

Министерство образования и науки Российской Федерации

Нижегородский государственный университет
им. Н.И. Лобачевского

Кузнецова И.А.

**Задания к практическим занятиям по курсу "Информатика и программирование"
(Pascal)**

учебное пособие для студентов
специальности «Прикладная информатика (в экономике)»

Нижний Новгород
2005 г.

УДК 681.3.068

ББК 32.973.26 – 081.1

Кузнецова И. А.

К89 Задания к практическим занятиям по курсу "Информатика и программирование" (Pascal). Учебное пособие для студентов специальности «Прикладная информатика (в экономике)» Н.Новгород: ННГУ, 2005. 57с.

Учебное пособие представляет собой сборник программ и задач для самостоятельного решения в среде разработки Pascal. Рассмотренные решения задач различной степени сложности демонстрируют возможности языка программирования.

Учебное пособие будет полезно студентам вузов технических специальностей, а также школьникам старших классов, учащимся колледжей, студентам техникумов.

Введение

В предлагаемом учебном пособии рассматриваются вопросы использования изучаемого теоретического материала по курсу информатики для решения на практических занятиях и при самостоятельном изучении. Настоящее пособие рекомендуется студентам при подготовке к занятиям, контрольным работам.

Тематическое планирование учебного материала проведено в соответствии с действующей программой. Распределение часов по темам изучаемого курса является условным, а преподаватель вправе изменить темы практических занятий и перераспределить учебное время, отведенное на каждое занятие.

В начале каждого практического занятия кратко приведен теоретический материал, необходимый для решения некоторых задач по данной теме. После него предлагается решение некоторых задач и список задач для самостоятельного выполнения.

Фактический материал учебного пособия неоднократно апробирован автором, а также учениками и студентами, и как показала практика, дает хороший результат при применении его на первых курсах и в старших классах.

Автор.

Практическая работа № 1

Тема: «Структура программы на языках PASCAL. Процедуры ввода-вывода, оператор присваивания»

Структура программы на языке Pascal

Структура программы на языке PASCAL состоит из трёх частей:

- 1) заголовка;
- 2) описательного блока;
- 3) исполнительного блока;

Program <имя программы> (**Input, Output**);

Uses <имя 1>[,<имя 2>...];

Label m1,m2,...;

Const [<константа 1 = значение 1>,...,< константа n = значение n >];

Type [<имя типа1 = тип>,...,< имя типа n = тип>];

Var <переменная 1>[,<переменная 2>,...,<переменная n>]:<тип>;

Procedure <имя процедуры>[(параметры)];

Begin

<тело процедуры>;

End;

Function <имя> (аргументы):<тип значения>;

Begin

<тело функции>;

End;

Begin

<тело программы>;

End.

Процедуры ввода/вывода языка Turbo Pascal

Для выполнения ввода/вывода информации существуют четыре стандартные процедуры:

Read (x1,x2,x3) - ввод переменных x₁, x₂, x₃.

Readln (x1,x2,x3) - ввод переменных x₁, x₂, x₃ с переходом курсора на новую строку.

Write (x1,x2,x3) - вывод на экран значения переменных x₁, x₂, x₃;

Writeln (x1,x2,x3) - вывод на экран значения переменных x₁, x₂, x₃ с переходом курсора на новую строку.

Например, **Write** ('x1=') - выводит на экран комментарий, заключённый в апострофах;

Процедура вывода также выводит на экран результат вычисления арифметического выражения, заключённого в скобках, например:

Write (x1+x2).

Оператор присваивания

Оператор присваивания – это один из основных операторов языка Turbo Pascal. В левой части указывается имя переменной, правая часть – это выражение того же типа, что и переменная. Символы «:=» связывают левую и правую части оператора присваивания и

означают «присвоить значение». Данные символы рассматриваются как один специальный символ и пишутся слитно.

Например: $a:= b + c$;

Примеры решений задач

1. Составить программу «Приветствие».

```
Program Hello;  
begin  
  writeln('Здравствуй, компьютер!');  
  write('Привет,');writeln('студент.')
```

end.

2. Найти сумму двух чисел

```
Program Summa;                                {заголовок программы}  
Var                                           {раздел объявления  
  X,Y, Summa: Real;                          переменных}  
Begin                                         {тело программы}  
  Write('Введите числа X и Y');             {вывод сообщения на  
  Readln (X,Y);                              экран}  
  Summa:=X+Y;                                {чтение двух чисел}  
  Writeln('Сумма чисел X и Y равна',      {определение суммы}  
Summa)                                       {вывод результата}  
End.
```

Практическая работа № 2

Тема: «Простые числовые типы данных»

Тип данных Integer

Действия с целыми числами

В типе данных **Integer** (целое) существует пять подтипов, различающихся:

- множеством значений;
- количеством занимаемой памяти;

<i>Тип</i>	<i>Название</i>	<i>Диапазон значений</i>	<i>Размер</i>
Shortint	Короткое целое	-128,127	8 бит
Integer	Целое	- 32768, 32767	16 бит
Longint	Длинное целое	-2147483648, 2147483647	32 бита
Byte	Байт	0, 255	8 бит
Word	Слово	0, 65535	16 бит

Операции над целыми числами

Арифметические операции применимы только к величинам целых и вещественных типов. У операций умножение, деление, деление нацело более высокий приоритет по сравнению со сложением и вычитанием, то есть они автоматически вычисляются первыми.

Операция **Div** – выполняет целочисленное деление. Операция **Mod** находит остаток от целочисленного деления.

Тип данных Real

Действия с действительными числами

Для представления вещественных чисел имеется следующий набор типов:

<i>Тип</i>	<i>Название</i>	<i>Диапазон значений</i>	<i>Размер</i>
Real	Вещественный	2.9E-39, 1.7E+38	6 байт
Single	Одинарный	1.5E-35, 3.4E+38	4 байта
Double	Двойной	5.0E-324, 1.7E+308	8 байт
Extended	Расширенный	3.4E-4932, 1.1E+4932	10 байт
Comp	Комплексный	-9.2E+18, 9.2E+18	8 байт

Стандартные функции и операции

Abs (x) соответствует $|x|$;

ArcTan (x) соответствует $\arctg(x)$;

Cos (x) соответствует $\cos(x)$;

Sin (x) соответствует $\sin(x)$;

Exp (x) соответствует e^x ;

Ln (x) соответствует $\ln(x)$;

Sqr (x) соответствует x^2 ;

Sqrt (x) соответствует \sqrt{x} ;

Frac (x) - дробная часть: $X - \text{INT}(X)$;

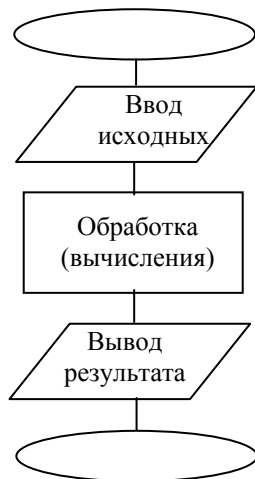
Int (x) возвращает целую часть числа;

Round (x) возвращает число равное целой части (округление по правилам арифметики);

Trunc(x) возвращает округленное число, отбрасывая дробную часть числа x.

Программы с линейной структурой

Программы с линейной структурой являются простейшими и используются, как правило, для реализации простых вычислений по формуле. В программах с линейной структурой инструкции выполняются последовательно. Алгоритм программы с линейной структурой может быть представлен в виде схемы:



Примеры решений задач

1. Составить программу, вычисляющую значение переменной t при данных i, j, k : $t = \frac{i+j}{k+1}$.

```
Program example_1;
```

```
Var i, j, k: Integer;
```

```
    m: Real;
```

```
Begin
```

```
    Write('Введите значения для i, j и k =>:');
```

```
    Readln(i, j, k);
```

```
    m:=(i+j)/(k+1);
```

```
    Writeln ('Значение для m=',m);
```

```
End.
```

Задачи для самостоятельной работы

1. Дан радиус окружности, подсчитать длину окружности.
2. Дан радиус окружности, подсчитать площадь круга.
3. Дан прямоугольный треугольник с катетами a и b . Найти гипотенузу c .
4. Дан произвольный треугольник. Известны стороны a и b и угол между ними α . Найти третью сторону c .
5. Дан произвольный треугольник со сторонами a, b и c . Найти площадь треугольника.
6. Вычислить объём шара радиуса R .
7. Найти среднее арифметическое и среднее геометрическое трёх заданных чисел.
8. Найти расстояние между двумя точками с данными координатами.

9. По ребру найти площадь грани, площадь боковой поверхности и объём куба.
10. Вычислить периметр и площадь правильного 10-угольника, вписанного в окружность заданного радиуса.
11. Для заданного целого числа a напечатать следующую таблицу:
- | | |
|-------|-------|
| a | |
| a^3 | a^6 |
| a^6 | a^3 |
| a | a |
12. Даны два действительных числа a и b . Получить их сумму, разность и произведение.
13. Дана длина ребра куба. Найти объём куба и площадь его боковой поверхности.
14. Даны два действительных положительных числа. Найти среднеарифметическое и среднегеометрическое этих чисел (или их модулей).
15. Даны катеты прямоугольного треугольника. Найти его гипотенузу и площадь.
16. Смешано V_1 литров воды температуры t_1 и V_2 литров воды температуры t_2 . Найти объём и температуру образовавшейся смеси.
17. Определить периметр правильного n -угольника, описанного около окружности радиуса r .
18. Дана сторона равностороннего треугольника. Найти площадь этого треугольника.
19. Даны гипотенуза и катет прямоугольного треугольника. Найти катет и радиус вписанной окружности.
20. Найти сумму членов арифметической прогрессии по данным значениям: a, d, n .
21. Треугольник задан длинами сторон. Найти:
- а) длины высот;
 - б) длины биссектрис;
 - в) длины медиан;
 - г) радиусы вписанной и описанной окружностей.
22. Вычислить расстояние между двумя точками с координатами x_1, y_1 и x_2, y_2 .
23. Даны целые числа k, m , действительные числа x, y, z . При $k < m/2$ или заменить модулем соответственно значения x, y или z , а два других уменьшить на 0.5.
24. Треугольник задан координатами своих вершин. Найти периметр и площадь треугольника.
25. Дано действительное число x . Получить целую часть x ; затем число x , округлённое до ближайшего целого; затем число без дробных цифр.
26. Даны действительные числа x, y . Вычислить расстояние от точки плоскости с координатами (x, y) до границы квадрата с вершинами:
- а) $(-0.5, -0.5), (-0.5, 0.5), (0.5, 0.5), (0.5, -0.5)$;
 - б) $(0, 0), (0, 1), (1, 1), (1, 0)$.
- Имеется в виду минимум расстояний от данной точки до точек квадрата.
27. Даны целые (либо вещественные) числа $x_1, y_1, x_2, y_2, x_3, y_3$. Известно, что точки с вершинами $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ являются тремя вершинами некоторого прямоугольника. Найти координаты четвёртой вершины.

Практическая работа № 3

Тема: «Условный оператор. Оператор многозначного ветвления»

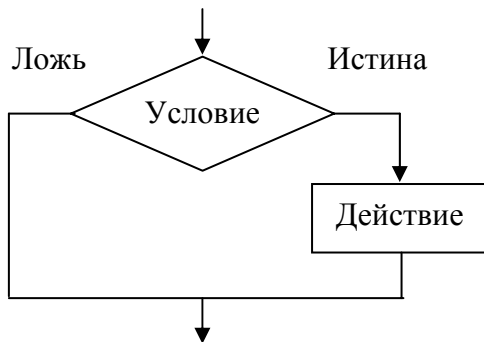
При описании разветвляющихся процессов обычно используют понятие условного и безусловного перехода. Если в программе требуется нарушить порядок выполнения операторов без предварительных проверок каких-либо условий, переход называется безусловным. Для реализации такого перехода служит оператор **goto n** (n – метка). В Паскале метка должна быть описана в разделе **label**, например:

label m, metka, 123

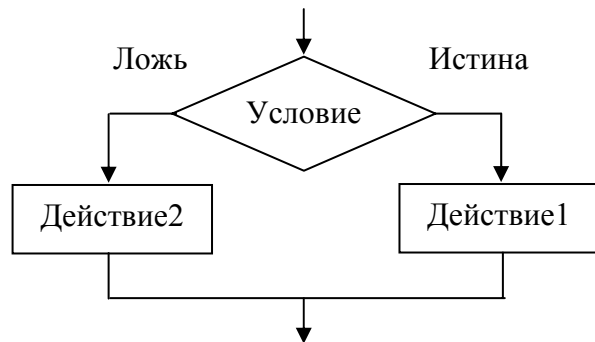
Однако современный стиль программирования предполагает как можно более редкое применение безусловного перехода, а еще лучше - полный отказ от него.

