**ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ**

**Государственное образовательное учреждение высшего профессионального образования** «**Пензенский государственный университет**»



**Д**. **В**. **Такташкин**, **А**. **Н**. **Ракова**, **Л**.**В**. **Гурьянов**

**Информатика**.

**Язык программирования** Turbo Pascal.

**Методические указания к лабораторным работам**

**по программированию на языке** Turbo Pascal

**Пенза**



**Издательство**

**Пензенского государственного**

**университета**

2009

**УДК** 681.3

**Ш**55

**Р е ц е н з е н т ы**:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Такташкин, Д. В.** |  |  |
| **Ш**55 | **Информатика**. | **Язык** | **программирования** |
|  | Pascal : **Методические указания к лабораторным работам** | | |
|  | **по программированию** | **на языке** | Turbo Pascal / |
|  | **Д**. **В**. **Такташкин**, **А**. **Н**. **Ракова**, **Л**. **В**. **Гурьянов**. – **Пенза** : | | |
|  | **Изд**-**во Пенз**. **гос**. **ун**-**та**, 2009.-85 **с**. | |  |

**Рассматриваются элементы диалоговой среды** Turbo Pascal, **структура про**-**грамм на языке** Turbo Pascal, **основные типы данных**, **преобразования типов данных и действия над ними**, **операторы языка**, **работа с массивами данных**, **работа с процедурами и функциями**. **Приведены примеры реализации про**-**грамм**. **Предложены задания для выполнения лабораторных работ**.

**Учебное пособие подготовлено на кафедре** «**Математическое обеспечение и применение ЭВМ**» **и предназначено для изучения курса информатики**.

**УДК 681.3**

© **Такташкин Д**. **В**., **Ракова А**. **Н**., **Гурьянов Л**. **В**., 2009

© **Издательство Пензенского государственного** **университета**, 2009

2

**ОГЛАВЛЕНИЕ**

**Введение** 4

**Общие сведения**. **Пакет программ** Turbo Pascal 5

**Лабораторная работа №** 1. **Программирование**

**алгоритмов линейной структуры** 8

**Лабораторная работа №** 2. **Программирование**

**алгоритмов разветвленной структуры** 16

**Лабораторная работа №** 3. **Организация циклов** 24

**Лабораторная работа №** 4. **Работа с одномерными**

**массивами** 39

**Лабораторная работа №** 5. **Работа с двухмерными**

**массивами** 50

**Лабораторная работа №** 6. **Процедуры и функции** 61

**Лабораторная работа №** 7. **Файлы и их применение в**

Turbo Pascal 75

**Рекомендованная литература** 86

3

**Введение**

* **настоящее время средства вычислительной техники используют**-**ся практически во всех сферах деятельности человека**, **и овладение навыками программирования этих вычислительных средств на одном из языков высокого уровня является обязательным элементом образо**-**вания и культуры каждого инженера**.

**Наибольшее распространение находят языки программирования**,

**имеющие высокий уровень формализации и позволяющие создавать хорошо структурированные программы**, **предназначенные для реше**-**ния обширного класса задач из самых разнообразных сфер инженер**-**ной деятельности**. **К таким языкам относятся** Turbo Pascal, C/C++ ,Modula, Ada **и т**. **д**.

**Одним из простых для первоначального изучения**, **но в тоже время достаточно универсальным языком программирования является** TurboPascal. **Данный язык**, **названный в честь французского математика и** **философа Блеза Паскаля** (1623-1662), **был создан как учебный язык программирования в** 1968-1971 **годах швейцарским ученым Никласом Виртом на кафедре информатики Стэнфордского университета** (**Цю**-**рих**). **В настоящее время** Turbo Pascal **имеет более широкую сферу применения**, **чем предусматривалось при его создании**. **Этот язык от**-**личается простотой понимания**, **стройностью и структурностью алго**-**ритмов**, **быстротой компилятора и удобными средствами создания и отладки программ**.

* **настоящий момент пользуются популярностью такие версии языка как** TMT Pascal, Free Pascal **и** GNU Pascal. **Продолжает исполь**-

**зоваться и** Turbo Pascal. **Развитием языка** Turbo Pascal **является** ObjectPascal – **версия языка Паскаль расширенная средствами объектно**-**ориентированного программирования**. **Последние версии** Turbo Pascal **лежат в основе среды программирования** Delphi.

4

**Общие сведения. Пакет программ Turbo Pascal**

**Пакет программ** Turbo Pascal **содержит среды для разработки про**-**грамм версий под** DOS **и под** Windows. **В пакет программ входят про**-**граммы компиляции**, **отладки**, **а также дополнительные утилиты**.

**Программная среда Turbo Pascal**

**Для запуска среды** Turbo Pascal **необходимо запустить bp.exe из папки BIN**.

**Верхняя строка представляет собой меню системы** , **доступ к кото**-**рому осуществляется мышью или нажатием клавиши F10**. **Используя мышь или клавиши курсора пользователь может выбрать нужный пункт меню**. **Некоторые команды меню доступны через** «**горячие кла**-**виши**». **Например**, **для выбора пункта меню File→Open достаточно нажать клавишу F3**.

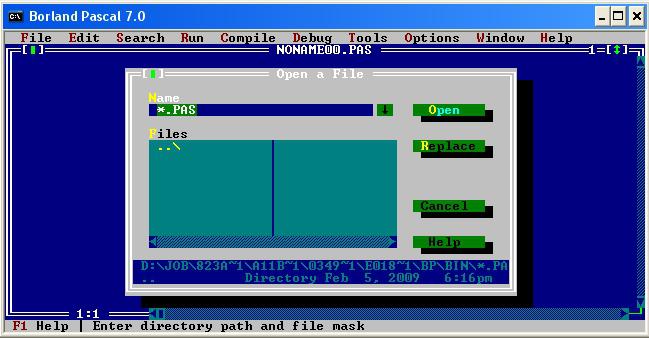
**Текст программы набирается в окне редактора** (**см**. **рисунок** 1), **на краях которого представлены следующие элементы**:

– **имя файла программы на диске**;

– **элементы управления окном мышью**;

– **полосы прокрутки**;

– **координаты положения курсора в тексте программы**.



**Рис**. 1 – **Интерфейс среды** Turbo Pascal

**Меню среды Turbo Pascal**

**Основное меню среды содержит следующие команды**:

**File** – **набор команд для работы с файлами** (**открытие**, **сохранение программы**);

5

**Edit** – **набор команд для работы с текстом**; **Search** – **команды поиска по тексту программы**;

**Run** – **команды**, **управляющие выполнением программы**;

**Compile** – **команды компиляции программы**, **т**.**е**. **получение испол**-**няемого файла**;

**Debug** – **команды отладки**;

**Tools** – **дополнительные утилиты**;

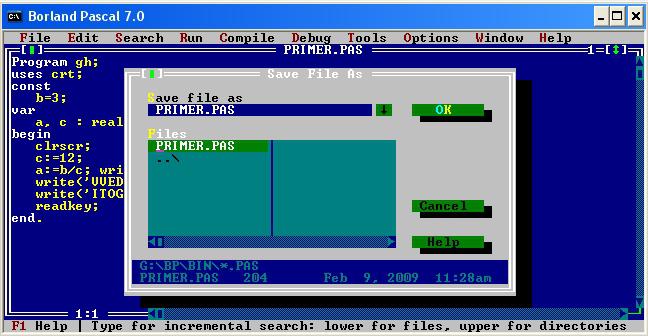
**Options** – **настройки среды**;

**Window** – **команды навигации между открытыми программами**; **Help** – **помощь**.

**Для использования помощи можно воспользоваться сочетанием клавиш Ctrl+F1** – **помощь по ключевому слову**, **Alt+F1** – **предыдущая справка**.

**Сохранение программы на диске**

**Набранный текс программы можно сохранить на диске**, **нажав кла**-**вишу F2 или выбрав пункт меню File→Save**. **Появится окно сохране**-**ния программы** (**см**. **рисунок** 2).



**Рис**. 2 – **Сохранение программы в среде** Turbo Pascal

* **появившемся окне следует выбрать каталог и ввести имя файла программы**, **под которым она будет сохранена на диске и нажать** OK.

**Запуск и отладка программы**

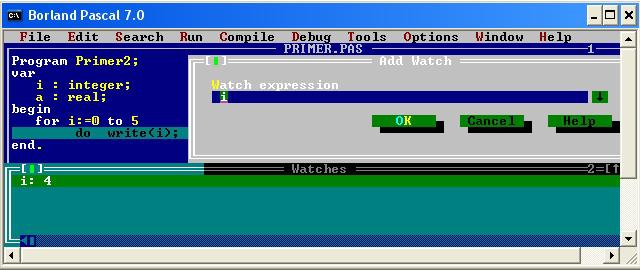
**Для запуска программы необходимо ее скомпилировать**, **т**.**е**. **полу**-**чить исполняемый** (**exe**) **файл**. **Это делается командой меню Compile→Make или клавишей F9**. **Запуск программы** – **командой ме**-**ню Run→Run** (**сочетание клавиш Ctrl+F9**).

6

* + **процессе компиляции программы выявляются синтаксические ошибки**. **При наличии таких ошибок среда сообщает тип ошибки и указывает место в тексте программы**, **где она присутствует**. **В случае затруднения пользователь может посмотреть помощь через пункт ме**-**ню Help→Topic Search или используя сочетание клавиш Ctrl+F1**.
  + **программах сложной структуры для выявления ошибки исполь**-**зуется отладка**. **Используя сочетание клавиш Ctrl+F8**, **на строке с кур**-**сором устанавливается контрольная точка** (**Breakpoint**). **В процессе выполнения программы на контрольной точке будет произведена ос**-**тановка выполнения программы и переход в режим ее отладки**. **Воз**-**можно два вида выполнения команд строки**: **F7** – **выполнение команд подпрограмм пошагово и F8** – **выполнение команд подпрограмм за один шаг**.

**На любом шаге возможен просмотр значений переменных**: **при на**-**жатии Ctrl+F7 появляется окно**, **в котором следует ввести имя интере**-**сующей переменной**. **При нажатии** OK **в отдельном окне появится имя**

* **значение этой переменной**. **При выполнении программы изменения значений выбранных переменных автоматически обновляются** (**см**. **рисунок** 3).



**Рис**. 3 – **Отладка программы в среде** Turbo Pascal

**При нажатии клавиши F9 выполнение программы будет продолже**-**но**. **Для просмотра результата работы программы следует использо**-**вать сочетание клавиш Alt+F5**. **Для прекращения работы программы следует использовать сочетание клавиш Ctrl+F2**.

**Для завершения работы со средой** Borland Pascal 7.0 **следует вос**-**пользоваться пунктом меню File→Exit или сочетанием клавиш Alt+X**.

7

**Лабораторная работа № 1. Программирование алгоритмов линейной струк-туры**

**Цель работы**: **изучение структуры программы на языке** TurboPascal; **типов данных**; **составление простых неразветвленных про**-**грамм**.

**Теоретические сведения**

**Программа на языке** Turbo Pascal **состоит из нескольких блоков**. **Порядок следования блоков определяется использованием их друг другом**.

**Блок объявления констант начинается служебным словом const**.

**После чего указывается имя константы**, **знак равенства и ее значение**.

* Turbo Pascal **допускается использование констант в виде выраже**-**ний**. **Пример объявления констант на языке** Turbo Pascal:

const a=5; b=7/(4+2.6);

c='this is constant';

**Блок объявления переменных начинается служебным словом var**. **Переменные перечисляются через запятую**, **в конце списка ставится двоеточие**, **после которого указывается их тип**. **Пример объявления переменных на языке** Turbo Pascal:

var

a, b : integer;

1. : real; c : char;

**Основные стандартные типы данных языка** Turbo Pascal:

**boolean** – **булевская переменная**, **может принимать значение** true(**истина**) **или** false (**ложь**);

**integer** – **целочисленная переменная** (**значение от** –32768 **до** 32767); **real** – **переменная с плавающей точкой** (**значение по модулю от**

2.9e–39 **до** 1.7e38);

**double** – **переменная с плавающей точкой** (**значение по модулю от** 5.0e–324 **до** 1.7e308);

**char** – **переменная одного символа**;

**string** – **строковая переменная**.

8

**Блок программы начинается ключевым словом begin и заканчива**-**ется ключевым словом end**. **Эти же ключевые слова используются для группировки команд**. **В программе последовательно перечисляются команды языка**. **Текст программы заканчивается символом** «**точка**».

**Основные операторы языка Turbo Pascal**

**Все команды языка** Turbo Pascal **разделяются символом** «**точка с запятой**» («;»).

**read**, **readln** – **ввод данных с клавиатуры** (**readln требует ввода данных через** Enter, **read** – **через пробел**), **в скобках перечисляются переменные**, **в которые будет осуществляться ввод данных**. **readln(x, y)**; **осуществляет ввод значения переменной** x, **затем** y;

**write**, **writeln** – **вывод текста и значений переменных на экран** (**writeln**, **в отличие от** **write**, **осуществляет переход на следующую** **строчку по завершении выполнения команды**), **write(x, ‘Абвгд’, y, ‘\_’, 2)**; **при значениях x = 4**, **y = 7 выводит на экран** «**4Абвгд7\_2**»;

**:=** – **оператор присваивания**. **x := 5 + 7**; **устанавливает значение пе**-**ременной x = 12**;

**Комментарии в программе записываются внутри фигурных скобок**

**{ комментарий } или внутри символов (\* комментарий \*)**.

**Математические операции языка Turbo Pascal**

|  |  |  |
| --- | --- | --- |
| \* | – **умножение** | (2\*3 **результат**: 6); |
| / | – **деление** | (30/2 **результат**: 1.5**Е**+01); |
| + | – **сложение** | (2+3 **результат**: 5); |
| – | – **вычитание** | (5–3 **результат**:2); |
| **div** | – **целочисленное деление** | (5 div 2 **результат**: 2); |
| **mod** – **остаток от деления** | | (5 mod 2 **результат**: 1). |

|  |  |  |  |
| --- | --- | --- | --- |
| **Операции отношения** | |  |  |
| > | – **больше**; | <> | – **не равно**; |
| < | – **меньше**; | >= | – **больше или равно**; |
| = | – **равно**; | <= | – **меньше или равно**. |

**Основные математические функции**

**Abs(Х)** – **возвращает абсолютное значение числа Х**;

**Sin(X), Cos(X)** – **возвращает синус и косинус числа Х**, **где Х** – **угол в радианах**;

**Exp(X)** – **возвращает число**, **равное** ex;

9

**Ln(X)** – **возвращает число**, **равное натуральному логарифму от числа Х**;

**Sqr(X)** – **возвращает число**, **равное квадрату числа Х**;

**Sqrt(X)** – **возвращает число равное квадратному корню из числа Х**;

**Frac(X)** – **возвращает число равное дробной части числа Х**;

**Int(X)** – **возвращает число**, **равное целой части числа Х**;

**Random(X)** – **возвращает случайное число в диапазон** [0,X-1].

**Содержание лабораторной работы**

* **лабораторной работе требуется разработать программу на языке программирования** Turbo Pascal, **которая вычисляет математическое выражение в соответствии с Вашим вариантом**. **В заданном выраже**-**нии переменная** *x* **задается пользователем**. **Отчет о проделанной ра**-**боте должен содержать**: **название и цель работы**; **номер варианта для выполнения задания и условие своего варианта**; **блок**–**схему решения задачи**; **тексты программ**; **полученные при расчетах численные ре**-**зультаты и выводы по проделанной лабораторной работе**.

