : C;

F3 : Text;

Тип-файл можно описать и непосредственно при введении файловых переменных:

Var

Z : File Of Integer;

Файловые переменные можно использовать лишь для выполнения операций с файлами (чтение, запись и т.д.).

Перед тем, как осуществлять ввод-вывод, файловая переменная должна быть связана с конкретным внешним файлом при помощи процедуры Assign.

Assign(<Имя файловой переменной>, <Имя файла>);

Имя файла задается либо строковой константой, либо через переменную типа String. После этого файл должен быть открыт одной из процедур Reset или Rewrite.

Reset(<Имя файловой переменной>); -открывает существующий файл для чтения, указатель текущей компоненты файла настраивается на начало файла. Если физического файла, соответствующего файловой переменной не существует, то возникает ситуация ошибки ввода-вывода.

Rewrite(<Имя файловой переменной>); - открывает новый пустой файл для записи, ему присваивается имя, заданное процедурой Assign. Если файл с таким именем уже существует, то он уничтожается.

После работы с файлом он, как правило, должен быть закрыт процедурой Close.

Close(<Имя файловой переменной>);

Это требование обязательно должно соблюдаться для файла, в который производилась запись.

Для ввода информации из файла, открытого для чтения, используется оператор Read.

Read(<Имя_файловой_переменной>,<Список ввода>);

Происходит считывание данных из файла в переменные, имена которых указаны в списке ввода. Переменные должны быть того же типа, что и компоненты файла.

Вывод информации производит оператор Write

Write (<Имя файловой переменной>, <Список вывода>);

Данные из списка вывода заносятся в файл, открытый для записи. Для текстовых файлов используются также операторы Readln и Writeln с соответствующими дополнениями, относящимися к файловому вводу-выводу.

Любой файл конечен и продолжать чтение из него информации можно лишь до определенного предела. Проверить, окончен ли файл, можно вызовом стандартной логической функции Eof:

Eof (<Имя файловой переменной>)

Функция Eof принимает значение True, если файл окончен, и False - в противном случае.

Пример 1.

Составить программу, которая записывает в файл фамилии учащихся, а затем считывает данные из файла и выводит все фамилии, кроме тех, которые начинаются с буквы 'Ш'.

Program PR1;

Var I, N : Integer; F : File Of String; S : String;

BEGIN

Assign(F,'Spis.lst'); *{Связываем переменную F с файлом Spis.lst}*

Writeln('Введите количество учащихся'); Readln(N);

Rewrite(F); *{Открываем файл для записи в него данных}*

For I:=1 To N Do Begin

Writeln('Введите фамилию'); Readln(S);

Write(F,S); *{Записываем в файл введенную фамилию}*

End;

Close(F); *{закрываем файл}*

Reset(F); *{Открываем файл для чтения}*

```

Writeln; Writeln('Список учащихся:');
While Not(Eof(F)) Do Begin      {пока не достигнут конец файла}
    Read(F,S);                  {считываем очередную фамилию}
    If S[1]<>'Ш' Then Writeln(S) {если 1-й символ не 'Ш', то выводим}
End;
Close(F) {закрываем файл}
END.

```

Пример 2.

Записать в файл квадраты первых 10 натуральных чисел. Прочитать данные из файла.

```

Program PrimFile;
Var f:file of integer; i,x:integer;
BEGIN
    assign(f,'kvadr.dat');      {Связываем переменную F с файлом 'kvadr.dat'}
    rewrite(f);                 {Открываем файл для записи в него данных}
    for i:=1 to 10 do begin
        x:=i*i;                 {вычисляем квадрат очередного натурального числа}
        write(f,x);             {Записываем в файл }
    end;
    close(f);                   {закрываем файл}
    reset(f);                   {Открываем файл для чтения}
    While not eof(f) do begin   {пока не достигнут конец файла}
        read(f,x);              {считываем очередное значение}
        write(x:4);             {выводим на экран}
    end;
    close(f);                   {закрываем файл}
    readln
END.

```

Пример 3.

