

**Елабужский государственный
педагогический университет**

**Кафедра информатики
и
дискретной математики**

P A S C A L

Лабораторный практикум

**Елабуга
2011**

УДК 681.14
ББК 32.973
И 21

Печатается по решению Редакционно-издательского совета
Елабужского государственного педагогического университета
Протокол № 32 от «19» февраля 2009 г.

Рецензенты:

Юрзанова Т.К., доцент кафедры информационных технологий филиала КГТУ им.
Туполева, кандидат педагогических наук

Кобелев И.А., ст. преподаватель кафедры информатики и дискретной математики
ЕГПУ

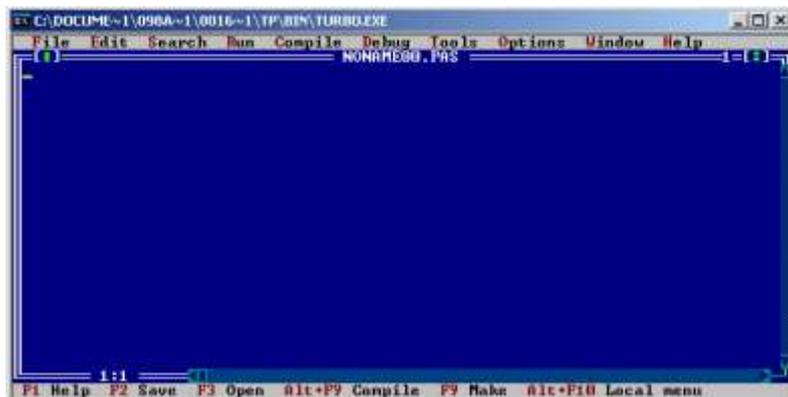
Пособие по языку программирования Pascal предназначено для практического изучения языка программирования различными способами.

СОДЕРЖАНИЕ

<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 1 ЗНАКОМСТВО С TURBO PASCAL 7.0.....</i>	<i>5</i>
<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 2 РАБОТА КОМПЬЮТЕРА В РЕЖИМЕ КАЛЬКУЛЯТОРА (ВЫЧИСЛЕНИЯ).....</i>	<i>10</i>
<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 3 РЕШЕНИЕ МАТЕМАТИЧЕСКИХ ВЫРАЖЕНИЙ.....</i>	<i>11</i>
<i>ЛИНЕЙНАЯ ПРОГРАММА.....</i>	<i>11</i>
<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 4 РЕШЕНИЕ ФИЗИЧЕСКИХ ЗАДАЧ.....</i>	<i>12</i>
<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 5 ОПЕРАТОР УСЛОВНОГО ПЕРЕХОДА.....</i>	<i>13</i>
<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 6 ОПЕРАТОР УСЛОВНОГО ПЕРЕХОДА (продолжение).....</i>	<i>14</i>
<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 7 ОПЕРАТОР УСЛОВНОГО ПЕРЕХОДА (продолжение).....</i>	<i>15</i>
<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 8 ОПЕРАТОР ВАРИАНТА.....</i>	<i>16</i>
<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 9 ОПЕРАТОР ВАРИАНТА (продолжение).....</i>	<i>17</i>
<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 10 ПОДГОТОВКА К КОНТРОЛЬНОЙ РАБОТЕ.....</i>	<i>18</i>
<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №11 ЦИКЛИЧЕСКИЕ ПРОЦЕССЫ.....</i>	<i>19</i>
<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 12 ЦИКЛИЧЕСКИЕ ПРОЦЕССЫ (продолжение).....</i>	<i>22</i>
<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 13 ЦИКЛИЧЕСКИЕ ПРОЦЕССЫ (продолжение).....</i>	<i>23</i>
<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 14 СУММА. ПРОИЗВЕДЕНИЕ. ТАБУЛИРОВАНИЕ.....</i>	<i>24</i>
<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 15 ТАБУЛИРОВАНИЕ (продолжение).....</i>	<i>25</i>
<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 16 ТАБЛИЦА ПИФАГОРА.....</i>	<i>26</i>
<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 17 ПРОЦЕДУРЫ.....</i>	<i>27</i>
<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 18 ПРОЦЕДУРЫ (продолжение).....</i>	<i>28</i>
<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 19 ПРОЦЕДУРЫ (продолжение).....</i>	<i>29</i>
<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 20 ФУНКЦИИ.....</i>	<i>31</i>
<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 21 ФУНКЦИИ (продолжение).....</i>	<i>32</i>
<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 22 ПОВТОРЕНИЕ. ПОДГОТОВКА К КОНТРОЛЬНОЙ РАБОТЕ.</i>	<i>32</i>
<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 23 ПОВТОРЕНИЕ.....</i>	<i>34</i>
<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 24 СИМВОЛЬНЫЕ И СТРОКОВЫЕ ФУНКЦИИ.....</i>	<i>35</i>
<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 25 СИМВОЛЬНЫЕ И СТРОКОВЫЕ ФУНКЦИИ (продолжение).....</i>	<i>36</i>
<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 26 ОБРАБОТКА СИМВОЛЬНЫХ И СТРОКОВЫХ ВЕЛИЧИН 1.....</i>	<i>37</i>
<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 27 ОБРАБОТКА СИМВОЛЬНЫХ И СТРОКОВЫХ ВЕЛИЧИН 2.....</i>	<i>39</i>
<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 28 ОБРАБОТКА СИМВОЛЬНЫХ И СТРОКОВЫХ ВЕЛИЧИН 3.....</i>	<i>39</i>
<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 29 ОБРАБОТКА СИМВОЛЬНЫХ ВЕЛИЧИН 4.....</i>	<i>40</i>
<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 30 ЗАПОЛНЕНИЕ И ВЫВОД МАССИВОВ.....</i>	<i>42</i>
<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 31 ЗАПОЛНЕНИЕ И ВЫВОД МАССИВОВ (продолжение).....</i>	<i>43</i>
<i>ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 32 ОБРАБОТКА ОДНОМЕРНЫХ ЧИСЛОВЫХ МАССИВОВ.....</i>	<i>44</i>

СУММИРОВАНИЕ ЭЛЕМЕНТОВ МАССИВА.....	44
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 33 ОБРАБОТКА ОДНОМЕРНЫХ ЧИСЛОВЫХ МАССИВОВ.....	45
СУММА ЭЛЕМЕНТОВ МАССИВА ПО УСЛОВИЮ.....	45
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 34 ОБРАБОТКА ОДНОМЕРНЫХ МАССИВОВ.....	46
ОПРЕДЕЛЕНИЕ ЧИСЛА ЭЛЕМЕНТОВ МАССИВА ПО УСЛОВИЮ.....	46
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 35 ОБРАБОТКА ОДНОМЕРНЫХ ЧИСЛОВЫХ МАССИВОВ.....	47
СУММА ДВУХ ОДИНАКОВЫХ ПО РАЗМЕРУ МАССИВОВ.....	47
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 36 ОБРАБОТКА ОДНОМЕРНЫХ МАССИВОВ.....	48
ОБЪЕДИНЕНИЕ ДВУХ МАССИВОВ С ЧЕРЕДОВАНИЕМ ЭЛЕМЕНТОВ.....	48
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 37 ОБРАБОТКА ОДНОМЕРНЫХ МАССИВОВ.....	49
ИНВЕРТИРОВАНИЕ МАССИВА.....	49
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 38 ОБРАБОТКА ОДНОМЕРНЫХ МАССИВОВ ФОРМИРОВАНИЕ МАССИВА ИЗ ЭЛЕМЕНТОВ ДРУГОГО МАССИВА УДОВЛЕТВОРЯЮЩИХ ЗАДАННОМУ УСЛОВИЮ.....	50
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 39 ОБРАБОТКА ОДНОМЕРНЫХ МАССИВОВ ПОИСК МАКСИМАЛЬНОГО ЭЛЕМЕНТА В МАССИВЕ С ЗАПОМИНАНИЕМ ЕГО ПОЛОЖЕНИЯ (ИНДЕКСА).....	51
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 40 ОБРАБОТКА ОДНОМЕРНЫХ МАССИВОВ ПОИСК ЗАДАННОГО ЭЛЕМЕНТА В МАССИВЕ.....	52
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 41 ОБРАБОТКА ОДНОМЕРНЫХ МАССИВОВ УПОРЯДОЧЕНИЕ МАССИВА.....	53
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 42 ОБРАБОТКА МАТРИЦ ВВОД И ВЫВОД МАТРИЦ.....	54
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 43 ОБРАБОТКА МАТРИЦ.....	55
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 44 ОБРАБОТКА МАТРИЦ.....	56
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 45 ПОДГОТОВКА К КОНТРОЛЬНОЙ РАБОТЕ.....	58
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 46. ЗАПИСИ.....	59
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 47. ФАЙЛЫ.....	61
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 48. ЗАДАЧИ ОПТИМИЗАЦИИ.....	63
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 49.....	64
ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 50. ЗНАКОМСТВО С IBM. ОПЕРАЦИОННАЯ СИСТЕМА. NORTON COMMANDER.....	66
ПРИЛОЖЕНИЕ.....	69
ЛИТЕРАТУРА.....	72

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 1 ЗНАКОМСТВО С TURBO PASCAL 7.0



Turbo Pascal 7.0 - реализация языка Pascal фирмой Borland International для IBM PC. Можно работать в Pascal ABC, заменяя `readkey` на `readln`. (см. приложение стр. 65, 66)
Цель разработки: обеспечение всесторонней поддержки всего технологического цикла создания программ производственного,

научного и коммерческого назначения.

ВХОД В СРЕДУ TurboPascal: Войти в меню пользователя, выбрать версию Turbo Pascal и нажать Enter. Либо найти файл **turbo.exe** в папке `c:\Pascal\turbo7\bin` и щелкнуть **Enter**.

ГЛАВНОЕ МЕНЮ



F10 - ВЫХОД В МЕНЮ

File - меню работы с файлами;

Edit - меню редактора;

Search - меню службы поиска;

Run - меню запуска программы;

Compile - меню компиляции;

Debug - меню отладки;

Options - меню установок;

Window - меню окон;

Help - меню справочной службы.

File: Open...F3 - открыть существующую на диске программу;

New - открыть новое окно редактора;

Save F2 - записать программу на диск с текущим именем;

Save as - записать программу на диск с указанием имени;

Exit - выход из среды или **Alt+X**

Ctrl+F2 - прекратить отладку (убрать голубую полосу); **Ctrl+F8** – красную полосу.

Ctrl+F4 - просмотр значения переменной из редактора;

ФУНКЦИИ КЛАВИШ

Ctrl,Alt,Shift - управляющие клавиши;

Esc - отмена команды;

BackSpace - удаление символа слева от курсора со сдвигом строки;

Insert - переключение режима вставки/замены;

Delete - удаление символа над курсором;

Home	- курсор в начало строки;
End	- курсор в конец строки;
Page Up	- курсор на страницу вверх;
Page Down	- курсор на страницу вниз;
Caps Lock	- переключения заглавные/строчные или Shift+буква;
Ctrl+Break+Break	- останов работы программы;
Ctrl+Y	- удалить строку;
Ctrl+N	- вставить строку;
Shift+↑ [↓, →, ←, Home, End, Page Up, Page Down]	- выделить фрагмент программы
Ctrl+K B	- отметить начало выделения фрагмента;
Ctrl+K K	- отметить конец выделения фрагмента;
Ctrl+K C	- скопировать выделенный фрагмент;
Ctrl+K V	- переместить выделенный фрагмент;
Ctrl+K Y	- удалить выделенный фрагмент;
Ctrl+K H	- снять/вернуть выделение;
Ctrl+F9	- выполнить компиляцию и запуск программы;

МОДУЛИ

Turbo Pascal содержит библиотеку модулей: SYSTEM, DOS, CRT, GRAPH и др.

Если работа производится в текстовом режиме, надо объявлять модуль CRT, если требуется выполнить графические работы, то объявляется графический модуль Graph.

Например,

```
Program Primer;
  USES CRT, GRAPH;
  Var ...
```

Чтобы увидеть результаты после выполнения программы, надо перед окончанием программы указать READLN или READKEY, без этих команд после выполнения программы осуществляется автоматический возврат в редактор, в этом случае для просмотра экрана пользователя надо нажать клавиши **Alt+F5**;



ЗАДАНИЯ

1. Выйти в редактор и получить на экране строки :

```
* : = - / ? ( ) [ { ] } ' ` , . " ; + < > % _ 1 2 0 9 4 8 7 6 3 5
abcdefghijklmnopqrstuvwxyz ABCDEFGHIJKLMNOPQRSTUVWXYZ
PROGRAM program CONST const type TYPE Begin End procedure
{ Программа вычисления функции Y=sin(x) }
```

2. В полученном на экране тексте раздвинуть строки.

3. Разделить первые три строки пополам, т.е. получить семь строк.
4. Удалить шестую строку.
5. Удалить пустые строки.
6. Скопировать первую строку в конец текста.
7. Поместить первую строку между 4 и 5 строками.
8. Переместить 3 и 4 строки в начало текста.
9. Подчеркнуть строки 1-ю - '*', 2-ю - '=', 3-ю - '-'.

ПРИЛОЖЕНИЕ К ПРАКТИЧЕСКОМУ ЗАНЯТИЮ ОСНОВНЫЕ ТИПЫ ЯЗЫКА ПАСКАЛЬ:

Integer - целый тип - целое число от -32768 до +32767, занимает 2 байта памяти
Real - вещественный тип - числа от 1E-37 до 15.9E37
Boolean - логический тип - значение высказывания true или false
char - символьный тип (один символ)
byte - стандартный байтовый тип - целые от 0 до 255, занимает 1 байт памяти
string - строковый тип.

Целые типы:

BYTE - целые положительные из интервала [0; 255];
WORD - целые положительные из интервала [0; 65535];
SHORTINT - целые из интервала [-128; 127];
INTEGER - целые из интервала [-32768; 32767];
LONGINT - целые из интервала [-2147483648; 214748647];

СТАНДАРТНЫЕ МАТЕМАТИЧЕСКИЕ ФУНКЦИИ

ABS(X) эбсолют - модуль $|x|$
SIN(X) синус - синус x , где x в радианах. Перевод град .в рад. $X=X*PI/180$
COS(X) косинус - косинус x , где x в радианах
ARCTAN(X) арктан - арктангенс x , где результат в радианах
SQR(X) сквэа - квадрат x **SQuaRe** -англ.
SQRT(X) сквэати - квадратный корень из x **SQuare RooT** -англ.
PI пи - константа ($\pi \approx 3.141592$)
ROUND(X) раунд - округление x . Результат - integer
TRUNC(X) транк - целая часть числа без округления. Результат - integer
INT(X) инт - целое, не превосходящее $+X$. Результат - real
ODD(X) од - нечетность. $ODD(7) = true$, $ODD(8) = false$
FRAC(X) фрак - дробная часть числа
EXP(X) эксп - экспонента функция $y=e$ ($e \approx 2.71828$)
LN(X) лн - натуральный логарифм $y=\ln x$ (по основанию e).