Условный оператор **IF** предназначен для изменения порядка выполнения операторов в зависимости от истинности или ложности некоторого условия. Он предписывает выполнять некоторое действие только в том случае, когда выполняется заданное условие. Это условие записывается в виде логического выражения, а действие, которое нужно выполнить, задается в виде последовательности операторов. Существует две конструкции оператора ветвления – простая и расширенная:

Простая конструкция



Расширенная конструкция



Полная развилка:

```
If < условие> then < оператор 1 >  
    else < оператор2 >;
```

Укороченная развилка

```
If < условие> then < оператор 1 > .
```

Оператор выбора CASE

В случае необходимости разветвить вычислительный процесс в зависимости от выполнения или невыполнения того или иного условия на более чем две ветви используется оператор выбора (случая, селектора, переключателя). Его использование оказывается более удобным по сравнению с использованием оператора if.

Case S of

C1: <Оператор1>

C2: <Оператор2>

.....

CN: <ОператорN>

Else <Оператор>

End;

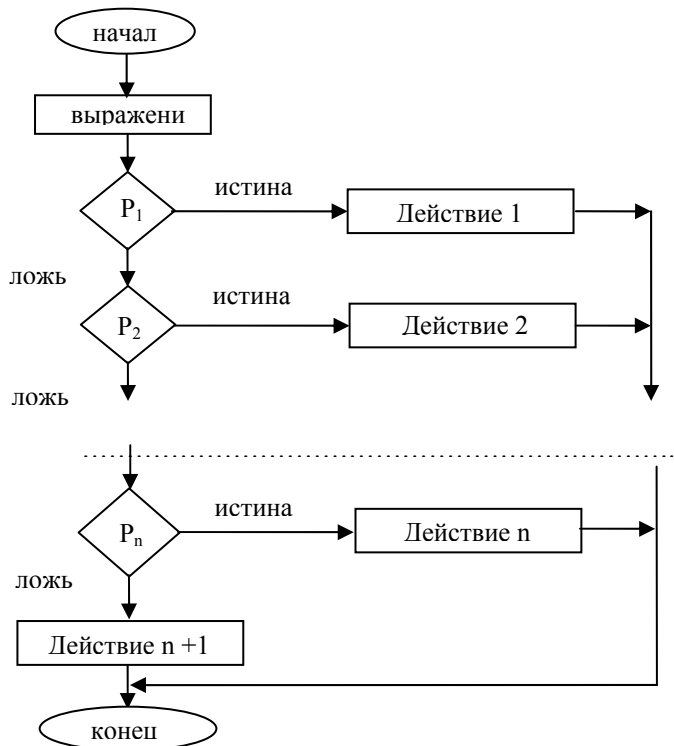
S - выражение порядкового типа значение которого вычисляется;

C1, C2, ..., CN – константы, с которыми сравнивается значение выражения S;

<Оператор1>, <Оператор2>, <Оператор N> - операторы, из которых выполняется тот, с константой которого совпадает значение выражения S. Ветвь оператора *else* является необязательной. Если она отсутствует и значение выражения S не совпадает ни с одной константой, весь оператор рассматривается как пустой.

Если для нескольких констант нужно выполнить один и тот же оператор, их можно перечислить через запятую, сопроводив их одним оператором.

Схематически такую конструкцию можно изобразить следующим образом:



Примеры решений задач

1. Даны действительные числа x и y . Получить $\max(x,y)$.

Program maximum;

Var

x,y :real;

Begin

Write('Введите x и y ');

Readln(x,y);

If $x > y$ **then** $m := x$ **else** $m := y$;

Write(' Максимальное $m =$ ' m);

End.

2. Ввести число от 1 до 100, если введенное число попадет в диапазон $[1..10]$ определить его четность.

Program Chisla;

Var

I : integer;

Begin

Write('Введите число');

Readln(i);

Case i of

2, 4, 6, 8: **Writeln**('Четная цифра')

1, 3, 5, 7, 9: **Writeln**('Нечетная цифра')

10..100: **Writeln**('Число от 10 до 100')

else

Writeln ('Отрицательное число или больше 100')

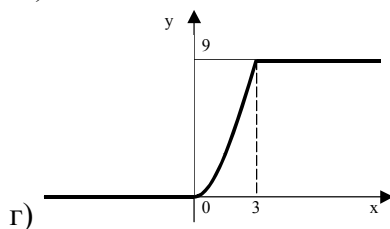
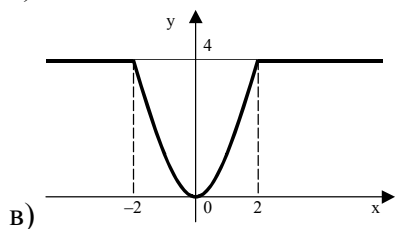
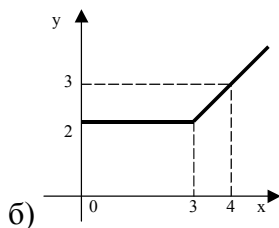
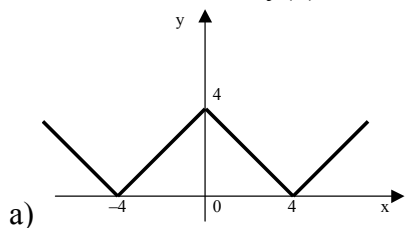
end;

end.

Список задач

1. Даны действительные числа x, y . Получить:
 - а) $\max(x, y)$,
 - б) $\min(x, y)$,
 - в) $\max(x, y), \min(x, y)$.
2. Даны действительные числа x, y, z . Вычислить:
 - а) $\max(x + y + z, x \cdot y \cdot z)$,
 - б) $\min^2(x + y + z/2, x \cdot y \cdot z) + 1$.
3. Даны действительные числа a, b, c . Проверить выполняется ли неравенство $a < b < c$.
4. Найти \min значение из трёх величин, определяемых арифметическими выражениями $a = \sin(x)$, $b = \cos(x)$, $c = \ln(x)$ при заданных значениях x .
5. Даны действительные числа a, b, c . Удвоить эти числа, если $a > b > c$ и заменить их абсолютными значениями, если это не так.
6. Даны два действительных числа. Заменить первое число нулём, если оно меньше или равно второму, и оставить числа без изменения иначе.
7. Даны действительные числа x, y . Меньшее из этих двух чисел заменить их полусуммой, а большее – их удвоенным произведением.
8. Даны действительные числа a, b, c, d . Если $a < b < c < d$, то каждое число заменить наибольшим из них; если $a > b > c > d$, то числа оставить без изменения; иначе все числа заменяются их квадратами.
9. Даны действительные числа a, b, c . Выяснить, имеет ли уравнение $ax^2 + bx + c = 0$ действительные корни. Если действительные корни имеются, то найти их. В противном случае ответом должно служить сообщение, что действительных корней нет.
10. Даны действительные положительные числа a, b, c, x, y . Выяснить, пройдёт ли кирпич с рёбрами a, b, c в прямоугольное отверстие со сторонами x и y . Просовывать кирпич в отверстие разрешается только так, чтобы каждое из рёбер было параллельно или перпендикулярно каждой из сторон отверстия.
11. Даны два действительных числа. Вывести первое число, если оно больше второго, и оба числа если это не так.
12. Даны три действительных числа. Выбрать из них те, которые принадлежат интервалу $(1, 3)$.
13. Даны три действительных числа. Возвести в квадрат те из них, значения которых не отрицательны.

14. Если сумма трёх попарно различных действительных чисел x, y, z меньше единицы, то наименьшее из этих трёх чисел заменить полусуммой двух других; иначе заменить меньшее из x и y полусуммой двух оставшихся значений.
15. Даны два числа. Если первое число больше второго по абсолютной величине, то необходимо уменьшить первое в 5 раз, иначе оставить числа без изменения.
16. Даны действительные положительные числа x, y, z .
- Выяснить, существует ли треугольник с длинами сторон x, y, z .
 - Если треугольник существует, то ответить - является ли он остроугольным, тупоугольным или прямоугольным.
17. Даны действительные числа $x_1, x_2, x_3, y_1, y_2, y_3$. Принадлежит ли начало координат треугольнику с вершинами $(x_1, y_1), (x_2, y_2), (x_3, y_3)$?
18. Составить программу определения большей площади из двух фигур круга или квадрата. Известно, что сторона квадрата равна a , радиус круга равен r . Вывести и напечатать значение площади большей фигуры.
19. Даны действительные, положительные числа a, b, c, d . Выяснить, можно ли построить четырёхугольник с такими длинами сторон.
20. Определить, является ли целое число чётным.
21. Определить, верно ли, что при делении неотрицательного целого числа a на положительное целое число b , получается остаток равный одному из двух заданных чисел r или s .
22. Вывести значение $y(x)$ в зависимости от введенного значения аргумента:



д) $y(x) = \begin{cases} x^2, & \text{при } |x| < 1 \\ x-1, & \text{при } |x| \geq 1 \end{cases}, \quad \text{е) } y(x) = \begin{cases} x^2 + 5, & \text{при } x < 0 \\ x, & \text{при } x \geq 0 \end{cases}$

Практическая работа № 4

Тема: «Операторы повтора»

При реализации многократного повторения некоторых операций линейной конструкцией необходимо снова и снова повторять одни и те же операторы. Для более компактной реализации этих операций во всех языках используются циклические конструкции, суть которых заключается в том, что вместо многократного переписывания одних и тех же строк программы управление в нужном месте передается предыдущим операторам с тем, чтобы они повторились.

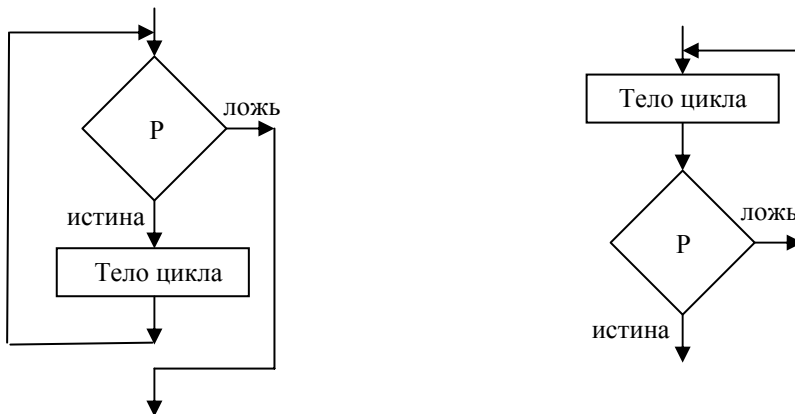
Имеется два вида циклических алгоритмов: цикл с предусловием (цикл ПОКА) и цикл с постусловием (цикл ДО). Оператор цикла **REPEAT** организует выполнение цикла, состоящего из любого числа операторов, с неизвестным заранее числом повторений. Тело цикла выполняется *хотя бы один раз*. Выход из цикла осуществляется при истинности некоторого логического выражения. Структура оператора:

repeat <тело цикла> **until**<условие>;

Оператор цикла **WHILE** организует выполнение одного оператора неизвестное заранее число раз. Выход из цикла осуществляется, если некоторое логическое выражение окажется ложным. Так как истинность логического выражения проверяется в начале каждого повтора, *тело цикла может не выполняться ни разу*. Структура оператора цикла имеет вид:

while <условие> **do** <тело цикла>;

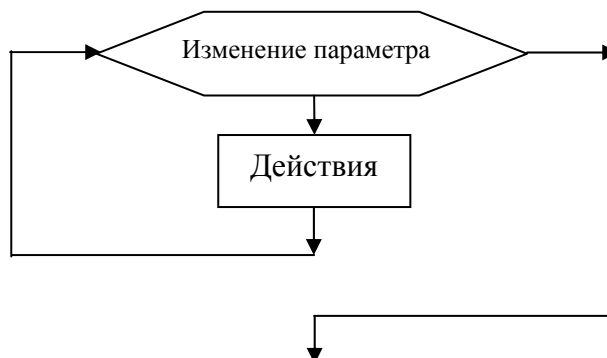
Блок-схемы циклических конструкций могут быть изображены следующим образом:



Оператор цикла **FOR** организует выполнение одного оператора заранее известное число раз. Существует два варианта оператора.

for <переменная>:=<нач. значение> **to** <кон. значение> **do**<оператор>;

for <переменная>:=<нач. значение> **downto** <кон. значение>
do <оператор>;



Выполнение очередного повтора включает в себя сначала выполнение оператора, а затем присваивание переменной цикла следующего большего значения (в первом случае) или следующего меньшего (во втором варианте). Особенностью арифметического цикла является то, что число повторений операторов цикла должно быть известно заранее.

Примеры решений задач.

1. *Вычислить N! (факториал).*

```
Program Faktorial;  
  Var n, i, f: integer;  
Begin  
  f:=1;  
  Write('Введите n=');  
  Readln(n);  
  For i:=2 to n do  
    f:=f*i;  
  Writeln(n,'!=',f);  
End.
```

2. *Найти сумму цифр в записи данного натурального числа;*

```
Program SUM;  
  Var a,b,s,k:Integer;  
Begin  
  Readln(a);  
  s:=0;  
  While a<>0 do  
    begin  
      b:=b mod 10;  
      s:=s+b;  
      a:=trunc(a/10); { a := a div 10 }  
    end;  
  Writeln(s);  
End.
```

Список задач

1. Дано натуральное число n . Получить все пифагоровы тройки натуральных чисел, каждое из которых не превосходит n , т.е. все такие тройки натуральных чисел a, b, c , что $a^2 + b^2 = c^2$.
2. Дано натуральное число n . Найти все меньшие n числа Мерсенна. Число Мерсенна – это простое число, представленное в виде $M_p = 2^p - 1$, где p – тоже простое число.
3. Два натуральных числа называют дружественными, если каждое из них равно сумме всех делителей другого, кроме самого этого числа. Найти все пары дружественных чисел, лежащих в диапазоне от 200 до 300.
4. Дано натуральное число n . Среди чисел 1, 2, ..., n найти все такие, запись которых совпадает с последними цифрами записи их квадрата.