Записать в файл следующие данные о студентах: ФИО, группа, оценка. Просмотреть данные из файла и найти среднюю оценку в 101 группе.

```

Program pr3;
Type
    vedom= record
        fio:string[30]; {фамилия}
        gr:string[5];   {группа}
        oc:byte;        {оценка}
    end;
Var
    f: file of vedom; {файловая переменная}
    n,s,i,k:integer; {n- общее количество студентов, S- суммарная оценка в 101 гр, k-количество студентов 101 гр }

```

```

x:vedom;    begin
assign(f,'vedomost.dat');  {Связываем переменную F с файлом 'vedomost.dat'}
rewrite(f);                {Открываем файл для записи в него данных}
write('N='); readln(N);    {вводим количество студентов}
for i:=1 to n do begin    {вводим данные о каждом студенте}
  write('fio:'); readln(x.fio);
  write('gruppa:'); readln(x.gr);
  write('ocenka:'); readln(x.oc);
  write(f,x); {}
end;
close(f);                  {закрываем файл}
reset(f);                  {Открываем файл для чтения}
s:=0; k:=0;
while not eof(f) do begin {пока не достигнут конец файла}
  read(f,x); {считываем очередную запись}
  if x.gr='101' then begin s:=s+x.oc; k:=k+1 end;
end;
if k<> 0 then writeln('Sred.ocenka 101 gr =', s/k:4:2) else writeln('101 gr net');
close(f);
readln;
END.

```

Задания для самостоятельной работы

1. Протабулировать функцию $y = \sin x + \cos x$ для $0 \leq x \leq 1.5$ с шагом 0.1, записать данные в файл и просмотреть данные из файла.
2. Протабулировать функцию $y = x^2 + 1/x$ для $1 \leq x \leq 5$ с шагом 0.2, записать данные в файл и просмотреть данные из файла.
3. Протабулировать функцию $y = \ln x - x^2$ для $3 \leq x \leq 4$ с шагом 0.1, записать данные в файл и просмотреть данные из файла.
4. Протабулировать функцию $y = e^x + e^{-x}$ для $-2 \leq x \leq 2$ с шагом 0.2, записать данные в файл и просмотреть данные из файла.
5. Протабулировать функцию $y = \sin x^2$ для $-\pi \leq x \leq \pi$ с шагом $\pi/10$, записать данные в файл и просмотреть данные из файла.
6. Протабулировать функцию $y = \sin(x + x^2)$ для $-2\pi \leq x \leq 2\pi$ с шагом $\pi/10$. Записать данные в файл и просмотреть данные из файла.
7. Создать файл, содержащий значения x ($0 < x < 2$, $h = 0,2$) и значения функций $y = \operatorname{tg}(x)$ и $y = \operatorname{tg}^2(x)$. Вывести на экран данные в виде таблицы.
8. Создать файл, содержащий числа a , b и их произведение. Вывести на экран данные в виде таблицы.
9. Создать файл, содержащий числа a , b и их частное. Вывести на экран данные в виде таблицы.
10. Создать файл, содержащий числа a , b и их сумму. Вывести на экран данные в виде таблицы.

11. Создать файл, содержащий числа x , y и их удвоенную сумму. Вывести на экран данные в виде таблицы.

12. Создать файл, содержащий числа a , b и квадрат их суммы. Вывести на экран данные в виде таблицы.

13. Создать файл, содержащий числа x , y и их полусумму. Вывести на экран данные в виде таблицы.

14. Создать файл, содержащий значения x ($1 < x < 4$, $h = 0,5$) и значения функции $y = \cos(x^2)$. Вывести на экран данные в виде таблицы и определить, при каком значении x функция принимает максимальное значение.

15. Создать файл, содержащий значения x ($0 < x < 10$, $h = 0,5$) и значения функции $y = \sin(x)$ и $y = \cos(x)$. Вывести на экран данные в виде таблицы.

16. Создать файл, содержащий значения x ($-3 < x < 3$, $h = 0,2$) и значения функции $y = \sin(e^x)$. Вывести на экран данные в виде таблицы и определить, при каком значении x функция принимает максимальное значение.

17. Создать файл, содержащий значения x ($-\pi < x < \pi$, $h = \pi/10$) и значения функции $y = \sin(x)$ и $y = \cos(x)$. Вывести на экран данные в виде таблицы.

18. Создать файл, содержащий значения x ($-2 < x < 7$, $h = 0,5$) и значения функции $y = \sin^2(\operatorname{tg} x)$. Вывести на экран данные в виде таблицы и определить, при каком значении x функция принимает максимальное значение.

19. Протабулировать функцию $y = |\sin x + \cos x|$ для $-\pi < x < \pi$ с шагом $\pi/10$, записать данные в файл и просмотреть данные из файла.