**Варианты заданий**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | *x* cos2 | | | | | | *x* | | | |  |  |  |  |  |  |  |  |  |  |  |  | *tg*3 *x* | | | | | | | | | | | | | | |  |
|  |  |  |  |  | + 3 | |  |  |  |  | *x* | |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| 1. | *y* = |  |  | 2 | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | . | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | 1+ *e* | | | | | | | | -*x*2 | | | |  |  |  | | | |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | 2 | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 18,3*e* | | | | | | -*x*2 | | | | | | | - 6,3*e* | | | | | | | | | | | -*x* | | | | | | | | | | | | | |  |
| 2. | *y* = | 2 | | |  |  |  |  | 3 | | | | | |  |  |  | , **где** *x* ¹ 0 . | | | | |  |
|  |  | 7,4*tg* | | | | | | | *x* | | | | - 3 | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | 2*x* | | | | | | | | |  |  |  |  | |  |  | |  |  |  |
|  |  |  |  | 2 | | | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  | *x*2 | |  | | |  |  |  |  |  |  | |  | | | | | | | | | | |  |  | |  |  |  | | |  | |  |  |  |
|  | *y* = *e* | | | - |  |  |  |  |  |  |  |  | *tg*3 *x*3 | | | | | | | | | | | | |  |  | |  |  |  | | |  | |  |  |  |
|  |  | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| 3. | 2 + 3 | | | | | | |  |  |  |  | | |  | | | | |  | | | |  |  |  | . | | | | | | |  | |  |  |  |
|  |  | 1+ | | | |  | | | | |  |  |  |  |  |  |
|  |  | *x* | | | |  |
|  |  |  |  |  |  | *e*-*x*2- | | | | | | | | | 4 |  | + cos | | | | | | | | | | | | | | | | | | | *x* | |  |  |  |
|  |  |  |  |  | *x* |  |
|  |  |  |  |  |  | |  |
| 4. | *y* =ln | | | | |  | | | |  |  |  |  |  |  |  |  |  |  |  | | | | | | | | | | | | | 2 | | | | | | . |  |
|  | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  | | | | | | | | | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  | 1 + *tg* 2 *x*2 | | | | | | | | | | | | | | | | | | | | |  |  | | | | | |  |  |



10

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | *x*2 |  |  |  | - *x* cos2 | | | | | | | | *x* | |  |  |  |  |  |  |
|  |  | *x* | |  |  |  |  |  |  |
| 5. | *y* = |  | 2 | | | . |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 4 | | 2 | |  | |  |  | 1+ *e*ln *x* | |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | | |  | | | | | | | | | *x* | |  | |  |  |  |  |
|  |  | *x* ln3 *x*3- | | | | | | | | sin 2 | | | | -1 | |  |  |  |  |
| 6. | *y* = | 2 | | , **где** *x* | > 0. |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |



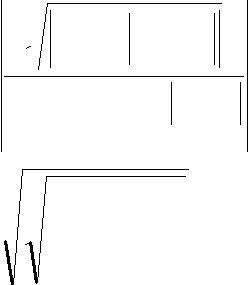
*e*sin *x*-*x*2 +1

1. *y* =ln *e*(-*a*2+*b*)+2*ab* ,**где**a, b > 0.

2sin 2 *ab* + *b*

* 1. - ln *x*2 -1

1. *y* = *tg*3 *x*3 + 2 ln *x* +1 , **где** *x* ¹ ±1 .



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 9. | *y* = | 4 | | 3 | |  |  | 1- *tg*3 *x*3 | | | | | | | | | | | | | | | | | . |  |  |  |  |  |  |  |  |  |
|  |  |  |  | 1+ *e*sin2 *x* | | | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  | |  |  | |  |  |  |  |  |  |  |  |  |  |  | | |  |  |  |  |  |  |  |  |  |
| 10. | *y* =ln | | | |  | *x*2 | | | | | | |  | + 3 |  |  |  |  |  |  |  | *e*(-1+*x*) | | | |  |  |  | , **где** x **≠** 0. | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
|  | 2*x* | | | | | |  |  |  |  | *tg* 2 *x*2+1 | | | | | | | | | | |  |  |
|  |  |  |  |  |  |  |  | 1 | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | 5 | | *e* | | | cos2 *x*+*tg* 2 | | | | | | | *x* | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 11. | *y* = | |  |  |  |  |  |  |  |  | 2 | |  |  |  |  | +1. | | | |  |  |  |  |  |  |  |  |  |
|  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 3 | |  | *tg* 2 *x*2+1 | | | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | |  |  |  |  |  |  | | |  |  |  |  |  |
|  |  | 3 | | |  |  |  | 6 - sin | | | | | | | |  |  |  | *x* | | |  |  |  | + 3*e* | -*tg* 2 *x* | | | | | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  | 2 | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |
| 12. | *y* = |  |  | 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | . | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  | 2 ln 2 (*x*2 +1) | | | | | | | | | | | | | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 13. | *y* =ln | | | | 5 | | |  |  |  |  |  |  | *tg* 3 *x*3-1 | | | | | | | | | | | |  |  |  |  |  |  | . | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | 2*e*-*x*2 | | | | |  | + 2 cos *x* | | | | | | | | | | | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |



11

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  | 4 - 3 | | | | | | | | | | |  |  |  |  |  |  |  | |  |  |  |  | *x* | | | - 30 | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  | cos3 | | | | | | | | | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 14. | *y* =2 ln | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 3 | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | , **где** *x* ¹ k**π**, k **€** z. | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1 | | | |  | +*x*2 | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | *etgx* | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  | |  | |  |  | |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 15. | *y* = | 3 | |  | cos | | | | | | |  |  |  |  |  |  |  |  |  | ln *x*2 -1 | | | | | | | | | | | | | | | | | |  |  |  | *x*) | | | | | | |  |  |  | , **где** *x* ¹ 0 . | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 1+ *ex*2 | | | | | | | | | | | | |  | - *tg*(3 | | | | | | | | | | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 1- ln | | | | | | | | | | | 3 | | |  |  | *x* | | | | |  |  |  |  |  |  |  |
| 16. | *y* =2*etgx* +*x*2 | | | | | | | | | | | | | | | |  | + ln | | | | | | | |  |  |  |  |  |  |  | , **где** *x* ¹ k**π**, k **€** z. | | |  |
|  |  |  |  |  | 2 | | | | | | |  |  |  |  |  |
|  |  |  |  |  | 1 | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | |  | |  |  | |  |  |  |  | |  |  |  |  |  |  |  |  |  | | | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 4 *tg* 2 | | | | | | | | | |  | *x* | | | +1 | | | | | | | | | | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 2 | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | 3 | | | 4 |  |  | 2 sin | | | | | | | | | | | | *x*2 | | | | | | + *tg* | | | | | | | | 2 | |  |  | *x* | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 17. | *y* = |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | , **где** *x* ¹ - | | | | | | | | 1 | . |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 2*x* +1 | | | | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 2 | | | | |  |  |
|  |  |  |  |  | æ cos3 (*x*3 )- 3 | | | | | | | | | | | | | | | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  | ö | | | |  |  |  |  |  |  |  |  |
|  |  |  |  |  | *tg* 2 *x* | | | | | | | | | | |  |  |  |  |  |  |  |  |
| 18. | *y* =5 | | |  | ç | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ÷ | | | |  | , **где** *x* > -2. | | | | | |  |
|  |  |  |  | -*x* | | | | | 2 |  |  | + 2 ln(*x* + 2) | | | | | | | | | | | | | | | | | | | | | | | | | |  |  |
|  |  |  |  |  | ç | |  |  |  |  |  |  | ÷ | | | |  |  |  |  |  |  |  |  |
|  |  |  |  |  | è *e* | | | | |  |  |  |  |  |  |  |  | ø | | | |  |  |  |  |  |  |  |  |
|  |  |  |  | 3,5*e* | | | | | | | |  | *x*2 | | | |  | - 8,3*e* | | | | | | | | | | | | *x*2 | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 19. | *y* = |  |  | 3 | | | |  |  | 4 | | |  |  |  |  | , **где** *x* ¹ 0 . | | | | | | | | | | | | | | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  | *x* | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | 3,3*tg* | | | | | | | | | | - 3 8,8*x* 2 | | | | | | | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  | | |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  | 6 | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  | *x* | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  | ln | | | | 2 |  |  |  | *x* | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 20. | *y* =3 | | |  | *tg*2 | | | | | |  | + 35 | | | | | | |  |  |  |  |  |  |  |  |  | 2 | | |  |  |  |  |  |  |  |  | , **где** *x* ¹ 2k**π**, k **€** z. | | | | | | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | æ *x* ö | | | | | | | | | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  | 2 | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 2*tg*ç | | | | | | | | | |  |  |  |  | ÷ | | |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 2 | | |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | è | | |  | ø | | |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  | | | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 21. | *y* =5 | | |  | ln | | |  |  |  |  |  |  |  |  |  |  |  | *x*2+5 | | | | | | | | | | | | | | |  |  |  |  |  |  |  | . | | |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | sin | | | | | | | |  | 2 | *x* | | | + cos2 | | | | | | | | | | | | | | | *x* | | | |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 2 | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |



12

*e*-3 *ax*+*x*2

*,* ***где*** *x ¹ 0 .*

1. *y* =3 5 1- 3sin 2 2*x* + 2*e*-*tg*3*x* .

**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | *x* | - | | 2cos | | | | 3 | |  | 1- |  |  | *x*2 | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| 23. | *y* =ln | |  |  | 3 | |  |  |  |  | 2 | | |  |  | . |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | 4 | |  |  | 5- *e*-*x*2cos *x* | | | | | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  | *e*-*tg* | *x* | | | |  |  |  |  |  |  |  |  |  |
| 24. | *y* = |  |  |  |  |  |  |  |  |  | 2 | |  |  |  |  |  |  | , **где** *x* ¹ ±1 , *x* ¹ *p* + 2*pk* , k **€** z . | | | |  |
| sin | | | | |  |  |  |  | 1- *x*2 | | | | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | cos3 *x*3 | | | | | |  | + 2 | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**

ln *x* - *e*- *x*2 + 3 *x* 25. *y* = *tg* 2 *x*3 - *x*2 + 3

**

26. *y* =2(ln *x* -4 *x* +1),**где** *x* ¹0.

**

+ 2*tgx*4

**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *y* = | 5 | | 1 + | | | | | 4 | *tg* | | | | | |  | *x*2-2 | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 27. |  |  |  |  |  |  |  | + ln(*x*2 ) , **где** *x* ¹ 0 . | | | | | | |  |
| *e* | |  | *x* |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |
| 28. | *y* = |  |  |  |  |  |  |  |  | *e*-*ax*2 | | | | | | | |  |  |  |  |  | . |  |
|  | ln | | 3 |  | sin 2 *x*2 - cos2 *x* | | | | | | | | | | | | | | |  |  |
|  |  |  |  |  |  |  |  |  |  |  | *e*-*x*2 | | | | |  |  |  |  |  |  |  |
| 29. | *y* =ln | | | | 4 |  | *x*2- *x*2 | | | | | | | | |  | + 5*e* | -*tg*3*x*3 | | | | | . |  |
|  |  |  |
|  |  |  |  | *e*-*x*2 | | | | | | |  |  |  |  |  |  |  |

sin(*x*3 +1)

30. *y* = *tg* 4 ( *x* -1)+ *e*- *x*2.

**

13

**Пример программы на языке Turbo Pascal**

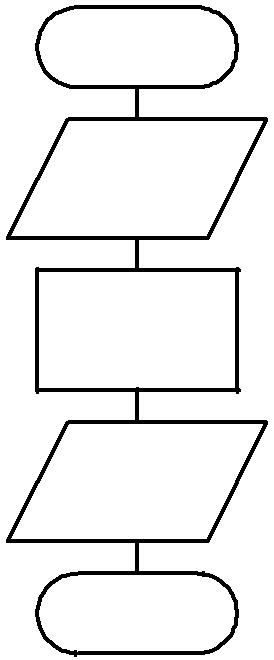
**Разработать алгоритм и составить программу вычисления значения**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **выражения** *y*= |  | *e*2 *x* | | |  |  |  |  | . **Вывести исходные данные и результат** |  |
| æ |  |  |  |  |  |  | ö |  |
|  | *x* | 2 | -1 |  |  |  |
|  | sinç |  |  |  | ÷ | |  |  |
|  | è |  |  |  |  |  | ø | |  |  |
|  |  |  |  |  |  |



**с пояснительным текстом**.

**Схема алгоритма работы программы представлена на рисунке** 4.



**Начало**

**Ввод**

**значения**

**переменной**

**Х**

y:=(exp(2\*x))/

sin(sqrt(abs(sqr(x)-

1)))

**Вывод на**

**экран**

**значения** Y

**Конец**

**Рис**. 4 – C**хема алгоритма работы программы**

Program Primer1;

var

x, y : real;

begin

write('**Введите значение переменной Х** ');

readln(x);

y:=exp(2\*x))/sin(sqrt(abs(sqr(x)–1));

writeln('**Значение выражения**');

writeln('y=exp(2\*x))/sin(sqrt(abs(sqr(x)–1))');

writeln('**равно** ', y:8:4);

end.

14

**Контрольные вопросы**

1. **Как пользоваться меню среды** Turbo Pascal?
2. **Что отображается в окне редактора среды** Turbo Pascal?
3. **Какие основные функции меню среды** Turbo Pascal?
4. **Как сохранить программу на диске**?
5. **Что такое компиляция программы**?
6. **Как запустить программу на выполнение**?
7. **Что такое синтаксическая ошибка программы**?
8. **Как находить синтаксические ошибки программы**?
9. **Как объявляются переменные в языке** Turbo Pascal?
10. **Какие основные типы данных в языке** Turbo Pascal?
11. **Как группируются команды**?
12. **Как пользоваться комментариями**?
13. **Как ввести данные с клавиатуры**?
14. **Как вывести данные на экран**?
15. **Что такое присваивание**?