5. Назовём натуральное число палиндромом, если его запись читается одинаково как с начала так и с конца (пример: 4884, 393, 1, 22).
 - а) найти все меньшие 100 натуральных числа, которые являются палиндромами;
 - б) определить, является ли заданное натуральное число палиндромом;
 - в) найти все меньшие 100 натуральных числа, которые при возведении в квадрат дают палиндром;
 - г) найти все меньшие 100 натуральных числа-палиндромы, которые при возведении в квадрат дают палиндром;
 - д) является ли число палиндромом с учётом чётных цифр;
 - е) верно ли, что это число содержит ровно три одинаковые цифры;
 - ж) верно ли, что все чётные цифры числа различны;
6. Дано натуральное число n ($n > 99$). Определить число сотен в нём.
7. Дано натуральное число n ($n < 99$). Выяснить, верно ли, что n^2 равно кубу суммы цифр числа n .
8. Дано натуральное число n ($n < 9999$).
 - а) сколько цифр в числе n ?
 - б) чему равна сумма его цифр?
 - в) найти последнюю цифру числа.
 - г) найти первую цифру числа.
 - д) найти предпоследнюю цифру числа (в предположении, что $n > 10$).
 - е) дано число m . Найти сумму m -последних цифр числа n .
 - ж) выяснить, входит ли цифра 3 в запись числа n .
 - з) поменять порядок цифр числа n на обратный.
 - и) переставить последнюю и первую цифры числа n .
 - к) приписать по единице в начало и конец записи числа n .
9. Является ли заданное натуральное число степенью двойки.
10. Разложить заданное число на простые множители.
11. Число, равное сумме всех своих делителей, включая единицу, называется совершенным. Найти и напечатать все совершенные числа в интервале от 2 до x .
12. Найти сумму квадратов чисел от m до n .
13. Найти сумму квадратов нечётных чисел в интервале, заданном значениями переменных m и n ;
14. Найти сумму квадратов чётных чисел в интервале, заданном значениями переменных m и n ;
15. Найти сумму целых положительных чисел, кратных 4 и меньших 100.
16. Определить k - количество трёхзначных натуральных чисел, сумма цифр которых равна n ($1, n, 27$). Операции деления, div и mod не использовать.

Вложенные циклы

17. Дано натуральное число n . Получить все натуральные числа, меньшие n и взаимно простые с ним.
18. Даны целые числа p и q . Получить все делители числа q , взаимно простые с p .
19. Дано натуральное число n . Получить все простые делители этого числа.
20. Найти 100 первых простых чисел.

21. Даны натуральные числа n, m . Получить все меньшие n натуральные числа, квадрат суммы цифр которых равен m .

22. Натуральное число называется совершенным, если оно равно сумме всех своих делителей за исключением самого себя. Например, $6=1+2+3$.

Дано натуральное число n . Получить все совершенные числа, меньшие n .

23. Дано пять различных целых чисел. Найти среди них два числа, модуль разности которых имеет:

а) наибольшее значение;

б) наименьшее значение.

24. Вывести на экран числовой ряд действительных чисел от 10 до 20 с шагом 0,2.

25. Дано натуральное число n . Вычислить

а) 2^n ; б) $n!$; в) a^n ; г) $a(a+1)\dots(a+n-1)$; д) $a(a-n)(a-2n)\dots(a-n\cdot n)$.

е) $\left(1 + \frac{1}{1^2}\right)\left(1 + \frac{1}{2^2}\right)\dots\left(1 + \frac{1}{n^2}\right)$;

ж) $\frac{1}{a} + \frac{1}{a(a+1)} + \dots + \frac{1}{a(a+1)\dots(a+n)}$;

з) $\frac{1}{a} + \frac{1}{a^2} + \frac{1}{a^4} + \dots + \frac{1}{a^{2^n}}$.

26. Даны действительные числа x, a , натуральное число n . Вычислить

$\underbrace{\left(\left(\dots\left(\left(x+a\right)^2+a\right)^2+\dots+a\right)^2+a\right)^2+a}_{n \text{ скобок}}$

27. Дано действительное число a . Найти:

а) среди чисел $1, 1 + \frac{1}{2}, 1 + \frac{1}{2}, 1 + \frac{1}{2} + \frac{1}{3}, \dots$ первое, большее a ;

б) среди чисел $1, 1 + \frac{1}{2}, 1 + \frac{1}{2} + \frac{1}{3}, \dots$ первое, меньшее a ;

28. Даны действительные числа n и m . Найти наибольший делитель этих чисел, используя алгоритм Евклида.

29. Дано натуральное n . Найти $\sum_{i=1}^n \frac{x_i}{1 + |y_i|}$.

30. Дано натуральное число n . Вычислить $1 \cdot 2 + 2 \cdot 3 + 3 \cdot 4 + \dots + n \cdot (n+1) + \dots + 2n$.

31. Вычислить $\frac{1}{1 + \frac{1}{3 + \frac{1}{5 + \frac{1}{\dots + \frac{1}{101 + \frac{1}{103}}}}}}$.

32. Даны натуральные числа n, k ($n \geq k \geq 0$). Вычислить $\frac{n(n-1)\dots[n-k+1]}{k!}$.

33. Вычислить а) $\sum_{i=1}^{100} \frac{1}{i^2}$; б) $\sum_{i=1}^{50} \frac{1}{i^3}$; в) $\sum_{i=1}^{10} \frac{1}{i!}$; г) $\sum_{i=1}^{128} \frac{1}{(2i)^2}$; д) $\prod_{i=1}^{52} \frac{i^2}{i^2 + 2i + 3}$;

е) $\prod_{i=1}^{10} \left(2 + \frac{1}{i!}\right)$; ж) $\prod_{i=2}^{100} \frac{i+1}{i+2}$; з) $\prod_{i=2}^{10} \left(1 + \frac{1}{i!}\right)^2$ и) $\sum_{k=1}^n \frac{1}{k}$; к) $\sum_{k=1}^n \frac{1}{k^5}$;

л) $\sum_{k=1}^n \frac{1}{(2k+1)^2}$; м) $\sum_{k=1}^n \frac{(-1)^k}{(2k+1)k}$; н) $\sum_{k=1}^n \frac{(-1)^{k+1}}{k(k+1)}$; о) $\sum_{k=1}^n \frac{(-1)^k (k+1)}{k!}$; п) $\sum_{k=1}^n \frac{k!}{\frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{k+1}}$.

34. Дано натуральное число n . Вычислить произведение первых n сомножителей:

а) $\frac{1}{2} \cdot \frac{3}{4} \cdot \frac{5}{6} \cdot \dots$; б) $\frac{1}{1} \cdot \frac{3}{2} \cdot \frac{5}{3} \cdot \dots$.

Практическая работа № 5

Тема: «Символьный тип данных»

Диапазон значений: значением переменной символьного типа может быть любой символ буквы, цифры, знаки операций, скобки, пробелы и специальные символы. Каждому символу алфавита соответствует индивидуальный код от 0 до 255.

Наиболее распространенной международной согласованной системой кодирования всех символов является система ASCII. Символы с кодами 0 до 127 представляют так называемую основную таблицу кодов ASCII. Эта часть идентична на всех IBM-компьютерах. Коды с символами от 128 до 255 представляют национальную часть.

Так как символы языка упорядочены, то к символьным данным применимы операции сравнения. Операция сравнения осуществляется следующим образом: из двух символов меньше тот, который встречается в таблице раньше. Функция **ORD** от величины типа **Char** дает код соответствующего символа. Величины символьного типа сравниваются по своим кодам.

Пример определения символьного типа в программе:

```
Var word: Char;
```

Примеры решений задач

1. Программа проверки введения клавиши 'Y'.

```
Program keyp;
```

```
Var ch: char;
```

```
Begin
```

```
    Repeat
```

```
        Readln(ch);
```

```
    Until (ch='y') or (ch='Y')
```

```
End.
```

2. Программа вывода последовательностей букв.

```
Program abc_xyz;
```

```
Var c,d: char;
```

```
Begin
```

```
    For c:='a' to 'z' do
```

```
        Begin
```

```
            For d:='a' to c do write(d);
```

```
            Writeln(' ');
```

```
        end
```

```
    end.
```

Список задач

1. Удалить группу символов, расположенных между круглыми скобками включая сами скобки.
2. Дано натуральное число n и n символов. Подсчитать, сколько раз среди данных символов встречается символ "+" и сколько раз символ "*".
3. Подсчитать общее число вхождений символов "+", "-", "*" в последовательности n символов.
4. Выделить символы, заключённые в фигурные скобки.
5. Удалить символы, заключённые в фигурные скобки.

6. Переставить местами группы символов While и Do.
7. Переставить местами группы символов Repeat и Until.
8. Подсчитать наибольшее число букв a, идущих подряд в данной последовательности символов.
9. Вычислить s- сумму порядковых номеров всех букв, входящих в слово SUM.
10. Напечатать текст, образованный литерами с порядковыми номерами 65, 71 и 69.
11. Если в заданный текст входит каждая из букв слова key, тогда напечатать yes, иначе no.
12. Напечатать заданный текст, удалив из него лишние пробелы, т.е. из нескольких подряд идущих пробелов оставить только один.
13. В заданный непустой текст входят только цифры и буквы. Определить, удовлетворяет ли он следующему свойству:
 - а) текст содержит (помимо букв) только одну цифру причём её числовое значение равно длине текста;
 - б) сумма числовых значений цифр, входящих в текст, равна длине текста;
14. Логической переменной b присвоить значение **true**, если между литерами 'a' и 'z' нет иных символов, кроме строчных латинских букв, и значение **false** иначе.

Практическая работа № 6

Тема: «Функции и процедуры»

Вспомогательные алгоритмы организуются двумя способами в зависимости от использования:

1) ПРОЦЕДУРЫ:

Программа процедура предназначена для выполнения какой-то законченной последовательности действий. Любая процедура начинается с заголовка. В отличие от основной программы заголовок в процедуре обязателен. Он состоит из зарезервированного слова **Procedure**, за которым следует идентификатор имени процедуры, а далее в круглых скобках список формальных параметров:

Procedure <имя процедуры> (список формальных параметров)

За заголовком могут идти такие же разделы, что и в основной программе. В отличие от основной программы процедура завершается не точкой, а точкой с запятой.

Procedure < имя процедуры >[(< аргументы >: < тип аргументов >]; **Var** <значение>:< тип значения >]);

Var < переменная 1 >[,< переменная 2 >,...]: тип;

Begin

< тело процедуры >;

End;

Для вызова процедуры из основной программы или другой подпрограммы следует записать оператор, состоящий из имени процедуры и списка фактических параметров, которые должны совпадать по количеству и типу с формальными параметрами процедуры.

2) ФУНКЦИИ:

Подпрограмма-функция предназначена для вычисления какого-либо параметра. У подпрограммы-функции два основных отличия от процедуры:

а) заголовок состоит из слова **function**, за которым следует имя функции, далее в круглых скобках - список формальных параметров, затем через двоеточие записывается тип функции - тип возвращаемого параметра. Функция может возвращать параметры следующих типов: любого порядкового, любого вещественного, стандартного строкового и символьного.

б) в теле функции хотя бы раз имени функции должно быть присвоено значение.

Function < имя функции >[(< аргументы >: < тип аргументов >):< тип значения функции >];

Var < переменная 1 >[,< переменная 2 >,...]: тип;

Begin

<тело функции>;

<имя функции>:=<выражение значение функции>;

End;

Примеры решений задач.

1. Пример функции для вычисления факториала.

Function Factorial(N:Byte):Longint;

Var

Fact:longint;

I:byte;

Begin

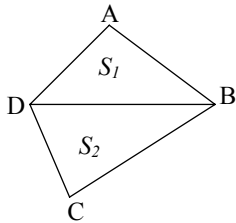
Fact:=n;

For i:=n-1 **downto** 2 **do**

Fact:=fact*i;

Factorial:=fact;

End;



2. Вычислить площадь произвольного четырехугольника, у которого заданы значения четырех сторон и диагонали. Для решения задачи необходимо разбить четырехугольник на два треугольника и вычислить их площади по формуле Герона.

Вариант 1. Использование процедуры без параметров.

Program Square1;

Var AB, BC, CD, DA, DB, a, b, c, p, s, S1, S2: Real;

Procedure geron;

Begin

p:=(a+b+c)/2; s:=sqrt(p*(p-a)*(p-b)*(p-a));

End;

Begin

WriteLn ('Введите длины сторон AB, BC, CD, DA, DB');

ReadLn (AB, BC, CD, DA, DB);

a:= AB; b:=DB; c:=DA;

geron; S1:=s;

a:= BC; b:=CD; c:=DB;

geron; S2:=s;

WriteLn ('Площадь равна:', S1+S2)

End.