20. Протабулировать функцию $y = \sin |(x - \cos x)|$ для $0 \leq x \leq 1,5$ с шагом 0,1, записать данные в файл и просмотреть данные из файла.

21. Протабулировать функцию $y = |3 \sin x^2|$ для $-2 \leq x \leq 1,5$ с шагом 0,1, записать данные в файл и просмотреть данные из файла.

22. Протабулировать функцию $y = \sin(x + 1/x)$ для $2 \leq x \leq 5$ с шагом 0,2, записать данные в файл и просмотреть данные из файла.

23. Протабулировать функцию $y = 1/(x + x^2)$ для $1 \leq x \leq 10$ с шагом 0,5, записать данные в файл и просмотреть данные из файла.

24. Протабулировать функцию $y = 1/e^x + 1/e^{-x}$ для $-2 \leq x \leq 1,5$ с шагом 0,1, записать данные в файл и просмотреть данные из файла.

25. Протабулировать функцию $y = e^x + \cos x$ для $-1 \leq x \leq 1,5$ с шагом 0,1, записать данные в файл и просмотреть данные из файла.

26. Записать в файл следующие данные о студентах: ФИО, группа, оценка. Просмотреть данные из файла и найти количество отличников в 301 группе. Используя созданный файл, сформировать два новых файла, содержащие сведения о студентах 301 и студентах 302 групп. Просмотреть данные из этих файлов.

27. Создать файл, содержащий данные: ФИО работника; год рождения; стаж работы. Просмотреть данные из файла. Определить фамилию работника, имеющего самый большой стаж работы.

28. Создать файл, содержащий данные: ФИО работника; пол; возраст. Просмотреть данные из файла. Определить средний возраст мужчин.

29. Создать файл, содержащий данные: ФИО ученика; пол; возраст. Просмотреть записи. Определить средний возраст девочек.

30. Создать файл, содержащий данные: ФИО ученика; класс; оценка. Просмотреть записи. Определить среднюю оценку в 9-м классе.

31. Создать файл, содержащий данные: ФИО студента; группа; оценка по информатике. Просмотреть записи. Определить среднюю оценку по информатике в 302 группе.

32. Создать файл, содержащий данные: ФИО работника; цех; стаж работы. Просмотреть записи. Определить средний стаж работы работников 1 - го цеха.

33. Создать файл, содержащий данные: ФИО работника; пол; количество детей. Просмотреть данные из файла. Определить количество женщин, имеющих более двух детей.

34. Создать файл, содержащий данные: ФИО ученика; пол; рост. Просмотреть данные из файла. Определить средний рост девочек.

35. Создать файл, содержащий данные: ФИО ученика; класс; пол. Просмотреть данные из файла. Определить количество мальчиков в 5 - классе.

36. Создать файл, содержащий данные: ФИО студента; группа; оценка по информатике. Просмотреть данные из файла. Определить количество '4' и '5' в 301 группе.

37. Создать файл, содержащий следующие данные: ФИО студента; пол; группа. Вывести данные из файла на экран. Подсчитать количество юношей и количество девушек 301 группы. Создать новый файл данных, содержащий сведения о студентах 302 группы (используя существующий файл данных).

38. Создать файл, содержащий следующие данные: ФИО студента; год рождения; группа; наличие золотой или серебряной медали. Вывести данные из файла на экран. Подсчитать количество медалистов. Создать новый файл данных, содержащий сведения о студентах, имеющих золотую медаль (используя существующий файл данных).

39. Создать файл, содержащий следующие данные: ФИО рабочего; количество отработанных дней; сумма зарплаты. Вывести данные из файла на экран. Подсчитать общую сумму зарплаты, выданной рабочим. Создать новый файл данных, содержащий сведения о рабочих, зарплата которых менее заданного значения.

40. Создать файл, содержащий следующие данные: наименование товара; количество; цена; сумма. Вывести данные из файла на экран. Подсчитать итоговую сумму. Создать новый файл данных, содержащий сведения о товарах, цена которых более заданной суммы.

41. Создать файл, содержащий следующие данные: автор; название книги; год издания. Вывести данные из файла на экран. Определить фамилию автора, чья книга вышла раньше других. Создать новый файл данных, содержащий сведения о книгах, издававшихся до 1991 года.

42. Создать файл, содержащий данные о багаже пассажиров: фамилия пассажира; количество вещей; общий вес. Записать в новый файл тех

пассажиров, общий вес вещей которых превышает 36кг (одновременно вывести данные на экран).