ФОРМУЛЫ ВОЗВЕДЕНИЯ В СТЕПЕНЬ

$$a^x = \text{EXP}(\text{LN}(a) * x) \quad - \text{ для } a > 0 \text{ и } x > 1$$

$$\sqrt[x]{a} = \text{EXP}(\text{LN}(\text{ABS}(a)) / x) \quad - \text{ для } 0 < x < 1, x - \text{ четное число}$$

$$\sqrt[x]{a} = \text{EXP}(\text{LN}(\text{ABS}(a)) / x) * \text{ABS}(a) / a \quad - \text{ для } 0 < x < 1, x - \text{ нечетное число}$$

ЗАПИСЬ МАТЕМАТИЧЕСКИХ ВЫРАЖЕНИЙ

Все выражения записываются в одну строку с использованием круглых скобок и знаков арифметических операций согласно **ПРИОРИТЕТА** действий:

- | | |
|-----------------------------------------------|------------------------|
| 1. Действия в скобках. | () |
| 2. Вычисление функций. | SIN, SQR, ABS и другие |
| 3. Целочисленное деление. Остаток от деления. | DIV MOD |
| 4. Умножение. Деление. | * / |
| 5. Сложение. Вычитание. | + - |
| 6. Операции отношений. | =, >, <, <>, <=, >= |
| 7. Булевы (логические) операции. | Not, And, Or |

ЗАПИСЬ МАТЕМАТИЧЕСКИХ ВЫРАЖЕНИЙ НА ПАСКАЛЕ:

В МАТЕМАТИКЕ:

$A + \frac{B}{C}$	$A + B / C$
$\frac{A + B}{C}$	$(A + B) / C$
$\frac{C}{A - B}$	$C / (A - B)$
$\frac{A}{B \cdot C}$	$A / B / C = A / (B * C)$
$\frac{A \cdot B}{C}$	$A * B / C = A / C * B = B / C * A$
$\frac{2 + \sqrt{ A - B }}{B \cdot C} - A$	$(2 + \text{sqrt}(\text{abs}(A - B))) / (B * C) - A$
$\text{Sin } 30^\circ$	$\text{sin}(30 * \text{PI} / 180)$
$\text{Arctg}(1)$	$\text{arctan}(1) = 0.785 \quad (\text{радиан})$ $\text{arctan}(1) * 180 / \text{pi} = 45 \quad (\text{градусов})$
$\frac{A}{-B}$	$A / (-B)$
$5 \cdot 10^{-6}$	$5e-6$
$\sin^2 \alpha^2$	$\text{sqr}(\text{sin}(A * A))$
$\text{tg} \alpha$	$\text{sin}(A) / \text{cos}(A)$
$\text{ctg } 60^\circ$	$\text{cos}(60 * \text{pi} / 180) / \text{sin}(60 * \text{pi} / 180)$ или $A := 60 * \text{pi} / 180$ затем $\text{cos}(A) / \text{sin}(A)$

СТРУКТУРА ПРОГРАММЫ

Программа на Паскале имеет имя и состоит из символов. В программе используются специальные зарезервированные слова, значение которых не меняется и они должны записываться без ошибок. Структура программы на Паскале:

Program Name; { название программы без пробелов }

{ разделы описаний записываются по мере надобности }

Uses - раздел объявления модулей;

label - раздел описания меток;

const - раздел описания констант;

type - раздел описания нестандартных типов;

var - раздел описания переменных (используется чаще всего);

function - раздел описания функций;

procedure - раздел описания процедур;

{ основная программа }

Begin

{ тело основной программы }

Readkey; или Readln; - ожидание нажатия клавиши (Enter) для просмотра результата.

End.

Зарезервированные слова нельзя использовать в качестве имен переменных, констант, функций, процедур и типов. Текст программы может записываться как большими, так и маленькими буквами. Русские буквы используются только для комментариев, которые записываются двумя способами: { текст комментария } или (* текст комментария *).

ОПЕРАТОР ВЫВОДА ИНФОРМАЦИИ НА ЭКРАН

write(a); - стандартная процедура вывода значения переменной «a» или результата

вычисления арифметического выражения на экран

writeln(a); - вывод с последующим переводом строки

writeln; - стандартная процедура перевода курсора на новую строку

write(x:8:4)- форматированный вывод (8-общее кол-во позиций, 4-дробная часть)

Например:	на экране:	если A:=2;
Write(a);	2	
Writeln(a);	2 и переход на следующую строку	
Write('a=',a);	a=2	
Write('a=', a, ' метров');	a=2 метров	
Write('Y=', 4+6);	Y=10	
Write('S=',a,'*',3,'=',a*3);	S=2*3=6,	если a=2
Write('z=',z:8:4);	z=135.6432,	если z=1,356432223e02
Writeln('погода');	погода	
Writeln('хорошая');	хорошая	
Write('погода ');		
Writeln('хорошая');	погода хорошая	

Для вывода в заданную позицию экрана перед командой Write используется команда GotoXY(C,R); которая устанавливает курсор в позицию C строки R. Размеры текстового экрана в Turbo Pascal 7.0 80x25. 80-количество позиций в строке, 25-количество строк.

Следовательно, C может меняться от 1 до 80, а R – от 1 до 25 (обязательно целые числа!).

Например: GotoXY(38,12); Write('Hello'); - выводит слово 'Hello' в центре экрана.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 2

РАБОТА КОМПЬЮТЕРА В РЕЖИМЕ КАЛЬКУЛЯТОРА (ВЫЧИСЛЕНИЯ)



ЗАДАНИЯ:

1. Набрать программу: {пояснения к программе:}
 Program Primer; uses crt; { -название программы 'Primer'. Объявлен
 модуль crt }

Begin { -начало программы}
 Clrscr; { -очистка экрана или окна вывода}
 write('Y=', 6+3); readkey; { -вывод на экран результата Y=9.}
 End. { -конец программы}

2. Запустить (исполнить) программу { -нажать Ctrl+F9}

3. Вычислить: $17+34$; $24,7+56$; $24:3$; $5-6$; $|-9|$; $12/3/4$; $\sqrt{36}$.

4. Вычислить: $-\frac{75-\sqrt{38}+55}{6^2-\sqrt{21}}$ **Ответ: -3.941621**

$\frac{5+8 \cdot (5^2 \cdot 3-7)+35}{(20-\sqrt{5}) \cdot 8}$ **Ответ: (≈ 4.1)**

$\frac{2,8-\sqrt{55,6}-4,2}{|3,2^2-\sqrt{32,4}| \cdot 7,8}$ **Ответ: -0.2496655**

1. Вычислить: $\frac{\sqrt{\sin 30^\circ} + \sqrt{|32-6 \cdot 8|}}{\sin^2(60^\circ)^2}$ и вывести ответ в центре экрана.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 3 РЕШЕНИЕ МАТЕМАТИЧЕСКИХ ВЫРАЖЕНИЙ

ЛИНЕЙНАЯ ПРОГРАММА

ЛИНЕЙНОЙ называют программу, реализующую конструкцию следования, когда инструкции расположены одна под другой и последовательно выполняются.

- :=** - символ присваивания переменным значений;
x:=3.4 - инструкция присваивания (переменной x присваивается значение 3.4).
readln(a); - стандартная процедура ввода значения переменной a с клавиатуры



ЗАДАНИЯ:

1. Набрать готовую программу вычисления площади прямоугольника $S=a*b$.
Во всех программах не забывать в начале Uses crt; и в конце readkey: или readln;

```

Program Area; Uses crt;      { Л И Н Е Й Н А Я  ПРОГРАММА (см. конспект) }
  var a,b,s: real;          { описание переменных. Тип вещественный }
Begin                       { начало программы S }
  ClrScr;                   { очистка экрана. От англ. CLear SCReen }
  write('a='); readln(a);   { ввод длины и переход на следующую строку }
  write('b='); readln(b);   { ввод ширины и переход на следующую строку }
  s:=a*b;                   { вычисление по формуле }
  write('S=',s);           { вывод результата на экран }
  readkey;                  { удержание изображения до нажатия клавиши }
End.                         { конец программы Area }

```

Запустить программу. Получить результат.

2. Добавить в программу комментарий к вводу переменных и выводу ответа
(Например: Введите сторону a= Площадь прямоугольника S=).
3. Составить программу вычисления площади круга: $S = \pi \cdot R^2$
4. Составить программу вычисления площади треугольника: $S = \frac{1}{2} \cdot A \cdot H$
 - 1) выделить целую часть результата;
 - 2) выделить дробную часть результата;
 - 3) получить целый остаток от деления;
 - 4) округлить результат
 - 5) округлить до целого, не превосходящего результат.
5. Составить программу вычисления выражения: $Y = \sin(a+b/2)$
при $a=15$, $b=30$ (градусов).

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 4 РЕШЕНИЕ ФИЗИЧЕСКИХ ЗАДАЧ

ОСОБЕННОСТИ:

- греческие буквы надо заменять на латинские буквы;
- описать назначение переменных;
- задача предварительно решается на бумаге, и конечная формула вводится в программу. На основании анализа результата делается вывод о правильности решения задачи;
- результат должен иметь указание единиц измерения;
- особого внимания требует правильный перевод единиц измерения физических величин;
- постоянные физические величины описываются в разделе описания констант.
- на Паскале нет различия между строчными и заглавными буквами: $G=g$, $A=a$.

Составить программу вычисления силы Всемирного тяготения: $F = G \cdot \frac{M1 \cdot M2}{R^2}$

```
Program Ff; Uses crt;
const g=6.672E-11;    { гравитационная постоянная }
var  m1,m2,           { массы взаимодействующих тел }
     r,               { расстояние между точечными телами }
     f: real; { сила Всемирного тяготения }
Begin
  ClrScr;
  write('Введите массу m1='); readln(m1);
  write('Введите массу m2='); readln(m2);
  write('Введите расстояние r='); readln(r);
  f:=g*m1*m2/(r*r);
  writeln;           { получение пустой строки }
  write('Сила притяжения F=',f:8:4,' Н');
  readkey;
End.
```



ЗАДАНИЯ:

1. Набрать программу вычисления силы Всемирного тяготения.
Запустить программу
Вычислить при $m1=m2=50$ кг и $r=1$ м. Ответ перевести в мН силы.
2. Составить программу вычисления кинетической энергии $E = \frac{M \cdot V^2}{2}$
3. Подчеркнуть результат звездочками.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 5 ОПЕРАТОР УСЛОВНОГО ПЕРЕХОДА

Неполная форма: **IF условие THEN серия;**

УСЛОВИЕ - это логическое выражение, принимающее значение истина или ложь.

Выполняется инструкция так: **если** условие истинно, **то** выполняется серия ,
иначе выполняется следующая за IF..THEN инструкция.

СЕРИЯ - один оператор языка.

Полная форма : **IF условие THEN серия1
ELSE серия2;**

Если условие истинно, то выполняется серия1, иначе выполняется серия2.

ВНИМАНИЕ! Перед словом ELSE точка с запятой не ставится !!

Найти большее из двух чисел а и b.

```
Program Vid; Uses crt;
  var a,b,max: integer;
Begin
  ClrScr;
  write(' Введите целое число a='); readln(a);
  write(' Введите целое число b='); readln(b);
  if a>=b then max:=a      { если условие =true,то max:=a }
    else max:=b;          { если условие =false,то max:=b }
  writeln('Max=',max);
  readkey;
End.
```

При нажатии цифры 1 печатается слово1, а при нажатии любой клавиши слово2.

```
Program Text; Uses crt;
  var x: integer;
Begin
  ClrScr;
  write('Введите цифру 1 или любую :'); readln(x); ClrScr;
  if x=1 then write(' В а с я')
    else write(' П е т я');
  readkey;
End.
```



ЗАДАНИЯ:

1. Набрать программу Vid и проверить ее работу для '+' и '-' чисел.
2. В программе Vid применить округление для вещественных чисел.
3. Набрать программу Text , получить результат.

4. Изменить программу Text так, чтобы второе слово выводилось только при нажатии цифры 2 , а при нажатии цифры кроме 1 и 2 выводилось сообщение "ERROR".
5. Составить программу вычисления для любого x функции $Y = 1/x$, при $x=0$ выдать сообщение ' функция не существует '.
6. Найти наибольшее из трех чисел.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 6 **ОПЕРАТОР УСЛОВНОГО ПЕРЕХОДА (продолжение)**

При сложных условиях применяются логические связки:

And - истинно, когда истинны все составляющие логическое выражение

Пример: $((x \geq 0) \text{ and } (x \leq 5))$ истинно для x из интервала $[0;5]$ и ложно при $x=6$

Not - истинно, когда ложно исходное логическое выражение

Пример: $\text{not}(x \geq 0)$ истинно для $x=-5$ и ложно для $x=6$

Or - истинно, когда истинно хотя бы одно из логических выражений

Пример: $((a < 5) \text{ or } (a > 9))$ истинно для $a=2$ и ложно для $a=6$

$$\text{Вычислить: } Y = \begin{cases} \frac{1}{x-1} & \text{при } x < 0 \\ \frac{1}{x-5} & \text{при } x \geq 0 \end{cases}$$

Program Primer; Uses crt;

var x: integer;

Begin

ClrScr;

write('Введите значение x='); readln(x);

if $x < 0$ then write(' Y1=', $1/(x-1)$)

else write(' Y1 - не существует по условию');

if $(x \geq 0) \text{ and } (x \leq 5)$ then write(' Y2=', $1/(x-5)$)

else write(' Y2 - не существует по условию');

readkey;

End.



ЗАДАНИЯ:

1. Набрать программу и проверить ее работу для X: -2, 1, 5, 6.
2. Эту же задачу решить для условия:
 - 1) $X < 0$ или $X = 5$ или $X = 8$
 - 2) $X \geq 0$ и $X \leq 5$ и $X \leq 6$

Проверить работу программы для X: -2, 0, 2, 5, 6, 8, 9.

3. Эту же задачу решить для условия: 1) $-2 \leq X \leq 5$ $X \neq 1$,
2) $-1 \leq X \leq 8$ $X \neq 5$

Проверить все контрольные точки: -3, -2, -1, 0, 1, 5, 8, 9.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 7 **ОПЕРАТОР УСЛОВНОГО ПЕРЕХОДА (продолжение)**

Если требуется после THEN или ELSE применить более одного оператора, то применяются **ОПЕРАТОРНЫЕ СКОБКИ BEGIN..END**

Решение КВАДРАТНОГО УРАВНЕНИЯ $a \cdot x^2 + b \cdot x + c = 0$

```
Program Kwur; Uses crt;
var a,b,c,x1,x2,d: real;
Begin
  ClrScr;
  write('Введите коэффициенты a,b,c :'); readln(a,b,c);
  d:=b*b-4*a*c;
  if d<0 then writeln('Нет корней.')
    else begin
      if d=0 then begin
        x1:=-b/(2*a);
        x2:=x1
      end
      else begin
        x1:=(-b+sqrt(d))/(2*a);
        x2:=(-b-sqrt(d))/(2*a)
      end;
      write(' x1=',x1:5:2,' x2=',x2:5:2)
    end;
  readkey;
End.
```



ЗАДАНИЯ:

1. Решить следующие уравнения:

а) $3x^2 + 6x - 8 = 0$

б) $x^2 - 2x - 1 = 0$

в) $5x^2 - 4x + 3 = 0$

г) $4x^2 + 4x - 1 = 0$

д) $5x^2 - 5x + 2 = 0$

2. Вставить в программу комментарии:
 - а) полное название программы;
 - б) автор программы и дату составления программы.
3. Сделать так, чтобы результат выводился на чистом экране с указанием значений коэффициентов.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 8

ОПЕРАТОР ВАРИАНТА

Оператор варианта является обобщением условного оператора: он дает возможность выполнить один из нескольких операторов в зависимости от значения некоторого выражения, называемого **СЕЛЕКТОРОМ**.

```

CASE <селектор> OF
    <список меток1> : <оператор1>;    { case (кейс) - выбор }
    <список меток2> : <оператор2>;    { of (оф) - из }
    ..... : .....;
    <список метокN> : <операторN>;
ELSE <оператор>
END;
```

селектор - выражение любого перечисляемого типа, кроме вещественного; integer, byte (-целые из [0;255]), char -символьный тип.