15

**Лабораторная работа № 2. Программирование алгоритмов разветвленной структуры**

**Цель работы**: **изучение составного оператора**, **оператора условия**, **оператора выбора**, **оператора безусловного перехода и составление программ с разветвленной структурой на языке** Turbo Pascal.

**Теоретические сведения**

**Программы с разветвленной структурой используется**, **когда в ал**-**горитме решения задачи предусмотрены альтернативные пути ее ре**-**шения**, **т**.**е**. **из нескольких альтернатив выбирается одна** , **в зависимости от условия** (**условием является логическое выражение**).

**Составной оператор**

**Составной оператор** – **это последовательность произвольных опе**-**раторов программы**, **заключенная в операторные скобки** – **зарезерви**-**рованные слова** begin … end. **При этом после** end **ставится точка с запятой** (;). **Язык** Turbo Pascal **не накладывает никаких ограничений на характер операторов входящих в составной оператор**. **Среди них мо**-**гут быть и другие составные операторы**.

begin

………..

begin

………..

end;

………..

end;

Pascal **допускает произвольную глубину их вложения**.

**Условный оператор**

**Условный оператор позволяет проверить некоторое условие и в за**-**висимости от результатов проверки выполнить то или иное действие**. **Таким образом**, **условный оператор** – **это средство ветвления вычис**-**лительного процесса**. **Структура условного оператора имеет следую**-**щий вид**:

if <**условие**> then <**оператор**\_1> else <**оператор**\_2>;

**где** if, then, else– **зарезервированные слова** (**если**, **то**, **иначе**);<**условие**> – **произвольное выражение логического типа**;

16

<**оператор**\_1>, <**оператор**\_2>– **любые операторы языка** Turbo Pascal.

**Условный оператор работает по следующему алгоритму**. **Вначале вычисляется условное выражение** <**условие**>. **Если результат есть** TRUE (**истина**), **то выполняется** <**оператор**\_1> , **а** <**оператор**\_2> **пропускается**; **если результат есть** FALSE (**ложь**), **наоборот**,<**опера**-**тор**\_1> **пропускается**, **а выполняется** <**оператор**\_2>. **Часть** else<**оператор**\_2> **условного оператора может быть опущена**. **Тогда при** **значении** TRUE **условного выражения выполняется** <**оператор**\_1>, **в противном случае этот оператор пропускается**.

**Оператор выбора**

**Оператор выбора позволяет выбрать одно из нескольких возмож**-**ных продолжений программы**. **Параметром**, **по которому осуществля**-**ется выбор**, **служит *ключ выбора*** – **выражение любого порядкового типа** (**кроме типов** REAL **и** STRING). **Структура оператора выбора имеет вид**:

case <**ключ**\_**выбора**> of <**список**\_**выбора**> else <**операторы**> end;

**Здесь** case, of, else, end– **зарезервированные слова** (**случай**, **из**, **иначе**, **конец**);

<**ключ**\_**выбора**> – **ключ выбора**;

<**список**\_**выбора**> – **одна или более конструкций вида**:

<**константа**\_**выбора**> : <**операторы**>;

<**константа**\_**выбора**> – **константа того же типа**,

**что и выражение** <**ключ**\_**выбора**>;

<**операторы**> – **произвольные операторы** Pascal.

**Оператор выбора работает следующим образом**. **Вначале вычисля**-**ется значение выражения** <**ключ**\_**выбора**>, **а затем в последователь**-**ности операторов** <**список**\_**выбора**> **отыскивается такой**, **которому соответствует константа**, **равная вычисленному значению**. **Если в спи**-**ске выбора не будет найдена константа**, **соответствующая вычислен**-**ному значению ключа**, **управление передается операторам**, **стоящим за словом** else. **Часть** else <**операторы**> **можно опустить**. **Тогда при отсутствии в списке выбора нужной константы оператор выбора про**-**сто завершит свою работу**.

17

**Метка и оператор безусловного перехода**

**Метка в языке** Pascal – **это произвольной идентификатор**, **позво**-**ляющий именовать некоторый оператор программы и таким образом ссылаться на него**. **Метка располагается непосредственно перед поме**-**ченным оператором и отделяется от него двоеточием** (:). **Оператор можно помечать несколькими метками**, **которые в этом случае отде**-**ляются друг от друга двоеточием** (:). **Перед тем как появиться в про**-**грамме**, **метка должна быть описана**. **Описание меток состоит из заре**-**зервированного слова** label(**метка**), **за которой следует список меток**:

var

label lb1;

begin

goto lb1;

………..

lb1 : ………..

………..

**Действие оператора** goto **состоит в передаче управления соответст**-**вующему помеченному оператору**.

**При программировании следует избегать злоупотреблением опера**-**торами безусловного перехода**, **так как это затрудняет понимание про**-**граммы**, **делает ее запутанной и сложной в отладке**.

**Содержание лабораторной работы**

* + **лабораторной работе требуется разработать программу на языке программирования** Pascal 7.0 **с разветвленной структурой в соответст**-**вии с Вашим вариантом**. **Отчет о проделанной работе должен содер**-**жать**: **название и цель работы**; **номер варианта для выполнения зада**-**ния и условие своего варианта**; **блок**–**схему решения задачи**; **тексты программ**; **полученные при расчетах численные результаты и выводы по проделанной лабораторной работе**.

**Варианты заданий**

1. **С клавиатуры ведены два числа**. **Если первое число больше второ**-**го на** 2, **то увеличить оба числа на** 3, **в противном случае добавить к введённым числам** –1.
2. **С клавиатуры ведены два числа**. **Если первое число меньше вто**-**рого на** 5, **то к меньшему числу прибавить** 3, **а от большего отнять** 1, **в противном случае оба числа возвести в квадрат**.

18

1. **С клавиатуры введены два числа**. **Если числа равны**, **то найти их сумму**. **В противном случае вычислить модуль разности**.
2. **Если во введенных с клавиатуры трёх числах есть отрицательное число**, **получить произведение этих чисел**, **в противном случае** – **их сумму**.
3. **Если второе из двух введенных чисел** A **и** B **равно** 8, **то вычислить** A+B, **а если нет**, **то получить** A–B.
4. **Если введенное с клавиатуры натуральное число кратно** 3, **вычис**-**лить его квадрат**, **а если нет**, **то** – **корень квадратный из числа**.
5. **Найти минимальное из трёх введённых с клавиатуры чисел** A, B,C.
6. **Если первое из введенных с клавиатуры чисел А и В равно** 5, **то вычислить А**\***В**, **а если нет**, **то получить А**/**В**
7. **Если введённое с клавиатуры натуральное число** n – **чётное**, **то вычислить квадрат**, **куб и четвертую степень и отпечатать эти зна**-**чения**. **В противном случае выдать информацию** «**вычислять не будем**».
8. **Если в последовательности из трёх чисел**, **введённых с клавиату**-**ры**, **есть число** 7, **то вычислить произведение этих чисел**, **в про**-**тивном случае получить их сумму**.
9. **Найти максимальное число в последовательности трёх чисел А**, **В**, **С**.
10. **Даны четыре числа** A, B, C, D. **Найти минимальное**. **Результат хранить в ячейке** minimum.
11. **В последовательности четырёх чисел**, **заданных с клавиатуры**, **найти количество положительных чисел**.
12. **Составить программу**, **упорядочивающую три числа А**, **В**, **С так**, **что в ячейке А находится минимальное число**, **в ячейке В** – **сред**-**нее**, **а в ячейке С** – **максимальное**.
13. **С клавиатуры введены три числа**. **Найти минимальное из них**. **Ес**-**ли оно больше нуля**, **вычислить произведение всех трёх чисел**, **в противном случае вычислить сумму всех трех чисел**. **При выводе организовать соответствующий комментарий**.
14. **Даны четыре числа А**, **В**, **С**, D. **Найти максимальное и результат записать в ячейку с именем** maximum.
15. **Составить программу**, **которая бы находила максимальное из трех чисел А**, **В**, **С и**, **если оно больше** 5, **увеличивала бы все числа в два раза**.

19

1. **С клавиатуры заданы два числа**. **Если их среднее арифметическое превышает их среднее геометрическое на** 3, **уменьшить каждое число на** 1, **в противном случае увеличить оба числа на** 1.
2. **Даны три числа**. **Найти произведение двух минимальных из них**.
3. **С клавиатуры заданы три числа**. **Найти среднее арифметическое положительных из них**.
4. **В последовательности четырёх целых чисел**, **заданных с клавиа**-**туры**, **найти количество нулей**.
5. **Даны два прямоугольных треугольника**, **катеты которых равны** a1,a2, b1, b2, **соответственно**. **Определить номер треугольника с** **большей гипотенузой**.
6. **С клавиатуры введены три числа**. **Расположить их в порядке убы**-**вания в тех же ячейках**.
7. **С клавиатуры введены три числа**. **Если их среднее арифметиче**-**ское в** 3 **раза больше их среднего геометрического**, **уменьшить значение каждого из них в** 2 **раза**. **В противном случае к каждому числу добавить** 1.
8. **С клавиатуры заданы два числа**. **Если первое из них больше или равно второму**, **удвоить их значения**, **в противном случае умень**-**шить каждое из них на единицу**.
9. **Найти минимальное число в последовательности их трёх чисел А**, **В**, **С**.
10. **Если сумма двух чисел**, **введённых с клавиатуры**, **превышает** 30, **увеличить первое число в** 5 **раз**, **второе** – **в три раза**, **в противном случае определить модуль их разности**.
11. **Если модуль разности двух введённых с клавиатуры чисел пре**-**вышает** 0,1, **определить корень квадратный из модуля произведе**-**ния этих чисел**, **в противном случае увеличить значение каждого числа в** 2 **раза**.
12. **Если три введенных с клавиатуры числа** – **отрицательные**, **опре**-**делить их произведение**, **в противном случае определить их сред**-**нее арифметическое**.
13. **Если два первых среди введённых с клавиатуры трёх чисел поло**-**жительны**, **увеличить все числа в** 3 **раза**. **В противном случае оп**-**ределить произведение всех чисел**.

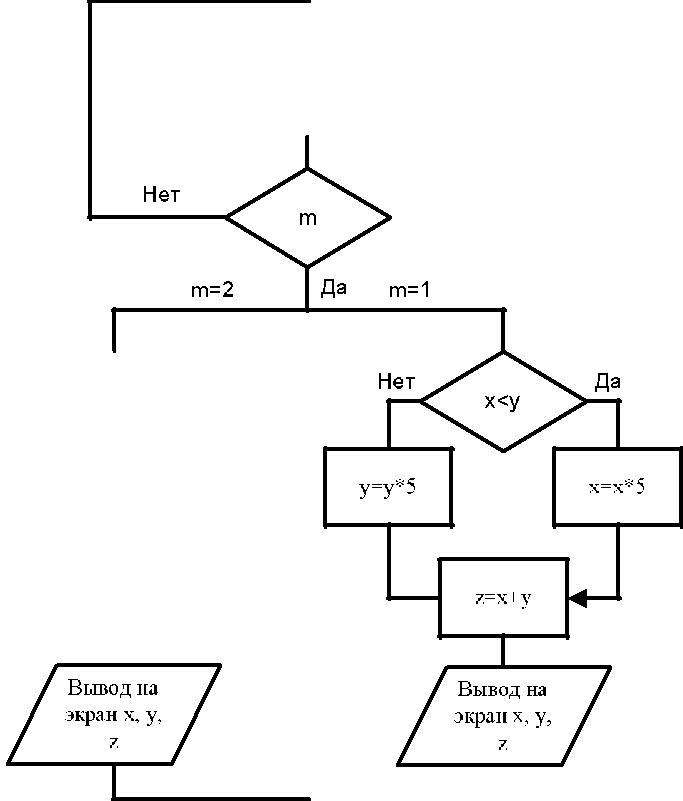
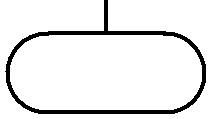
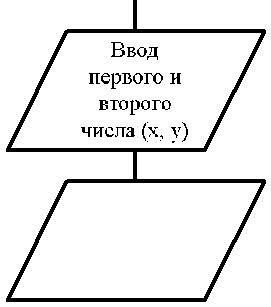
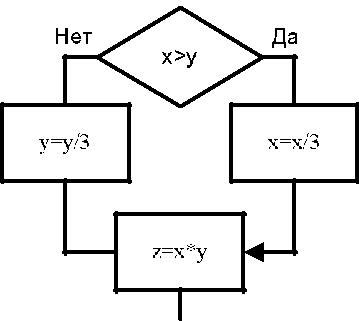
**Пример программы на языке Pascal**

**Разработать алгоритм и составить программу**, **в которой пользова**-**тель вводит с клавиатуры два числа**, **после чего ему предоставляется**

20

**следующий выбор**: **если первое число меньше второго**, **то увеличить наименьшее из введенных чисел в** 5 **раз и вычислить их сумму**, **в про**-**тивном случае**, **уменьшить наибольшее из них в** 3 **раза и вычислить их произведение**.

**Схема алгоритма работы программы представлена на рисунке** 5.



**Рис**. 5 – **Схема алгоритма работы программы**

21

Program Primer2;

label A;

var x, y, z : real;

1. : integer;

begin

write('**Введите первое число** '); readln(x); write('**Введите второе число** '); readln(y);

A : writeln('**Нажмите** 1 **чтобы увеличить наименьшее из них** **в** 5 **раз и сложить два числа**');

writeln('**Нажмите** 2 **чтобы уменьшить наибольшее из них** **в** 3 **раза и умножить два числа**');

readln(m); case m of 1 : begin

if x<y then x:=x\*5 else y:=y\*5;

z:=x+y;

writeln('**Первое число равно** ', x:8:4, ' **Второе число** **равно** ', y:8:4, ' **Их сумма равна** ', z:8:4);end;

2 : begin

if x>y then x:=x/3 else y:=y/3;

z:=x\*y;

writeln('**Первое число равно** ', x:8:4, ' **Второе число** **равно** ', y:8:4, ' **Их произведение равна** ', z:8:4);end;

else goto A;

end;

**end.**

**Контрольные вопросы**

1. **Что понимают под составным оператором**?
2. **Особенности организации составных операторов**?
3. **Что понимают под алгоритмом ветвления**?
4. **Привести примеры случаев ветвления**.
5. **Как обозначается ветвление в блок**–**схемах**?
6. **Какие операторы ветвления существуют в языке** Turbo Pascal?
7. **Структура условного оператора**?
8. **Какой формат имеет оператор** IF?