43. Создать файл, содержащий данные об учениках школы: фамилия, имя; класс (1,2,...11); буква (А,Б,В,...). Определить в файле, созданном при решении задачи, сколько учащихся в 5Б и 10А классах. Записать в новый файл список учеников 11А класса. Параллельно вывести этот список на экран.

44. Создать файл, содержащий сведения об автомобилях: марка; номер в виде четырёхзначного числа; фамилия владельца. Определить, сколько автомобилей имеют номера, начинающиеся с двух нулей. Вывести данные об этих автомобилях на экран. Записать в новый файл сведения о владельцах и номерах для автомобилей BMW. Параллельно вывести список на экран.

45. Создать файл, содержащий сведения об учениках школы: ФИО; класс; отметки по предметам (русский язык, математика, чтения, рисования, музыка). Определить, сколько отличников в школе. Вывести на экран их фамилии и класс. Записать в новый файл оценки учеников 1А класса. Для визуального контроля выводите параллельно сведения об учениках на экран.

46. Создать файл, содержащий сведения о днях рождения: ФИО; число; месяц; год. В созданном файле определить, кто отмечает день рождения в июле. Записать в новый файл данные о людях, родившихся весной.

47. Создать файл, содержащий сведения о книгах: фамилия автора; книга; год издания. В созданном файле определить количество книг автора Фаронова. Вывести на экран сведения об этих книгах. Записать в новый файл данные о книгах с названием “Информатика”. Параллельно вывести эти данные на экран.

48. Создать файл, содержащий сведения о телефонах: фамилия имя отчество; телефон; адрес. В созданном файле определить телефон по фамилии. Записать в новый файл данные о телефонах начинающихся с цифры 5. Параллельно вывести эти данные на экран.

49. Создать файл, содержащий сведения о кубиках: размер ребра; цвет; материал. В созданном файле определить количество деревянных кубиков с ребром 3см и количество металлических кубиков с ребром более 5см. Записать в новый файл данные о деревянных кубиках. Параллельно вывести эти данные на экран.

50. Создать файл, содержащий сведения об экспортируемых товарах: наименование товара; страна импортирующая товар; объем поставки в штуках. В созданном файле найти страны, в которые экспортируется данный товар. Записать в новый файл сведения только о тех товарах, общий суммарный объем экспорта которых превышает 1000000 шт.

ПРОГРАММИРОВАНИЕ МАТЕМАТИЧЕСКИХ ЗАДАЧ.

Пример 1.

Найти значение суммы $\frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \dots$ с точностью $E=0.00001$ и определить количество слагаемых этой суммы.

Найти сумму с заданной точностью - это значит добавлять к сумме по одному слагаемому до тех пор, пока разница между полученным и предыдущим значением суммы не станет меньше точности ϵ .

$$\sum_{k=1}^n \frac{1}{k!} - \sum_{k=1}^{n-1} \frac{1}{k!} < \epsilon$$

Разность между полученным и предыдущим значением суммы – это последнее слагаемое, которое было добавлено к сумме.

$$\sum_{k=1}^n \frac{1}{k!} - \sum_{k=1}^{n-1} \frac{1}{k!} = \frac{1}{n!}$$

Таким образом, если очередное слагаемое больше заданной точности ($\frac{1}{n!} > \epsilon$), то его следует добавить к сумме.

Program Pr1;

Var S, E: real;

n, k: integer;

{функция, вычисляющая факториал числа k}

Function Fakt(k: integer): real;

Var i: integer; f: real;

Begin

f:=1;

For i:=1 to k do

f:=f*i;

Fakt:=f;

End;

BEGIN

S:=0; n:=0; k:=2; E:=0.00001;

While 1/Fakt(k)>=E do begin

S:=S+1/Fakt(k);

n:=n+1; k:=k+1;

end;

Writeln('S=', s:8:5, ' n=', n)

END.

Пример 2.

Решение квадратного уравнения вида $Ax^2 + Bx + C = 0$.

Program pr2;

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

BEGIN

Write('a, b, c='); readln(a, b, c); *{ввод коэффициентов квадратного уравнения}*

d:=sqr(b)-4*a*c; *{вычисляем дискриминант}*

if d>0 then *{если дискриминант положительный, то вычисляем два корня}*

begin

x1:=(-b+sqrt(d))/(2*a); writeln('x1=', x1:6:2);

```

        x2:=(-b-sqrt(d))/(2*a); writeln('x2=', x2:6:2)
    end
    else if d=0 then      { иначе, если дискриминант равен 0, то
                           вычисляем один корень}
        begin
            x1:=-b/(2*a); writeln('x=', x1:6:2)
        end
        else writeln('Корней нет'); {иначе выводим сообщение
                                     'Корней нет' }

    readln
END.