оператор - любой оператор языка, в том числе и составной;

список меток - список разделенных запятыми значений выражения 'селектор' или одно его значение; тип метки и селектора одинаков;

Оператор варианта выбирает для исполнения тот ОПЕРАТОР, одна из меток которого равна текущему значению выражения СЕЛЕКТОР. Если ни одна из меток не равна текущему значению селектора, то никакие операторы не выполняются, либо выполняются операторы, следующие за зарезервированным словом ELSE (если такое имеется).

```
Program Primer_Case1; Uses crt;
```

```
    var n:integer;
```

```
Begin
```

```
    clrscr; write('Нажмите цифру 1..8'); readln(n);
```

```
    case n of
```

```
        1,7: write('В а н я');
```

```
        2: write('В а с я');
```

```
        3: write('В а л я');
```

```
        4: write('В а р я');
```

```
        5: write('В а д я');
```

```
        6: write('в а з а')
```

```
    else write(' Нет такой метки')
```

```
    end; readkey;
```

```
End.
```

```
Program Primer_Case2; Uses crt;
```

```

var n:char;      { char - символьный тип ,занимает 1 байт памяти }
Begin
  clrscr; write('Нажмите букву а..е'); readln(n);
  case n of
    'a': write('Вы ввели символ а');
    'b': write('Вы ввели символ b');
    'c': write('Вы ввели символ c');
    'd': write('Вы ввели символ d')
    else write('Вы ввели символ, которого нет среди меток')
  end; readkey;
End.

```



ЗАДАНИЯ:

1. Набрать приведенные выше программы и выполнить их.
2. Во второй программе после <clrscr> ввести МЕНЮ для выбора меток.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 9 **ОПЕРАТОР ВАРИАНТА (продолжение)**

Вычислить площади фигур, выбирая их из списка (меню) :

```

Program PLFigur; Uses crt;
  var s,a,b,h,r:real;
      n:integer;
Begin
  ClrScr;
  writeln('      П Л О Щ А Д И');
  writeln('      =====');
  writeln(' 1. Прямоугольника. ');
  writeln(' 2. Трапеции. ');
  writeln(' 3. Круга. ');
  writeln(' 4. Конец работы. ');
  writeln;
  write('Введите номер строки :'); readln(n);
  case n of
    1: begin
      clrscr;
      write('Введите стороны прямоугольника a,b :');
      readln(a,b);
      s:=a*b;
      write('Площадь прямоугольника S=',s)
    end;
    2: begin

```

```

clrscr;
write('Введите основания и высоту трапеции a,b,h :');
readln(a,b,h);
s:=(a+b)*h/2;
write('Площадь трапеции S=',s)
end;
3: begin
  clrscr;
  write('Введите радиус круга r=');
  readln(r);
  s:=pi*r*r;
  write('Площадь круга S=',s)
end
else write(' ***** Работа окончена *****');
end;
readkey;
End.

```



ЗАДАНИЯ:

1. Набрать программу и выполнить ее для четырех строк меню.
2. Самостоятельно дописать в программу вычисление площади треугольника.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 10 **ПОДГОТОВКА К КОНТРОЛЬНОЙ РАБОТЕ**

1. Составить программу вычисления силы взаимодействия двух точечных электрических зарядов: $F = k \cdot \frac{|q1| \cdot |q2|}{\epsilon \cdot r^2}$, где:

F - сила взаимодействия зарядов по закону Кулона (Ньютон),
 $q1, q2$ - неподвижные точечные электрические заряды (Кулон),

$k=9 \cdot e9$ - коэффициент ($\frac{H \cdot m^2}{Kл^2}$),

ϵ - относительная диэлектрическая проницаемость среды (табличное значение),
 r - расстояние между точечными электрическими зарядами (метр).

2. Вычислить значение функции:

$$Y = \begin{cases} \frac{1}{x-3}, & \text{при } x \neq 3 \text{ и } -2 \leq x < 8 \\ \frac{2}{x+5}, & \text{при } x \neq -5 \text{ и } x = -10 \text{ и } x = 0 \text{ и } x = 10 \end{cases}$$

3. Повторить к контрольной работе:

- а) Принципы фон Неймана;
- б) экспоненциальная форма представления числа;
- в) графическое представление разветвляющихся алгоритмов;
- г) перевод алгебраических выражений для ввода в программу;
- д) решение математических и физических задач;
- е) разветвляющиеся вычислительные процессы.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №11 ЦИКЛИЧЕСКИЕ ПРОЦЕССЫ

Циклические конструкции используются в тех случаях, когда необходимо повторять какие-либо действия. Различают два вида циклов: с заданным количеством шагов и циклы, в которых количество шагов заранее неизвестно, а цикл завершается при выполнении условия.

1. ЦИКЛ С ПРЕДУСЛОВИЕМ или цикл "ПОКА".

Сначала проверяется условие и тело цикла выполняется ПОКА УСЛОВИЕ ИСТИННО,

если условие становится ЛОЖНЫМ, то происходит выход из цикла.

Реализуется этот цикл инструкцией:

WHILE <условие> DO <тело цикла>

while - (вайл) - пока, do - (ду) – делать (выполнять).

В тело цикла входит только один оператор! Если нужно объединить в тело цикла несколько операторов, то применяются операторные скобки **begin-end**.

Условие может быть простым или сложным, если применить слова: NOT, AND, OR.

ПРИМЕРЫ ПРОСТЕЙШИХ ЦИКЛИЧЕСКИХ ПРОЦЕССОВ.

Пример 1. Вывести 20 одинаковых символов на экран.

```

Program Simwol; Uses crt;
  var i,h,n: integer;
Begin
  {Действительно ли на экране 20 символов?}
  clrscr;
  {пронумеруем символы:}
  i:=1; h:=1; n:=20;
  while i<=n do
  begin
    write('* ');
    i:=i+h;
  end;
  readkey;
End.
```

Пример 2. Вывести на экран числа от 1 до 20.

Пример 3. Вывести на экран числа от 20 до 1.

```

Program Ch; Uses crt;
  var i,h,n: integer;
```

```

Program Ch; Uses crt;
  var i,h,n: integer;
```

```

Begin
  Clrscr;
  i:=1; h:=1; n:=20;
  while i<=n do
    begin
      write(i, ' ');
      i:=i+h;
    end;
  readkey;
End.

```

```

Begin
  clrscr;
  i:=20; h:=-1; n:=1;
  while i>=n do
    begin
      write(i, ' ');
      i:=i+h;
    end;
  readkey;
End.

```

ПРИЛОЖЕНИЕ К ПРАКТИЧЕСКОМУ ЗАНЯТИЮ № 11

Пример 4. Вывести на экран числа, оканчивающиеся на 7 на интервале от -100 до 100.

```

Program Ch;      Uses crt;
  var i,h,n: integer;
Begin
  clrscr;
  i:=-100; h:=1; n:=100;
  while i<=n do
    begin
      if i mod 10=7 then write(i, ' ');
      i:=i+h;
    end;
  readkey;
End.

```

нечетные:	if odd(i) then write(i, ' ');
четные:	if not(odd(i)) then write(i, ' ');
делящиеся на 7:	if i mod 7=0 then write(i, ' ');
нечетные и делящиеся на 7:	if (odd(i)) and (i mod 7=0) then write(i, ' ');

Пример5. Найти сумму ряда: 1, 2, ..20.

Тип переменной S может быть integer, если сумма чисел не более 32767.

```

Program Summa; Uses crt;
  var i,n,h,s: integer;
Begin
  clrscr;
  i:=1; n:=20; h:=1; s:=0;      { s:=0 -обнуление суммы }
  while i<=n do
    begin
      s:=s+i;                    { процесс суммирования }
      i:=i+h;
    end;
  write('Сумма чисел натурального ряда от 1 до 20 равна ',s);

```

readkey;

End.

Пример 6. Найти сумму ряда: 0.1, 0.3, 0.5, .. , 32.5

```

Program Summa; Uses crt;           { s - сумма }
var x,h,n,s: real;                 { h - шаг }
Begin                               { x – начальное значение параметра цикла}
  ClrScr;                           { n - конечное значение параметра цикла.}
  s:=0; x:=0.1; h:=0.2; n:=32.5;
  while (x<=n) do
  begin
    s:=s+x;   { -суммирование очередного элемента }
    x:=x+h;   { -изменение параметра цикла на шаг }
  end;
  writeln('Сумма ряда: 0.1, 0.3,..., 32.5 S=',s); readkey; End.

```

ПРИЛОЖЕНИЕ К ПРАКТИЧЕСКОМУ ЗАНЯТИЮ № 11

продолжение

Пример 7. Вывести на экран таблицу функции $Y=\sin(x)$,

где x изменяется от 10 до 90 с шагом 5 градусов.

```

Program Tabl; Uses crt;
var x, h, n, y: real;
Begin
  ClrScr; writeln(' Таблица функции y=sin(x)');
  x:=10; h:=5; n:=90;
  while x<=n do
  begin
    y:=sin(x*pi/180);
    writeln('X=',x,'      Y=',y);
    x:=x+h;
  end;
  readkey;
End.

```



ЗАДАНИЯ:

1. Найти сумму ряда: **5.6, 5.9, .. ,9.7 2**
2. Получить таблицу функции $y=\cos(5 \cdot x)$, где x изменяется от 1 до 10 с шагом 0.5
3. Найти произведение чисел от **6.7** до **7.8** с шагом **0.4**
4. Сравнить, что больше сумма ряда **8.5, 8.6, ..., 40.9** или произведение чисел от **1** до **3** с шагом **0.25**

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 12 **ЦИКЛИЧЕСКИЕ ПРОЦЕССЫ (продолжение)**

2. ЦИКЛ С ПОСТУСЛОВИЕМ или ЦИКЛ "ДО"

Сначала выполняется тело цикла, затем проверяется условие. Выход из цикла осуществляется, когда УСЛОВИЕ СТАНЕТ ИСТИННЫМ , если условие ЛОЖНО, то

цикл повторяется.

REPEAT

тело цикла

UNTIL условие;

repeat - (рипид) - повторять

until - (антил) - ДО тех пор пока

Проверка условия производится после выполнения тела цикла, что делает применение цикла "ПОКА" более предпочтительным, когда требуется пропустить цикл вообще.

Вычислить $Y = \frac{1}{X}$ для любого X, с проверкой корректности ввода.

```
Program Func; Uses crt;
  var x,y: real;
Begin
  repeat
    ClrScr;
    write('X='); readln(x);
  until (x<>0);
  y:=1/x;
  writeln('x=',x,' y=',y);
  readkey;
End.
```



ЗАДАНИЯ:

1. Вычислить сумму ряда: **0.6, 0.7, 0.8, 0.9**.
2. Вывести таблицу функции $y=\cos(x)$, где x изменяется в градусах от 10 до 45 с шагом 5.
3. Определить, что больше сумма ряда: **1.1 , 1.3 , ... , 93.1** или произведение: **5*7*...*11**.
4. Вычислить значение функции $y=1/(x-5)$, на отрезке от 0 до 10 с шагом 1 и с проверкой отсутствия деления на 0.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 13 ЦИКЛИЧЕСКИЕ ПРОЦЕССЫ (продолжение)

3. ЦИКЛ С ЗАДАНЫМ ПАРАМЕТРОМ (ДЛЯ)

Применяется, если известно сколько раз требуется выполнить тело цикла.

FOR <параметр цикла> := <выражение1> **TO** <выражение2> **DO** <оператор>
[**DOWNTO**]

for - (фо) - для; to - (ту) - до; { шаг=1 };

downto - (даунту) - вниз до; { шаг=-1 }; do - (ду) - выполнить;

параметр цикла - переменная любого скалярного типа, кроме вещественного
(управляющая переменная);

выражение1 - скалярное выражение - начальное значение параметра цикла;

выражение2 - скалярное выражение - конечное значение параметра цикла;

оператор - тело цикла;

Напечатать числа от 1 до 20.

Program Chisla; Uses crt;

var i:integer; { управляющая (параметр) цикла }

Begin

clrscr;

for i:=1 to 20 do write(i, ' '); { 1 -начальное значение параметра }

End. { 20 -конечное значение параметра }

{ write(i, ' ') -тело цикла }

После каждого выполнения тела цикла параметр **i** увеличивается на **1**.

Напечатать числа от 20 до 1.

Program Chisla; Uses crt;

var i:integer;

Begin

clrscr;

for i:=20 downto 1 do write(i, ' ');

End.

После каждого выполнения тела цикла параметр **i** увеличивается на **-1**.

Найти произведение чисел делящихся на 2 от 10 до 20.

Program Mult; Uses crt;

var x:integer; P:real;

Begin

clrscr;

p:=1; { начальное значение произведения }

for x:=10 to 20 do

if x mod 2=0 then p:=p*x;

write('Произведение чисел, делящихся на 2 от 10 до 40 равно P=',p)

End.



ЗАДАНИЯ:

1. Вычислить сумму чисел из промежутка $[1;1000]$.
2. Вывести на экран 56 символов & .
3. Получить на экране значения функции $Y=\text{SIN } X$, где x берется из промежутка $[25;45]$ и измеряется в градусах.
4. Вывести на экран все числа, делящиеся на 13, от 1 до 100.
5. Вывести на экран все числа от 500 до 300.
6. Найти значение дроби: в числителе - сумма чисел от 1 до 100 делящихся на 7, а в знаменателе - сумма чисел от 100 до 500, делящихся на 17.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 14 **СУММА. ПРОИЗВЕДЕНИЕ. ТАБУЛИРОВАНИЕ**

Рассмотрим суммирование числового ряда, если элемент ряда является алгебраическим выражением с помощью циклов ДЛЯ, ПОКА, ДО:
Найти сумму ряда для i от 1 до 15, каждый элемент которого выражается по формуле $i^2 + 5$, то есть найти сумму: $6+9+14+\dots$

```
Program Summa; Uses crt;
  var s,i,n: integer;
Begin
  clrscr;
  s:=0; n:=15;
  FOR i:=1 TO n DO s:=s+(i*i+5);
  write('Сумма числового ряда S=',s);
End.
```

```
Program Summa; Uses crt;
  var s,i,n,h: integer;
Begin
  clrscr;
  s:=0; i:=1; n:=15; h:=1;
  REPEAT
    s:=s+(i*i+5);
    i:=i+h;
  UNTIL i>n;
  write('Сумма числового ряда S=',s);
End.
```

```
Program Summa; Uses crt;
  var s,i,n,h: integer;
Begin
  clrscr;
  s:=0; i:=1; n:=15; h:=1;
  WHILE i<=n DO
    begin
      s:=s+(i*i+5);
      i:=i+h;
    end;
  write('Сумма ряда S=',s);
End.
```

Найти произведение чисел ряда, каждый элемент которого определяется по формуле $(i+1)/i$ и i изменяется от 1 до 20 с шагом 0.2, т.е. найти произведение чисел: $P=2*1.8333*1.714*1.625*\dots$

```
Program Mult; Uses crt;
```

```

var i,p,n,h: real;
Begin
  clrscr;
  p:=1; i:=1; n:=20; h:=0.2;
  while i<=n do begin p:=p*(i+1)/I; i:=i+h; end;
  write('Произведение ряда чисел P=',p);
End.

```



ЗАДАНИЯ:

1. Найти сумму чисел ряда, каждый элемент которого определяется по формуле $(i+5)/i$ и i изменяется от 1 до 20 с шагом 0.4.
2. Получить таблицу для построения графика функции $Y = \frac{x^2 + 5 \cdot x + 4}{x - 6}$ на отрезке $[7;15]$ с шагом 0.4.
3. Найти сумму ряда чисел, каждый элемент которого определяется по формуле $(i+5)/(i*i)$ и $i=1,2,3,4,\dots$. Суммировать до элемента равного $6E-3$.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 15 ТАБУЛИРОВАНИЕ (продолжение)

Для получения качественного вывода таблицы применяется указание **ШИРИНЫ ПОЛЯ ВЫВОДА**, которая пишется после переменной через двоеточие. Например, **a:5** или **b:6**. Для числа $\langle a \rangle$ выделяется поле из пяти символов. Но следует иметь в виду, что целые и вещественные числа при таком выводе в таблице будут расположены нестандартно.