22

1. **Какие различия между полной и сокращенной формой условного оператора**?
2. **Структура оператора выбора**?
3. **Что такое ключ выбора**, **список выбора и константа выбора**?
4. **Какие различия между полной и сокращенной формой записи опе**-**ратора выбора**?
5. **Что такое метка в языке** Turbo Pascal?
6. **Как описываются в программе метки**?
7. **Что такое оператор безусловного перехода**?

23

**Лабораторная работа № 3.**

**Организация циклов**

**Цель работы** : **изучение оператора** «**цикла с параметром**», «**цикла с предусловием**», «**цикла с постусловием**» **и составление программ с циклической структурой на языке** Turbo Pascal.

**Теоретические сведения**

**Циклы предназначены для многократного выполнения некоторой последовательности операторов**, **что значительно сокращает объем программы и повышает ее читабельность**. **В языке** Turbo Pascal **реали**-**зованы три оператора цикла**. **Формально эти операторы называют** «**цикл с параметром**», «**цикл с предусловием**» **и** «**цикл с постуслови**-**ем**». **Для каждого из этих циклов существуют свои области примене**-**ния**. **Эти области рассмотрим при подробном рассмотрении каждого из операторов цикла**.

**Цикл с параметром**

**Обобщенный синтаксис этого оператора следующий**:

for <**счетчик** > := <**нач**\_**знач**> to <**кон**\_**знач**> do <**тело**\_**цикла**>; **Здесь**:

<**счетчик**> – **параметр цикла** – **переменная типа** Integer. **Эта пере**-**менная при выполнении тела цикла при каждом выполнении последо**-**вательно принимает значения от** <**нач**\_**знач**> **до** <**кон**\_**знач**>;

<**нач**\_**знач**>, <**кон**\_**знач**> – **начальное и конечное значение счет**-**чика**;

<**тело**\_**цикла**> – **последовательность операторов**, **которая выпол**-**няется в цикле**. **Тело цикла может содержать один оператор или не**-**сколько**. **В случае нескольких операторов их заключают в оператор**-**ные скобки** (**составной оператор**).

**При выполнении программы процесс выполнения цикла выглядит следующим образом**:

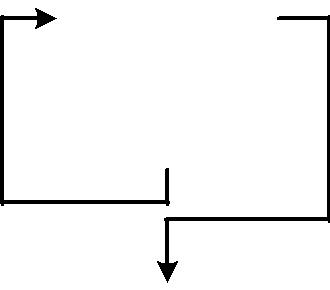
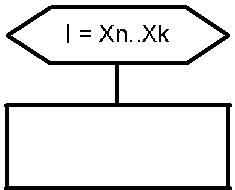
* **первоначально** (**до цикла**) **значение переменной**–**счетчика в общем случае не определено**;
* **в момент первого выполнения оператора** for **счетчик принимает значение** <**нач**\_**знач**>;
* **далее выполняются операторы** <**тело**\_**цикла**>. **При выполнении этих операторов значение счетчика не изменяется**;

24

* **при достижении последнего оператора в теле цикла управление возвращается на оператор** for, **где значение счетчика увеличивает**-

**ся на** 1 **и сравнивается с** <**кон**\_**знач**>. **Если счетчик стал больше**, **чем** < **кон**\_**знач**>, **управление передается на следующий после цикла оператор**.

**Таким образом**, **можно сделать следующий вывод**: **цикл** for **выпол**-**няется столько раз**, **сколько различных значений есть между** <**нач**\_**знач**> **и** <**кон**\_**знач**> **включительно**. C**хема алгоритма цикла с** **параметром приведена на рисунке** 6.



**Рис**. 6 – **Схема алгоритма цикла с параметром**

**Если начальное значение равно конечному**, **цикл выполнится один раз**. **Если начальное значение больше конечного**, **цикл не выполнится ни разу**.

* **том случае**, **если требуется менять значения счетчика в сторону уменьшения** – **от большего к меньшему** – **то применяется несколько иная форма цикла**:

for <**счетчик** > := <**нач**\_**знач**> downto <**кон**\_**знач**> do <**те**-

**ло**\_**цикла**>;

* **этом случае при каждом выполнении оператора** for **значение счетчика уменьшается на** 1.

**Данный вид цикла применяется чаще всего для перебора элементов массива**, **и в тех случаях**, **когда количество проходов заранее известно**.

**Цикл с предусловием**

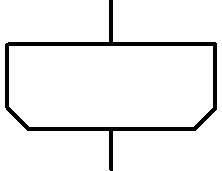
**Обобщенный синтаксис этого оператора следующий**:

while <**условие**> do <**тело**\_**цикла**>;

25

<**Условие**> **является логическим выражением**. **Это означает**, **что** **результатом вычисления этого выражения является одно из значений** TRUE **или** FALSE.

**Тело цикла выполняется до тех пор**, **пока значением условия явля**-**ется** TRUE. **Обратите внимание**, **что возможна ситуация**, **когда усло**-**вие всегда будет истинным**. **В этом случае цикл будет выполняться бесконечно**. **Если условие изначально ложно**, **тело цикла не выпол**-**нится ни разу**, **и управление сразу будет передано на оператор**, **сле**-**дующий за телом цикла**. **Схема алгоритма цикла с предусловием при**-**ведена на рисунке** 7.



A

**Рис**. 7 – **Схема алгоритма цикла с предусловием**

**Этот цикл применяется в тех случаях**, **когда число итераций зара**-**нее неизвестно**. **Типичный пример** – **различные задачи поиска**, **или расчеты методом последовательных приближений**.

**Цикл с постусловием**

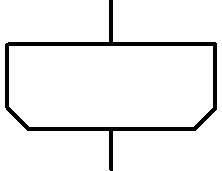
**Обобщенный синтаксис этого оператора следующий**:

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

**Как и в случае цикла с предусловием**, **условие является логическим выражением**. **В отличие от цикла** while, **этот цикл выполняется**, **пока условие ложно**. **Как только условие принимает истинное значение**, **выполнение цикла прекращается**.

**Схема алгоритма цикла с постусловием приведена на рисунке** 8.

26



**уровень** 1

**Рис**. 8 – C**хема алгоритма цикла с постусловием**

**Наиболее часто данный цикл применяется при вводе исходных данных и контроле корректности введенных данных**.

**Содержание лабораторной работы**

* + **лабораторной работе требуется разработать программу на языке программирования** Turbo Pascal **с циклической структурой в соответ**-**ствии с Вашим вариантом**. **Отчет о проделанной работе должен со**-**держать**: **название и цель работы**; **номер варианта для выполнения за**-**дания и условие своего варианта**; **блок**–**схему решения задачи**; **тексты программ**; **полученные при расчетах численные результаты и выводы по проделанной лабораторной работе**.

**Варианты заданий**

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **Вычислить функцию** *y*=0,5*x*+sin2*x* **на отрезке** [0,3*p*] **с шагом**

*h* = *p* / 2.**На каждом шаге выводить на печать значения переменных**

1. **и** *y*. **Для данной задачи написать программы**, **используя операторы цикла** *for*,*while*,*repeat*.

**б**) **Определить знак произведения функции** *y*=(3-sin 2*x*)cos3*x* **на интервале изменения аргумента от** *xn* **до** *xk* **с шагом** *h*.

1. **Разработать программу на языке программирования** Turbo Pascal:

27

**а**) **Вычислить функцию** *y*=*x*2+5*x*-6 **на отрезке** [-7,7]**с шагом** *h* =0,5.**На каждом шаге выводить на печать значения переменных** *x*

* *y* .**Для данной задачи написать программы**,**используя операторы****цикла** *for*,*while*,*repeat*.

**б**) **Найти**, **во сколько раз среднее арифметическое положительных значений отличается от среднего геометрического этих же значений функций** *y*=(3*x*-cos*x*)sin 2*x* **на интервале изменения аргумента от**

*xn* **до** *xk* **с шагом** *h* .

1. **Разработать программу на языке программирования** Turbo Pascal:

**а**) **Вычислить функцию** *y*= *x*(*x* -3)4**на отрезке**[1,6]**с шагом** *h* =0,25.**На каждом шаге выводить на печать значения переменных**



1. **и** *y*. **Для данной задачи написать программы**, **используя операторы цикла** *for*,*while*,*repeat*.

|  |  |
| --- | --- |
| **б**) **Найти среднее арифметическое значений** | **функции** |
| *y* =(1- *e*- *x*2)sin 2*x* **на интервале изменения аргумента от** *x* | **до** *x* **с** |
| *n* | *k* |
| **шагом** *h*. |  |

1. **Разработать программу на языке программирования** Turbo Pascal:

**а**) **Вычислить функцию** *y*= 6*x*2 - *x* **на отрезке** [-6,6] **с шагом** *h* =0,5.**На каждом шаге выводить на печать значения переменных** *x*

* *y* .**Для данной задачи написать программы**,**используя операторы****цикла** *for*,*while*,*repeat*.

**б**) **Найти среднее арифметическое положительных значений функ**-

**ции** *y*=(1-*e*-*x*2) cos 2*x* **на интервале изменения аргумента от** *xn* **до** *xk* **с шагом** *h* .

1. **Разработать программу на языке программирования** Turbo Pascal:

**а**) **Вычислить функцию** *y*=*x*+1+sin(*x*-1) **на отрезке** [0,4*p*]**с шагом** *h*=*p*6 . **На каждом шаге выводить на печать значения пере**-менных *x* и *y* . Для данной задачи написать программы, используя операторы цикла *for*, *while*, *repeat* .

28

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **б**) **Определить** | **знак максимального значения** |  | **функции** |  |
| *y* =(1- *ctg* 3*x*)*e*-*x*2 | **на интервале изменения аргумента от** | *x* | **до** *x* **с** |  |
| **шагом** *h*. |  | *n* | *k* |  |
|  |  |  |  |

1. **Разработать программу на языке программирования** Turbo Pascal:

**а**) **Вычислить функцию** *y*=*x*sin*x*+cos2*x* **на отрезке** [-4*p*,4*p*] **с**

**шагом** *h*=*p* 3 . **На каждом шаге выводить на печать значения пере**-

**менных** *x* **и** *y*. **Для данной задачи написать программы**, **используя операторы цикла** *for*,*while*,*repeat*.

**б**) **Найти максимальное значение функции** *y*=(1-*e*-*x*2)sin 3*x* **на интервале изменения аргумента от** *xn* **до** *xk* **с шагом** *h*.

1. **Разработать программу на языке программирования** Turbo Pascal:

**а**) **Вычислить функцию** *y*= 1+ *x*3 **на отрезке** [0,5] **с шагом** *h* =0,2.**На каждом шаге выводить на печать значения переменных** *x*

* *y* .**Для данной задачи написать программы**,**используя операторы****цикла** *for*,*while*,*repeat*.

**б**) **Определить знак минимального значения** **функции**

*y* =(1- *tg* 2*x*)*e*-*x* **на интервале изменения аргумента от** *xn* **до** *xk* **с**

**шагом** *h*.

1. **Разработать программу на языке программирования** Turbo Pascal:

**а**) **Вычислить функцию** *y*=ln*x*+1*x* **на отрезке** [0,2;2] **с шагом** *h* =0,1.**На каждом шаге выводить на печать значения переменных** *x*

* *y* .**Для данной задачи написать программы**,**используя операторы****цикла** *for*,*while*,*repeat*.

**б**) **Найти минимальное значение функции** *y*=(1-*e*-*x*)cos2*x* **на интервале изменения аргумента от** *xn* **до** *xk* **с шагом** *h*.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **Вычислить функцию** *y*=cos*x* **на отрезке** [-3*p*,3*p*] **с шагом**

*h* = *p* 4.**На каждом шаге выводить на печать значения переменных**



29

1. **и** *y*. **Для данной задачи написать программы**, **используя операторы цикла** *for*,*while*,*repeat*.

**б**) **Найти среднее геометрическое положительных значений функ**-

**ции** *y*=(*e*-*x*-2*x*)cos*x* **на интервале изменения аргумента от** *xn* **до** *xk* **с шагом** *h* .

1. **Разработать программу на языке программирования** Turbo Pascal:

**а**) **Вычислить функцию** *y*= cos2 *x* +1 **на отрезке** [0,4*p* ] **с шагом** *h* = *p*4 . **На каждом шаге выводить на печать значения переменных**



1. **и** *y*. **Для данной задачи написать программы**, **используя операторы цикла** *for*,*while*,*repeat*.

**б**) **Найти количество положительных и количество отрицательных значений функции** *y*=(1-*tg*2*x*)sin*x* **на интервале изменения аргу**-**мента от** *xn* **до** *xk* **с шагом** *h*.

11. **Разработать программу на языке программирования** Turbo Pascal:

**а**) **Вычислить функцию** *y*= 1+ *x*2 **на отрезке** [-1,1] **с шагом** *h* =0,1.**На каждом шаге выводить на печать значения переменных** *x*

* *y* .**Для данной задачи написать программы**,**используя операторы****цикла** *for*,*while*,*repeat*.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **б**) |  | **Определить сумму положительных значений** | | | **функции** |  |
| æ 1 | |  | -*x*2 | ö |  |  |
| *y* = ç |  | - *e* |  | ÷sin 3*x* **на интервале изменения аргумента от** | *xn* **до** *xk* **с** |  |
| 2 |  |  |
| è |  |  | ø |  |  |

**шагом** *h*.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **Вычислить функцию** *y*=*x*3+sin*x* **на отрезке** [-2*p*,2*p*] **с ша**-

**гом** *h*=*p*6 . **На каждом шаге выводить на печать значения перемен**-**ных** *x* **и** *y*. **Для данной задачи написать программы**, **используя опе**-**раторы цикла** *for*,*while*,*repeat*.

**б**) **Определить произведение положительных значений функции** *y* =(1- *e*-*x*2)sin2*x* **на интервале изменения аргумента от** *xn* **до** *xk* **с****шагом** *h*.

30

1. **Разработать программу на языке программирования** Turbo Pascal:

**а**) **Вычислить функцию** *y*=*x*-*tg x* **на отрезке** [0,*p*2] **с шагом** *h* = *p* 10 . **На каждом шаге выводить на печать значения переменных**

1. **и** *y*. **Для данной задачи написать программы**, **используя операторы цикла** *for*,*while*,*repeat*.