```

Пример 3.

Перевод числа из десятичной системы счисления в двоичную.

```

Program pr2;
Var
    N,r:integer;
    n2,St:string;
BEGIN
    Write('N=');Readln(N);
    While N>0 do begin
        R:= n mod 2;  str(r,St);
        n2:=st+n2;  n:=n div 2;
    end;
    write(n2);
    readln
END.

```

В программе реализован алгоритм перевода путем деления десятичного числа N на 2 с остатком до тех пор, пока N остается больше 0. Полученные при делении остатки R преобразуются в строковое значение St и добавляются к строке n2. Полученная строка n2 и является записью числа N в двоичной системе счисления.

Пример 4.

Два треугольника длинами своих сторон. Определить, площадь какого из них больше (создать функцию для вычисления площади треугольника по длинам его сторон).

Для решения задачи используем формулу Герона

$$S = \sqrt{p \cdot (p-x) \cdot (p-y) \cdot (p-z)}, \text{ где } x, y, z - \text{ стороны треугольника, } p - \text{ полупериметр)}$$

```

Program pr3;
var
    a1, b1, c1, s1, a2, b2, c2, s2: real;
    {функция вычисления площади треугольника со сторонами x,y,z}
Function PTr(x ,y, z: real): real;
Var

```

```

    p: real;
Begin
    p:=(x + y + z)/2;           {полупериметр треугольника}
    PTr:=Sqrt(p*(p-x)*(p-y)*(p-z));
End;
BEGIN           {основная программа}
    {ввод исходных данных}
    Write('Стороны 1-го треугольника:'); Readln(a1, b1, c1);
    Write('Стороны 2-го треугольника:'); Readln(a2, b2, c2);
    S1:=PTr(a1, b1, c1);{вызов функции для нахождения площади 1-го треугольника}
    S2:=PTr(a2, b2, c2);{вызов функции для нахождения площади 2-го треугольника}
    if S1>S2 then Writeln('S1>S2')
        else if S1<S2 then Writeln('S1<S2')
            else Writeln('S1=S2');

    Readln;
END.

```

Пример 5.

Создать функцию нахождения наибольшего общего делителя двух натуральных чисел.

Решение:

Program Pr5;

Var

a,b:integer;

Function NOD(a,b:integer):integer;

begin

while a<>b do {пока числа не равны}

{заменяем большее из них разностью большего и меньшего}

if a>b then a:=a-b else b:=b-a;

NOD:=a;

end;

BEGIN

write('a,b='); readln(a,b); {ввод исходных данных}

write('NOD=', NOD(a,b)); {вызов функции и вывод ее значения}

readln;

END.

Пример 6

Рассчитать период (в месяцах) роста банковского вклада в **m** раз при известной ежемесячной процентной ставке **pr** (1% соответствует **pr=1**).

Program Pr10;

Var

pr,m,s,s0: real;

N: word;

BEGIN

```

Repeat
  Write ('Введи проценты по вкладу:'); readLn(pr);
  Write ('Введи коэффициент роста вклада:'); readLn(m)
Until (m>1) and (pr>0);      {проверка вводимых данных}
N:=0;                        {счетчик месяцев}
s0:=1000;                     {начальная сумма вклада – произвольно}
s:=s0;                        {текущая сумма вклада}
  Repeat                      {перебор по месяцам}
    N:=N+1;                   {еще один месяц прошел}
    s:=s*(1+pr/100);          {сумма через месяц}
  Until s/s0>=m;             {до тех пор, пока сумма не превысит начальную в m раз}
WriteLn ('количество месяцев вклада:',N);
ReadLn
END.

```

Задания для самостоятельной работы

1. Заданы стороны двух треугольников ABC и PMN. Сравнить площади этих треугольников (создать функцию пользователя для вычисления площади треугольника по длинам его сторон с помощью формулы Герона).

2. Вычислить площадь P произвольного плоского шестиугольника, заданного координатами вершин.