Для получения более качественного вывода таблицы применяют **GOTOXY(C,R)**.

C - номер символа в строке, изменяется от 1 до 79, { Column }

R - номер строки, изменяется от 1 до 24. { Row }

При выводе таблицы значение **C** не изменяется, а значение **R** должно, после каждой выведенной строки, увеличиваться на 1.

Рассмотрим программу решения задания #2 практического занятия #14.

```

Program Tab; Uses crt;
  var x,h,n: real; r:byte;
Begin
  clrscr;
  x:=7; h:=0.4; n:=15; r:=1;   { r - номер строки }
  repeat
    gotoxy(1,r); write('X=',x:2,' Y=',(x*x+5*x+4)/(x-6));
    x:=x+h; r:=r+1;
  until x>n;
End.

```

```
until x>n;  
End.
```

ПРИМЕЧАНИЯ:

1. Выравнивание разрядов производится if-then-else
2. Если строк в таблице больше, чем строк на экране, то применяется постранный вывод с очисткой экрана (см. WHEREY занятие № 17).

Встречаются таблицы, в которых при некоторых значениях аргументов значения функции не определяются. В этом случае необходимо исключить прерывание программы и выдать на экран сообщение об отсутствии функции.

Вывести таблицу значений функции $Y = \frac{1}{X-5}$ на интервале 1,2..15.

```
Program Tab; Uses crt;  
  var x, n: integer;  
Begin  
  clrscr; n:=15;  
  for x:=1 to n do if x=5 then writeln('X=',x:2,' Y-не определяется')  
                  else writeln('X=',x:2,' Y=',1/(x-5) );  
End.
```



ЗАДАНИЯ:

1. Получить таблицу для построения графика функции $y=\sin x$, используя указание ширины поля вывода.
2. Получить эту же таблицу с использованием **gotoxy(c,r)**;
3. Вывести таблицу в центральной части экрана.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 16 ТАБЛИЦА ПИФАГОРА

Если телом цикла является циклическая конструкция, то такие циклы называют **ВЛОЖЕННЫМИ**.

```
Program Pifagor; Uses crt;  
  var i,j: integer;  
Begin  
  clrscr;  
  gotoxy(20,1);writeln('Таблица Пифагора');  
  
  FOR I:=1 TO 9 DO  
    BEGIN  
      FOR J:=1 TO 9 DO  
        BEGIN
```

```

gotoxy(i*4,j*2);
write(i*j:4);
END;
WRITELN;
END;

```

```

for i:=1 to 41 do
begin
gotoxy(i,3);write('-');
if i<=19 then begin
gotoxy(9,i);
write('!');
end;

end;
gotoxy(5,22);
readkey;
End.

```



ЗАДАНИЯ:

1. Набрать программу и получить на экране таблицу.
2. Получить максимально возможную квадратную таблицу.
3. Получить максимально возможную прямоугольную таблицу.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 17 ПРОЦЕДУРЫ

Программу, состоящую только из операторов, называют **МОНОЛИТНОЙ**.
Программу, содержащую процедуры и функции называют **МОДУЛЬНОЙ**.
ПРОЦЕДУРА - часть программы, имеющая имя и предназначенная для решения определенной задачи. Каждая процедура определяется только однажды, но может использоваться многократно. Процедуры определяются в разделе описаний вслед за разделом переменных.

PROCEDURE Name; procedure - служебное слово;
BEGIN Name - идентификатор отражающий смысл данной задачи;
ИНСТРУКЦИЯ Описание процедуры само по себе никакого действия не вызывает. Чтобы исполнить процедуру, необходимо в
END; нужном месте программы поместить обращение к ней, указав Name; .

Например: Напечатать сорок любых символов.

В программе используется функция **WhereX** – она определяет позицию курсора в строке (принимает значения от 1 до 80, изменяется автоматически при выводе

информации на экран). С помощью этой функции можно проверить, где находится курсор. Аналогичная функция **WhereY** – определяет строку, в которой в данный момент находится курсор (принимает значения от 1 до 25). Условие (wherex<40) означает, что при выводе на экран еще не достигнута 40 позиция в текущей строке.

```
Program Stroka; Uses crt;           { Stroka - имя программы }

procedure Str_Simw;                 { Str_Simw - имя процедуры }
begin
  while wherex<40 do write('$');    { - тело процедуры }
  writeln
end;

Begin
  clrscr;
  Str_Simw; Str_Simw;               { - вызов процедуры (два раза) }
  Readkey;
End.
```



ЗАДАНИЯ:

1. Набрать и исполнить пример.
 2. Взять любую программу суммирования числового ряда и часть программы описать с помощью процедуры.
 3. Применить процедуру Str_Simw для подчеркивания.
- Процедуры позволяют избежать повторения фрагментов программ и таким образом сократить размеры программы.
- Процедуры позволяют разбить программу на законченные блоки.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 18 **ПРОЦЕДУРЫ (продолжение)**

Переменные, описанные VAR после слова PROGRAM , называются **ГЛОБАЛЬНЫМИ**.

Глобальные переменные передаются в любую часть программы, в том числе и в процедуры.

```
Program Stroka; Uses crt;
var i: integer;           { i - глобальная переменная }
procedure St_Sim;
begin
  for i:=1 to 40 do write('$');
  writeln;
```

```

end;
Begin
  clrscr;
  St_Sim; St_Sim;
End.

```

Переменные, описанные в процедуре, называют **ЛОКАЛЬНЫМИ**, так как они используются только в данной процедуре. Доступа к ним из программы нет.

```

Program Stroka; Uses crt;
  procedure St_Sim;
    var i: integer;      {i - локальная переменная}
  begin
    for i:=1 to 40 do write('-');
  end;
Begin
  clrscr;
  St_Sim;
  Readkey;
End.

```

Нежелательно, чтобы глобальные и локальные переменные совпадали.



ЗАДАНИЯ:

1. Набрать и исполнить данные программы.
2. Решить задачу: Найти сумму чисел ряда 1,2,3,...,60 с использованием процедуры нахождения суммы и процедуры подчеркивания результата. В обеих процедурах использовать локальные переменные. Количество символов в строке задать глобальной переменной.
3. С помощью процедур получить вертикальную и горизонтальную линии. Линия проводится многократной печатью символа. Заключить слово в рамку из полученных линий.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 19 ПРОЦЕДУРЫ (продолжение)

Часто процедура содержит список параметров.

procedure имя (СПИСОК ПАРАМЕТРОВ);

...

Список параметров - перечень имен для обозначения исходных данных и результатов работы процедуры с указанием их типов. Параметры, перечисляемые в списке, называют **ФОРМАЛЬНЫМИ**.

При обращении к процедуре, имеющей список параметров, в скобках указываются конкретные значения (выражения) и имена, подставляемые на место формальных параметров процедуры при ее выполнении. При вызове процедуры

формальные параметры, указанные в ее заголовке, заменяются значениями в порядке их следования. Количество и тип фактических и формальных параметров должно совпадать.

Вывести на экран строку содержащую задаваемое с клавиатуры число символов.

```

Program Stroka; Uses crt;
var m: integer;           { m – глобальная переменная }
procedure Str_Sim( n: integer; c: char); { n, c - параметры-значения }
var i:integer;           { i - локальная переменная }
begin
  for i:=1 to n do write(c);           { конструкция печати символов }
  writeln;
end;
Begin
  clrscr;
  Str_Sim(64,'*');           { вызов процедуры с указанием числа звездочек }
}
  writeln; writeln;
  write('Сколько символов напечатать? '); readln(m);
  Str_Sim(m,'=');           { вызов процедуры с указанием числа знаков '=' }
End.

```

Параметры-значения нельзя использовать для передачи результатов из процедуры в основную программу ! Если возникла такая необходимость, то применяют формальные параметры-переменные , перед которыми в списке формальных параметров стоит **var**.

```

Program Pr1; Uses crt;
var x: integer;
procedure Izm(y: integer);
begin
  y:=1
end;
Begin
  x:=0;
  Izm(x);
  write(x)
End.

```

Результат: x=0.

```

Program Pr2; Uses crt;
var x: integer;
procedure Izm(var y: integer);
begin
  y:=1
end;
Begin
  x:=0;
  Izm(x);
  write(x)
End.

```

Результат: x=1.



ЗАДАНИЯ:

1. Набрать и исполнить приведенные примеры.

2. Найти сумму чисел ряда 1,2,3,4,... от числа n до числа m. Суммирование оформить процедурой с параметрами. Значения n и m программа должна запрашивать.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 20

ФУНКЦИИ

ФУНКЦИЯ - это подпрограмма, результат выполнения которой есть единственное скалярное значение, присваиваемое имени этой функции. Функция - частный случай процедур.

ОТЛИЧИЯ от процедур:

1. В описании указывается тип результата функции.
2. Результат выполнения функции - ОДНО значение.
3. Имени функции обязательно присваивается результат.
4. Результат передается в программу как значение имени этой функции.
5. Функция записывается в выражениях, а не отдельной командой как процедура.

Описание функции аналогично описанию процедуры:

```
FUNCTION имя ( список формальных параметров ): тип;
  [var локальные переменные: тип;]
BEGIN
  блок функции
END;
```

Допускается описание функции без параметров:

```
FUNCTION имя : тип;
```

...

Найти сумму чисел ряда 1,2,3,... ,используя функцию.

```
Program Summa; uses crt;
```

```
var n:integer;           { n -глобальная переменная }
function Sum(m: integer): integer; { m -параметр-значение функции}
  var i,s: integer;      { i,s - локальные переменные}
begin
  s:=0;
  for i:=1 to m do s:=s+i;
  Sum:=s                 {- присваивание имени функции значения}
end;
Begin
  clrscr;
  write('До какого числа суммировать? '); readln(n);
  write('Сумма чисел ряда S=',Sum(n)) { -вызов функции Sum с парам.}
  readkey;
End.
```



ЗАДАНИЯ:

1. Набрать и исполнить программу.
2. Изменить программу так, чтобы была найдена сумма чисел от m до числа n , вводимых с клавиатуры.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 21 **ФУНКЦИИ (продолжение)**

Составить программу вычисления степени числа X с целым показателем n , используя функцию без параметров.

```

Program Step; uses crt;
var n:integer;
    x,y:real;
function St:real;
var i: integer;
begin
y:=1;
for i:=1 to n do y:=y*x;
St:=y
end;
Begin
clrscr;
write('Введите число :'); readln(x);
write('Введите степень :'); readln(n);
if n<0 then begin n:=-n; x:=1/x end;
write('Число ',x,' в степени ',n,' равно ',St)
readkey;
End.

```



ЗАДАНИЯ:

1. Набрать и исполнить программу.
2. Изменить программу, применив функцию с параметрами.
3. Вычислить $Y = \frac{X^6 - X^4}{X^{-5} + X^{-3}}$, при $X=3$, используя функцию с параметрами.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 22 **ПОВТОРЕНИЕ. ПОДГОТОВКА К КОНТРОЛЬНОЙ РАБОТЕ.**

1. Вычислить $Y=A+B+C$, применив:
 - а) - процедуру ввода данных;
 - процедуру вычисления Y ;
 - процедуру вывода результата;
 - глобальные переменные;

- б) - заменить глобальные переменные на формальные;
- в) - заменить процедуру вычисления суммы на функцию.

ВТОРОЙ ВАРИАНТ КОНТРОЛЬНОЙ РАБОТЫ

Получить таблицу значений аргумента и функции $Y = \frac{\sqrt{x+8} + 2 \cdot (\sqrt{x+8} - 2)}{4 \cdot (5 - \sqrt{x+8}) - \sqrt{x+8}}$ на интервале от -5 до 15 с шагом 2
Повторяющуюся часть описать в функции, а получение таблицы описать в процедуре.

```
Program Primer; Uses crt;
  Function R(x:integer):real;      { x – формальная переменная }
  Begin
    R:=sqrt(x+8);
  End;
  Procedure Tab;
    Var x, n, h: integer;          { x,n,h – локальные переменные }
  Begin
    X:=-5; n:=15; h:=2;
    While x<=n do
      Begin
        Writeln('x=',x,'      Y=',(r(x)+2*(r(x)-2))/(4*(5-r(x))-r(x)) )
        X:=x+h;
      End;
    End;
  End;
Begin
  Clrscr;
  Tab;
  Readkey;
End.
```

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 23 ПОВТОРЕНИЕ

Вычислить сумму четных или нечетных чисел натурального ряда до 100, используя выбор из меню:

1. Сумма четных чисел ряда 1,2,3,..., 100.
2. Сумма нечетных чисел ряда 1,2,3,...,100.
3. Конец работы.

Выбор осуществить с помощью if-then-else.

В программе обязательно использовать процедуры или функции.

Применить CASE

Если успеете, то подчеркните результат, применив процедуру подчеркивания, и покажите результат в центральной части экрана.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 24 СИМВОЛЬНЫЕ И СТРОКОВЫЕ ФУНКЦИИ

Различают величины:

1. **Константы** - последовательность символов в апострофах.
2. **Символьные переменные** типа **CHAR** (кэр, чар) - переменные принимают значения символов: ...var a:char;...a:='м';...
3. **Строковые переменные** типа **STRING** - цепочки символов:..var a:string;
Максимальное число символов в цепочке равно 255.
1. **LENGTH(S)** -(лэнкс) - определяет и присваивает ДЛИНУ строковой величины S
(количество символов, включая пробел). Результат – целое число.
2. **CONCAT(S1,S2)** -(конкат) - **ОБЪЕДИНЕНИЕ** (склеивание) символьных и строковых
величин S1 и S2. Конкатенация производится также с помощью знака + ,например S:='Ва'+'ся' Результат - строка 'Вася'.
3. **POS(ST,S)** -(позишэн) - определяет и присваивает **НОМЕР ПОЗИЦИИ**,
с которой подстрока ST входит в строку S.
Результат - число, например pos('о','колесо') =2.
4. **COPY(S,P,D)** -(копи) - **ВЫБИРАЕТ** часть строковой величины S,
начиная с позиции P, длиной D символов.
Результат - выбранная строка или один символ.

ПРИМЕРЫ:

```
Program Primer1; Uses crt;
  var b: string;
      y: integer;
Begin
  clrscr;
  b:='информатика';
  y:=length(b);
  writeln('Длина строки =',y);
End.
```

```
Program Primer3; Uses crt;
  var a: string;
      y: integer;
Begin
  clrscr;
  a:='информатика';
  writeln(a);
  y:=pos('п',a);
  writeln('п на ',y,' месте');
End.
```

```
Program Primer2; Uses crt;
  var a,b,c: string;
Begin
  clrscr;
  a:=' погода';
  b:=' хорошая';
  c:=a+b;
  writeln(c)
End.
```

```
Program Primer4; Uses crt;
  var a,b: string;
Begin
  clrscr;
  a:='информатика';
  b:=copy(a,3,5);
  writeln(a);
  writeln(b)    { форма }
  End.
```



ЗАДАНИЯ:

1. Набрать и запустить данные программы.
2. Определить число символов в предложении: 22 сентября - день осеннего равноденствия. Вырезать слово 'день' и присвоить его переменной b. Вывести предложение и это слово на экран.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 25 СИМВОЛЬНЫЕ И СТРОКОВЫЕ ФУНКЦИИ (продолжение)

CHR(X) -(си эйч а) - ПРИСВАИВАЕТ СИМВОЛ соответствующий коду X.
CHR(65) соответствует латинской заглавной букве А.