AB, BC, CD, DA, DB, a, b, c, p, s, S1, S2 – глобальные переменные;

Вариант 2. Использование процедуры без параметров.

Program Square2;

Var AB, BC, CD, DA, DB, a, b, c, s, S1, S2: Real;

Procedure geron;

Var p:Real;

Begin

p:=(a+b+c)/2; s:=sqrt(p*(p-a)*(p-b)*(p-a));

End;

Begin

WriteLn ('Введите длины сторон AB, BC, CD, DA, DB');

ReadLn (AB, BC, CD, DA, DB);

a:= AB; b:=DB; c:=DA;

geron; S1:=s;

a:= BC; b:=CD; c:=DB;

geron; S2:=s;

WriteLn ('Площадь равна:', S1+S2)

End.

AB, BC, CD, DA, DB, a, b, c, s, S1, S2 – глобальные переменные;

p – локальная переменная.

Вариант 3. *Использование процедуры с параметрами.*

```
Program Square3;  
Var AB, BC, CD, DA, DB, s, S1, S2: Real;  
Procedure geron (a, b, c);  
Var p:Real;  
Begin  
  p:= (a+b+c)/2; s:= sqrt(p*(p-a)*(p-b)*(p-a));  
End;  
Begin  
  WriteLn ('Введите длины сторон AB, BC, CD, DA, DB');  
  ReadLn (AB, BC, CD, DA, DB);  
  geron (AB, DB, DA); S1:=s;  
  geron (BC, CD, DB); S2:=s;  
  WriteLn ('Площадь равна:', S1+S2)  
End.  
AB, BC, CD, DA, DB – фактические параметры;  
a, b, c – формальные параметры.
```

Вариант 4. *Использование процедуры с возвращением значения.*

```
Program Square4;  
Var AB, BC, CD, DA, DB, S1, S2: Real;  
Procedure geron (a, b, c; Var s:Real);  
Var p:Real;  
Begin  
  p:= (a+b+c)/2; s:= sqrt(p*(p-a)*(p-b)*(p-a));  
End;  
Begin  
  WriteLn ('Введите длины сторон AB, BC, CD, DA, DB');  
  ReadLn (AB, BC, CD, DA, DB);  
  geron (AB, DB, DA, S1);  
  geron (BC, CD, DB, S2);  
  WriteLn ('Площадь равна:', S1+S2)  
End.
```

Вариант 5. *Использование процедуры-функции.*

```
Program Square5;  
Var AB, BC, CD, DA, DB, S: Real;  
Function geron (a, b, c: Real): Real;  
Var p:Real;  
Begin  
  p:= (a+b+c)/2; geron:= sqrt(p*(p-a)*(p-b)*(p-a));  
End;  
Begin  
  WriteLn ('Введите длины сторон AB, BC, CD, DA, DB');  
  ReadLn (AB, BC, CD, DA, DB);  
  S:= geron (AB, DB, DA) + geron (BC, CD, DB);  
  WriteLn ('Площадь равна:', S)  
End.
```

Список задач

1. Даны действительные числа s, t . Получить:
 $f(t, -2s, 1.17) + f(2.2, t, s - t)$, где $f(a, b, c) = \frac{2a - b - \sin c}{5 + |c|}$
2. Даны действительные числа s, t . Получить:
 $g(1/2, s) + g(t, s) - g(2s - 1, st)$, где $g(a, b) = \frac{a^2 + b^2}{a^2 + 2ab + 3b^2 + 4}$
3. Даны действительные числа a, b, c . Получить:
 $\frac{\max(a, a + b) + \max(a, b + c)}{1 + \max(a + bc, 1, 15)}$
4. Даны действительные числа a, b . Получить:
 $u = \min(ab), v = \min(ab, a + b), f = \min(u + v^2, 3.14)$
5. Даны натуральные числа n, m , целые числа $a_1, a_2, \dots, a_n, b_1, \dots, b_m, c_1, \dots, c_{10}$. Получить: $L = \min(b_1, \dots, b_m) + \min(a_1, \dots, a_m) + \min(c_1, \dots, c_{10})$.
6. Даны натуральные числа k, l, m , действительные числа $x_1, \dots, x_k, y_1, \dots, y_l, z_1, \dots, z_m$. Получить $L = \max(y_1, \dots, y_l) + \max(z_1, \dots, z_m)$, если $\max(x_1, \dots, x_k) > 30$.
7. Дано натуральное число n . Среди чисел $1, 2, 3, \dots, n$ найти все те, которые можно представить в виде сумм квадратов двух натуральных чисел. Определить процедуру, позволяющую распознавать полные квадраты.
8. Даны действительные числа $x_1, y_1, x_2, y_2, \dots, x_{10}, y_{10}$. Найти периметр десятиугольника, вершины которого имеют соответственно координаты $(x_1, y_1), (x_2, y_2), \dots, (x_{10}, y_{10})$. Определить процедуру вычисления расстояния между двумя точками, заданными своими координатами.
9. Даны действительные числа a, b, c, d, e . Найти площадь пятиугольника. Определить процедуру вычисления площади треугольника по трём сторонам.
10. Даны натуральное число n , действительные числа $x_1, y_1, x_2, y_2, \dots, x_n, y_n$. Найти площадь n -угольника, вершины которого при некотором последовательном обходе имеют координаты $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$. Определить процедуру вычисления площади n -угольника по координатам его вершин.
11. Дано натуральное число n . Выяснить, имеются ли среди чисел $n, n+1, \dots, 2n$ близнецы, т.е. простые числа, разность между которыми равна. Определить процедуру, позволяющую распознать простые числа.
12. Составить процедуру, результатом работы которой является истинное значение, если символ, заданный при обращении к процедуре, буква, и ложное значение в противном случае.
13. Составить процедуру, результатом работы которой является символ, заданный при обращении к процедуре, если этот символ не является буквой, и соответствующая строчная (малая) буква в противном случае.
14. Составить процедуру, заменяющую в исходной строке символов все единицы нулями и все нули единицами. Замена должна выполняться, начиная с заданной позиции строки.

Практическая работа № 7

Тема: «Строковый тип данных»

Тип-строка - последовательность символов произвольной длины (до 255). Строку можно рассматривать как массив символов, однако в связи с широким использованием строк и некоторыми особенностями по сравнению со стандартными массивами они выделены в отдельный тип данных.

У типа-строки в квадратных скобках может быть указан его размер (от 1 до 255). Если размер строки не указан, он считается равным 255, например:

Var

```
Str: string[80];  
MaxStr: string;
```

Const

```
January: string[10]='Январь';
```

Процедуры и функции обработки строковых переменных:

Length(S: String): Integer; - функция определения длины строкового выражения.

Пример использования в программе:

Var

```
S: String;
```

begin

```
Readln (S);  
Writeln("", S, "");  
Writeln('length = ', Length(S));
```

end.

Copy(S: String; Index: Integer; Count: Integer): String; - функция выделения подстроки из строки.

Пример использования в программе:

```
Var S: String;
```

begin

```
S := 'ABCDEF';  
S := Copy(S, 2, 3); { 'BCD' }
```

end.

Concat(s1 [, s2,..., sn]: String): String; - функция склейки строк; аналогично операции «+» - последовательное соединение строк.

Пример использования в программе:

Var

```
S: String;
```

begin

```
S := Concat('ABC', 'DEF'); { 'ABCDE' }
```


end.

Delete(Var S: **String**; Index: **Integer**; Count: **Integer**); - процедура удаления подстроки из строки.

Пример использования в программе:

```
Var  
s: string;  
begin  
  s := 'Honest Abe Lincoln';  
  Delete(s,8,4);  
  Writeln(s); { 'Honest Lincoln' }  
end.
```

Insert(Source: **String**; Var S: **String**; Index: **Integer**); - процедура вставки в строку *Word2* подстроки *Word1* с позиции n.

Пример использования в программе:

```
Var  
  S: String;  
begin  
  S := 'Honest Lincoln';  
  Insert('Abe ', S, 8); { 'Honest Abe Lincoln' }  
end.
```

Pos(Substr: **String**; S: **String**): **Byte**; - функция определения позиции включения одной строки в другой;

Пример использования в программе:

```
Var S: String;  
begin  
  S := ' 123.5';  
  while Pos(' ', S) > 0 do  
    S[Pos(' ', S)] := '0';  
end.
```

Примеры решений задач

1. Объединение двух строк.

```
Program assign;  
Var  
  Str, str1, str2: string[80];  
Begin  
  Str1 := 'Turbo';  
  Str2 := 'Pascal';  
  Str := str1 + str2;  
  Writeln (str);  
End.
```

Список задач

1. Дана строка, подсчитать сколько раз встречается буква a.
2. Даны натуральное число n , символы S_1, S_2, \dots, S_n . Подсчитать, сколько раз среди данных символов встречается буква b.
3. Даны натуральное число n , символы S_1, S_2, \dots, S_n . Подсчитать:
 - а) сколько раз среди данных символов встречается символ "+" и сколько раз символ "*";
 - б) общее число вхождений символов "+", "-" и "*" в последовательность S_1, S_2, \dots, S_n .
4. Дана последовательность S_1, S_2, \dots, S_n , заменить в ней:
 - а) все восклицательные знаки точками;
 - б) каждую точку многоточием;
 - в) каждую из групп стоящих рядом точек одной точкой;
 - г) каждую из групп стоящих рядом точек многоточием.
5. Даны натуральное число n , символы S_1, S_2, \dots, S_n . Выяснить, имеются ли в последовательности S_1, S_2, \dots, S_n такие члены последовательности S_i и S_{i+1} , что S_i - это запятая, S_{i+1} - это тире.
6. Даны натуральное число n , символы S_1, S_2, \dots, S_n . Получить первое натуральное i , для которого каждый из символов S_i и S_{i+1} , совпадают с буквой A. Если такой пары в последовательности S_1, S_2, \dots, S_n нет, то ответом должно быть число 0.
7. Даны натуральное число n , символы S_1, S_2, \dots, S_n . Известно что среди S_1, S_2, \dots, S_n есть по крайней мере одна запятая. Найти такое натуральное i , что:
 - а) S_i - первая по порядку запятая;
 - б) S_i - последняя по порядку запятая.
8. Даны натуральное число n , символы S_1, S_2, \dots, S_n . Преобразовать удалив каждый символ "*" и повторив каждый символ отличный от "*".
9. Даны натуральное число n , символы S_1, S_2, \dots, S_n , среди которых есть двоеточие.
 - а) получить все символы, расположенные до первого двоеточия включительно.
 - б) получить все символы, расположенные после первого двоеточия включительно.
 - в) получить все символы, расположенные между первым и вторым двоеточиями. Если второго двоеточия нет, то получить все символы после первого двоеточия.
10. Даны натуральное число n , символы S_1, S_2, \dots, S_n .
 - а) подсчитать наибольшее количество идущих подряд пробелов;
 - б) выяснить, верно ли, что в последовательности S_1, S_2, \dots, S_n имеются пять идущих подряд букв C.
11. Даны натуральное число n , символы S_1, S_2, \dots, S_n . Группы символов, разделённых пробелами (одним или несколькими) и не содержащим пробелов внутри себя будем называть словами.
 - а) подсчитать количество букв "а" в последнем слове данной последовательности.
 - б) найти количество слов, начинающихся с буквы "с".
 - в) найти количество слов, у которых первый и последний символы совпадают.
 - г) подсчитать количество слов в данной последовательности.
 - д) найти какое-нибудь слово, начинающиеся с буквы "а".
 - е) преобразовать данную последовательность, заменяя всякое вхождение слова "это" на слово "то".
 - ж) найти длину самого короткого слова.

- з) найти длину самого длинного слова.
 - и) удалить все символы, не являющиеся буквами.
 - к) заменить все малые буквы одноимёнными большими.
12. Найти первое слово самое короткое предложения.
 13. Найти последнее слово самое короткое предложения.
 14. Найти самое длинное слово в предложении.
 15. Найти первое симметричное слово в предложении.
 16. Заменить заданное слово предложения на другое слово.
 17. Найти в предложении слова, которые начинаются на одну и ту же букву.
 18. Напечатать предложение, удалив из него повторное вхождение слов.
 19. Напечатать различные слова предложения, указав для каждого из них число его вхождений.
 20. Напечатать предложение после удаления средней буквы для слов нечётной длины.
 21. Напечатать слова предложения в алфавитном порядке.

Практическая работа № 8

Тема: «Одномерные массивы»

Массив это упорядоченная совокупность конечного числа данных одного типа. Простейшим примером массива может служить линейная таблица. Значения, образующие линейную таблицу, являются элементами массива, а их порядковые номера в таблице называются индексами. Можно сказать, что одномерный массив соответствует понятию вектора. Индекс определяет положение элемента массива относительно его начала.

Общая форма описания переменной:

Var <имя>: **Array** |<тип-индексов>| **of**<тип-элементов>

Выбор отдельной компоненты одномерного массива осуществляется указанием идентификатора массива, за которым в квадратных скобках следует индексное выражение. Индексное выражение должно давать значения, лежащие в диапазоне, определяемом типом индекса.