**б**) **Найти среднее арифметическое положительных значений функ**-

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| æ | *x* | ö | -*x* |  |  |  |
| **ции** *y*= çcos |  | - *tg x* ÷*e* |  | **на интервале изменения аргумента от** | *xn* **до** |  |
| 3 |  |  |
| è | ø |  |  |  |  |

*xk* **с шагом** *h* .

1. **Разработать программу на языке программирования** Turbo Pascal:

**а**) **Вычислить функцию** *y*=ln*x*2+3*x* **на отрезке** [1,5] **с шагом** *h* =0,4.**На каждом шаге выводить на печать значения переменных** *x*

* *y* .**Для данной задачи написать программы**,**используя операторы****цикла** *for*,*while*,*repeat*.

**б**) **Определить сумму отрицательных значений** **функции**

*y* =(0,2- *e*- *x* )cos3*x* **на интервале изменения аргумента от** *xn* **до** *xk*

**с шагом** *h*.

1. **Разработать программу на языке программирования** Turbo Pascal:

**а**) **Вычислить функцию** *y*= *ctg x* +0,5**на отрезке**[-*p* 2,*p* 2]**с**



**шагом** *h*=*p* 10 . **На каждом шаге выводить на печать значения пере**-**менных** *x* **и** *y*. **Для данной задачи написать программы**, **используя операторы цикла** *for*,*while*,*repeat*.

**б**) **Определить**, **во сколько раз модуль суммы отрицательных зна**-**чений функции отличается от суммы положительных значений функ**-

**ции** *y*=(*x*2+1)cos3*x* **на интервале изменения аргумента от** *xn* **до** *xk* **с шагом** *h*.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **Вычислить функцию** *y*=*x*sin*x*+*ex* **на отрезке** [-4*p*,4*p*] **с ша**-

**гом** *h*=*p* 3 . **На каждом шаге выводить на печать значения перемен**-**ных** *x* **и** *y*. **Для данной задачи написать программы**, **используя опе**-

**раторы цикла** *for*,*while*,*repeat*.

31

**б**) **Определить**, **на сколько модуль суммы отрицательных значений отличается от суммы положительных значений функции**

*y* =(sin 3*x* +2)sin2*x* **на интервале изменения аргумента от** *xn* **до** *xk* **с шагом** *h*.

1. **Разработать программу на языке программирования** Turbo Pascal:

**а**) **Вычислить функцию** *y*=3 1- *x*3 **на отрезке** [-5,5] **с шагом** *h* =0,2.**На каждом шаге выводить на печать значения переменных** *x*



* *y* .**Для данной задачи написать программы**,**используя операторы****цикла** *for*,*while*,*repeat*.

**б**) **Определить произведение отрицательных значений функции** *y* =(sin 2*x* +2*x*)cos *x* **на интервале изменения аргумента от** *xn* **до** *xk* **с шагом** *h*.

1. **Разработать программу на языке программирования** Turbo Pascal:

**а**) **Вычислить функцию** *y*= cos *x* **на отрезке** [0,4*p* ] **с шагом** *h* = *p*4 . **На каждом шаге выводить на печать значения переменных**



1. **и** *y*. **Для данной задачи написать программы**, **используя операторы цикла** *for*,*while*,*repeat*.

**б**) **Определить**, **во сколько раз отличается количество положитель**-

**ных** **от количества отрицательных значений** **функции**

*y* =(cos3*x* -1)*e*- *x* **на интервале изменения аргумента от** *xn* **до** *xk* **с****шагом** *h*.

1. **Разработать программу на языке программирования** Turbo Pascal:

**а**) **Вычислить функцию** *y*=*x*1- *x*2 **на отрезке** [-1,1] **с шагом** *h* =0,1.**На каждом шаге выводить на печать значения переменных** *x*

* *y* .**Для данной задачи написать программы**,**используя операторы****цикла** *for*,*while*,*repeat*.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **б**) **Найти количество отрицательных значений** | **функции** | | | |  |
| *y* =(2*xe*- *x* - *x*)sin *x* **на интервале изменения аргумента от** *x* | *n* | **до** *x* | *k* | **с** |  |
|  |  |  |  |

**шагом** *h*.

1. **Разработать программу на языке программирования** Turbo Pascal:

32

**а**) **Вычислить функцию** *y*=ln(*x*+ *x* )**на отрезке**[0,8; 2*p* ]**с ша**-**гом** *h*=0,1. **На каждом шаге выводить на печать значения перемен**-**ных** *x* **и** *y*. **Для данной задачи написать программы**, **используя опе**-**раторы цикла** *for*,*while*,*repeat*.

**б**) **Найти среднее арифметическое максимального и минимального значений функции** *y*=(2*x*-*e*-*x*2)cos2*x**x* **на интервале изменения аргумента от** *xn* **до** *xk* **с шагом** *h*.

1. **Разработать программу на языке программирования** Turbo Pascal:

**а**) **Вычислить функцию** *y*=*ax*+ *x*2+1**на отрезке**[1,2]**с шагом** *h* =0,5.**На каждом шаге выводить на печать значения переменных** *x*

* *y* .**Для данной задачи написать программы**,**используя операторы****цикла** *for*,*while*,*repeat*.

**б**) **Определить**, **на сколько отличается количество положительных**

**от количества отрицательных значений функции** *y*=(*e*-3*x*-*x*)cos3*x* **на интервале изменения аргумента от** *xn* **до** *xk* **с шагом** *h*.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **Вычислить функцию** *y*=ln(*x*+7*x* +1,65)**на отрезке**[0,7]**с**



**шагом** *h*=0,1. **На каждом шаге выводить на печать значения пере**-**менных** *x* **и** *y*. **Для данной задачи написать программы**, **используя операторы цикла** *for*,*while*,*repeat*.

**б**) **Определить**, **во сколько раз произведение положительных значе**-**ний функции отличается от модуля произведения отрицательных зна**-

**чений функции** *y*=(*x*3-1)sin 2*x* **на интервале изменения аргумента от** *xn* **до** *xk* **с шагом** *h*.

1. **Разработать программу на языке программирования** Turbo Pascal:

**а**) **Вычислить функцию** *y*=sin2 *a x* **при** *a* =20,3**на отрезке**[0,5;2] **с шагом** *h* = 0,1. **На каждом шаге выводить на печать значе**-



33

**ния переменных** *x* **и** *y*. **Для данной задачи написать программы**, **ис**-**пользуя операторы цикла** *for*,*while*,*repeat*.

**б**) **Определить знак суммы значений функции** *y*=(*x*2+1)cos 2*x* **на интервале изменения аргумента от** *xn* **до** *xk* **с шагом** *h*.

1. **Разработать программу на языке программирования** Turbo Pascal:

**а**) **Вычислить функцию** *y*= 2,5*x*2 + 0,4sin *x* +1 **на отрезке** [-1,1]



* **шагом** *h*=0,2. **На каждом шаге выводить на печать значения пере**-**менных** *x* **и** *y*. **Для данной задачи написать программы**, **используя операторы цикла** *for*,*while*,*repeat*.

**б**) **Найти разность максимального и минимального значений функ**-

**ции** *y*=(1-*e*-*x*)sin2*x* **на интервале изменения аргумента от** *xn* **до** *xk* **с шагом** *h*.

1. **Разработать программу на языке программирования** Turbo Pascal:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **а**) **Вычислить функцию** *y*= | ln3 | | *x* + *x* |  | **на отрезке** | [0,2;2] **с шагом** |  |
|  |  |  |  |  |
|  | *x* | + 2,2 |  |
|  |  |  |  |  |  |



*h* =0,1.**На каждом шаге выводить на печать значения переменных** *x*

* *y* .**Для данной задачи написать программы**,**используя операторы****цикла** *for*,*while*,*repeat*.

**б**) **Найти произведение максимального и минимального значений**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | æ | 1 |  | -*x*2 | ö |  |
| **функции** | *y* = ç |  | - *e* |  | ÷cos3*x* **на интервале изменения аргумента от** |  |
| 2 |  |  |
|  | è |  |  | ø |  |
| *xn* **до** *xk* | **с шагом** *h*. | | | |  |  |

1. **Разработать программу на языке программирования** Turbo Pascal:

**а**) **Вычислить функцию** *y*=  *x*3-0,3 *x*2+0,7*x* **на отрезке**[-3,2]



* **шагом** *h*=0,1. **На каждом шаге выводить на печать значения пере**-**менных** *x* **и** *y*. **Для данной задачи написать программы**, **используя операторы цикла** *for*,*while*,*repeat*.

34

**б**) **Найти корень квадратный из модуля произведения максималь**-**ного и минимального значений функции** *y*=(0,3*e*-*x*-2)sin*x* **на ин**-**тервале изменения аргумента от** *xn* **до** *xk* **с шагом** *h*.

1. **Разработать программу на языке программирования** Turbo Pascal:

**а**) **Вычислить функцию** *y* = sin *x*3 - cos *x*2 + *x* **на** **отрезке**



[-2*p* ,2*p* ] **с шагом** *h* = *p*2 . **На каждом шаге выводить на печать зна**-**чения переменных** *x* **и** *y*. **Для данной задачи написать программы**, **используя операторы цикла** *for*,*while*,*repeat*.

**б**) **Определить знак суммы максимального и минимального значе**-**ний функции** *y*=(3-cos 2*x*)sin3*x* **на интервале изменения аргумента от** *xn* **до** *xk* **с шагом** *h*.

1. **Разработать программу на языке программирования** Turbo Pascal:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **а**) **Вычислить функцию** *y*= |  | *x*2 | -1 |  | **на отрезке** [-10,10] **с шагом** |  |
|  |  |  |
|  |  |  |  |  |
|  | 2*x* | |  |  |
|  |  |  |  |  |



*h* =1,5.**На каждом шаге выводить на печать значения переменных** *x*

* *y* .**Для данной задачи написать программы**,**используя операторы****цикла** *for*,*while*,*repeat*.

**б**) **Определить**, **является ли функции положительным числом**. **Функция имеет вид** *y*=(1-*e*-*x*3)sin*x* **на интервале изменения аргу**-**мента от** *xn* **до** *xk* **с шагом** *h*.

29. **Разработать программу на языке программирования** Turbo Pascal:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **а**) **Вычислить функцию** *y*= |  | 2cos2 *x* |  | **на отрезке** [0,3*p*] **с ша**- |  |
| sin *x* + cos *x* |  |  |



**гом** *h*=*p*2 . **На каждом шаге выводить на печать значения перемен**-**ных** *x* **и** *y*. **Для данной задачи написать программы**, **используя опе**-**раторы цикла** *for*,*while*,*repeat*.

35

**б**) **Определить знак максимального значения** **функции**

*y* =(3-sin *x e*- *x* )cos *x* **на интервале изменения аргумента от** *xn* **до**

*xk* **с шагом** *h* .

1. **Разработать программу на языке программирования** Turbo Pascal:

**а**) **Вычислить функцию** *y*=3 *x*(3*x*3+7)**на отрезке**[-2,0]**с шагом** *h* =0,2.**На каждом шаге выводить на печать значения переменных** *x*



* *y* .**Для данной задачи написать программы**,**используя операторы****цикла** *for*,*while*,*repeat*.

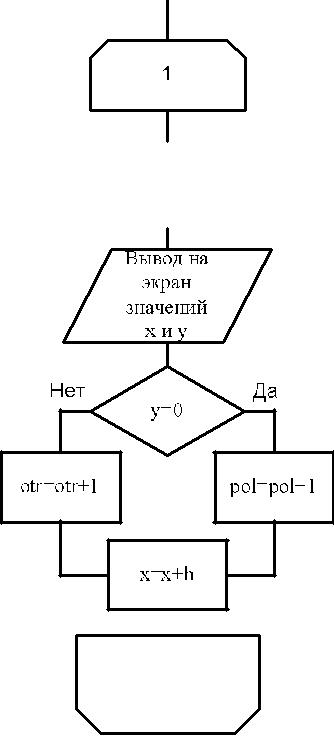
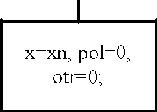
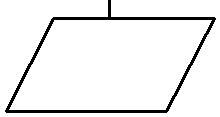
**б**) **Определить**, **превышает ли количество положительных значений функции** *y*=(1-*e*-2*x*)cos 2*x* **на интервале изменения аргумента от** *xn* **до** *xk* **с шагом** *h* .

**Пример программы на языке Turbo Pascal**

**Определить превышает ли количество положительных значений функции** *y*=sin(3*x*)+cos(*x*2) **количество ее отрицательных значе**-**ний на интервале изменения аргумента от** *xn* **до** *xk* **с шагом** *h*.

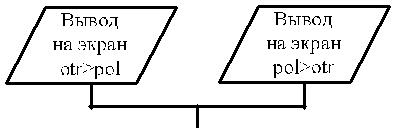
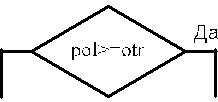
**Схема алгоритма работы программы представлена на рисунке** 9.

36



x>xk

1



**Рис**. 9 – **Схема алгоритма работы программы**

37

Program Primer3;

var

y, x, xn, xk, h : real;

i, pol, otr : integer;

begin

write('**Начальное значение переменной х**='); readln(xn);

write('**Конечное значение переменной х**='); readln(xk); write('**Расчетный шаг** h='); readln(h);

x:=xn; pol:=0; otr:=0;

repeat

y:=sin(3\*x)+cos(sqr(x));

writeln('**Текущие значения**: x=', x:8:4, ' y=', y:8:4); if y>=0 then pol:=pol+1 else otr:=otr+1;

x:=x+h;

until x >= xk;

if pol>=otr then writeln('**Количество** Y\_**пол**=', pol, ' Y\_**отт**=', otr, ' Y\_**пол**>Y\_**отр**') else writeln('**Количество** Y\_**пол**=', pol, ' Y\_**отт**=', otr, ' Y\_**пол**<Y\_**отр**');

end.

**Контрольные вопросы**

1. **Что такое** «**цикл**»?
2. **Какие виды циклов есть в Паскале**?
3. **Обобщенный синтаксис оператора цикла с параметром**?
4. **Обобщенный синтаксис оператора цикла с предусловием**?
5. **Обобщенный синтаксис оператора цикла с постусловием**?
6. **Чем отличается цикл с предусловием от цикла с постусловием**?
7. **Сколько раз минимум может выполниться цикл с предусловием**?
8. **Сколько раз минимум может выполниться цикл с постусловием**?
9. **Какой тип переменной используется для цикла с параметром**?
10. **Какая форма цикла с параметром применяется для цикла** «**от большего к меньшему значению**»?
11. **Какой тип должно иметь условие для цикла с постусловием**?
12. **Каков шаг изменения переменной**–**счетчика в цикле с параметром**?
13. **Для решения каких задач применяется цикл с параметром**?
14. **Для решения каких задач применяется цикл с предусловием**?
15. **Для решения каких задач применяется цикл с постусловием**?