ORD(S) -(ода) -ПРИСВАИВАЕТ НОМЕР символа из упорядоченной последовательности символов (см. ASCII), например: ORD('B')=66.

ПРОЦЕДУРЫ НАД СТРОКОВЫМИ ВЕЛИЧИНАМИ

1. **DELETE(S,POS,NUM)** -(дилит) - УДАЛЯЕТ из строки S участок, содержащий NUM символов с позиции POS.
2. **INSERT(OBJ,S,POS)** -(инсет) - ВСТАВИТЬ строку OBJ (переменную или константу) в строку S(переменная) с позиции POS (целое).

ПРОЦЕДУРЫ ПРЕОБРАЗОВАНИЯ ТИПОВ:

3. **STR(VALUE,ST)** -(стр) - ПРЕОБРАЗУЕТ числовое значение в СТРОКУ символов и запоминает результат в строке ST.
4. **VAL(ST,VAR,CODE)** -(вэл) - ПРЕОБРАЗУЕТ строковую величину ST в ЧИСЛОВУЮ величину (в зависимости от VAR). CODE - целая величина. Code=0 ,если нет ошибки, иначе Code равно Pos ошибочного символа, начиная с апострофа.

Program Primer5; Uses crt;

```
var a: string;
Begin
  clrscr;
  a:='Информатика';
  writeln(a);
  delete(a,3,7);
  writeln(a);
End.
```

Program Primer7; Uses crt;

```
var a,b: string;
    c: integer;
Begin
  clrscr;
  c:=100;
  a:='Цена руб.';
```

Program Primer6; Uses crt;

```
var a: string;
Begin
  clrscr;
  a:='Инка';
  writeln(a);
  insert('формати',a,3);
  writeln(a);
End.
```

Program Primer8; Uses crt;

```
var a: string;
    r,x,y: integer;
Begin
  clrscr;
  a:='100 лет';
  val(a,x,r);
```

```
str(c:4,b);
insert(b,a,5);
writeln(a);
End.
```

```
y:=5*x;
write('Y=',y);
readkey;
End.
```



ЗАДАНИЯ:

1. Исполнить пример 5, затем используя delete, получить слова: "фомка" или "нотка".
2. Набрать и исполнить пример 6, затем используя delete, получить из слова "информатика" слово "математика".
3. Исполнить пример 7, затем получить фразу "Цена 10000 руб.". Для нахождения места вставки числа применить pos.
4. Исполнить пример 8, затем используя str и конкатенацию, получить фразу "2000 лет".

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 26 **ОБРАБОТКА СИМВОЛЬНЫХ И СТРОКОВЫХ ВЕЛИЧИН 1**

Изменить цену товара, применив: copy, delete, val, str, insert.

```
Program Primer9; uses crt;
```

```
var a,b: string;
    x,y,p,r: integer;
```

```
Begin
```

```
clrscr;
```

```
a:='Цена товара 100 руб.'; {начальная фраза}
```

```
p:=pos('1',a); {позиция цифры 1}
```

```
b:=copy(a,p,3); {вырезка числа 100}
```

```
delete(a,p,3); {удаление числа 100 из фразы}
```

```
val(b,x,r); {преобразование '100' в число 100}
```

```
y:=2*x; {изменение числа 100}
```

```
str(y:4,b); {преобразование 200 в константу '200'}
```

```
insert(b,a,p); {вставка '200'}
```

```
writeln(a); {печать полученной фразы}
```

```
readkey;
```

```
End.
```



ЗАДАНИЯ:

1. Набрать и исполнить программу.
2. Уменьшить начальную цену до 1 и получить 2000.
3. Из фразы "Сашеньке вчера исполняется 120 лет" получить фразу "Саше завтра исполняется 12 лет", используя процедуры и функции.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 27 ОБРАБОТКА СИМВОЛЬНЫХ И СТРОКОВЫХ ВЕЛИЧИН 2

ЗАГАДКА

Задать пользователю загадку и похвалить его за правильный ответ.

```
Program Primer10; uses crt;
  var otg,name,a: string;
Begin
  clrscr;
  write('Здравствуйте. Как Вас зовут ? '); readln(name);
  clrscr; writeln;
  writeln(name,' , отгадайте загадку: ');
  a:='Висит груша,нельзя скушать.';
  repeat
    write(a,' Отгадка: '); readln(otg); clrscr; {только READLN}
  until (otg='лампа') or (otg='лампочка');
  writeln;
  writeln;
  writeln('Молодец, ',name,' ! До свидания !');
  readkey;
End.
```



ЗАДАНИЯ:

1. Набрать и исполнить программу.
2. Ввести свою загадку и ответ. Предусмотреть различные способы ввода отгадки.
3. "Спрятать" в программе отгадку и вызывать ее с помощью строковых процедур и функций.
4. Установить пароль для защиты текста от неразрешённого доступа.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 28 ОБРАБОТКА СИМВОЛЬНЫХ И СТРОКОВЫХ ВЕЛИЧИН 3

Получить перевертыш: ОГОРОД - ДОРОГО.

```
Program Primer11; uses crt;
  var  a: string;
       i,l: integer;
Begin
  clrscr;
  write('Введите слово :'); readln(a);
  clrscr;                               { for i:=length(a) downto 1 do write(a[i]); }
  writeln(a);
  l:=length(a);
  for i:=l downto 1 do writeln(copy(a,i,1));
  readkey;
```

End.

Очень давно слово "АБРАКАДАБРА" записывали определенным образом на ткани и это заклинание носили в виде шарфика вокруг шеи. Считалось, что заклинание защищает от малярии и других болезней. Получить эту запись в виде треугольника.

Program Primer12; uses crt;

```
var a: string;  
    i, k: integer;
```

Begin

```
clrscr;  
a:='a b r a k a d a b r a';  
k:=0;  
for i:=1 to 11 do  
begin  
gotoxy(30+i,9+i);  
delete(a,22-k,2);  
write(a);k:=k+2  
end;  
readkey;
```

End.

Программу можно написать и другими способами.

Можете подумать на досуге.



ЗАДАНИЯ:

1. Набрать и исполнить программы. Разобраться в работе программ.
2. Ввести слово произвольной длины и напечатать его в виде треугольника: 1)Л 2) ЛИПА 3) ЛИПА 4) ЛИПКА

```
ЛИ      ЛИП      ИПА      ИПК  
ЛИП     ЛИ      ПА      П  
ЛИПА    Л      А
```

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 29 ОБРАБОТКА СИМВОЛЬНЫХ ВЕЛИЧИН 4

Задача: Ввести с клавиатуры предложение.

1. Подсчитать в нем количество всех символов.
2. Подсчитать количество букв "а".
3. Заменить запятые пробелами.
4. Осуществить движение первых двух символов вниз.

Program Simwol; Uses crt;

```
var a,          { начальное предложение }  
    b,          { промежуточная переменная }
```

```

c: string;      {конечное предложение}
i,             {переменная цикла      }
k,             {счетчик символов "a"}
l: integer; {счетчик символов}

```

Begin

```

clrscr;
write('Введите предложение :'); readln(a);
l:=length(a);
writeln('В предложении содержится ',l,' символов');
k:=0; b:=""; c:="";
for i:=1 to l do begin b:=copy(a,i,1); if b='a' then k:=k+1 end;
writeln('В предложении содержится ',k,' символов "a"');
for i:=1 to l do
begin
  b:=copy(a,i,1);
  if b<>' ' then c:=c+b else c:=c+' '
end;
delay(6000);           {временная задержка вывода (пауза)}
clrscr; write(c);
b:=copy(c,1,2);
for i:=2 to 22 do
begin
  gotoxy(1,i); write(b); delay(500); {печать двух символов }
  gotoxy(1,i); write(' ')           {стирание этих символов }
end; readkey;

```

End.



ЗАДАНИЯ:

1. Набрать и исполнить программу.
2. Разобраться в работе программы.
3. Заменить движение первых двух символов на движение слова.
4. Заменить движение слова по вертикали на движение по наклонной.
5. Заменить движение слова на перемещение букв этого слова по одной.

МАССИВЫ

ПОСТАНОВКА ЗАДАЧИ - существует большое число задач, требующих много однотипных переменных. Например, присвоить переменным значения среднесуточных температур в течение года.

Работать с таким количеством переменных очень трудно: процесс описания, присваивания, способы обработки переменных и т.д.

РЕШЕНИЕ ЗАДАЧИ - введем новое понятие - массив.

МАССИВ - это упорядоченный по номерам набор значений, объединенных общим типом и именем. Упорядоченность в том, что элементы массива располагаются в последовательных, пронумерованных ячейках памяти. Массив

имеет фиксированные: имя, тип и количество значений. Массиву выделяется место в памяти.

Элементы массива - это переменные с индексами. Индексы можно вычислять.

ОПРЕДЕЛЕНИЕ МАССИВА - массивы, как и простые переменные, подлежат описанию в разделе переменных:

VAR <имя>: ARRAY[<min индекс..max индекс>] OF <тип элементов>;
array (эгрэй) - массив. of (оф) - из.

Тип индексов - любой скалярный тип кроме real. Индексы можно вычислять.

Тип элементов - любой для данной версии языка.

[min..max] - размер массива. min - нижняя, max - верхняя границы индексов. min<=max. Границы можно задать с помощью переменных.

Можно задавать постоянный массив:

const x: array [1..10] of integer=(4,6,3,5,2,1,7,8,9,13);

const x: array [1..8] of char=('a','b','c','d','e','f','g','h');

Элементы таких массивов являются константами.

Если массивы одинаковы и отличаются лишь именами, то их можно описать в одной строке:

var a,b,c :array[1..n] of real; , где n -верхняя граница индекса и n описывается в разделе констант.

ВИДЫ МАССИВОВ:

Одномерные - ряд элементов. В[5] - 5-ый элемент массива.

Двумерные - таблицы. В[3,5] - элемент таблицы В ,стоящий в 3 строке и в 5 столбце.

Многомерные - ряд таблиц (изучаются в Вузе).

ЗАПОЛНЕНИЕ МАССИВА:

1. С помощью CONST.
2. С помощью READ и циклической конструкции.
3. Часто требуется обнуление ячеек массива. **For I:=1 to n do A[I]:=0;**
4. Для отладки программ и для программирования игр, тестов применяют заполнение массива случайными числами. **randomize; For I:=1 to n do A[I]:=random(q-p+1)+p; q≥p**
5. Присваивание значений. A[1]:=7; A[2]:=77; A[3]:=33; и т. д.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 30 ЗАПОЛНЕНИЕ И ВЫВОД МАССИВОВ

Заполнить и вывести на экран одномерный массив целых чисел.

Program Primer1; uses crt;

var a: array[1..5] of integer;

i: integer;

Begin

clrscr;

{***** заполнение массива с клавиатуры ***** }

for i:=1 to 5 do

begin

```

write('a['i,']='); readln(a[i]);
end;
clrscr;
{***** ВЫВОД МАССИВА НА ЭКРАН *****}
Write('Массив A: '); for i:=1 to 5 do write(a[i], ' ');
readkey;
End.

```



ЗАДАНИЯ:

1. Набрать и исполнить программу.
2. Увеличить число элементов массива.
3. Описать размерность массива с помощью переменных.
4. Оформить программу с помощью процедуры ввода массива и процедуры вывода массива на экран.
5. Заполнить массив нулями.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 31 **ЗАПОЛНЕНИЕ И ВЫВОД МАССИВОВ (продолжение)**

Заполнить и вывести на экран одномерный массив символов.

```

Program Primer2; uses crt;
const a: array[1..5] of char=('a','b','c','d','e');

```

```

procedure Print(n: integer);
var i: integer;
begin
for i:=1 to n do write (a[i]:2)
end;
Begin
clrscr;
write('Массив A: '); Print(5);
readkey;
End.

```



ЗАДАНИЯ:

1. Набрать и исполнить программу.
2. Вывести на экран первые 3 буквы.
3. Вместе с тремя буквами вывести на экран последнюю букву.
3. Заполнить массив дробными и отрицательными значениями.

4. Заполнить и вывести на экран одномерный массив, содержащий 200 одинаковых элементов.
5. Вывести элементы построчно (используйте **wherex**, см. практическое занятие 17).
6. Вывести элементы в указанной с клавиатуры строке экрана.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 32
ОБРАБОТКА ОДНОМЕРНЫХ ЧИСЛОВЫХ МАССИВОВ.
СУММИРОВАНИЕ ЭЛЕМЕНТОВ МАССИВА

ПОСТАНОВКА ЗАДАЧИ: Найти сумму элементов одномерного массива.
 Найти среднее арифметическое.

Program Summa; uses crt;

const n=5;

var a: array[1..n] of integer;

m,i,s: integer;

procedure Input;

begin

for i:=1 to n do

begin

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

end

end;

procedure Print;

begin

write('Массив целых чисел A: ');

for i:=1 to n do write(' ',a[i])

writeln; writeln; { пропуск строки }

end;

procedure Sum(m: integer);

begin

s:=0;

for i:=1 to m do s:=s+a[i];

end;

Begin

ClrScr;

Input; { заполнение массива из 5 элементов }

ClrScr;

Print; { печать заполненного массива }

write('Сколько элементов суммировать: '); readln(m);

Sum(m); writeln; { суммирование m элементов массива }

writeln('Сумма ',m,' элементов массива равна: ', s);

{ суммирование всех элементов массива }

Sum(n); writeln;

writeln('Сумма всех ',n,' элементов массива равна: ', s);

Автоматическое заполнение
случайными числами от 1 до 20

Procedure Input;

begin

p:=1; q:=20; randomize;

for i:=1 to n do a[i]:=random(q-p+1)+p;

end;

Write ('Среднее арифметическое всех элементов равно ', s/n);
 Readkey;
 End.



ЗАДАНИЯ:

1. Набрать и исполнить программу.
2. Увеличить число элементов массива.
3. Найти сумму 8 элементов массива.
4. Осуществить ввод количества элементов массива с клавиатуры.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 33 ОБРАБОТКА ОДНОМЕРНЫХ ЧИСЛОВЫХ МАССИВОВ СУММА ЭЛЕМЕНТОВ МАССИВА ПО УСЛОВИЮ

ПОСТАНОВКА ЗАДАЧИ: Заполнить и вывести массив на экран.

Ввести условие суммирования ($a[i] < u$).

Найти сумму и вывести на экран результат.

```
Program SumUs1; uses crt;
const a: array[1..10] of integer=(3,5,2,1,6,3,8,7,5,9);
var i, s, u: integer;
procedure Su;
begin
  s:=0;
  for i:=1 to 10 do if a[i]<u then s:=s+a[i]
  write('Сумма элементов меньших ',u,' равна ',s);
end;
Begin
  ClrScr;
  Print;          { смотри занятие 32 }
  writeln; writeln;
  write('Введите условие суммирования : '); readln(u);
  Su;
  Readkey;
End.
```



ЗАДАНИЯ:

1. Изменить условие суммирования.
2. Написать процедуру Input.
3. Найти произведение элементов по условию.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 34
ОБРАБОТКА ОДНОМЕРНЫХ МАССИВОВ
ОПРЕДЕЛЕНИЕ ЧИСЛА ЭЛЕМЕНТОВ МАССИВА ПО УСЛОВИЮ

ПОСТАНОВКА ЗАДАЧИ:

Заполнить массив и вывести его на экран.

Ввести с клавиатуры условие.

Подсчитать количество элементов удовлетворяющих условию.