Например: M[1],M[2],...,M[N].

Для организации автоматического ввода значений следует воспользоваться функцией библиотеки CRT **Random**. Эта функция возвращает случайное число из диапазона от 0 до n. Для этого необходимо в основной программе инициализировать датчик случайных чисел командой **Randomize**.

Формат вызова функции **Random**:

Random (m), где m – значение, указывающее на правую границу диапазона выдаваемых значений.

Например: создание линейной таблицы размерностью 10, числами из диапазона от 0 до 10.

```
Program Vector_Full;  
Const n=10;  
Type vector = array [1..n] of Integer;  
Var v:vector;  
Procedure Enter (Var vect: vector);  
Var i: Integer;  
Begin  
  For i:=1 to n do  
    vect[i]:=Random(10);  
End;  
Begin  
  Randomize;  
  Enter (v);  
End.
```

Примеры решений задач

1. Дана последовательность символов s_1, s_2, \dots, s_{30} . Требуется определить, совпадает ли начальная часть последовательности s_1, s_2, \dots, s_{15} с ее концевой частью $s_{15}, s_{16}, \dots, s_{30}$.

```
Program Double;
  Label 1;
  Type t:array[1..15];
  Var x,t;y :char; t:integer;
Begin
  For i:=1 to 15 do read(x[i]);
  For i:=1 to 15 do
    Begin read(y);
      if x[i]<>y then
begin write('не ');
goto 1;
end;
      end;
1:   write ('совпадают')
end.
```

2. Найти максимальное число в последовательности.

```
program maximum;
Var
a:array [1..10] of integer;
n: byte;
  m: integer;
begin
  for n:=1 to 10 do
readln (a[n]);
  m:=a[1];
  for n:=2 to 10 do
    if a[n]>m then m:=a[n];
  writeln ('maximum = ', m);
  readln
end.
```

Список задач

- Даны натуральные числа n, A_1, \dots, A_n . Определить количество членов A_k последовательности A_1, \dots, A_n :
 - являющихся нечётными числами;
 - кратных 3 и некрратных 5;
 - являющихся квадратами чётных чисел;
- Даны натуральные числа n, A_1, \dots, A_n . Найти те элементы A_k последовательности n, A_1, \dots, A_n , которые:
 - являются удвоенными нечётными числами;
 - при делении на 7 дают остаток 1, 2 или 5;

3. Даны целые числа A_1, \dots, A_{20} . Получить сумму трёх членов данной последовательности, которые:
 - а) кратны 5;
 - б) нечётны и отрицательны;
 - в) удовлетворяют условию $|A_i| < i^2$.
4. Даны натуральное число n , целые числа A_1, \dots, A_n . Найти количество и сумму тех членов последовательности, которые делятся на 5 и не делятся на 7.
5. Даны целые числа p, q, A_1, \dots, A_{17} ($p > q > 0$). В последовательности заменить нулями элементы, модуль которых при делении на p даёт в остатке q .
6. Даны натуральные числа n, p , целые числа A_1, \dots, A_n . Получить произведение элементов последовательности, кратных p .
7. Даны натуральное число n , действительные числа A_1, \dots, A_n . В последовательности получить удвоенную сумму всех положительных элементов.
8. Даны натуральное число n , действительные числа A_1, \dots, A_n . В последовательности все отрицательные числа увеличить на 0.5, а все неотрицательные на 0.1.
9. Даны натуральное число n , действительные числа A_1, \dots, A_n . В последовательности все элементы, меньше 2, заменить нулями. Кроме того, получить сумму элементов, принадлежащих отрезку $[3, 7]$, а также число таких элементов.
10. Даны натуральное число n , действительные числа A_1, \dots, A_n . В последовательности все неотрицательные элементы, не принадлежащие отрезку $[1, 2]$, заменить на 1. Кроме того, получить число отрицательных элементов и число элементов, принадлежащих отрезку $[1, 2]$.
11. Даны натуральное число n , целые числа A_1, \dots, A_n . Получить сумму положительных и число отрицательных элементов последовательности.
12. Даны натуральное число n , целые числа A_1, \dots, A_n . Заменить все, большие 7, элементы последовательности числом 7. Вычислить количество таких элементов.
13. Даны целые числа A_1, \dots, A_{15} . Получить число отрицательных элементов последовательности A_1, \dots, A_{10} и число нулевых элементов всей последовательности A_1, \dots, A_{15} .
14. Даны натуральное число n , целые числа A, X_1, \dots, X_n . Если в последовательности есть хотя бы один элемент, равный A , то получить сумму всех элементов, следующих за первым таким элементом; в противном случае ответом должно быть число 10.
15. Даны целые числа A_1, \dots, A_{20} . Получить последовательность B_1, \dots, B_{20} , которая отличается от исходной тем, что все нечётные элементы удвоены.
16. Даны натуральное число n , целые числа A, X_1, \dots, X_n . Определить, каким по счёту идёт в последовательности элемент, равный A . Если такого элемента нет, то ответом должно быть число 0.
17. Даны натуральное число n , действительные числа A_1, \dots, A_n . Получить:
 - а) $\max(A_1, \dots, A_n)$;
 - б) $\min(A_1, \dots, A_n)$;
 - в) $\max(A_2, A_4, \dots)$;
 - г) $\min(A_1, A_3, \dots)$;
 - д) $\min(A_2, A_4, \dots) + \max(A_1, A_3, \dots)$;
 - е) $\max(|A_1|, \dots, |A_n|)$;

- ж) $\max(-A_1, A_2, -A_3, \dots, (-1)^n A_n)$;
- з) $(\min(A_1, \dots, A_n))^2 - \min(A_1^2, \dots, A_n^2)$.
18. Дано натуральное число n . Выбросить из записи числа n цифры 0 и 5, а оставив прежним порядок остальных цифр. Например, из числа 59015509 должно получиться 919.
19. Даны натуральное число n , целые числа A_1, \dots, A_n . Найти:
- наименьшее из чётных чисел, входящих в последовательность A_1, \dots, A_n .
 - наибольшее из нечетных и количество чётных чисел, входящих в последовательность A_1, \dots, A_n, A_{n+1} .
20. Даны натуральное число n , действительные числа A_1, \dots, A_n . В последовательности определить число соседств:
- двух положительных чисел;
 - двух чисел разного знака;
 - двух чисел одного знака, причём модуль первого числа должен быть больше модуля второго числа.
21. Даны целые числа C_1, \dots, C_{15} . Имеются ли в последовательности:
- два идущих подряд нулевых элемента;
 - три идущих подряд нулевых элемента.
22. Даны натуральное число n . Получить все такие натуральные q , что n делится на q^2 и не делится на q^3 .
23. Даны натуральные числа m, n . Получить все их натуральные общие кратные, меньше mn .
24. Даны целые положительные числа m, n . Получить все их общие делители.
25. Даны натуральное число n , действительные числа A_1, \dots, A_n . Выяснить, является ли последовательность упорядоченной по убыванию.
26. Даны натуральное число n , целые числа A_1, \dots, A_n .
- Выяснить, какое число встречается в последовательности раньше - положительное или отрицательное. Если все элементы последовательности равны нулю, то сообщить об этом.
 - Найти номер первого чётного элемента последовательности; если чётных элементов нет, то ответом должно быть число 0.
 - Найти номер последнего нечётного элемента последовательности; если нечётных элементов нет, то ответом должно быть число $n+1$.

Алгоритмы циклической структуры с использованием одномерных массивов.

27. В массиве $\{A_j\}$, $j=1, 2, 10$ есть хотя бы один отрицательный элемент. Вычислить произведение элементов в массиве до первого отрицательного.
28. В массиве есть хотя бы один нуль.
- Вычислить произведение элементов массива до первого нуля.
 - Вычислить сумму элементов массива до первого нуля.
29. В массиве существуют отрицательный и положительный элементы. Вычислить:
- сумму положительных элементов;
 - сумму отрицательных элементов;
 - количество положительных элементов;
 - количество отрицательных элементов;

- д) произведение положительных элементов;
 - е) произведение отрицательных элементов;
30. В массиве подсчитать количество элементов, больших 3.
31. Составить программу для вычисления суммы S элементов числовой последовательности A_1, A_2, \dots, A_{10} по формуле $S=A_1+A_2+\dots+A_{10}$.
32. Составить программу для вычисления суммы элементов последовательности целых чисел P_1, P_2, \dots, P_{10} , имеющих чётные индексы и произведение элементов последовательности P_1, P_2, \dots, P_{10} , имеющих нечётные индексы.

**Вычисление с хранением
последовательности значений**

33. Даны действительные числа $A_1, \dots, A_n, B_1, \dots, B_n$. Вычислить $(A_1+B_n) \times (A_2+B_{n-1}) \times \dots \times (A_n+B_1)$.
34. Даны действительные числа A_1, A_2, \dots, A_{2n} . Получить:
- а) $A_1, A_{n+1}, A_2, A_{n+2}, \dots, A_n, A_{2n}$;
 - б) $A_1, A_{2n}, A_2, A_{2n-1}, A_3, \dots, A_n, A_{n+1}$;
 - в) $A_1+A_{2n}, A_2+A_{2n-1}, \dots, A_n+A_{n+1}$.
35. Даны действительные числа A_1, A_2, \dots, A_{17} . Получить:
- а) $A_{17}, A_1, A_2, \dots, A_{16}$;
 - б) $A_{11}, A_{12}, \dots, A_{17}, A_1, A_2, \dots, A_{10}$;
 - в) $A_{11}, A_{12}, \dots, A_{17}, A_{10}, A_9, \dots, A_1$;
 - г) $A_1, A_3, \dots, A_{17}, A_2, A_4, \dots, A_{16}$.
36. Даны действительные числа A_1, \dots, A_n . Если в результате замены отрицательных элементов последовательности их квадратами элементы будут образовывать неубывающую последовательность, то получить сумму элементов исходной последовательности; в противном случае их произведение.
37. Даны действительные числа A_1, \dots, A_{10} . Все числа попарно различны. Поменять в этой последовательности местами:
- а) наибольший и наименьший элементы;
 - б) наибольший и последний элементы.
38. Дано натуральное число n . Сколько различных цифр встречается в его десятичной записи.
39. Даны действительные числа A_1, \dots, A_{10} . Вычислить $A_1+A_2^2+\dots+A_{10}^{10}$.

Практическая работа № 9

Тема: «Сортировка одномерных массивов»

Рассмотрим массив целых или действительных чисел A_1, \dots, A_n . Пусть требуется переставить элементы этого массива так, чтобы после перестановки они были упорядочены по не убыванию $A_1 \leq A_2 \leq \dots \leq A_n$. Эта задача называется задачей сортировки или упорядочения массива. Для решения этой задачи можно воспользоваться следующими алгоритмами:

I. Сортировка выбором

Найти элемент массива, имеющий наибольшее значение, переставить его с первым элементом, затем проделать то же самое, начав со второго элемента и т.д.

II. Сортировка обменами (метод пузырька)

Последовательным просмотром чисел A_1, \dots, A_n найти наименьшее i такое, что $A(i) > A(i+1)$. Поменять местами $A(i)$ и $A(i+1)$, возобновить просмотр с элемента $A(i+1)$ и т.д. Тем самым наибольшее число передвинется на последнее место. Следующие просмотры начинать опять сначала, уменьшая на единицу количество просматриваемых элементов. Массив будет упорядочен после просмотра, в котором участвовали только первый и второй элементы.

III. Сортировка простыми вставками

Просматривать последовательно A_2, \dots, A_n и каждый новый элемент A_i вставлять на подходящее место в уже упорядоченную совокупность $A(1) \dots A(i-1)$. Это место определяется последовательным сравнением A_i с упорядоченными элементами $A(1), \dots, A(i-1)$.

Примеры решений задач

I. Сортировка отбором.

Procedure otbor (**Var** x:arr; dlina:integer);

Var mal: dano;

 Min, k, j: integer;

Begin

For k:=1 **to** dlina-1 **do**

Begin

 Mal:=x[k];

 Min:=k;

For j:=k+1 **to** dlina **do**

If x[j]<mal **then**

Begin

 Mal:=x[j];

 Min:=j;

End;

 X[Min]:=x[k];

 X[k]:=mal

End

end

2. *Сортировка методом пузырька.*

Procedure Bubble (**Var** X:massiv; DLINA:integer);

Var A:dannoe;

J,R: integer;

ZAMENA: boolean;

Begin

Repeat

ZAMENA:=false;

For J:=2 to R **do**

If X[j-1]>X[J] **then**

Begin

A:=X[J-1];

X[J-1]:=X[J];

X[J]:=A;

ZAMENA:= true

End;

R:=R-1

Until not ZAMENA;

End.

3. *Сортировка методом простых вставок.*

Procedure simpins (**Var** a:arr);

Var j,j,k: word;

T: integer;

Begin

For i:=2 to n **do**

Begin

T:=a[i];

J:=1;

While (t>=a[j]) **and** (j<i) **do** inc(j);

If j<>i **then**

For k:=j-1 **downto** j **do** a[k+1]=a[k];

A[j]:=t

End;

End;

End.