38

**Лабораторная работа № 4. Работа с одномерными массивами**

**Цель работы**: **изучение принципов работы с одномерными масси**-**вами на языке программирования** Pascal. **Получение навыков приме**-**нения основных алгоритмов для решения задач с использованием од**-**номерных массивов**.

**Теоретические сведения**

***Массив*** –**это конечная поименованная совокупность элементоводинакового типа**. **Число элементов в массиве называется размерно**-**стью массива**. **Каждый элемент массива задается своим порядковым номером в массиве** – **так называемым *индексом***. **Примером может служить список фамилий студентов одной группы**, **где каждый сту**-

**дент однозначно определяется своим порядковым номером в списке** (**индексом в массиве**).

**Перед использованием массив**, **как и любая переменная в** Pascal– **программах**, **должен быть объявлен в разделе объявления перемен**-**ных**. **В общем виде объявление массива выглядит так**:

<**имя**> : array [<**нижний**\_**индекс**> .. <**верхний**\_**индекс**>] of

<**тип**>;

**Здесь**:

<**имя**> – **имя переменной массива**;

array – **ключевое слово**, **обозначающее**, **что переменная является** **массивом**;

<**нижний**\_**индекс**> **и** <**верхний**\_**индекс**> – **целые числа**, **опреде**-**ляющие диапазон изменения индексов** (**номеров**) **элементов массива и**, **неявно**, **количество элементов** (**размер**) **массива**;

<**тип**> – **тип элементов массива**.

**Примеры объявления массивов**:

temper: array[1..31] of real;

koef: array[0..2] of integer;

name: array[1..30] of string[25];

**Размещение массива в памяти происходит до выполнения про**-**граммы**, **поэтому при описании индекса можно применять только кон**-**станты или константные выражения**. **Использовать для этого пере**-**менные нельзя**!

**Обычно при описании массива верхняя граница его индекса зада**-**ется в виде именованной константы**. **Именованная константа объявля**-

39

**ется в разделе описания констант**, **который располагается перед разде**-**лом объявления переменных**. **Например**, **массив названий команд** –

**участниц чемпионата по футболу может быть объявлен следующим образом**:

const

NT=18; {**число команд**}

SN=25; {**предельная длина названия команды**} var

team: array[1..NT] of string[SN]

**Массив может быть также описан следующим образом**:

type <**имя**\_ **типа**> = array[<**нижний**\_**индекс**> .. <**верх**-**ний**\_**индекс**>] of <**тип**>;

**где**

type – **зарезервированное слово**, **используемое для создания поль**-**зовательского типа данных**.

**Примеры описания типа массивов**:

Type

M=array[1..10] of integer;

AR=array[0..15] of real;

VT=array[–5..5] of string[10]

* **первом операторе описан тип массива из целых чисел**, **которые нумеруются от** 1 **до** 10. **Во втором операторе элементами массива яв**-**ляются вещественные числа пронумерованные от** 0 **до** 15. **Соответст**-

**венно в третьем примере описан тип массива строковых переменных с изменением индексов от** –5 **до** 5.

**После задания типа массива переменные этого типа описываются обычным образом**:

var a:m; b:ar; c:vt;

**Тип элементов массива может быть любым**, **кроме файлового**, **тип индексов** – **интервальным**, **перечисляемым или** byte.

**С массивами в целом можно выполнять только одну операцию**: **присваивание**. **При этом массивы должны быть одного типа**, **напри**-**мер**:

d:=a;

40

* **отдельными элементами массива выполняются все остальные действия**. **Для обращения к элементу массива после имени массива указывается номер элемента массива в квадратных скобках**:

a[3]

* **элементом массива можно делать все**, **что допустимо для пере**-**менных этого же типа**. **Присвоение значения элементам массива мож**-**но делать до начала выполнения программы присвоением**, **можно вво**-**дить значения как исходные данные**, **а можно получать в результате расчетов**.

**К типичным операциям с массивами можно отнести**: **вывод масси**-**ва**; **ввод массива**; **сортировка массива**; **поиск в массиве заданного эле**-**мента**; **поиск в массиве максимального или минимального элемента**.

**Содержание лабораторной работы**

* + **лабораторной работе требуется разработать программу на языке программирования** Turbo Pascal 7.0 **реализующую алгоритмы обра**-**ботки одномерных массивов в соответствии с Вашим вариантом**. **От**-**чет о проделанной работе должен содержать**: **название и цель работы**; **номер варианта для выполнения задания и условие своего варианта**; **блок**–**схему решения задачи**; **тексты программ**; **полученные при рас**-

**четах численные результаты и выводы по проделанной лабораторной работе**.

**Варианты заданий**

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **В массиве** t(n), **где** n – **заданное число**, **сформированном случай**-

**ным образом**, **найти количество элементов меньших** 0,3.

**б**) **В массив произвольного размера внесены суммы месячного за**-**работка в порядке возрастания табельного номера**. **Найти количество работающих**, **чья зарплата ниже средней и количество работающих**, **чья зарплата выше средней**.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **Массив** a(30) **сформирован случайным образом**. **Найти в нём ко**-

**личество элементов меньших среднего арифметического положитель**-**ных элементов этого массива**.

**б**) **В массив внесена стоимость книг в порядке возрастания их но**-**меров в каталоге**. **Найти самую дорогую и самую дешевую книги** (**их номера**), **если всего книг** 30.

1. **Разработать программу на языке программирования** Turbo Pascal:

41

**а**) **В массиве** s(20) **найти нулевые элементы и заменить их средним арифметическим положительных элементов массива** t(15).

**б**) **В массив занесено население** 12–**ти городов в соответствии с их кодировочными номерами от** 1 **до** 12. **Найти номер города**, **население которого минимально**. **Вывести числовое значение населения этого города**.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **Массив** b(25) **сформирован случайным образом**. **Поменять мес**-

**тами в этом массиве максимальный и минимальный элементы**.

**б**) **В массив внесены суммы вкладов вкладчиков**, **которые зашиф**-**рованы номерами**. **Найти среднюю сумму вкладов**. **Определить сколь**-**ко вкладчиков имеют вклады выше средней суммы**.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **В массиве** y(60), **сформированном случайным образом**, **произве**-

**сти сортировку**, **расположив элементы в порядке возрастания**.

**б**) **В массив внесен месячный баланс** 12 **подразделений предпри**-**ятия**. **Найти количество подразделений**, **имеющих отрицательный ба**-**ланс**. **Найти номер подразделения с наибольшей суммой баланса**.

1. **Разработать программу на языке программирования** Turbo Pascal:

**а**) **В массиве** y(25), **сформированном случайным образом**, **найти среднее геометрическое модулей всех ненулевых элементов**.

**б**) **В массив внесен годовой баланс предприятия**. **Найти среднее значение суммы баланса за год**. **Определить** , **в какие месяцы** (**по номе**-**рам**) **баланс был меньше средней величины**.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **В массиве** b(20), **сформированном случайным образом**, **произве**-

**сти сортировку**, **расположив элементы в порядке убывания**.

**б**) **В массив внесены оценки ученика по определенному предмету** (**их** 10). **Найти среднюю оценку ученика по этому предмету**. **Опреде**-**лить сколько пятерок получил ученик в четверти**.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **В массиве** d(12) **найти среднее геометрическое положительных**

**элементов массива**.

**б**) **В массив внесены результаты контрольной работы в классе по математике**. **Отсутствующие ученики кодируются оценкой** 0. **Опреде**-**лить количество отсутствующих и получивших** 4 **и** 5 (**вместе**).

1. **Разработать программу на языке программирования** Turbo Pascal:

42

**а**) **Элементы массива** y **сформированы по следующему закону**:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ì | 2 | 2 |  |  |
|  | *i*, ***если*** *i*- ***четное***, **найти среднее арифметическое положи**- | |  |
| ïcos |  |  |
| *yi* = í | *i* |  | , ***если*** *i* - ***нечетное***, |  |
| sin |  |  |
| ï |  | 2 |  |  |
| î |  |  |  |



**тельных элементов массива**, **где** *i* ***€*** *[1..n]*. **Определить в полученном массиве** y **количество положительных и отрицательных элементов**.

**б**) **В массив внесены результаты контрольной работы в классе по физике**. **Найти средний балл учащихся**, **полученный по этой кон**-**трольной**. **Подсчитать количество пятерок**, **четверок**, **троек и двоек** (**в классе** 25 **учащихся**).

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **В массиве** *t*(30), **сформированном случайным образом**, **найти ко**-

**личество элементов**, **стоящих на четных местах**, **удовлетворяющих условию** *a*<*ti*<*b*, **где** *a* **и** *b*– **заданные числа**.

**б**) **В массиве содержатся результаты соревнований по плаванию**, **введенные по возрастанию номеров участников**. **Определить номер участника**, **показавшего лучший результат**, **если их было** 25.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **В массиве** *y*(30), **сформированном случайным образом**, **найти**

**максимальный элемент и его номер**.

**б**) **В массиве хранится** 10 **случайных чисел в интервале** [-1,1]. **Най**-**ти количество положительных и отрицательных чисел**. **Определить** , **во сколько суммарное количество положительных чисел превышает от**-**рицательных**.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **В массиве** *b*(40), **сформированном случайным образом**, **найти**

**количество элементов**, **стоящих на нечетных местах**, **удовлетворяю**-**щих условию** *d*<*bi*<*t*, **где** *d* **и** *t*– **заданные числа**.

**б**) **В массиве содержится сумма заработной платы работающих в цехе по порядку их номеров в ведомости** (**их всего** 20). **Найти сред**-**нюю заработную плату**. **Определить сколько работающих получают больше средней заработной платы**, **а сколько меньше**.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **Элементы массива** *a* **сформированы следующим образом**:

|  |  |  |  |
| --- | --- | --- | --- |
| ì | 2 | *i*, ***если*** *i*- ***нечетное***, **Найти среднее арифметическое положи**- |  |
| ïtg |  |  |
| *ai* = í |  | *i* , ***если*** *i* - ***четное***. |  |
| sin | |  |
| ï |  | 2 |  |
| î |  |  |



**тельных элементов этого массива**, **где** *i* ***€*** *[1..n].*

43

**б**) **В массив внесен рост учеников класса из** 20 **человек**. **Найти са**-**мый максимальный и самый минимальный рост**.

1. **Разработать программу на языке программирования** Turbo Pascal:

**а**) **Сформировать массив** c **по следующему** **принципу**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ì |  |  | 2 | , ***если*** *i* - ***нечетное***, |  |
| ïtg*i* | | |  |  |
| *ci* = í |  |  |  | *i* Î[1..*n*].**Перенести положительные эле**- |  |
| ï | 2 | , ***если*** *i* - ***четное***. | | |  |
| î*i* | |  |

**менты массива в массив** y (**подряд**) **а отрицательные элементы в мас**-**сив** x (**подряд**).

**б**) **В массиве содержится сумма заработной платы работающих в отделе по порядку их номеров в ведомости** (**всего работающих** 15). **Найти номера в ведомости с максимальной и минимальной заработ**-**ной платой**.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **В массиве** b(25), **сформированном случайным образом**, **найти**

**количество элементов**, **удовлетворяющих условию** *l*<*bi*<*m*, **где** *l, m*– **заданные числа**.

**б**) **В массив внесены результаты соревнований по бегу в порядке возрастания номеров участников** (**их всего** 20). **Найти значение самого лучшего результата и самого худшего**.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **Даны два массива**. **Массив** t(20) **задан произвольно**. **Элементы**

**массива** d(20) **сформированы** **следующим** **образом**:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ì*i* +1 | | | | | cos *ti* | | , ***если*** *i* - ***четное***, |  |  |
| ï |  |  |  |  | **Заменить все отрицательные элементы** |  |
|  | 2 |  |  |  |
| ï |  |  |  |  |  |  |  |
| *di* = í | |  |  |  |  | *ti* | 2 |  |  |
| ï |  | - |  |  |  | , ***если*** *i* - ***нечетное***. |  |  |
| 1 | | 2sin | | |  |  |  |
| ï |  |  |  |  |  | 2 |  |  |  |
| î |  |  |  |  |  |  |  |  |

**массива** d(20) **средним арифметическим положительных элементов массива** t. **Вывести новый массив** d.

**б**) **В массиве хранятся оценки студента в сессию** (**результаты пяти экзаменов**). **Определить к какой категории** (**отличник**, **занимающийся на** 4 **и** 5, **троечник**) **относится данный студент**.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **В массиве** t(10) **найти среднее геометрическое положительных**

**элементов и среднее арифметическое отрицательных элементов**.

**б**) **Экзаменационная ведомость может содержать** n **фамилий с оценками по физике**. **Определить количество абитуриентов**, **получив**-**ших двойки и пятерки**.

44

1. **Разработать программу на языке программирования** Turbo Pascal:

**а**) **Даны два массива** . **Массив** d(16) **задается произвольно**. **Элемен**-**ты массива** c(16) **формируются следующим образом**:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | ì*d* | | *i* | cos *di* , ***если*** *i* - ***кратно*** 3 | | |  |  |
|  | ï |  | **Расположить элементы массива** c **в** |  |
| ***с****i* | 2 | |  |
| = í |  |  |  |  |  |
|  | ï |  |  |  | 2 | , ***во всех других случаях***. |  |  |
|  | îtg *d* | | | |  |  |  |

**порядке убывания**.

**б**) **Товарная ведомость содержит** n **позиций со стоимостью товаров**.

**Найти количество товаров**, **цена которых выше некоторой величины**.

1. **Разработать программу на языке программирования** Turbo Pascal:

**а**) **Даны два массива** t(20) **и** s(20). **Найти среднее геометрическое положительных элементов массива** t **и заменить им все отрицательные элементы массива** s.

**б**) **Товарная ведомость содержит** n **позиций со стоимостью товаров**.

**Расположить суммы стоимости в порядке убывания**.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **Даны два массива**. **Элементы массива** d(15) **заданы произвольно**,

* **элементы массива** a(15) **сформированы случайным образом**. **Сло**-**жить массивы** d **и** a. **Найти в новом массиве второй отрицательный элемент и вывести его на печать**. **Если количество отрицательных элементов меньше двух**, **то дать об этом сообщение**.

**б**) **Товарная ведомость содержит** n **позиций со стоимостью товаров**. **Расположить суммы стоимости в порядке их возрастания**.