Program KolUsl; uses crt;

const m=1; { min индекс }

n=10; { max индекс }

var i, k, usl: integer;

a: array[m..n] of integer;

... { процедуры Input и Print см. занятие 32 }

procedure Kol_po_Usl;

begin

write('Введите ограничение :'); readln(usl);

k:=0;

for i:=m to n do if a[i]>usl then k:=k+1

write('Количество элементов больших ',usl, ' равно ', k);

end;

Begin

ClrScr;

Input;

ClrScr;

Print;

writeln; writeln;

Kol_po_Usl;

Readkey;

End.



ЗАДАНИЯ:

1. Самостоятельно написать процедуры Input, Print и исполнить программу.
2. Изменить условие подсчета элементов.
3. Подсчитать сумму элементов по условию.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 35 ОБРАБОТКА ОДНОМЕРНЫХ ЧИСЛОВЫХ МАССИВОВ СУММА ДВУХ ОДИНАКОВЫХ ПО РАЗМЕРУ МАССИВОВ

В языке Pascal программист может описывать свои собственные типы и объекты, относящиеся к этому типу. Тип массив относится к *структурированным* типам. Описание типа помещают в раздел определения типов после раздела описания констант и до раздела описания переменных (var).

Например: **TYPE massiv=array[1..100] of integer;**
var a, b, c: massiv;

Описание типа позволяет применить формальный параметр типа massiv.

ПОСТАНОВКА ЗАДАЧИ:

Заполнить и вывести на экран два одинаковых по размеру массива: А и В.

Найти массив С ,каждый элемент которого равен сумме соответствующих элементов массивов А и В: $C[i]=A[i]+B[i]$.

```
Program Summa_Mass; uses crt;
Type mas= array[1..100] of integer;
var a, b, c: mas;
    I, n: integer;
procedure Input(var x: mas);
begin
    for i:=1 to n do begin write('Элемент ['i,']='); readln(x[i]); end
end;
procedure Print(x: mas); begin for i:=1 to n do write(' ',x[i]); end;
procedure SumMass; begin for i:=1 to n do c[i]:=a[i]+b[i]; end;
Begin
    ClrScr; n:=20;
    writeln('Ввод массива А:'); Input(a);
    writeln('Ввод массива В:'); Input(b);
    ClrScr;
    write('  А: '); Print(a);
    write('  В: '); Print(b);
    SumMass;
    write('  С: '); Print(c);
    readkey;
End.
```



ЗАДАНИЯ:

1. Набрать и исполнить программу.
2. Найти сумму элементов, стоящих на четных местах в массиве С.
3. Найти сумму четных элементов в массиве С.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 36 ОБРАБОТКА ОДНОМЕРНЫХ МАССИВОВ ОБЪЕДИНЕНИЕ ДВУХ МАССИВОВ С ЧЕРЕДОВАНИЕМ ЭЛЕМЕНТОВ

ПОСТАНОВКА ЗАДАЧИ:

Объединить два массива А и В ,содержащих по N элементов в один массив с, который будет содержать 2*N элементов, т.е. получить массив С=(a1,b1,a2,b2,...,an,bn).

```
Program Ob_Mass; uses crt;
type massiv=array[1..20] of integer;
var a,b,c: massiv;
    I, n: integer;
procedure Input(var x: massiv; n: integer);
begin
  for i:=1 to n do
    begin
      write('Элемент [' ,i, '='); readln(x[i]);
    end
end;
procedure Print(x: massiv; n: integer);
...

procedure Ob(n: integer);
begin
  for i:=1 to n do
    begin
      c[2*i-1]:=a[i];
      c[2*i]:=b[i]
    end
end;
Begin
  ClrScr; n:=10;
  Input(a,n); writeln;
  Input(b,n);
  ClrScr;
  write('Массив A: '); Print(a, n); writeln;
  write('Массив B: '); Print(b, n); writeln;
```

```

Ob(n);
write('Массив C: '); Print(c, 2*n);
readkey;
End.

```



ЗАДАНИЯ:

1. Набрать и исполнить программу.
2. Изменить количество элементов в массивов.
3. Объединить массивы так, чтобы элементы одного массива следовали за элементами другого: $c=(a_1, a_2, a_3, \dots, a_n, b_1, b_2, b_3, \dots, b_n)$.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 37 ОБРАБОТКА ОДНОМЕРНЫХ МАССИВОВ ИНВЕРТИРОВАНИЕ МАССИВА

ПОСТАНОВКА ЗАДАЧИ:

Требуется изменить порядок следования элементов массива C , состоящего из n элементов, на обратный, используя вспомогательную переменную p .

Исходный массив C : 1 2 3 4 5

Конечный массив C : 5 4 3 2 1

Перестановка производится по одному элементу через промежуточную переменную:

- 1) 1-ый элемент передается в ячейку p ;
- 2) последний элемент ставится на место 1-го;
- 3) элемент из ячейки p ставится на последнее место.

Количество таких перестановок равно $\text{trunc}(n/2)$ или $(n \text{ div } 2)$ – деление нацело.

Program Inwert; uses crt;

{ описание переменных и массива }

{ описание процедуры ввода массива }

{ описание процедуры вывода массива }

procedure Inw;

var m: integer; p: real;

begin

m:=trunc(n/2); { определение числа циклов }

for i:=1 to m do

begin

p:=c[i]; { перестановка i элемента в p }

c[i]:=c[n-i+1]; { перестановка на i место элемента n-i+1 }

c[n-i+1]:=p; { перестановка элемента из p на n-i+1 место }

end

end;

Begin

```

ClrScr; n:=20;
Input;
ClrScr;
Print; writeln;
Inw;
Print; writeln;
Inw;
Print; writeln;
Readkey;
End.

```



ЗАДАНИЯ:

1. Набрать и исполнить программу.
2. Увеличить число элементов.
3. Заменить числовой массив на символьный.
Получить из слова ГОРОД слово ДОРОГ и другие перевертыши.
4. Произвести инвертирование с сохранением исходного массива.
Подсказка: используйте новый массив.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 38 ОБРАБОТКА ОДНОМЕРНЫХ МАССИВОВ ФОРМИРОВАНИЕ МАССИВА ИЗ ЭЛЕМЕНТОВ ДРУГОГО МАССИВА УДОВЛЕТВОРЯЮЩИХ ЗАДАННОМУ УСЛОВИЮ

ПОСТАНОВКА ЗАДАЧИ: Требуется из данного массива A , состоящего из N элементов, выбрать элементы, удовлетворяющие заданному (пусть $A(i) > T$), и сформировать из них массив B .

Исходный массив A : 4 6 8 2 23 1 9 Условие: $A(i) \geq 8$

Новый массив B : 8 23 9

Особенность решения: Индексы элементов массивов A и B не совпадают, так как не все элементы массива A включаются в массив B .

Введем переменную, формирующую индексы и являющуюся счетчиком элементов нового массива.

Program Massiv_po_Usloviu; Uses crt;

```

...
procedure Mass_Usl;
begin
  write('Введите параметр условия: '); readln(t);
  k:=0;           {k- счетчик найденных элементов   }
  for i:=1 to n do if a[i]>=t then begin k:=k+1; b[k]:=a[i]; end;
end;
Begin
  ClrScr; n:=20;

```

```

Input;           {заполнение исходного массива А}
ClrScr;
Print(a, n); writeln; {печать исходного массива А}
Mass_Usl;       {формирование массива по условию}
Print(b, k); writeln; {вывод на экран нового массива В}
Readkey;
End.

```



ЗАДАНИЯ:

1. Набрать и исполнить программу.
2. Изменить условие.
3. Вывести на экран сообщение о количестве найденных элементов.
4. Найти среднее арифметическое элементов нового массива.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 39 ОБРАБОТКА ОДНОМЕРНЫХ МАССИВОВ ПОИСК МАКСИМАЛЬНОГО ЭЛЕМЕНТА В МАССИВЕ С ЗАПОМИНАНИЕМ ЕГО ПОЛОЖЕНИЯ (ИНДЕКСА)

ПОСТАНОВКА ЗАДАЧИ: Найти в массиве А, состоящем из n элементов, наибольший

элемент и запомнить его индекс (К).

```

Program MaxElMassiva; uses crt;
  {описание массива и переменных }
  {процедура ввода массива}
  {процедура вывода массива }

  procedure MaxEl;
  begin
    max:=a[1]; k:=1;
    for i:=1 to n do if max<a[i] then begin max:=a[i]; k:=i; end;
    writeln('Максимальный элемент массива А равен ',max);
    writeln('Максимальный элемент массива стоит на ',k, ' месте);
  end;
Begin
  ClrScr; n:=20;
  Input;
  ClrScr;
  Print; writeln;
  MaxEl;

```

Readkey;;
End.



ЗАДАНИЯ:

1. Набрать и исполнить программу.
2. Найти минимальный элемент и его индекс.
3. Найти минимальный и максимальный элементы, и их индексы.
4. Заменить найденный минимальный элемент числом 100 и показать на экране массив с этой заменой.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 40 ОБРАБОТКА ОДНОМЕРНЫХ МАССИВОВ ПОИСК ЗАДАННОГО ЭЛЕМЕНТА В МАССИВЕ

ПОСТАНОВКА ЗАДАЧИ: Определить, есть ли в заданном массиве элемент равный L.

Имеется массив A: 3 1 2 5 3 4 8 L=5

В массиве A имеется элемент равный 5

В данной программе можно использовать "флажок" - специальную переменную, которая принимает значения "истина" или "ложь" в зависимости от того найден элемент или нет.

```
Program Poisk_El_Massiva; uses crt;
...
procedure Poisk(L: integer);
  var fl: boolean;
begin
  fl:=false;
  for i:=1 to n do if a[i]=L then fl:=true;
  if fl then write(' Элемент равный ',L,' имеется   ')
    else write(' Элемент равный ',L,' отсутствует');
end;
Begin
  ClrScr; n:=20;
  Input;
  ClrScr;
  Print; writeln; writeln;
  write('Введите значение искомого элемента: '); readln(L);
  Poisk(L);
  Readkey;End.
```



ЗАДАНИЯ:

1. Набрать и исполнить программу, т. е. найти 3-4 элемента.
2. Если в массиве несколько одинаковых элементов, то найти первый из них и указать его индекс.
3. Подсчитать количество одинаковых элементов в массиве.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 41 ОБРАБОТКА ОДНОМЕРНЫХ МАССИВОВ УПОРЯДОЧЕНИЕ МАССИВА

ПОСТАНОВКА ЗАДАЧИ: Расположить элементы массива в порядке возрастания.

Из множества сортировок рассмотрим поиск MIN и перестановку.

Исходный массив: 3 2 -5 2 0 8 4

Конечный массив: -5 0 2 2 3 4 8

Program SortirowkaMassiva; uses crt;

```

...
procedure Sort;
  var min: real;
begin
  for i:=1 to n do
    begin
      min:=a[i]; k:=i;
      for j:=i to n do
        if a[j]<min then begin min:=a[j]; k:=j; end;
      a[k]:=a[i];
      a[i]:=min
    end
  end;
Begin
  ClrScr;
  Input;
  ClrScr;
  write('Исходный массив: '); Print; writeln;
  Sort;
  write('Конечный массив: '); Print;
  readkey;
End.
```



ЗАДАНИЯ:

1. Набрать и исполнить программу.
2. Отсортировать исходный массив по убыванию элементов.
3. Отсортировать исходный массив по возрастанию и убыванию элементов.
4. Отсортировать массив латинских букв.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 42 ОБРАБОТКА МАТРИЦ ВВОД И ВЫВОД МАТРИЦ

В Паскале двумерные массивы называют матрицами.

TYPE MATRICA=ARRAY[1..N,1..M] OF REAL; - определение типа двумерного массива, где N -число строк, M - столбцов.

ПОСТАНОВКА ЗАДАЧИ: Заполнить матрицу и вывести ее на экран.

Заполнение матрицы может производиться также как и одномерного массива, только с помощью вложенных циклов.

```
Program Wwod_Wywod_Mat; uses crt;
```

```
const n=3;
```

```
var a: array[1..n,1..n] of integer;
```

```
    i,j: integer;
```

```
Begin      { ввод матрицы по строкам }
```

```
ClrScr;
```

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

```
  for j:=1 to n do
```

```
    begin
```

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

```
    end;
```

```
      { вывод матрицы по строкам
```

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

```
  begin
```

```
    for j:=1 to n do write(a[i,j], ' ');
```

```
    writeln { перевод курсора на новую строку }
```

```
write(a[i,j]:3);
```

```
  end;
```

```
  readkey;
```

```
End.
```

```
      вывод в указанном месте }
```

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

```
  begin
```

```
    gotoxy(10,5+i);
```

```
    for j:=1 to n do
```

```
  end;
```

```
  readkey;
```

```
End.
```

Можно задать постоянную матрицу:

```
const a: array[1..3,1..3] of integer=((1,2,3),  
                                       (4,5,6),  
                                       (7,8,9));
```



ЗАДАНИЯ:

1. Набрать и исполнить программу.
2. Описать процедуры ввода и вывода матриц.
3. Задать постоянную матрицу.
4. Вывести на экран матрицу по столбцам.
5. Вывести матрицу в указанное место экрана.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 43 ОБРАБОТКА МАТРИЦ

1. СУММА ЭЛЕМЕНТОВ МАТРИЦЫ

ПОСТАНОВКА ЗАДАЧИ: Для двумерного массива B размером $N \times M$ необходимо вычислить сумму всех элементов матрицы.

```

Program Summa_Mat; uses crt;
...
procedure Sum;
begin
  s:=0;
  for i:=1 to n do
    for j:=1 to m do
      s:=s+b[i,j]
    end;
end;
Begin
  ClrScr;
  Input_Mat; { если массив вводится с клавиатуры }
  ClrScr;
  Print_Mat; writeln;
  Sum;
  write('Сумма элементов матрицы равна ',s);
  readkey;
End.

```



ЗАДАНИЯ:

1. Набрать и исполнить программу.
2. Ввести дробные и отрицательные числа.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 44 ОБРАБОТКА МАТРИЦ

2. ВЫЧИСЛЕНИЕ СЛЕДА МАТРИЦЫ

СЛЕД МАТРИЦЫ - сумма диагональных элементов матрицы $N \times N$.

ПОСТАНОВКА ЗАДАЧИ: Для матрицы B размером $N \times N$ надо вычислить сумму диагональных элементов $B[i,i]$.

```
...  
procedure Sled_Mat; { главная диагональ }  
begin  
  s:=0;  
  for i:=1 to n do s:=s+b[i,i]  
end;
```

3. ВЫЧИСЛЕНИЕ СУММЫ ЭЛЕМЕНТОВ СТРОК МАТРИЦЫ

ПОСТАНОВКА ЗАДАЧИ: Вычислить сумму элементов каждой строки матрицы B размером $N \times M$. Результат получить в виде вектора D .

```
...  
procedure Sum_Strok_Mat;  
begin  
  for i:=1 to n do  
    begin  
      s:=0;  
      for j:=1 to m do  
        s:=s+b[i,j];  
      d[i]:=s  
    end  
  end;
```

```
...  
...
```



ЗАДАНИЯ:

1. Набрать и исполнить программу.
2. Получить сумму элементов побочной диагонали.
3. Напечатать вектор суммы строк вертикально справа от матрицы.
4. Получить вектор суммы столбцов матрицы.