4. *Задача поиска места элемента. Для решения задачи полезен алгоритм, который называется алгоритмом деления пополам.*

Пусть дан упорядоченный по неубыванию массив целых или действительных чисел $A_1 \leq A_2 \leq A_n$. Пусть дано некоторое число B (соответственно целое или действительное), для которого нужно найти такое место среди чисел A_1, \dots, A_n , чтобы после вставки числа B

на это место упорядоченность не нарушилась. Если вследствие равенства между собой некоторых элементов массива число V можно вставлять на разные места, то требуется определить ближайшее к началу массива место.

Список задач

1. Даны действительные числа A_1, \dots, A_n, P , натуральное число k ($A_1 \leq \dots \leq A_n, k \leq n$). Удалить из A_1, \dots, A_n элемент с номером k (т.е. A_k) и вставить элемент, равный P , так, чтобы не нарушилась упорядоченность.
2. Даны действительные числа $C_1, \dots, C_p, D_1, \dots, D_q$. ($C_1 \leq C_2 \leq \dots \leq C_p, D_1 \leq D_2 \leq \dots \leq D_q$). Ввести единицу упорядоченности, получив F_1, F_2, \dots, F_{p+q} , что $F_1 \leq F_2 \leq \dots \leq F_{p+q}$. Число сравнений не должно превосходить $p+q$.
3. Пусть дан массив A_1, \dots, A_n . Требуется переставить A_1, \dots, A_n так, чтобы в начале массива шла группа элементов, больших того элемента, который в исходном массиве располагался на первом месте, затем сам этот элемент, потом - группа элементов, меньших или равных ему.
4. Даны натуральное число n , целые числа A_1, \dots, A_n . Найти наибольшее значение, встречающееся в последовательности A_1, \dots, A_n , после выбрасывания одного из членов со значением $\max(A_1, \dots, A_n)$.
5. Дана действительная матрица размером $m \times n$; упорядочить строки матрицы:
 - а) по убыванию значений первых элементов строк;
 - б) по возрастанию сумм элементов строк;
 - в) по убыванию значений наименьших элементов строк;
 - г) по возрастанию значений наибольших элементов строк.В заданиях б, в, г разрешается дополнительно определить числовой массив A_1, \dots, A_n .

Практическая работа № 10

Тема: «Двумерные массивы. Преобразование и построение матриц»

Двумерные массивы являются аналогами матриц. Первый индекс элемента двумерного массива определяет номер строки, а второй – номер столбца, на пересечении, которых расположен элемент. Строки и столбцы нумеруются либо от единого заранее установленного минимального значения индекса, либо от граничного значения, заданного одновременно с объявлением массива.

Для задания значений двумерного массива можно воспользоваться процедурой с использованием датчика случайных чисел, данной в лабораторной работе №8 .

Например:

```
Program Array_Full;  
Const n=10;  
Type mas = array [1..n, 1..n] of Integer;  
Var m:mas;  
Procedure Enter (Var tabl: mas);  
Var i, j: Integer;  
Begin  
  For i:= 1 to n do  
    For j:= 1 to n do  
      tabl[i,j]:=Random(10);  
  End;  
Begin  
  Randomize;  
  Enter (m);  
End.
```

С помощью процедуры Enter осуществляется ввод двумерного массива размерностью 10×10 случайными числами из диапазона от 0 до 10. Задание диапазона значений определяется константой n=10.

Описание двумерного массива производится следующим образом:

```
Const  
  n=<ранг матрицы>;  
Type  
  matr=array [1..n, 1..n] of <тип элементов матрицы>;  
Var  
  <имя матрицы>: matr;
```

Примеры решений задач

1. Процедура ввода матрицы.

```
Procedure Enter;  
Var  
  ii, jj: integer;  
Begin
```

```

For jj:=1 to n do
  For ii:=1 ot n do
    write('g[';ii;',';jj;']=>');readln (g[ii,jj]);
End;

```

2. Процедура вывода матрицы на экран. (В данной процедуре используется процедура GotoXY, содержащаяся в модуле CRT. Поэтому необходимо при использовании этой процедуры указать имя библиотеки CRT в разделе описания библиотек Uses)

Program Name_Program;

Uses Crt;

...

Procedure List;

Var

ii, jj: integer;

Begin

For jj:=1 **to** n **do**

For ii:=1 **to** n **do begin**

gotoxy(3*ii, jj);

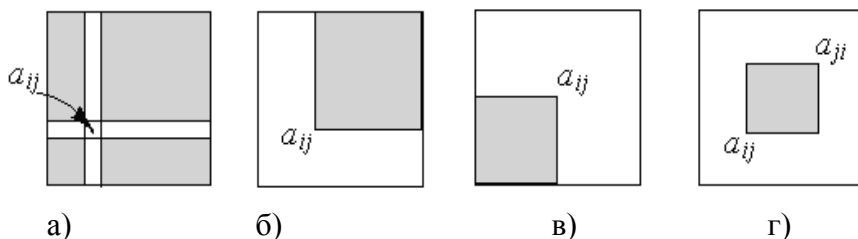
write (g[ii,jj]);

end;

End;

Список задач

1. Даны действительная матрица размера $n \times (n+1)$, действительные числа $A_1, \dots, A_{n+1}, B_1, \dots, B_{n+1}$, натуральные числа p, q ($p \leq n, q \leq n+1$). Образовать новую матрицу размера $(n+1) \times (n+2)$ вставкой после строки с номером p данной матрицы новой строки с элементами A_1, \dots, A_{n+1} , и последующей вставкой после столбца с номером q нового столбца с элементами B_1, \dots, B_{n+1} .
2. Даны целые числа A_1, \dots, A_{10} , целочисленная квадратная матрица порядка n . Заменить нулями в матрице те элементы с четной суммой индексов, для которых имеются равные среди A_1, \dots, A_{10} .
3. Даны действительные числа A_1, \dots, A_n , действительная квадратная матрица порядка n ($n \geq 6$). Получить действительную матрицу размера $n \times (n+1)$, вставив в исходную матрицу между пятым и шестым столбцами новый столбец с элементами A_1, \dots, A_n .
4. Дана целочисленная матрица размера 6×9 . Найти матрицу, получающуюся из данной:
 - а) перестановкой столбцов - первого и последнего, второй с предпоследним и т.д.;
 - б) перестановкой строк - первого и последнего, второй с предпоследним и т.д.;
5. Дана действительная матрица $[A_{ij}]$, $i, j=1, \dots, n$. Получить действительную матрицу $[B_{ij}]$, $i, j=1, \dots, n$, элемент B_{ij} который равен сумме элементов данной матрицы, расположенных в области, определяемой индексами i, j так, как показано на рис. (область заштрихована).



- Сходным образом можно рассмотреть вместо суммы элементов их произведение, наибольшее значение, наименьшее значение.
6. Дана действительная квадратная матрица порядка n . Преобразовать матрицу по правилу: строку с номером n сделать столбцом с номером n , столбец с номером n сделать строкой с номером n .
 7. Даны две действительные квадратные матрицы порядка n . Получить новую матрицу:
 - а) умножением элементов каждой строки 1-й матрицы на наибольшее из значений элементов соответствующей второй матрицы;
 - б) прибавлением к элементам каждого столбца 1-й матрицы произведения элементов соответствующих строк второй матрицы.
 8. В данной действительной матрице $m \times n$ ($n \geq 3, m \geq 3$) поменять местами:
 - а) строки с номерами 2 и $n-1$;
 - б) столбцы с номерами 3 и $n-2$.
 9. Даны целочисленная матрица $n \times 3$, целые числа k, l ($1 \leq k \leq n, 1 \leq l \leq n, k < l$). Преобразовать матрицу так, чтобы строка с исходным номером k непосредственно следовала за строкой с исходным номером l , сохранив порядок следования остальных строк.
 10. В данной действительной квадратной матрице порядка n найти наибольший по модулю элемент. Получить квадратную матрицу порядка $n-1$ путем выбрасывания из исходной матрицы какой-нибудь строки и столбца, на пересечении которых расположен элемент с найденным значением.
 11. Дана действительная квадратная матрица порядка n , все элементы которой различны. Найти наибольший элемент среди стоящих на главной и побочной диагоналях и поменять его местами с элементом, стоящим на пересечении этих диагоналей.
 12. Построить квадратную матрицу порядка $2n$.

$$\begin{pmatrix} 1 & 1 & \dots & 1 & 2 & 2 & \dots & 2 \\ 1 & 1 & \dots & 1 & 2 & 2 & \dots & 2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & \dots & 1 & 2 & 2 & \dots & 2 \\ 3 & 3 & \dots & 3 & 4 & 4 & \dots & 4 \\ 3 & 3 & \dots & 3 & 4 & 4 & \dots & 4 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 3 & 3 & \dots & 3 & 4 & 4 & \dots & 4 \end{pmatrix}$$
 13. Дано действительное число x . Получить квадратную матрицу порядка $n=10$. Середина заполняется нулями.

$$\begin{array}{l}
 \text{e)} \left| \begin{array}{cccc|c} 1 & 1 & \cdot & 1 & 1 \\ 1 & & & & 1 \\ 1 & & 0 & & 1 \\ 1 & & & & 1 \\ 1 & 1 & \cdot & 1 & 1 \end{array} \right| \quad \text{ж)} \left| \begin{array}{cccc|c} 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & \\ 3 & 3 & 3 & & \\ \cdot & \cdot & \cdot & \cdot & 0 \\ n & & & & \end{array} \right| \quad \text{з)} \left| \begin{array}{cccc|c|c} 1 & 1 & 1 & \cdot & 1 & 1 & 1 \\ & 1 & 1 & \cdot & 1 & 1 & \\ & & 1 & \cdot & 1 & & \\ 0 & & 1 & \cdot & 1 & & \\ & & 1 & 1 & \cdot & 1 & 1 \\ 1 & 1 & 1 & \cdot & 1 & 1 & 1 \end{array} \right|
 \end{array}$$

$$\begin{array}{l}
 \text{и)} \left| \begin{array}{cccc|c} n & & & & \\ n-1 & n & & 0 & \\ n-2 & n-1 & n & & \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & 2 & 3 & n & \end{array} \right| \quad \text{к)} \left| \begin{array}{ccc|c} 1 & 0 & 1 & \\ 1 & 1 & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & \end{array} \right| \quad \text{п)} \left| \begin{array}{cccc|c} 1 & 2 & 3 & \cdot & n \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ n-2 & n-1 & n & & \\ n-1 & n & & 0 & \\ n & & & & \end{array} \right|
 \end{array}$$

$$\begin{array}{l}
 \text{м)} \left| \begin{array}{ccc|c} 1 & 0 & & n \\ & 2 & n-1 & \\ 0 & \cdot & \cdot & 0 \\ & 2 & n-1 & \\ 1 & 0 & & n \end{array} \right| \quad \text{н)} \left| \begin{array}{cccc|c} 1 & 2 & 3 & \cdot & n-1 & n \\ 2 & 1 & 2 & \cdot & n-2 & n-1 \\ 3 & 2 & 1 & \cdot & n-3 & n-2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ n-1 & n-2 & n-3 & \cdot & 1 & 2 \\ n & n-1 & n-2 & \cdot & 2 & 1 \end{array} \right|
 \end{array}$$

Практическая работа № 11

Тема: «Матричная алгебра»

В данной лабораторной работе рассматриваются действия, совершаемые над матрицами.

Задача: найти произведение двух матриц.

$$A = \begin{pmatrix} -1 & -2 & -4 \\ -1 & -2 & -4 \\ 1 & 2 & 4 \end{pmatrix} \text{ и } B = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{pmatrix}.$$

Решение: Вычислим произведения АВ. Согласно правилу умножения матриц элемент матрицы АВ, стоящий в *i*-ой строке и *j*-м столбце (c_{ij}) равен сумме произведений элементов *i*-й строки матрицы А на соответствующие элементы *j*-го столбца матрицы В. Так, например,

$$C_{23} = (-1) \cdot 3 + (-2) \cdot 6 + (-4) \cdot 9 = -51$$

Подсчитав таким образом все элементы матрицы АВ, находим:

$$AB = \begin{pmatrix} -17 & -34 & -51 \\ -17 & -34 & -51 \\ 17 & 34 & 51 \end{pmatrix}.$$

Мы видим на этом примере, что произведение матриц зависит от порядка сомножителей.

Примеры решений задач

1. Составить программу нахождения произведения двух матриц А и В размером 2x3 и 3x3 соответственно.

Элементы результирующей матрицы С (размером 2x3) определяются по

формуле: $c_{ij} = \sum_{k=1}^m a_{ik} b_{kj}$; $i = 1, 2, \dots, n$; $j = 1, 2, \dots, p$, где n - число строк матрицы А; m - число

столбцов матрицы А и число строк матрицы В; p - число столбцов В. В общем случае результирующая матрица С имеет n строк и p столбцов.