3. Два треугольника заданы координатами своих вершин A1, B1, C1 и A2, B2, C2. Вычислить площади треугольников, используя известную формулу из аналитической геометрии на плоскости, и определить, какой треугольник имеет большую площадь. Вычисление площади треугольника оформить в виде функции пользователя.

4. Сократить дробь вида a/b (a, b – вводимые натуральные числа), создав функцию для нахождения наибольшего общего делителя двух натуральных чисел.

5. Найти наибольший общий делитель чисел A, B, C, создав функцию для нахождения НОД двух натуральных чисел.

6. Найти НОК(наименьшее общее кратное) двух натуральных чисел.

7. Чего больше: всех возможных трехзначных чисел, записанных цифрами 2, 3, 4, 5 или всех четырехзначных чисел, записанных цифрами 1, 3, 7, 8, 9? Подсчет производить по формуле размещений:

$$A_n^k = \frac{n!}{(n-k)!}$$

и оформить в виде подпрограммы.

8. Чего больше: всех возможных трехзначных чисел, записанных цифрами 1, 2, 3, 4, 5 или всех двухзначных чисел, записанных цифрами 2, 4, 6, 8? Подсчет производить по формуле размещений:

$$A_n^k = \frac{n!}{(n-k)!}$$

и оформить в виде подпрограммы.

9. Сколькими способами можно отобрать команду в составе 5 человек из 8 кандидатов; из 10 кандидатов; из 11 кандидатов? Подсчет количества способов отбора оформить в виде подпрограммы по формуле сочетаний

$$C_n^k = \frac{n!}{k!(n-k)!}$$

10. Приписать к числу 523 три такие цифры справа, чтобы полученное число делилось на 7, 8 и 9.

11. Найти трехзначное число abc, для которого $a!+b!+c! = abc$.

12. Найти все трехзначные числа, которые являются полными квадратами и записываются четными цифрами.

13. Найти четырехзначное число, являющееся точным квадратом, у которого первые две цифры одинаковые и две последние тоже одинаковые.

14. Найти трехзначное число, квадрат которого оканчивается тремя одинаковыми цифрами, отличными от 0.

15. Приписать к числу 523 три такие цифры справа, чтобы полученное число делилось на 7, 8, 9.

16. Квадрат трехзначного числа оканчивается тремя цифрами, которые как раз составляют взятое число. Найти все такие числа.

17. В трехзначном числе, все цифры которого нечетны, зачеркнули среднюю цифру. Оказалось, что полученное двузначное число является делителем исходного числа. Найти все такие трехзначные числа.

18. Четырехзначное число, а также число, записанное теми же цифрами в обратном порядке, оба являются точными квадратами. Найти эти числа.

19. Приписать к числу 999 слева три такие цифры, чтобы полученное шестизначное число делилось на 13, 17 и 19.

20. Найти все натуральные числа, не превосходящие 600, у которых сумма цифр является делителем самого числа.

21. Найти все трехзначные числа, которые являются полными квадратами и записываются тремя нечетными цифрами.

22. Найти четырехзначное число, равное квадрату суммы двух двузначных чисел, образованных двумя первыми и двумя последними цифрами числа.

23. Рассмотрим произвольное натуральное число и найдем сумму его цифр, затем сумму цифр полученного числа и так далее, пока не получим однозначное число. Назовем это число цифровым корнем. Напишите программу, запрашивающую с клавиатуры натуральное число N и вычисляющую его цифровой корень.

24. Натуральное число из n цифр является числом Амстронга, если сумма его цифр, возведенных в n-ю степень, равна самому числу (например,

$153=1^3+5^3+3^3=1+125+27$). Напишите программу, печатающую все числа Амстронга, состоящие из двух, трех и четырех цифр.

25. Составить программу определяющую, какие натуральные числа от 2 до 10 являются простыми, а какие - составными (создать процедуру, определяющую простым или составным является данное число). Число называется простым, если оно не имеет делителей кроме 1 и самого себя. Если у него есть другие делители, то число – составное.

26. Натуральное число называется совершенным, если оно равно сумме всех своих делителей, за исключением самого себя. Например, число 6 - совершенное, так как $1+2+3=6$, а 8 не является совершенным ($1+2+4$ не равно 8). Составить программу, запрашивающую с клавиатуры натуральное число и печатающую все совершенные числа, не превышающие введенного числа (создать процедуру, которая определяет является ли совершенным данное число).

27. Составить программу для перевода заданного натурального числа N из десятичной системы счисления в двоичную.