21. **Разработать программу на языке программирования** Turbo Pascal: **а**) **Даны два массива** x(12) **и** y(10). **В массиве** y **все нулевые элемен**-

**ты заменить средним арифметическим элементов массива** x.

**б**) **Балансовый отчет содержит** n **позиций с указанием величины и знака баланса**. **Распечатать вначале значения всех положительных ба**-**лансов**, **а затем значения всех отрицательных балансов**.

22. **Разработать программу на языке программирования** Turbo Pascal: **а**) **Произвести сортировку массива** t(30), **записав положительные**

**элементы в массив** y **подряд**, **а отрицательные** – **в массив** x **подряд**.

**б**) **Ведомость заработной платы содержит** n **позиций**. **Определить сколько человек получают зарплату на** 60 **процентов больше средней**, **сколько человек получают зарплату на** 50 **процентов ниже средней**.

23. **Разработать программу на языке программирования** Turbo Pascal:

45

**а**) **Элементы массива** y(20)

ì2 tg *i*2 , ***если*** *i* - ***четное*** ï

*yi* =ïícos22*i* -1,***если*** *i* -***кратно***5

ï

ïî*i*2-10,***во всех других случаях***.

**сформированы следующим образом**:

**Найти максимальный и минимальный**

**элементы массива** y **и поменять их местами**.

**б**) **В списке указаны суммы вкладов** n **вкладчиков**. **Расположить эти суммы в порядке убывания сумм вкладов**.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **Найти разность элементов двух массивов** *t*(12) **и** *n*(12) **и опреде**-

**лить номер строки**, **для которой эта разность максимальна**.

**б**) **Ведомость заработной платы сотрудников учреждения содержит** n **позиций**. **Найти минимальную заработную плату и определить на**-**сколько она ниже средней**.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **Даны два массива** *k*(15) **и** *d*(5). **Из отрицательных элементов мас**-

**сивов** *k* **и** *d* **сформировать массив** *z*(20), **содержащий только отрица**-**тельные элементы**. **Если отрицательных элементов в массивах** *k* **и** *d* **окажется меньше** 20, **дополнить оставшиеся места** –1.

**б**) **Результаты соревнований по плаванию оформлены в виде про**-**токола**, **содержащего информацию о времени прохождения дистан**-**ции**. **Определить сколько из** n **участников заплыва заслуживают при**-**своения** I **разряда** (**результат ≤**t) **и сколько** –II **разряда** (t< **результат ≤**l).

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **Массив** *b*(28) **сформирован случайным образом**. **Найти макси**-

**мальный элемент в нем**. **Рассортировать элементы после этого эле**-**мента в порядке убывания**.

**б**) **Балансовый отчет содержит** n **позиций с указанием величины и знака баланса** . **Распечатать вначале значения всех положительных ба**-**лансов**, **затем значения всех отрицательных балансов**.

1. **Разработать программу на языке программирования** Turbo Pascal:

**а**) **Найти сумму элементов двух массивов** *a*(15) **и** *b*(15) **и опреде**-**лить номер строки**, **для которой эта сумма максимальна**.

**б**) **Результаты месячной работы** n **участников цеха сведены в таб**-**лицу**, **где указан баланс доходов**–**расходов**. **Определить количество участников**, **которые имели отрицательный баланс**.

1. **Разработать программу на языке программирования** Turbo Pascal:46

**а**) **Известны данные по продаже компьютеров в течении недели**.

**Найти общее количество проданных компьютеров**.

**б**) **Подсчитать среднемесячную зарплату сотрудника предприятия и найти зарплату**, **которая наиболее близка к средней**. **В качестве ре**-**зультата вывести среднюю зарплату**, **наиболее близкую и ее номер в массиве**.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **Курс доллара в течение года менялся в диапазоне от** 28 **руб**. **до**
2. **руб**. **Найти наибольшее значение курса доллара**. **В качестве резуль**-**тата вывести номер месяца и значение курса доллара**.

**б**) **Известен месячный план выпуска некоторой продукции и объе**-**мы выпущенной продукции заводом за год** (**помесячно**). **Определить месяц**, **в котором было максимальное отклонение от плана**. **В качестве результата вывести номер месяца и отклонение**.

30. **Разработать программу на языке программирования** Turbo Pascal: **а**) **Даны результаты сдачи экзамена по информатике группы из** 15

**студентов**. **Подсчитать количество студентов**, **не сдавших экзамен**, **в численном и в процентном соотношении**.

**б**) **Известны данные по продаже компьютеров в течение недели**. **Расположить эти данные в порядке возрастания**.

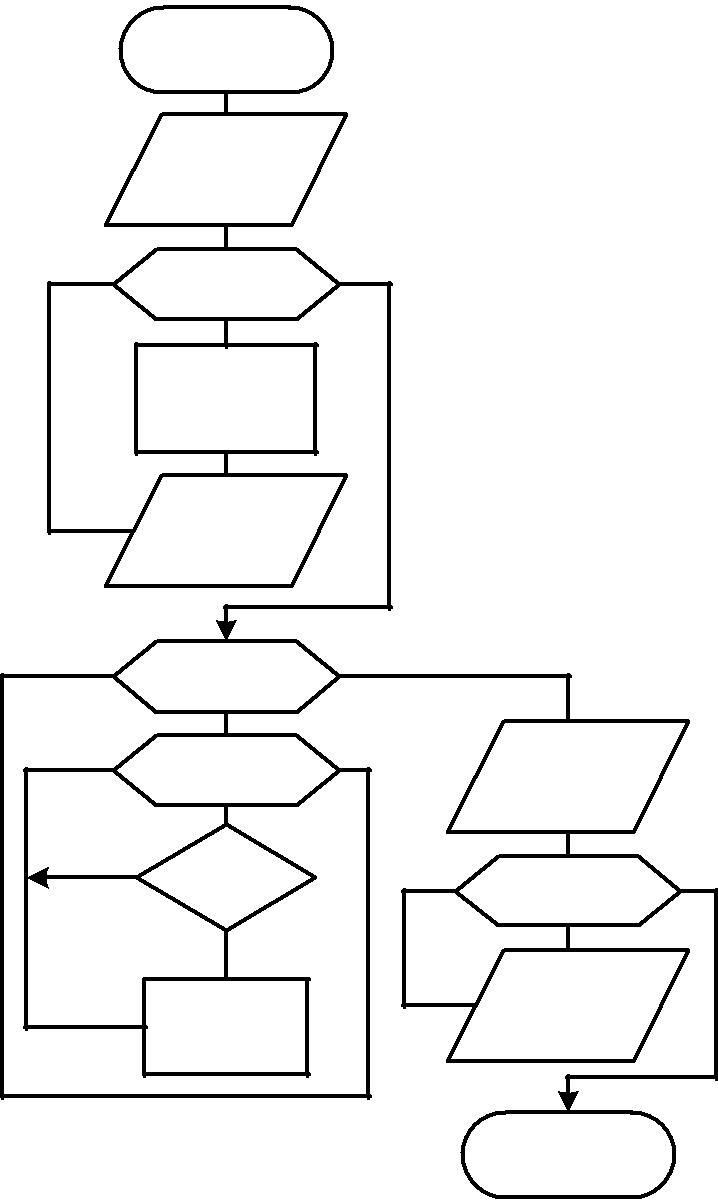
**Пример программы на языке Turbo Pascal**

**Предположим**, **известны результаты соревнований по стрельбе**, **в которых принимали участие** 9 **человек**. **Расположить данные результа**-**ты в порядке возрастания набранных при стрельбе очков**.

**Схема алгоритма работы программы представлена на рисунке** 10.

47

|  |  |  |
| --- | --- | --- |
| **Начало** |  |  |
| '**Неупоряд** |  |  |
| **оченный** |  |  |
| **массив**:' |  |  |
| I = 1 .. 9 |  |  |
| rez[i] = |  |  |
| random(100) |  |  |
| **Вывод на** |  |  |
| **экран** |  |  |
| **значений** |  |  |
| rez[i] |  |  |
| I = 1 .. 8 |  |  |
| j = i+1 .. 9 | '**Упорядоч** |  |
| **енный** |  |
|  | **массив**:' |  |
| **Нет** |  |  |
| rez[i]>rez[j] | I = 1 .. 9 |  |
|  |  |
| **Да** | **Вывод на** |  |
|  |  |
| s:=rez[j] | **экран** |  |
| rez[j]:=rez[i] | **значений** |  |
| rez[i]:=s | rez[i] |  |
|  | **Конец** |  |



**Рис**. 10 – **Схема алгоритма работы программы**

Program Primer4;

var

rez : array[1..9] of integer;

i, j, s : integer;

begin

48

randomize;

writeln;

writeln('**Неупорядоченный массив**:');

for i:=1 to 9 do begin rez[i]:=random(101); write(rez[i]:5); end; writeln;

for i:=1 to 8 do

for j:=i+1 to 9 do

if rez[i]>rez[j] then

begin

s:=rez[j]; rez[j]:=rez[i]; rez[i]:=s;

end;

writeln('**Упорядоченный массив**:'); for i:=1 to 9 do write(rez[i]:5);

end.

**Контрольные вопросы**

1. **Что понимают под массивом данных**?
2. **Что называют размерностью массива**?
3. **Что понимают под индексом элемента массива**?
4. **Какой массив называется одномерным**?
5. **Приведите примеры одномерных массивов**.
6. **Как описываются одномерные массивы на языке** Turbo Pascal?
7. **Как задается диапазон изменения индексов массива**?
8. **Как обозначаются индексы массивов на языке** Turbo Pascal?
9. **Какого типа могут быть элементы массива**?
10. **Какого типа могут быть индексы элементов массива**?
11. **Какие стандартные алгоритмы по работе с одномерными массива**-**ми Вы знаете**?
12. **Какими способами может быть заполнен массив**? **Приведите при**-**меры**.
13. **Как будет выглядеть блок схема для решения задачи сортировки одномерного массива**?
14. **Как будет выглядеть блок схема для решения задачи поиска в од**-**номерном массиве заданного элемента**?
15. **Как будет выглядеть блок схема для решения задачи поиска в од**-**номерном массиве максимального или минимального элемента**?

49

**Лабораторная работа № 5. Работа с двухмерными массивами**

**Цель работы**: **изучение принципов работы с двухмерными масси**-**вами на языке программирования** Turbo Pascal. **Получение навыков применения основных алгоритмов для решения задач с использовани**-**ем двухмерных массивов**.

**Теоретические сведения**

**Исходные данные для решения многих задач удобно представить в виде таблицы**. **Колонки и строки таблицы**, **как правило**, **содержат од**-**нородную информацию**, **если использовать терминологию** TurboPascal – **данные одинакового типа**. **Поэтому в программе для хранения**

* **обработки табличных данных можно использовать совокупность одномерных массивов или двухмерный массив**.
  + **общем виде описание двумерного массива выглядит следующим образом**:

<**Имя**> : array [<**нижняя**\_**граница**\_**индекса**1> .. <**верх**-**няя**\_**граница**\_**индекса**1>, <**нижняя**\_**граница**\_**индекса**2> .. <**верх**-**няя**\_**граница**\_**индекса**2>] of <**тип**>;

**Здесь**:

<**Имя**> – **имя массива**;

array – **ключевое слово**, **показывающее**, **что объявляемый элемент** **данных является массивом**;

<**нижняя**\_**граница**\_**индекса**1>, <**верхняя**\_**граница**\_**индекса**1>, <**нижняя**\_ **граница**\_**индекса**2>, <**верхняя**\_**граница**\_**индекса**2> –

**целые константы**, **определяющие диапазоны изменения индексов и**, **следовательно**, **число элементов массива**;

<**тип**> – **тип элементов массива**.

**Пример объявления двумерного массива**:

Product: array[1..3,1..4] of integer;

**Количество элементов массива может быть вычислено по формуле**:

(<**верхняя**\_**граница**\_**индекса**1> – <**ниж**-**няя**\_**граница**\_**индекса**1>+ 1)\*(<**верхняя**\_**граница**\_**индекса**2>–<**нижняя**\_**граница**\_**индекса**2> + 1),

**Таким образом**, **массив** Product **состоит из** 12 **элементов типа** integer ((3–1+1)\*(4–1+1)=3\*4=12).

**Приведем еще примеры описания двумерных массивов в** Pascal– **программах**:

50

Type MATR=array[1..4,1..5] of integer;

Type B= array[2..9,0..6] of real;

Type C= array[–1..4,–1..4] of char.

**Также допускается указание имени другого типа массива в качест**-**ве типа элементов массива**, **например**:

Type VEC= array[1..4] of real;

MAS= array[1..5] of vec.

* **результате приведенного выше описания тип массива** MAS **будет объявлен как тип двумерного массива**, **первый индекс которого будет меняться от** 1 **до** 5, **а второй индекс** – **от** 1 **до** 4, **т**.**е**. **размерность мас**-**сива составит** 5\*4 **элементов**.

**Чтобы использовать элемент массива**, **нужно указать имя массива и индексы элемента**. **Первый индекс обычно соответствует номеру строки таблицы**, **второй** – **номеру колонки**. **Двумерные массивы**, **в ко**-**торых диапазоны индексов начинаются с** 1, **также называются иногда матрицами**. **Если число строк матрицы равняется числу столбцов**, **то матрицы такого вида называются квадратными**. **Элементы квадратной матрицы вида** B[1,1], B[2,2]< B[3,3] **составляют главную диагональ матрицы**. **Иногда вводят понятие побочной диагонали квадратной матрицы**, **которую составляют элементы** B[1,N], B[2,N–1], B[3,N–2], ..B[N,1], **где** N – **число строк** (**столбцов**) **матрицы**.

**При вводе и выводе значений элементов двумерных массивов ис**-**пользуются вложенные циклы**, **в которых внешний оператор цикла**, **как правило**, **задает изменение строк массива**, **внутренний оператор цикла** – **изменение столбцов**.

**Пример использования вложенных циклов**:begin

for i:=1 to 3 do for j:=1 to 4 do begin

writeln(‘**Введите элемент а**[‘,i,’,’,j,’]’); readln(a[i,j]);

end;

end;

**К типичным операциям с матрицами и двухмерными массивами можно отнести**: **вывод массива**; **ввод массива**; **сортировка массива**; **поиск в массиве элементов удовлетворяющих определенному усло**-

51

**вию**; **поиск в массиве максимального или минимального элемента**; **нахождение суммы элементов строк и столбцов матрицы**.