Program UM;

Var

A,C,:array [1..2,1..3] of real;

B:array [1..3,1..3] of real;

I,J,K: integer;

Begin

For I:=1 to 2 do

For J:=1 to 3 do

Read (A[I,J]);

For I:=1 to 3 do

For J:=1 to 3 do

Read (B[I,J]);

For I:=1 to 2 do

For J:=1 to 3 do

Begin

C[I,J]:=0;

```

For K:=1 to 3 do
    C[I,J]:=C[I,J]+A[I,K]*B[K,J];

```

```

End;

```

```

Begin

```

```

Writeln('Массив C');

```

```

    For J:=1 to 3 do

```

```

        Write (C[I,J]:12:5)

```

```

    End

```

```

End.

```

2. Составьте программу вычисления скалярного произведения двух векторов U и V , состоящих из четырех элементов каждый, по формуле:

$$S = \sum_{i=1}^4 u_i v_i . \text{ Вычислите длину вектора } D_v = \sqrt{\sum_{i=1}^4 v_i^2} .$$

```

Program SP;

```

```

Type

```

```

N:1..4;

```

```

Var

```

```

S,DV:real;

```

```

    U,V:array [N] of real;

```

```

    I:N;

```

```

Begin

```

```

For I:=1 to 4 do

```

```

    Read (V[I]);

```

```

For I:=1 to 4 do

```

```

    Readln (U[I]);

```

```

    S:=0;

```

```

    DV:=0;

```

```

    For I:=1 to 4 do

```

```

        Begin

```

```

            S:=S+U[I]*V[I];

```

```

            DV:=DV+U[I]*V[I];

```

```

        End;

```

```

    DV:=sqrt(DV);

```

```

    Writeln ('Скалярное произведение=',S);

```

```

    Writeln ('Длина вектора=');DV

```

```

End.

```

Список задач

1. Даны матрицы A и B размера $k \times m$ и $m \times l$ соответственно. Найти произведения AB .
2. Дана квадратная матрица порядка n . Получить матрицу A^2 .
3. Даны матрицы A и B порядка n . Получить матрицу $AB-BA$.

4. Дана квадратная матрица A порядка n . Получить матрицу AB ; элементы матрицы B вычисляются по формуле:
- а) $d_{ij}=1/(i+j-1)$; б) $b_{ij} = \begin{cases} 1/(i+j-1), & \text{если } i \leq j \\ 1/(i+j+1), & \text{иначе} \end{cases}$
- в) $b_{ij} = \begin{cases} 1/(i+j-1), & \text{если } i < j \\ 0, & \text{если } i = j \\ 1/(i+j+1), & \text{если } i > j \end{cases}$
5. Даны квадратная матрица A порядка n и вектор c с n элементами. Получить вектор:
- а) Ab ;
 б) A^2b ;
 в) $(A-E)b$, где E - единичная матрица порядка n .
6. Дана квадратная матрица порядка n . Получить вектор Ab , где вектор b - вектор, элементы которого вычисляются по формуле:
- а) $b_i=1/(i^2+2)$; б) $\begin{cases} 1/(i^2+2), & \text{если } i - \text{четное} \\ 1/i, & \text{иначе} \end{cases}$
7. Даны квадратная матрица A порядка n и вектор x и y с n элементами. Получить вектор $A(x+y)$.
8. Даны квадратная матрица A , B , C порядка n . Получить матрицу $(A+B)C$.
9. Даны квадратные матрицы A и B порядка n . Получить матрицу $A(B-E)+C$, где E - единичная матрица порядка n , а элементы матрицы C вычисляются по формуле: $c_{ij}=1/(i+j)$, $i, j=1, 2, \dots, n$.
10. Дана матрица A размера $m \times n$. Получить транспонированную матрицу A ($m \times n$).
11. Дана матрица A :
- а) размера $m \times m$;
 б) размера $m \times n$;
 Получить матрицу AA ($m \times m$).
12. След квадратной матрицы равен сумме элементов, расположенных на главной диагонали. Даны квадратная матрица порядка m , натуральное число n . Вычислить следы матриц A , A^2 , ..., A^n .
13. Даны целые числа A_1, A_2, A_3 . Получить целочисленную матрицу $[B_{ij}]$, $i, j = 1, 2, 3$, для которой $B_{ij}=A_i-3 \cdot A_j$.
14. Получить $[A_{ij}]$, $i=1, 10$, $j=1, 12$, где $A_{ij}=i+2 \cdot j$.
15. Получить $[A_{ij}]$, $i=1, n$, где $A_{ij}=1/(i+j)$.
16. Даны натуральное число n , действительная матрица $n \times 9$. Найти среднеарифметическое:
- а) каждого из столбцов;
 б) каждого из столбцов, имеющих четные номера.
17. Дана действительная матрица $n \times m$. Определить числа B_1, \dots, B_m , равные соответственно:
- а) сумме элементов строк;
 б) произведениям элементов строк;
 в) наименьшим значениям элементов строк;
 г) значениям среднеарифметических элементов строк;
 д) разностям наибольших и наименьших значений строк.
18. Дана матрица $n \times n$. Получить последовательность главной диагонали.
19. Дана матрица 6×9 . Найти среднеарифметическое наибольшего и наименьшего значений ее элементов

20. Дана матрица $n \times m$. Найти сумму наибольших значений элементов ее строк.
21. В данной действительной квадратной матрице $n \times n$ найти сумму элементов строки, в которой расположен элемент с наибольшим значением. Предполагается, что такой элемент единственный.

Практическая работа № 12

Тема: «Тип данных множество»

Типом множеством называется множество-степень исходного множества объектов порядкового типа, т.е. множество всевозможных сочетаний объектов исходного множества.

Число элементов исходного множества не может быть больше 256, а порядковые номера элементов должны находиться в пределах от 0 до 255.

Type

<имя множества> = **set of** *<тип компонент>*;

Var

<переменная>:*<имя множества>*;

Допустимые операции с множествами:

‘+’ объединение;

‘-’ разность;

‘*’ пересечение;

‘=’, ‘<=’, ‘>=’ проверка эквивалентности двух множеств;

‘<>’ проверка неэквивалентности двух множеств;

in логический оператор проверки присутствия компоненты в множестве.

Примеры решений задач

1. Ввести строку символов, состоящую из латинских букв, символов и пробелов. Осуществить проверку правильности введенных символов.

Program stroka;

Var

Str: **string**;

L: **byte**;

Tru: **boolean**;

Begin

Writeln (‘Введите строку’)

Readln (str);

L:=**Length** (Str);

Tru:= L>0;

While Tru and (L>0) **do**

Begin

 Tru:=Str[L] **in** [‘0’,’9’,’A’..’Z’,’a’..’z’,’ ‘];

Dec (L)

End;

If Tru **then**

WriteLn (‘Правильная строка’);

Else

WriteLn (‘Неправильная строка’);

End.

Список задач

1. Дан текст из строчных латинских букв, за которым следует точка. Определить, каких букв - гласных (a, e, i, o, u) или согласных - больше в этом тексте.
2. Напечатать в возрастающем порядке все цифры, не входящие в десятичную запись натурального числа n.
3. Дан текст из строчных латинских букв, за которым следует точка. Напечатать:
 - а) все буквы, входящие в текст не менее двух раз;
 - б) все буквы, входящие в текст по одному разу;
 - в) первые вхождения букв в текст, сохраняя их исходный взаимный порядок.
4. В возрастающем порядке напечатать целые числа из диапазона 1.. 10000, представимые в виде $n^2 + m^2$, где $n, m > 0$
5. Напечатать слова предложения в алфавитном порядке.
6. Дана последовательность целых чисел. Определить, является ли эта последовательность перестановкой заданного отрезка элементов натурального ряда.
7. Подсчитать количество чётных цифр в исходной символьной строке и распечатать все, кроме пробелов, знаков операций и знаков препинания.
8. Сформировать множество, в которое входят только латинские буквы, встретившиеся во входной строке, и множество знаков препинания из входной строки.

Практическая работа № 13

Тема: «Перечисляемый и интервальный (диапазоны) типы данных»

Перечисляемый тип - это определяемый пользователем тип данных, при котором количество всех возможных значений ограничено. Перечислимые данные должны иметь синтаксис идентификаторов, и поэтому не могут перечисляться цифры, символы, строки.

Type

<имя типа>:(*<элемент 1>*, [*<элемент 2>*, ..., *<элемент n>*]);

Функции обработки данных перечисляемого типа:

Succ (x) – возвращает следующее за x значение в перечислимом типе.

Pred (x) - возвращает предыдущее значение в перечислимом типе.

>, <, >=, <=, <>, = - операции сравнения.

Интервальный тип - это определяемый пользователем тип данных, который содержит значения только из ограниченного поддиапазона некоего базового типа. Базовым типом, из которого вычлняются диапазоны, может быть любой целочисленный тип, тип Char и любой из введенных программистом перечислимых типов.

Type

<имя типа>:*<интервал>*

Пример:

Type

Days_weeks: 1..7;

Примеры решений задач

1. Для целого числа *k* от 1 до 99 напечатать фразу "мне *k* лет", учитывая при этом, что при некоторых значениях *k* слово "лет" заменить на слово "год" или "года";

Program how_old_year;

Const t=10;

Var a,b:Integer;

k:1..99;

Begin

Readln(k);

If (k>=5) and (k<=19) **then WriteLn**(k,'лет');

a:=k div t;

b:=k-a*t;

If (b>=5) and (b<=9) **then WriteLn**(k,'лет');

If b=0 **then Write**(k,'лет');

If (b>=2) and (b<=4) **then WriteLn**(k,'года');

If b=1 **then Write**(k,'год');

End.

Список задач

1. Type

падеж=(им, род, дат, вин, твор, предл);

слово=(степь, боль, тетрадь, дверь);

Var w:слово;

p:падеж;

Напечатать слово w в падеже p и единственном числе (например, при $w = \text{степь}$ и $p = \text{твор.}$ надо напечатать слово "степью");

2. Переменной d присвоить количество дней в месяце m (год считать не високосным);
3. По дате d, m, y определить $d1, m1, y1$ -дату следующего дня (**Var** $d,d1:1..31; m,m1:\text{месяц}; y:1901..2000; y1:1901..2001;$);
4. Определить k – порядковый номер того дня високосного года, который имеет дату d,m ;
5. Пусть значение функции $f(n)$ равно количеству букв в записи числа n русскими словами: $f(1)=4$ (один). $f(3)=3$ (три), $f(42)=8$ (сорок два) и т. п. Напечатать все натуральные числа n , меньшие 100, для которых $f(n)=n$.
6. В старояпонском календаре был принят 60-летний цикл, состоящий из пяти 12-летних подциклов. Подциклы обозначались названиями цветов: зелёный, красный, жёлтый, белый и чёрный. Внутри каждого подцикла годы носили названия животных: крысы, коровы, тигра, зайца, дракона, змеи, лошади, овцы, обезьяны, курицы, собаки и свиньи. (1984 год – год зелёной крысы – был началом очередного цикла).
Написать программу, которая выводит номер некоторого года нашей эры и печатает его название по старояпонскому календарю.

Практическая работа № 14

Тема: «Тип данных запись»

Тип запись включает ряд компонент, называемых полями, которые могут быть разных типов. При задании типа-записи после зарезервированного слова **record** следует перечислять все поля типа записи с указанием через двоеточие их типов и завершить задание типа словом **end**. Поля отделяются друг от друга точкой с запятой. Количество полей записи может быть любым.

Тип данных запись описывается следующим образом:

Type

Record

<имя поля 1>:<тип поля 1>;

<имя поля 2>:<тип поля 2>;

. . .

<имя поля n>:<тип поля n>;

End;

Если тип нескольких полей совпадает, то имена полей могут быть просто перечислены. После объявления в программе переменной типа «запись» к каждому ее полю можно обратиться, указав сначала идентификатор переменной-записи, а затем через точку – имя поля. Поле записи может иметь практически любой тип. Доступ к вложенным элементам таких структур осуществляется по тем же правилам, как и обычно.

Переменная типа «запись» может участвовать только в операциях присваивания. Но поле записи может принимать участие во всех операциях, применимых к типу этого поля.

Присвоение значений переменным типа запись производится одним из двух способов:

1) *<имя переменной>.<имя поля 1>:=<выражение>;*

<имя переменной>.<имя поля 2>:=<выражение>;

. . .

<имя переменной>.<имя поля n>:=<выражение>;

2) для облегчения работы с полями записей вводится оператор присоединения.

With *<имя переменной>* **do**

begin

<имя поля 1>:=<выражение>

<имя поля 2>:=<выражение>

. . .

<имя поля n>:=<выражение>

end;

Примеры решений задач

1. Для каждого студента указаны фамилия и оценки в баллах по пяти дисциплинам. Требуется вычислить средний балл.

```

Program BAL;
    Type STR=
Record
    Fam: Array [1..15] of char;
    B1, B2, B3, B4, B5: 2..5;
    SB: real;
    End;
Var
    Tbl: array[1..25] of str;
    I: integer;
Begin
    For I:=1 to 25 do
        Read (TBL [I].FAM, TBL [I].B1, TBL [I].B2,
        TBL [I].B3, TBL [I].B4, TBL [I].B5);
    For I:=1 to 25 do
        begin
            TBL[I].SB:=(TBL[I].B1+TBL[I].B2+TBL[I].B3+
            + TBL[I].B4+TBL[I].B5)/5;
            Writeln(TBL[I].SB);
        end;
End.