28. Составить программу для перевода заданного натурального числа N из десятичной системы счисления в троичную.

29. Составить программу для перевода заданного натурального числа N из десятичной системы счисления в 8-ричную.

30. Составить программу для перевода числа из двоичной системы счисления в десятичную (двоичное число вводится как строка).

31. Составить программу для перевода числа из 8-ричной системы счисления в десятичную (8-ричное число вводится как строка).

32. Составить программу для перевода числа из 16-ричной системы счисления в десятичную (16-ричное число вводится как строка).

33. Составить программу, содержащую функцию вычисления экспоненты в виде бесконечного ряда с точностью 10^{-9} .

$$\exp(x) = 1 + \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^N}{N!} + ..$$

В основной программе организовать вычисление этого ряда для двух значений x, запрашиваемых с клавиатуры, и проверку получаемых результатов путем сравнения с системной функцией EXP(X).

34. Составить программу, содержащую функцию вычисления косинуса в виде бесконечного ряда с точностью 10^{-7} .

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots + (-1)^N \frac{x^{2N}}{(2N)!} + ..$$

В основной программе организовать вычисление этого ряда для двух значений x, запрашиваемых с клавиатуры, и проверку получаемых результатов путем сравнения с системной функцией COS(X).

35. Составить программу, содержащую функцию вычисления синуса в виде бесконечного ряда с точностью 10^{-8} .

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots + (-1)^N \frac{x^{2N+1}}{(2N+1)!} + \dots$$

В основной программе организовать вычисление этого ряда для двух значений x , запрашиваемых с клавиатуры, и проверку получаемых результатов путем сравнения с системной функцией $\text{SIN}(X)$.

36. Составить программу, содержащую функцию вычисления гиперболического синуса в виде бесконечного ряда с точностью 10^{-8} .

$$\text{sh}(x) = \frac{e^x - e^{-x}}{2} = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \dots + \frac{x^{2N+1}}{(2N+1)!} + \dots$$

В основной программе организовать вычисление этого ряда для двух значений x , запрашиваемых с клавиатуры, и проверку получаемых результатов путем сравнения с системной функцией $(\exp(X) - \exp(-X))/2$.

37. Составить программу, содержащую функцию вычисления гиперболического косинуса в виде бесконечного ряда с точностью 10^{-7} .

$$\text{ch}(x) = \frac{e^x + e^{-x}}{2} = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \frac{x^6}{6!} + \dots + \frac{x^{2N}}{(2N)!} + \dots$$

В основной программе организовать вычисление этого ряда для двух значений x , запрашиваемых с клавиатуры, и проверку получаемых результатов путем сравнения с системной функцией $(\exp(X) + \exp(-X))/2$.

38. Найти значение суммы $S = \frac{1}{3!} + \frac{2}{4!} + \dots$ с точностью $\varepsilon = 10^{-6}$.

39. Найти значение суммы $S = \frac{1}{2!} + \frac{3}{4!} + \dots$ с точностью $\varepsilon = 10^{-6}$.

40. Найти значение суммы $S = \frac{1}{1+2} + \frac{2}{1+2+3} + \dots$ с точностью $\varepsilon = 10^{-4}$.

41. Составить программу расчета роста по месяцам в течение одного года банковского вклада. Программа запрашивает с защитой от неверного ввода данных следующую информацию:

- начальный размер вклада (2000...20000),
- размер процентной ставки по вкладу (1% ... 3% в месяц).

Вывести таблицу роста вклада по месяцам, а также определить количества месяцев, необходимые для роста вклада в полтора раза.

42. Составить программу расчета роста по месяцам в течение двух лет банковского вклада. Программа запрашивает с защитой от неверного ввода данных следующую информацию:

- начальный размер вклада (5000...30000),
- размер процентной ставки по вкладу (0.5% ... 2.5%),
- размер периодических платежей (от 0.5% до 5% от начального вклада).

Вывести таблицу роста вклада по месяцам, а также определить количества месяцев, необходимые для роста начальной суммы вклада на 50%.

43. Составить программу расчета роста по месяцам банковского вклада. Программа запрашивает с защитой от неверного ввода данных следующую информацию:

- начальный размер вклада (6000...60000),
- размер процентной ставки по вкладу (0.6% ... 6%),
- количество месяцев, в течение которого проводится расчет (от 6 до 24).

Вывести таблицу роста вклада по месяцам, в которую включить дополнительный столбец роста вклада в предположении отсутствия периодических платежей.