ПРИЛОЖЕНИЕ**ПРОЦЕДУРЫ ОБРАБОТКИ МАТРИЦ:****Сумма указанной строки:**

S:=0; i:=3;

For J:=1 to m do S:=S+b[i,j];

Транспонирование квадратной матрицы:

For i:=1 to n-1 do

 For j:=1 to n do

 Begin

 P:= a[i,j];

 a[i,j]:=a[j,i];

 a[j,i]:=p;

 end;

Удаление строки из матрицы:

n:=n-1;

For i:=1 to n do

 For j:=1 to m do b[i,j]:=b[i+1,j];

Включение строки в матрицу:

i:=n;

while i>=k do

 begin

 for j:=1 to m do b[i+1,j]:=b[i,j];

 i:=i-1;

 end;

for j:=1 to m do b[k,j]:=c[j];

n:=n+1;

Перестановка строк матрицы:

For k:=1 to k do c[k]:=a[i,k];

For k:=1 to k do a[i,k]:=a[j,k];

For k:=1 to k do a[j,k]:=c[k];

Поиск минимального элемента матрицы:

Min:= a[1,1]; k:=1; L:=1;

For i:=1 to n do

 For j:=1 to m do

 IF min>a[i,j] then begin min:=a[i,j]; k:=i; l:=j; end;

Сложение строк матрицы:

For j:=1 to m do

 a[k,j]:=a[k,j]+a[l,j]*b;



ЗАДАНИЯ:

1. Изменить значения указанной строки.
2. Найти минимальные значения в каждой строке.
3. Найти максимальные значения в каждом столбце.

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 45 ПОДГОТОВКА К КОНТРОЛЬНОЙ РАБОТЕ

ПОСТАНОВКА ЗАДАЧИ:

1. ПОВТОРИТЬ ТЕМУ "МАССИВЫ (ВЕКТОРА И МАТРИЦЫ)".
2. В КОНТРОЛЬНОЙ РАБОТЕ 2 ЗАДАНИЯ:
ОБРАБОТКА ОДНОМЕРНОГО МАССИВА (ВЕКТОРА);
а) 1 -я обработка; б) 2 -я обработка.
3. ПРОГРАММА ДОЛЖНА ИМЕТЬ БЛОЧНУЮ СТРУКТУРУ.

ПРИМЕР ЗАДАНИЯ: Заполнить одномерный числовой массив и **выбрать** из него отрицательные элементы. **Найти** среднее арифметическое отрицательных элементов массива. Отрицательные элементы **отсортировать** по возрастанию.

ПРИМЕР РЕШЕНИЯ:

```
Program Vektor; uses crt;
type mas=array[1..40] of integer;
var i, k, s, n, p, q: integer;
    a, d: mas;
procedure Input;
begin
    randomize; p:=-10; q:=10; for i:=1 to n do a[i]:=random(q-p+1)+p;
end;
procedure Print (y: mas; x: integer);
begin
    for i:=1 to x do write(y[i], ' '); writeln; writeln;
end;
procedure Vyor;
begin
    k:=0; for i:=1 to n do if a[i]<0 then begin k:=k+1; d[k]:=a[i]; end;
end;
function Sredn_Arifm: real;
begin
    s:=0;
    if k<>0 then begin for i:=1 to k do s:=s+d[i]; Sredn_Arifm:=s/k; end
    else Sredn_Arifm:=0
end;
```

```

procedure Sort;
  var l,j,min: integer;
begin
  k:=0; l:=0;
  for i:=1 to k-1 do begin
    min:=d[i];
    for j:=i+1 to k do
      if d[j]<min then begin
        min:=d[j]; l:=j;
        d[l]:=d[i]; d[i]:=min
      end
    end
  end;
begin
  ClrScr;
  write('Число элементов(<40) : '); readln(n); clrscr; Input;
  write('Исходный массив A: ');      Print(a,n); vybor;
  write('Массив D: '); Print(d,k);
  writeln('Среднее арифметическое отрицательных элементов массива равно
',Sredn_Arifm);
  if Sredn_Arifm=0 then writeln('Нет отрицательных элементов');
  Sort;
  if Sredn_Arifm<>0 then
    begin
      write('Отсортированный массив по возрастанию: ');
      Print(d,k)
    end;
  readkey;
end.

```

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 46. ЗАПИСИ

При работе с массивами существует один существенный недостаток – элементы в массиве одного типа. На практике часто требуется обрабатывать данные с элементами разного типа. Если одновременно вести работу с несколькими массивами разного типа, то программа получается очень громоздкой и требуется много времени для отладки такой программы.

В Паскале предусмотрена возможность описывать сложные типы и работать с ними. **ЗАПИСЬ** – упорядоченное множество значений, называемых **полями записи**, которые могут иметь разные типы.

ЗАПИСЬ – структура данных, построенная из компонент, имеющих в общем случае разные типы данных.

Тип записи

TYPE NAME=RECORD

поле_1: тип;

поле_2: тип;

.....
поле_N:тип;
END;

VAR NAMREC: NAME;

Record – служебное слово – запись,

NAME – имя типа записи,

Поле_N – поле записи указанного типа, в качестве поля может быть запись,

NAMREC – имя записи – переменная определенного TYPE типа;

Имена полей не должны совпадать,

Если полей одного типа несколько, то их имена перечисляются через запятую:

Record x,y: real; end;

Доступ к элементу записи производится с помощью *компонентных переменных*, имеющих форму **R.T**, где R – переменная типа записи, T – имя поля.

Если полей много, то обращение можно произвести с помощью *оператора над записью* или *оператора присоединения WITH* имя_записи **DO**

Задача

Для группы учащихся, имеющих итоговые оценки по 5 предметам вычислить средний балл и упорядочить список группы по убыванию среднего балла.

Нарисуем таблицу с указанием имен полей, их типов.

Затем приступим к описанию переменных и написанию программы.

ТАБЕЛЬ УСПЕВАЕМОСТИ

Фамилия	Матем	Русский	Физика	Химия	История	Средн балл
Fam:	Mat:	Rus:	Fiz:	Xim:	Ist: byte	Sb:real
string[15]	byte	byte	byte	byte		
Иванов	4	5	3	4	3	3.8
Петров	5	3	4	3	5	4
Сидоров	3	4	3	3	4	3.4

Program Sr_Ball; uses crt;

type grup=record

fam:string[15];

mat,rus,xim,fiz:byte;

sb:real;

end;

var tbl:array[1..20] of grup;

y:grup;

i,j,k,m:integer;

x:real;

Begin

clrscr;

write('Количество учащихся в группе: '); readln(m);

writeln;

for i:=1 to m do

with tbl[i] do

```

begin
  write('fam[',i,']='); readln(fam);
  write('mat='); readln(mat);
  write('rus='); readln(rus);
  write('xim='); readln(xim);
  write('fiz='); readln(fiz);
end;
for i:=1 to m do with tbl[i] do sb:=(mat+rus+xim+fiz)/4;
for i:=1 to m do
begin
  k:=i; x:=tbl[i].sb;
  for j:=i to m do
    if tbl[j].sb>x then begin k:=j; x:=tbl[j].sb end;
  y:=tbl[k];
  tbl[k]:=tbl[i];
  tbl[i]:=y;
end;
clrscr;
writeln('Пн | Фамилия      | Mat | Rus | Xim | Fiz ');
writeln('-----');
for i:=1 to m do
  with tbl[i] do
    writeln(i:2,'|',fam:15,'|',mat:2,'|',rus:2,'|',xim:2,'|',fiz:2);
  readln; clrscr; writeln('Пн | Фамилия      | Sredn ball ');
  for i:=1 to m do with tbl[i] do writeln(i:2,'|',fam:15,'|',sb:3:1); readkey;
End.

```

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 47. ФАЙЛЫ

Файл – форма хранения информации вне среды.

Файл удобен:

- данные из файла можно использовать в нескольких программах;
- файл сохраняется после окончания работы программы;
- данные часто не убираются в ОЗУ (игры);
- с файлами можно выполнять множество полезных манипуляций.

Типы файлов:

- 1) Последовательные – чтение/запись от начала к концу.
- 2) Файлы прямого доступа – доступ к записи по адресу.
- 3) Индексированные файлы – доступ к записи по адресу и по ключу.

Порядок работы с файлами.

1. Объявить файл (var f: file of тип или var f: text для текстового файла).
2. Связать файловую переменную с конкретным файлом на устройстве.
3. Открыть файловую переменную для записи или чтения.
4. Закрыть файловую переменную для записи или чтения.

По организации работы с данными различают три вида файлов: *типизированные, текстовые и нетипизированные*.

Основные процедуры и функции обработки файлов (общие).

- Assign(f,name);** - связь файловой переменной f с внешним файлом.
Reset(f); - открыть существующий файл для чтения.
Rewrite(f); - открыть файл для записи (если существует – стирается!).
Close(f); - закрыть файл для чтения/записи.
Eof(f) - функция «конец файла» (true, если из файла всё считано).

Для типизированных файлов:

- Read(f,a);** - чтение информации из файла в переменную a.
Write(f,a); - запись информации (значения переменной a) в файл.
Seek(f,num); - пропуск заданного количества записей в файле.
Truncate(f); - удаление части файла с текущего компонента до конца.
FilePos(f) - функция номера текущей записи в файле.
FileSize(f) - функция - количество записей в файле (размер файла).

Для текстовых файлов:

- Readln(f,a);** - чтение строки из файла.
Writeln(f,a); - запись строки в файл.
Append(f); - открыть файл для добавления в конец файла.
Eoln(f) - функция конец строки файла.
SeekEoln(f) - функция конец строки файла(не замечает пробелов в конце).
SeekEof(f) - функция конец файла(не замечает пробелов в конце файла).

Задача 1. Присвоить переменным числовые значения и сохранить их в файле. Прочитать файл, передав числовые значения другим переменным.

Задача 2. Создать текстовый файл данных из нескольких строк. Прочитать файл. Прочитать числовое значение из текстового файла в числовую переменную и изменить полученное значение.

```
Program fil; Uses crt;
```

```
  var f: file of integer;  
      b,c,d,x,y,z,i:integer;  
      t:text; a:string;
```

```
Begin
```

```
  clrscr; b:=3; c:=4; d:=5;  
  writeln('b=',b,' c=',c,' d=',d);  
  assign(f,'c:r.dat');  
  rewrite(f);  
  write(f,b,c,d);  
  close(f);  
  reset(f);  
  read(f,x,y,z);  
  write('x=',x,' y=',y,' z=',z);  
  close(f);  
  readkey; clrscr; assign(t,'c:\pascal\turbo7\work\prob.dat');  
  reset(t);  
  for i:=1 to 4 do  
    begin
```

```

readln(t,a);
writeln(a);
end;
readln(t,x); x:=x*2; writeln('x=',x);
close(t);
readkey;
End.

```

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 48. ЗАДАЧИ ОПТИМИЗАЦИИ

Проблема оптимизации встречается во всех сферах человеческой деятельности - от самого высокого общегосударственного уровня до личного (распределение средств). Задача сводится к поиску наилучшего варианта с точки зрения намеченной цели.

При решении проблемы следует оформить ее как математическую задачу, придав количественный смысл словам "лучше", "хуже".

Многие задачи оптимизации сводятся к поиску **МАХ**, **МИН** некоторой функции, которую принято называть **ЦЕЛЕВОЙ ФУНКЦИЕЙ** или **КРИТЕРИЕМ КАЧЕСТВА**.

Задачи оптимизации, в которых целевая функция является линейной функцией независимых переменных, а условия, определяющие допустимые значения этих переменных, имеют вид линейных уравнений и неравенств относят к **ЛИНЕЙНОМУ ПРОГРАММИРОВАНИЮ**.

НАЧАЛО: 1938г. Ленинградский Государственный Университет. **Л.В. Канторович** приступил к решению задачи о распределении работы 8 станков. Канторович нашел общий метод решения. Позже Канторович получил Ленинскую и Нобелевскую премии.

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ

Этапы моделирования:

1. Определение цели и формирование **ЦЕЛЕВОЙ ФУНКЦИИ**.
2. Составление **СПИСКА ПЕРЕМЕННЫХ** и ранжировка их по степени влияния на целевую функцию. Выделение значимых переменных.
3. Составление математического описания процесса.
4. Разработка алгоритма реализации математического описания на ЭВМ.
5. Проверка соответствия математической модели моделируемому процессу.
6. Исследование объекта моделирования с помощью математической модели.

НЕКОТОРЫЕ ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ ЗАДАЧА О ИСПОЛЬЗОВАНИИ РЕСУРСОВ

Выпускаются стулья двух типов:

Переменная	Цена(у.е.)	Доски(м)	Ткань(кв.м)	Рабочее время(чел.час)
X1	8	2	0,5	2
X2	12	4	0,25	2,5
Ресурсы:		440	65	320

Какие стулья выпускать и в каком количестве чтобы выручка от продажи стульев была наибольшей?

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ

$$2 \cdot x_1 + 4 \cdot x_2 \leq 440$$

$$0,5 \cdot x_1 + 0,25 \cdot x_2 \leq 65 \quad \text{Целевая функция } P = 8 \cdot x_1 + 12 \cdot x_2 \rightarrow \max$$

$$2 \cdot x_1 + 2,5 \cdot x_2 \leq 320$$

$$x_1 \geq 0, x_2 \geq 0$$

После построения и исследования графиков получим, что оптимальные значения: $x_1=60$, $x_2=80$ стульев.



ЗАДАНИЯ:

1. Найти выручку, если выпускать стулья только x_1 .
2. Найти выручку, если выпускать стулья только x_2 .
3. Найти выручку, если выпускать стулья $x_1=x_2$

ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 49. БИОЛОГИЧЕСКАЯ ЗАДАЧА

Для производства вакцины на заводе выращивают культуру бактерий. ИЗВЕСТНО, что если масса бактерий X г, то через день она увеличится на $(A - B \cdot X) \cdot X$ г, где A и B - коэффициенты рождение и гибель бактерий. Ежедневно для нужд производства забирается M г бактерий.

ИССЛЕДОВАТЬ, как изменяется масса бактерий по дням от 1 до 365.

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ

Исходные данные: коэффициенты A, B

начальная масса X_0

масса для производства M

Результаты: масса бактерий через 1,2,3,...,365 дней

Из условия получим рекуррентную формулу: $X_n = X_{n-1} + (A - B \cdot X_{n-1}) \cdot X_{n-1} - M$

Если масса бактерий станет равной или меньшей 0, то необходимо сигнализировать о ЧП - чрезвычайном происшествии и остановить производство.

ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ

$A=1, B=0.0001, X_0=12000, M=2000$.

Масса бактерий сначала быстро убывает, но на 11 день становится равной 7236 г и после этого практически не изменяется.

Увеличим начальную массу бактерий до $X_0=17000$. Масса резко снижается, но к 27 дню стабилизируется на уровне 7236 г.

Возьмем 18000 г, то уже через 2 дня произойдет ЧП. Бактерии гибнут.

НАЙТИ:

1. Минимальное первоначальное количество бактерий.
2. Максимальное первоначальное количество бактерий.
3. Определить какую максимальную массу можно забирать, чтобы завод бесперебойно работал в течение года при начальной массе $X_0=12000$ грамм.

ЭКОЛОГИЧЕСКАЯ ЗАДАЧА

В результате аварийного или преступного сброса в реку загрязненных стоков в ней резко увеличилась концентрация вредных веществ. Требуется ОПРЕДЕЛИТЬ, каков будет уровень загрязненности реки через сутки, двое и т.д. и через какое время уровень загрязненности станет меньше допустимого, если ИЗВЕСТНО, что за сутки он

уменьшается в определенное количество раз.

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ

В каждом городе, районе имеется СЭС - санитарно-эпидемиологическая станция, которая, в частности, следит за состоянием воды в реках, источниках, прудах и т.д. Лаборант СЭС может выезжать на место загрязнения по сообщению жителей, рыбаков, туристов и т.д. На месте загрязнения берется проба воды сегодня, завтра и с помощью лабораторных исследований определяется коэффициент, который показывает, во сколько раз за сутки уменьшается загрязненность. Коэффициент $K > 1$, так как при $0 < K < 1$ - загрязненность увеличивается, а при $K = 1$ - не изменяется.