```

Список задач

1. Сформировать переменную типа запись, в которой расположены данные о каждом отдельном ученике в следующем порядке: имя (15 символов), фамилия (15 символов), год обучения (целое число), буква (символ). Требуется перенести эти данные в другую переменную выводя первую букву имени и фамилию ученика:
И. Петров
П. Иванов
и т. д.
2. Переменная содержит сведения об учениках некоторой школы (см. задачу 1).
 - а) Собрать в сведения об учениках девярых классов школы,
 - б) Выяснить, на сколько человек в восьмых классах больше, чем в девярых.
3. Багаж пассажира характеризуется количеством вещей и общим весом вещей. Сформировать переменную *Bagaj*, содержащую сведения о багаже нескольких пассажиров. Сведения о багаже каждого пассажира представляют собой запись с двумя полями: одно поле целого типа (количество вещей) и одно-действительное (вес в килограммах).
 - а) Найти багаж, средний вес одной вещи в котором отличается не более, чем на 0,3 кг от общего среднего веса одной вещи.
 - б) Найти число пассажиров, имеющих более двух вещей и число пассажиров, количество вещей которых превосходит среднее число вещей.
 - в) Выяснить, имеется ли пассажир, багаж которого состоит из одной вещи весом менее 30 кг.

4. Упорядочить сведения о багаже, записанные в переменной `bagaje` (см. предыдущую задачу) по невозрастанию веса багажа. Предполагается, что число пассажиров, зарегистрировавших багаж, известно заранее и равно p (некоторая константа), при этом p – не слишком велико. Указание. Перенести сведения о багаже из переменной `багаж` в массив V_1, \dots, V_p , затем упорядочить этот массив, используя то, что для переменных x, y одного и того же комбинированного типа можно использовать оператор присваивания $x:=y$. После этого переписать элементы массива V_1, \dots, V_p в переменную `Bagaje`.
5. Требуется удалить из данной переменной `Bagaje` сведения о багаже, общий вес вещей в котором меньше, чем 10 кг. Использовать вспомогательную переменную F .
6. Переписать сведения о багаже из переменной `Bagaj` в переменную `Bag`. В переменной `Bag` сведения о багаже каждого пассажира представляются массивом из двух целых чисел – числа вещей и общего веса вещей, выраженного в граммах. Составить также программу обратного преобразования: переписи сведений о багаже из переменной `Bag` в переменную `Bagaj`.
7. Сформирована переменная `bibl`, содержащий сведения о книгах. Сведения о каждой из книг – это фамилия автора, название и год издания.
 - а) Найти названия книг данного автора, изданных с 1960 года
 - б) Определить имеется ли книга с названием "Информатика". Если да, то сообщить фамилию автора и год издания. Если таких книг несколько, то сообщить сведения обо всех этих книгах.
8. Дана переменная T , которая содержит номера телефонов сотрудников учреждения: Указывается фамилия сотрудника, его инициалы и номер телефона. Найти номер телефона сотрудника по его фамилии и инициалам.
9. Сформирована переменная типа запись, содержащая различные даты. Каждая дата – это число, месяц и год. Найти:
 - а) год с наименьшим номером.
 - б) все весенние даты.
 - в) самую позднюю дату.
10. Сформировать переменную `Tovar`, содержащую сведения об экспортируемых товарах: Указывается наименование товара, страна импортирующая товар, и объем поставляемой партии в штуках. Составить список стран, в которые экспортируется данный товар, и общий объем его экспорта.
11. Сформирована переменная `Assortim`, содержащая сведения об игрушках: указано название игрушки, ее стоимость в рублях, и возрастные границы. Получить следующие сведения:
 - а) название игрушек цена которых не превышает 4 руб., и которые подходят детям 5 лет.
 - б) цену самого дорогого конструктора.
 - в) Название наиболее дорогих игрушек. (цена которых отличается не более чем на 1 руб. от самой дорогой.)
 - г) название игрушек которые подходят как детям 4 лет так и детям 10 лет.
 - д) можно ли подобрать игрушку, любую кроме мяча, подходящую ребенку 3 лет, и дополнительно мяч, так, чтобы суммарная стоимость игрушек не превосходила 5 руб.?

Практическая работа № 15

Тема: «Тип данных файл»

У понятия файл есть две стороны. С одной стороны, файл – это область памяти на внешнем носителе, в котором хранится некоторая информация. Файл в таком понимании называют физическим файлом, т.е. существующим физически на некотором материальном носителе информации. С другой стороны, файл – это одна из структур данных, используемых в программировании. Файл в таком понимании называют логическим файлом, т.е. существующим в нашем логическом представлении при написании программы.

Структура физического файла представляет собой простую последовательность байт памяти носителя информации. Структура логического файла – это способ восприятия файла в программе.

Любой файл имеет следующие характеристики-требования:

- у него есть имя (набор из восьми, допустимых для имени файла, символов плюс расширение, указываемое после точки в имени файла состоящее из трех символов);
- он должен содержать данные одного типа (любой тип Турбо Паскаля, кроме типа Файл, то есть не существует типа «Файл файлов»);

Длина создаваемого файла никак не регламентируется при создании файла и ограничивается только емкостью носителя информации.

Работа с файлами в Турбо Паскале осуществляется следующим образом: сначала объявляется переменная файлового типа, с указанием свойств переменной (то есть типом содержимого), затем данная файловая переменная связывается («ассигнуется») с именованным дисковым пространством (то есть непосредственно с конкретным файлом, содержащим или, который будет содержать данные того же типа, что и связываемая переменная-файл) или логическим устройством)

Переменная файлового типа может быть объявлена одной из следующих строк:

<имя> = **file of** <тип>;

<имя> = **text**;

<имя> = **file**;

где <имя> – имя переменной-файла;

file of – зарезервированные слова (файл, из);

text – имя стандартного типа текстовых файлов;

<тип> – имя любого стандартного типа Турбо Паскаля, кроме типа файл.

Например файл, содержащий список учеников и их возраст:

type

```
pupil = record  
  surname : string;  
  name : string;  
  age : word
```

end;

Var

```
journal : file of pupil;
```

В зависимости от способа объявления можно выделить три вида файлов:

- ◆ типизированные файлы (задаются предложением **file of**);
- ◆ текстовые файлы (определяются типом **text**);
- ◆ нетипизированные файлы (определяются типом **file**).

Файловые переменные имеют специфическое назначение. Такие операции, как присвоение значения, сравнение и т.д. над переменными типа файл осуществлять нельзя.

Текстовые файлы – это файлы, содержащие символы, разделенные на строки. Причем в конце каждой строки стоит признак конца строки. Текстовые файлы не имеют прямого доступа. При чтении и записи числа преобразуются автоматически. К ним применима процедура Append(<имя переменной текстового файла>). Она открывает текущий файл, с которым связана данная переменная, текущий указатель помещает в конец для добавления новой информации.

Нетипизированные файлы предназначены для низкоуровневой работы с файлами. С их помощью можно обратиться к файлу любого типа и логической структуры. За одно обращение считывается/записывается число байт, приблизительно равное величине буфера ввода/вывода. В качестве буфера может выступать любая переменная. Для записи и чтения используются процедуры BlockRead, BlockWrite.

Процедуры и функции обработки файловых переменных

Assign (<имя файловой переменной>,'<путь и имя файла на диске>') – связь переменной файлового типа с конкретным внешним файлом.

Reset (f) – процедура открытия существующего файла и подготовка к чтению файла. Указатель текущей позиции файла устанавливается в его начало.

Rewrite (f) – процедура создания нового физического файла и подготовка к записи файла. Если такой файл существует, то он удаляется, и на этом месте создается новый пустой файл. Указатель текущей позиции файла устанавливается в его начало.

Get (f) - присваивание значения компоненты буферной переменной;

Put (f) - создание в файле новой компоненты;

Readln (f) - пропуск строки файла до начала следующей;

Writeln (f) - запись признака конца строки и переход на следующую;

Read (f, x) – процедура чтения компоненты файла. Данные выводятся из файла.

Write (f, x) – процедура записи значения переменной в файл, который хранится на диске. Указатель перемещается на следующий элемент. Если указатель текущей позиции файла находится за последним элементом, т.е. в конце файла, то файл расширяется.

Eof - признак конца файла- логическая функция для определения, достигнут ли конец файла.

Close (f) - процедура закрытия файла.

Примеры решений задач

1. Прочитать из текстового файла А все записанные в него целые числа, преобразовать их в вещественные и вывести в текстовый файл В по 4 числа в строку.

Program File1;

Var F1,F2: **text**;

X: **real**;

```

I:integer;
Begin
Assign (F1, 'A');
Reset(F1);
Assign (F2, 'B');
Rewrite (F2);
Repeat
    For I:=1 to 4 do
        If not seekeof(F1) then
            Begin
                Read (F1,x);
                Write (F2,x:18)
            End;
        Writeln(F2);
Until seekeof (F1);
Close (F1);
Close (F2);
End
End.

```

Список задач

1. Дан файл *f*, компоненты которого являются действительными числами. Найти:
 - а) сумму компонент файла;
 - б) произведение компонент файла;
 - в) сумму квадратов компонент файла;
 - г) модуль суммы и квадрат произведения компонент файла;
 - д) последнюю компоненту файла;
2. Дан файл *f*, компоненты которого являются действительными числами. Найти:
 - а) из значений компонент;
 - б) наименьшее из значений компонент с чётными номерами;
 - в) наибольшее из значений модулей компонент с нечётными номерами;
 - г) сумму наибольшего и наименьшего из значений компонент;
 - д) разность первой и последней компонент файла.
3. Дан файл *f*, компоненты которого являются целыми числами. Найти:
 - а) количество чётных чисел среди компонент;
 - б) количество удвоенных нечётных чисел среди компонент;
 - в) количество квадратов нечётных чисел среди компонент;
4. Дан символьный файл *f*. Получить копию файла в файле *g*.
5. Даны символьные файлы *f1* и *f2*. Переписать с сохранением порядка следования компоненты файла *f1* в файл *f2*. Использовать вспомогательный файл *h*.
6. Дан символьный файл *f*. В файле не менее двух компонент. Определить являются ли два первых символа файла цифрами. Если да, то установить, является ли число, образованное этими цифрами чётными.

7. Дан файл f , компоненты которого являются целыми числами. Получить в файле g все компоненты файла f :
 - а) являющиеся чётными числами;
 - б) делящиеся на 3 и не делящиеся на 7;
 - в) являющиеся точными квадратами;
8. Записать в файл g все чётные числа файла f , а в файл h = все нечётные. Порядок следования чисел сохраняется.
9. Дан символьный файл f . Записать в файл g компоненты файла f в обратном порядке.
10. Дан файл, содержащий различные даты. Каждая дата - это число, месяц и год. Найти:
 - а) год с наименьшим номером;
 - б) все весенние даты;
 - в) самую позднюю дату;
11. Даны символьные файлы f и g . Записать в файл h сначала компоненты файла f , затем компоненты файла g с сохранением порядка.
12. Дан файл f , компоненты которого являются целыми числами. Никакая из компонент файла не равна нулю. Файл f содержит столько же отрицательных чисел, сколько и положительных. Используя вспомогательный файл h , переписать компоненты файла f в файл g так, чтобы в файле g :
 - а) не было двух соседних чисел с одним знаком;
 - б) сначала шли положительные, потом отрицательные числа.
 - в) числа шли в следующем порядке: два положительных два отрицательных, два положительных, два отрицательных и т. д. (предполагается, что число компонент в файле f делится на 4).

Литература

- 1) Абрамов С.А.; Зима Е.В. Начала информатики. – М: Наука, 1989 – 256 с. (Б-чка программиста).
- 2) Дагене В.А. и др. Сто задач по программированию: кн. для уч-ся. / Пер. с лит. А.Д. Шмелева, В.А. Дагине, Г.К. Григас, К.Ф. Аугутис. – М.: Просвещение, 1993 – 255 с.
- 3) Кушнеренко А.Г. и др. Основы информатики и вычислительной техники: Пробный учебник для средних учебных заведений./ А.Г. Кушнеренко, Г.В. Лебедев, Р.А. Сворень, - 3 изд. – М.: Просвещение, 1993, 223 с.
- 4) Поляков Д.Б., Круглов И.Ю. Программирование в среде ТурбоПаскаль.
- 5) Фаронов В.В. Программирование на персональных ЭВМ в среде Турбо-Паскаль. – М.: Изд-во МГТУ, 1990. – 580 с.
- 6) Фролов Г.Д., Кузнецов Э.И. Элементы информатики: [Учеб. пособие для пед. ин-тов]. – М.: Выс. шк., 1989. – 303 с.

Содержание

Введение	1
Лабораторная работа № 1	4
Лабораторная работа № 2	6
Лабораторная работа № 3	9
Лабораторная работа № 4	133
Лабораторная работа № 5	18
Лабораторная работа № 6	200
Лабораторная работа № 7	24
Лабораторная работа № 8	28
Лабораторная работа № 9	33
Лабораторная работа № 10	36
Лабораторная работа № 11	41
Лабораторная работа № 12	45
Лабораторная работа № 13	47
Лабораторная работа № 14	49
Лабораторная работа № 15	52
Литература	56