44. Вывести в виде таблицы значения функции $Y = \sin X^2 - \cos X$ для $X=0, 0.1, 0.2, \dots, 2$. Значения X выводить с 1 десятичным знаком, а значения Y – с 3 десятичными знаками. После таблицы вывести значения Y_{\max} .

45. Вывести в виде таблицы значения функции $Y = e^x - 2 \cos X$ для $X=0, 0.05, 0.1, \dots, 1$. Значения X выводить с 2 десятичными знаками, а значения Y – с 4 десятичными знаками. Определить среднее арифметическое отрицательных значений функции.

46. Вывести в виде таблицы значения функции $Y = \cos(X + X^2)$ для $X=-1, -0.9, \dots, 1$. Значения X выводить с 1 десятичным знаком, а значения Y – с 4 десятичными знаками. Найти сумму положительных значений функции.

47. Вывести в виде таблицы значения функции $Y = \sin X - (X+1)^2$ для $X=0, 0.1, 0.2, \dots, 2$. Значения X выводить с 2 десятичными знаками, а значения Y – с 3 десятичными знаками. Найти среднее арифметическое положительных значений функции.

48. Вывести в виде таблицы значения функции $Y = \sin X^2 - e^x$ для $X=0, 0.05, 0.1, \dots, 1$. Значения X выводить с 2 десятичными знаками, а значения Y – с 4 десятичными знаками. Определить максимальное значение функции.

49. Вывести в виде таблицы значения функции $Y = \sin X + \cos(X + X^2)$ для $X=0, \pi/20, \dots, \pi$. Значения X выводить с 1 десятичным знаком, а значения Y – с 3 десятичными знаками. Определить минимальное значение функции.

50. Вывести в виде таблицы значения функции $Y = \sin X^2 + X$ для $X=0, \pi/10, \dots, 2\pi$. Значения X выводить с 3 десятичными знаками, а значения Y – с 5 десятичными знаками. После таблицы вывести значения Y_{\max} .

ЛИТЕРАТУРА

1. Аляев Ю.А. и др. Практикум по алгоритмизации и программированию на языке Паскаль: учеб. пособие / Ю.А. Аляев, В.П. Гладков, О.А. Козлов.- Москва: Финансы и статистика, 2004. -528С.
2. Васильев П.П. Турбо Паскаль в примерах и задачах. Освой самостоятельно: учеб. пособие.-Москва: Финансы и статистика, 2003. -496С.
3. Васюкова Н.Д., Тюляева В.Р. Практикум по основам программирования языка Паскаль. Учебное пособие. М.: Высшая школа. 1991.
4. Вирт Н. Алгоритмы и структуры данных. – М.,1989.
5. Грызлов В.И. Турбо Паскаль 7.0. / В.И. Грызлов, Т.М. Грызлова - 4-е изд., испр.-Москва: ДМК Пресс, 2005. -416С.
6. Д. Праймс. Программирование на языке Паскаль. Практическое руководство М.: Мир, 1997.
7. Немнюгин С.А. Turbo Pascal. Программирование на языке высокого уровня: учебник для вузов / С.А. Немнюгин - 2-е изд.-Санкт-Петербург: Питер, 2005.
8. Немнюгин С.А. Turbo Pascal.: практикум.-Санкт-Петербург: Питер, 2001. - 256С.
9. Немнюгин С.А. Turbo Pascal.: учеб. пособие для вузов.-Санкт-Петербург: Питер, 2003. -492С.
10. Немнюгин С.А. Изучаем Turbo Pascal / С.А. Немнюгин, Л.В. Перколаб.- Санкт-Петербург: Питер, 2006. -313С.
11. Программирование на языке Паскаль: задачник / Под ред. Усковой О.- Санкт Петербург: Питер, 2002. -336С.
12. Турбо Паскаль 7. п.: Самоучитель. - 16-е изд. / Под ред. В. Кораблева.- Санкт-Петербург: Питер; Киев: ВНУ, 2004. -479С.
13. Фаронов В.В. Турбо Паскаль 7.0. Практика программирования: Учебное пособие.-Москва: ОМД Групп, 2003. -432С.
14. Фаронов В.В. Turbo Pascal.-Санкт-Петербург: БВХ-Петербург, 2003. -1056С.
15. Фаронов В.В. Турбо Паскаль 7.0. Начальный курс: Учебное пособие.- Москва: ОМД Групп, 2003. -616С.