Обозначим: C_0 - начальная концентрация (лабораторные исследования);

C_d - допустимая концентрация (табличные значения);

C_n - искомая концентрация через n суток.

Естественно предположить, что $C = C_n / K$.

Организуя итерационный процесс с выводом результата на экран можно получить уровень загрязненности через сутки, двое и т.д.

ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ

ВЕЩЕСТВО	C_0 (мг/л)	C_d (мг/л)	K	n (суток)
СВИНЕЦ	10	0.03	1.12	
МЫШЬЯК	5	0.05	1.05	
ФТОР	8	0.05	1.01	

Написать программу и найти значения n для свинца, мышьяка и фтора - очень опасных веществ для человека, если концентрация их значительно превышает норму.

**ПРАКТИЧЕСКОЕ ЗАНЯТИЕ № 50.
ЗНАКОМСТВО С IBM. ОПЕРАЦИОННАЯ СИСТЕМА.
NORTON COMMANDER**

IBM - **International Business Machines Corporation** - американская корпорация, разработчик и изготовитель ЭВМ, внешних устройств и программного обеспечения.

Intel Corporation - американская фирма, разрабатывающая и производящая полупроводниковые приборы. Ею были разработаны первые микропроцессоры: Intel 4004 и Intel 8008, которые использовались в калькуляторах, затем были разработаны и изготовлены серии: 80080, 80086, 80088, 80286, 80386, 80486, Pentium, Pentium Pro, -MMX, -II, -III, -4, Pentium D.

Microsoft - американская фирма, разрабатывающая и выпускающая программное обеспечение для ПЭВМ; ведущий разработчик для IBM PC.

IBM PC AT - персональный компьютер фирмы IBM.

PC - Personal Computer, AT - Automatic Translation.

IBM PC состоит из оборудования:

- системный блок;
- дисплей (VGA, SVGA);
- клавиатура (Keyboard);
- мышь (Mouse);
- принтер (Printer);
- CD/DVD ROM;
- модем (Modem);
- сканер (Scanner).

УПРАВЛЕНИЕ: нажать кнопку включения (POWER); (через 10 сек. после ВЫКЛ.)

TURBO - кнопка переключения скорости выполнения операций.

RESET (сброс) - кнопка для перезагрузки ЭВМ без отключения питания.

ОПЕРАЦИОННАЯ СИСТЕМА (ОС) - система для работы с программами, файлами и оглавлениями данных на ЭВМ.

MS DOS - Дисковая Операционная Система фирмы MicroSoft; требует знания специальных команд и точного их ввода. Например, Copy имя файла prn - распечатать текст файла на принтере.

NORTON COMMANDER - программный пакет (программная оболочка) для работы с MS DOS; команды ОС расположены в меню. Создана фирмой Symantec Corporation. - выполняет все команды DOS; не требуется запоминать множество команд,

WINDOWS - операционная система с графической оболочкой, позволяющая исполнять команды ОС.

КРАТКАЯ СПРАВКА ПО NORTON COMMANDER

После включения ЭВМ автоматически загружается с винчестера ОС и программная оболочка NORTON COMMANDER. О ее загрузке можно узнать по синим панелям (окнам) на экране монитора.

Если панели не появились, то в MS DOS набрать команду ps и нажать Enter.

Активное окно - то, в котором находится подсветка (highlight).

Смена окна производится нажатием Tab.

Под панелями расположена строка для ввода команд MS DOS.

При работе в NC строка автоматически заполняется командами MS DOS.

КАТАЛОГ (catalog) - структура данных, обеспечивающая поиск объекта по текстовому имени. В каталог входит перечень имен файлов и имен других каталогов

(подкаталогов), объединенных по какому-либо признаку и хранящихся на устройствах памяти ЭВМ. Имена каталогов записываются заглавными буквами.

ФАЙЛ (file) - форма хранения информации вне среды. Имена файлов записываются строчными буквами по правилу 8.3. Файлы хранятся на каком-либо носителе информации, например на магнитном диске.

СРЕДА (environment) - программные средства, с которыми взаимодействует программа и аппаратура, на которой она выполняется. Например, среда программирования - Pascal.

Файлы имеют **РАСШИРЕНИЕ** от 0 до 4 букв. Расширение помогает систематизировать файлы. Файлы с расширениями: exe, com, bat - выполняемые программы;

.txt, .doc, .lst - текстовые файлы;

.bat - командные файлы ОС; .pas - программы на Паскале;

.bas - программы на Бейсике; .bak - старые версии файлов.

ФУНКЦИИ КЛАВИШ:

F1 - справка по NC;

F2 - меню пользователя - набор названий программ, часто используемых в работе;

F3 - просмотр текста файла;

F4 - редактирование файлов;

F5 - копирование файлов в другой каталог или устройство;

F6 - переименование файлов, перемещение;

F7 - создание нового каталога или подкаталога;

F8 - удаление текущего файла или каталога (пустого);

F9 - дополнительное меню;

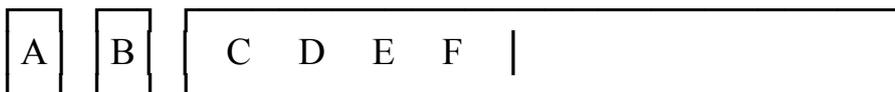
Enter - вход в каталог, выход из подкаталога (курсор на ..), запуск программ ;

Insert - выделить/снять выделение файла, на котором находится курсор;

ПОИСК ФАЙЛОВ:

Alt+F1 - вывод информации о устройствах в левую панель;

Alt+F2 - вывод информации о устройствах в правую панель;



A - НГМД 5.25" B - НГМД 3.5"

C - винчестер(НЖМД); D,E,F - логические диски на винчестере.

Выбор устройства подтверждается нажатием **Enter**.

Перемещаясь по каталогам и подкаталогам, ведем поиск нужного файла. Если файлов много, то такой поиск утомителен.

Можно осуществить быстрый поиск файла на **ТЕКУЩЕМ ДИСКОВОДЕ**, используя команду Alt+F7. Поиск можно производить по первым буквам файла, а также использовать *. Например, найти файлы *.lst

ЗАПУСК ПРОГРАММ

-Если файл имеет расширение .exe или .com, то такой файл запускается непосредственно из Norton Commander.

-Если файл создан в среде, создающей свои специфические файлы, то надо запустить данную среду, затем открыть и запустить файл.

Например, файл с расширением .pas можно запустить после входа в среду Pascal.

ПУТЬ - указание имен каталогов и подкаталогов ведущих к файлу.

Дисковод:\путь\имя файла -> c:\pascal\turbo7\work\life.pas

c: - дисковод (устройство – жесткий диск);

pascal - каталог; life - имя файла;

turbo7, work - подкаталоги; .pas - расширение имени файла.

В верхней строке автоматически показывается путь от родительского каталога к файлу.



ЗАДАНИЯ:

1. Перейти в соседнюю панель и обратно.
2. Войти в любой каталог, посмотреть его содержание и выйти из него.
3. Найти файл turbo.exe разными способами.
4. Посмотреть каталог диска a: на левой и правой панелях. Вернуться к каталогу c: на обеих панелях.
5. В правую панель вывести каталог другого устройства:
6. Вывести содержимое левой панели в виде:
 - только имена; (Brief)
 - по убыванию размера; (Size)
 - в алфавитном порядке; (Name)
 - в алфавитном порядке расширений; (eXtension)
 - по дате создания; (tiMe)
 - с полной информацией; (Full)
 - краткий формат без сортировки. (Unsorted)
7. Создать каталог PRIMER.
8. Скопировать в полученный каталог текстовые файлы.
9. Удалить из каталога PRIMER все файлы.
10. Определить тип ЭВМ, дисплея, процессора, подключенных устройств и содержимое оперативной памяти ЭВМ.

ПРИЛОЖЕНИЕ

Особенности работы в средах Borland Pascal 7.0 и Pascal ABC

Pascal. ВИРТ Никлаус. Швейцария.1971 Turbo Pascal. Филип Кан. 1983 г.	Pascal ABC. Мехмат РГУ. 2002-2006 Станислав Малкович. База - Delphi Pascal
MS DOS	Windows
Ctrl+F9 запуск программы	F9 или ► на панели инструментов
Ctrl+Break+Break -останов	Stop - на панели инструментов или ☒
Выделение: Shift+клавиша, или ЛК	ЛК провести по фрагменту *
Ctrl+K C или V или H или Y	ПК через контекстное меню *
Вывод на отдельном экране вывода	Вывод также, но при открытом экране ввода
Сохранить файл: F10 - FILE - Save as...	Файл - Сохранить как ... или F2
Открыть файл F3	F3
Текстовый и числовой режим: Uses CRT;	Uses CRT;
Графический модуль: Uses Graph;	Uses GraphABC;
	Очистка окна вывода: Ctrl+Del или ☒
Integer: -32768 .. 32767	нет
longint:	integer: -2147483648 .. 2147483647 4 байта
real: 2.9E-39 .. 1.7E38	нет
double	real: 5E-324 .. 1.7E308 8 байт
Пауза delay(ms);	sleep(ms); delay(ms);
Очистка окна вывода ClrScr;	ClrScr; или Cls;
Режим замена имеется Insert	нет
Возведение x^y :real; exp(y*ln(x))	power(x,y)
	FileExists(name) - true, если файл на диске
Ожидание нажатия: readkey; или readln;	Readln; readkey - функция
Выход из среды Alt+X	Закрыть стандартное окно Windows ☒

*ЛК – левая кнопка мыши.

*ПК – правая кнопка мыши.

Программа *.pas исполняется после
входа в среду и вызова из среды
редактора
данного файла в окно редактора.

Щелчок по имени файла запускает среду
и считывание текста файла в окно

КОМАНДЫ ГРАФИЧЕСКОГО МОДУЛЯ

SetColor(C); -цвет линии, шрифта	SetPenColor(CIBrown); -цвет линии;
	SetFontColor(CIBlue); -цвет шрифта;
SetFillStyle(P,C); -цвет фона	SetBrushColor(CIPurple); -цвет фона;
SetLineStyle(Ls,Pa,Th); -стиль линии	SetPenStyle(psClear); -стиль пера;
	(psSolid, psDash, psDot, psDashDot,..)
	SetPenWidth(sz); - толщина линии;
SetTextStyle(F,D,Sz); -парам. шрифта	SetFontSize(sz); -размер шрифта;
	SetFontName('Times'); - вид шрифта;
OutTextXY(X,Y,S); - вывод текста	TextOut(X,Y,S); - вывод текста
PutPixel(X,Y,C); вывод точки	SetPixel(X,Y,C);
GetX, GetY – координаты точки;	GetPixel(X,Y) – координаты точки;
MoveTo(X,Y);	MoveTo(X,Y); не рисуй до точки;
Line(X1,Y1,X2,Y2); линия	Line(X1,Y1,X2,Y2); линия
LineTo(X,Y); - линия до точки с координатами X, Y	
Rectangle(X1,Y1,X2,Y2); прямоугольник	Rectangle(X1,Y1,X2,Y2);
	RoundRect(X1,Y1,X2,Y2,W,H); w,h - оси
Bar(X1,Y1,X2,Y2); полоса	FillRect(X1,Y1,X2,Y2); -закраска текущим цветом
FloodFill(X,Y,B); B- цвет границы	FloodFill(X,Y,C); - закраска цветом C
Circle(X,Y,R); - окружность	Circle(X,Y,R); - окружность
Ellipse(X,Y,Sa,Ea,Xr,Yr); эллипс	Ellipse(X1,Y1,X2,Y2);
Arc(X,Y,a1,a2,R); - дуга (градусы)	Arc(X,Y,R,a1,a2);
Sector(X,Y,Sa,Ea,Xr,Yr); сектор эллипса	Pie(X,Y,R,a1,a2); - сектор окружности
PieSlice(X,Y,Sa,Ea,R); - закрашенный сектор	
FillEllipse(X,Y,Xr,Yr); -закрашенный эллипс	Chord(X,Y,R,a1,a2); - сегмент;
	LoadWindow('* .bmp'); -включить картинку
	ClearWindow; – очистка окна текущим цветом
	MaximizeWindow; - максимальное окно

СПИСОК ЦВЕТОВ:

0	чёрный	clBlack
1	синий	clNavy
2	зелёный	clGreen
3	голубой	clBlue
4	красный	clMaroon
5	фиолетовый	clPurple

6	коричневый	<code>clBrown</code>
7	светло-серый	<code>clLtGray</code> (<code>clMedGray</code> – серый)
8	тёмно-серый	<code>clDkGray</code>
9	ярко-синий	<code>clSkyBlue</code>
10	ярко-зелёный	<code>clLime</code>
11	ярко-голубой	<code>clAqua</code> - бирюзовый
12	розовый	<code>clFuchsia</code> - сиреневый
13	малиновый	<code>clRed</code>
14	жёлтый	<code>clYellow</code>
15	белый	<code>clWhite</code> (<code>clCream</code> – кремовый)

ЛИТЕРАТУРА

1. Р. Форсайт «Паскаль для всех» Москва. Машиностроение. 1986 г.
2. Г. Джонстон «Учитесь программировать» Москва. Финансы и статистика. 1989 г.
3. В.С. Новичков, Н.И. Парфилова, А.Н. Пылькин «Алгоритмические языки в техникуме. Паскаль» Москва. Высшая школа. 1990 г.
4. Е.А. Зуев «Язык программирования Turbo Pascal 6.0» Москва. «Унитех» 1992 г.
5. М.В. Дмитриева, А.А. Кубенский «Элементы современного программирования» Санкт-Петербург изд-во С-Петербургского университета. 1991 г.
6. В.В. Фаронов «Основы турбо-паскаля» МВТУ-Фесто Дидактик Москва. 1992 г.
7. А.М. Епанешников, В.А. Епанешников «Программирование в среде Turbo Pascal 7.0» Москва. Диалог МИФИ.1995 г.
8. С.А. Абрамов, Г.Г. Гнездилова, Е.Н. Капустина, М.И. Селюн «Задачи по программированию» Москва. «Наука». 1988 г.
9. В.Н. Пильщиков «Сборник упражнений по языку Паскаль» Москва. Наука. 1989 г.
10. Л.З. Шауцукова «Информатика 10-11» Просвещение. 2000 г.
11. С.А. Абрамов, Е.В. Зима «Начала информатики» Москва. Наука. 1989 г.
12. П.Семакин И.Ф. Основы программирования: учебник / И. Ф. Семакин, А. П. Шестаков. - М.: Мастерство, 2002.
13. Фаронов В.В, Turbo Pascal 7.0. Начальный курс: учебное пособие / В.В. Фаронов. - М.: Нолидж, 2000.
14. Васильев П.П. Турбо Паскаль в примерах и задачах. Освой самостоятельно: учебное пособие / П. П. Васильев. – М.: Финансы и статистика, 2003.
15. Макарова Н.В., Николайчук Г.С., Титова Ю.Ф. под ред. Макаровой Н.В. Информатика и ИКТ (базовый уровень) 10 Питер Пресс.
16. Макарова Н.В., Николайчук Г.С., Титова Ю.Ф. под ред. Макаровой Н.В. Информатика и ИКТ (базовый уровень) 11 Питер Пресс.
17. Семакин И.Г., Хеннер Е.К. Информатика и ИКТ (базовый уровень) 10-11 БИНОМ.
18. Фиошин М.Е., Рессин А.А., Юнусов С.М. под ред. Кузнецова А.А. Информатика и ИКТ (профильный уровень) 10-11 Дрофа.
19. Гейн А.Г., Ливчак А.Б., Сенокосов А.И. и др. Информатика и информационные технологии (базовый и профильный уровни) 10 Просвещение.