**Содержание лабораторной работы**

* + **лабораторной работе требуется разработать программу на языке программирования** Turbo Pascal **реализующую алгоритмы обработки двухмерных массивов в соответствии с Вашим вариантом**. **Отчет о проделанной работе должен содержать**: **название и цель работы**; **но**-**мер варианта для выполнения задания и условие своего варианта**; **блок**–**схему решения задачи**; **тексты программ**; **полученные при рас**-

**четах численные результаты и выводы по проделанной лабораторной работе**.

**Варианты заданий**

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **В матрице** *c*(8,5) **поменять местами третий и пятый столбцы**. **б**) **В матрице** *d*(8,8) **заменить отрицательные элементы**, **располо**-

**женные выше главной диагонали**, **средним геометрическим положи**-**тельных чисел**.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **В матрице** *t*(7,8) **найти среднее арифметическое элементов**,

**расположенных в четных столбцах**.

**б**) **Из элементов матрицы** *a*(5,5), **удовлетворяющих условию** - 5 £ *a* (*i*, *j*) £ 5 , **построить вектор** *b*(25) , **заменив недостающие эле**-**менты нулями**.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **В матрице** *a*(8,7) **поменять местами третий и шестой столбцы**.

**Найти среднее арифметическое всех элементов матрицы**.

**б**) **Найти сумму положительных элементов строк матрицы** *d*(8,7).

**Результат поместить в вектор** *c*(8).

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **В матрице** *a*(6,8) **найти количество положительных элементов**.

**б**) **Сложить две матрицы** *a*(7,3) **и** *b*(7,3) **и найти наибольший эле**-**мент в полученной сумме**.

1. **Разработать программу на языке программирования** Turbo Pascal:

52

**а**) **В матрице** *a*(6,8) **найти количество положительных и отрица**-

**тельных элементов**. **Определить**, **каких элементов больше и на сколь**-**ко**.

**б**) **Сложить две матрицы** *a*(7,3) **и** *b*(7,3), **найти наибольший и**

**наименьший элементы в полученной сумме**. **Найденные элементы по**-**менять местами**.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **В матрице** *d*(5,7) **найти среднее арифметическое отрицательных**

**элементов и количество положительных элементов**.

**б**) **Из матрицы** *t*(7,8) **выбрать положительные элементы и размес**-**тить их подряд в вектор** *z*(56). **Если положительных элементов в** *z* **будет меньше**56,**дополнить их**+1.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **В матрице** *c*(7,7) **найти количество положительных элементов**,

**расположенных на главной диагонали**.

**б**) **В матрице** *d*(8,6) **найти максимальный и минимальный элемен**-**ты и их координаты**.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **В матрице** *d*(7,8) **найти среднее арифметическое отрицательных**

**элементов**.

**б**) **Произвести сортировку матрицы** *b*(7,8), **записав ее положи**-**тельные элементы в вектор** *x*(**подряд**), **а отрицательные** – **в вектор**

* 1. (**подряд**).

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **В матрице** *d*(7,6) **поменять местами третью и шестую строки**.

**б**) **В матрице** *a*(6,6) **выбрать все отрицательные элементы**, **распо**-

**ложенные выше главной диагонали**, **и поместить их в вектор** *z* **под**-**ряд**.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **В матрице** *d*(7,7) **найти количество положительных элементов**,

**расположенных на главной диагонали**.

**б**) **Найти суммы отрицательных элементов столбцов матрицы**

*t* (6,8)**и поместить результат в вектор** *m* .

1. **Разработать программу на языке программирования** Turbo Pascal:

53

**а**) **Найти среднее арифметическое отрицательных элементов мат**-

**рицы** *t*(9,6).

**б**) **В матрице** *d*(8,8) **найти минимальный элемент среди элементов**, **расположенных ниже главной диагонали**.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **В матрице** *t*(7,5) **поменять местами первый и последний столб**-

**цы**, **а затем вторую и первую строки**.

**б**) **В матрице** *d*(8,8) **заменить отрицательные элементы**, **располо**-

**женные ниже главной диагонали**, **нулями**, **а положительные** – **едини**-**цами**.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **Найти среднее геометрическое положительных элементов мат**-

**рицы** *t*(10,10).

**б**) **В матрице** *c*(12,12) **найти среднее арифметическое отрицатель**-**ных элементов**, **расположенных выше побочной диагонали**.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **В матрице** *t*(6,7) **найти сумму положительных элементов и сум**-

**му элементов третьей строки**. **Результат вывести с пояснительным текстом**.

**б**) **Произвести вычитание из матрицы** *a*(7,3) **матрицы** *c*(7,3) **и в полученной разности найти минимальный элемент и его номер**.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **В матрице** *s*(12,14) **найти среднее арифметическое отрицатель**-

**ных элементов**, **а также сумму элементов шестой строки**.

**б**) **В матрице** *d*(9,9) **найти количество положительных элементов**, **расположенных ниже побочной диагонали**.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **В матрице** *k*(3,8) **найти среднее геометрическое положительных**

**элементов и сумму элементов седьмого столбца**.

**б**) **Из матрицы** *m*(7,8) **выбрать элементы**, **удовлетворяющие усло**-**вию** *k*<*mi*,*j*<*l*, **и поместить их в вектор** *z*(56). **Если таких элементов меньше** 56, **то на оставшиеся места поместить** 0.

1. **Разработать программу на языке программирования** Turbo Pascal:

54

**а**) **В матрице** *b*(7,7) **поменять местами первую и седьмую строки и найти среднее арифметическое положительных элементов**.

**б**) **Из отрицательных элементов вектора** *d*(9) **сформировать мат**-**рицу** *a*(3,3). **Если отрицательных элементов в** *d* **меньше** 9, **дополнить оставшиеся места** – 1. **Транспонировать полученную матрицу**.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **В матрице** *d*(5,7) **найти среднее геометрическое положительных**

**элементов**, **расположенных в четных строках**.

**б**) **Из элементов вектора** *b*(49), **сформированных случайных обра**-**зом**, **построить матрицу** *d*(7,7).

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **В матрице** *d*(8,3) **найти среднее арифметическое положитель**-

**ных элементов каждого столбца и сформировать из них вектор**.

**б**) **Умножить матрицу** *a*(*n*,*m*) **на вектор** *b*(*m*), **где** *n*=5,*m*=6.

* **полученном произведении произвести сортировку элементов по убыванию** (*n*,*m*)<11.

20. **Разработать программу на языке программирования** Turbo Pascal: **а**) **В матрице** *t*(7,8) **поменять местами третью и пятую строки и**

**найти среднее арифметическое элементов**, **расположенных в четных столбцах**.

**б**) **Записать на место отрицательных элементов матрицы** *a* (10,10)**нули**,**а на место положительных**–**единицы**.**Вывести на пе**-

**чать нижнюю треугольную матрицу в общепринятом виде**.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **В матрице** *d*(6,8) **поменять местами третий и пятый столбцы и**

**найти среднее геометрическое элементов**, **расположенных под глав**-**ной диагональю**.

**б**) **Рассортировать элементы матрицы** *d*(*n*,*n*), **где** *n=5*, **так**, **чтобы**

* **массиве** *x* **были расположены подряд только положительные и ну**-**левые элементы**, **а в массиве** *y*(**подряд**) – **только отрицательные**. **Расположить элементы массивов** *x* **и** *y* **в порядке возрастания их ве**-

**личины**.

1. **Разработать программу на языке программирования** Turbo Pascal:

55

**а**) **Транспонировать матрицу** *f*(10,8) **и вывести на печать элемен**-

**ты главной диагонали и диагонали**, **расположенной под главной**. **Ре**-**зультаты разместить в двух строках**.

**б**) **Найти среднее геометрическое положительных элементов каж**-**дого столбца матрицы** *d*(*n*,*m*). **Поместить их в массив** *x*(*n*,*m*<12).

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **Найти среднее геометрическое положительных элементов мат**-

**рицы** *d*(*n*,*n*), **где** *n=6*, **расположенных на побочной диагонали**.

**б**) **Из положительных элементов массивов** *a*(15) **и** *b*(5) **сформиро**-**вать матрицу** *c*(4,5). **Если положительных элементов в массиве будет меньше** 20, **заполнить свободные места числом** +1.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **В матрице** *c*(7,8) **найти среднее арифметическое элементов ка**-

**ждой строки и поместить эти значения в массив** *m*(7).

**б**) **Вычислить сумму и число положительных элементов матрицы** *b*(*n*,*n*),**находящихся над главной диагональю**(*n*<11).

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **Произвести транспонирование матрицы** *t*(8,6) **и поменять мес**-

**тами третью и шестую строки**.

**б**) **Для целочисленной матрицы найти для каждой строки число элементов**, **кратных** 5, **и наибольший из полученных результатов**.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **В матрице** *a*(9,9) **поменять местами третий и пятый столбцы и**

**найти среднее геометрическое положительных элементов**, **располо**-**женных под главной диагональю**.

**б**) **В матрице** *s*(*n*,*m*) **в каждой строке расположить элементы в по**-**рядке их возрастания** *n*,*m*<10.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **Найти наибольший элемент матрицы** *a*(13,13), **расположенный**

**выше главной диагонали**, **и номера столбца и строки**, **где он находит**-**ся**.

**б**) **Из положительных элементов вектора** *z*(60) **сформировать мат**-**рицу** *b*(10,6). **Если положительных элементов в векторе окажется меньше** 60, **дополнить оставшиеся места числом** +1.

56

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **Найти минимальный элемент матрицы** *c*(5,7) **и обнулить строку**,

**где он располагается**.

**б**) **В матрице** *k*(8,12) **найти сумму отрицательных элементов каж**-**дой строки и разместить значения этих сумм в вектор** *d*.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **Найти максимальный элемент матрицы** *d*(7,7) **и поместить в**

**строку и столбец**, **где он находится**, **число** +1.

**б**) **Из положительных элементов матрицы** *a*(6,6), **расположенных выше главной диагонали**, **сформировать вектор** *m*.

1. **Разработать программу на языке программирования** Turbo Pascal: **а**) **Найти среднее арифметическое отрицательных элементов мат**-

**рицы** *c*(7,6) **и заменить им все нули матрицы**.

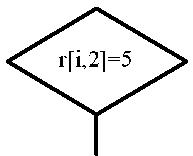
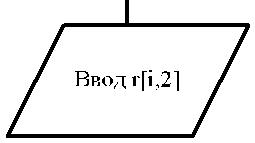
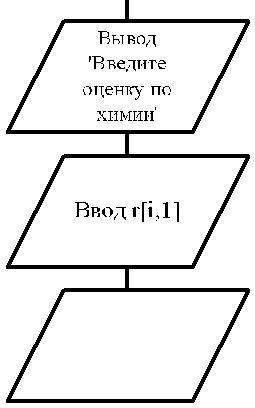
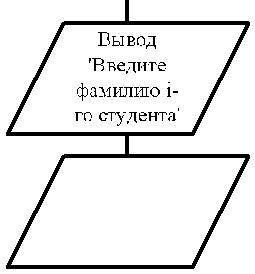
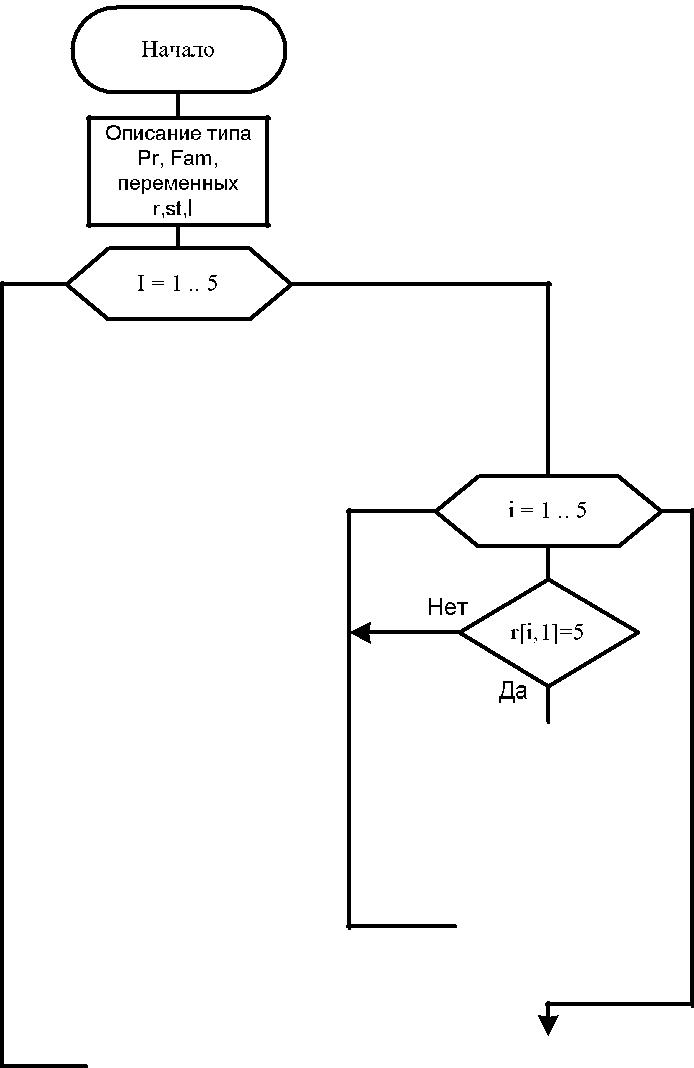
**б**) **Из отрицательных элементов вектора** *z*(70) **сформировать мат**-**рицу** *k*(7,10). **Если отрицательных элементов в векторе окажется меньше** 70, **дополнить оставшиеся места нулями**.

**Пример программы на языке Turbo Pascal**

**Известны результаты** 5 **студентов по итогам экзаменов по химии и информатике**. **Найти фамилии студентов**, **сдавших оба экзамена на отлично**.

**Схема алгоритма работы программы представлена на рисунке** 11.

57



**Рис**. 11 – **Схема алгоритма работы программы**

Program Primer5;

type pr=array [1..5,1..2] of integer; fam=array[1..5] of string[10];

var r:pr;

st:fam;

i,j:integer;

begin

58

for i:=1 to 5 do

begin

writeln('**Введите фамилию** ',i,'–**го студента**'); readln(st[i]);

writeln('**Введите оценку данного студента по химии**

(**от** 2 **до** 5)');

readln(r[i,1]);

writeln('**Введите оценку данного студента по информатике**

(**от** 2 **до** 5)');

readln(r[i,2]);

end;

for i:=1 to 5 do

if (r[i,1]=5) and (r[i,2]=5) then writeln('**Студент** – **отличник** – ',st[i]);

end.

**Контрольные вопросы**

1. **Какой массив называется двухмерным**?
2. **Приведите примеры двухмерных массивов**.
3. **Как может быть вычислено количество элементов двухмерного массива**?
4. **Дайте понятие квадратной матрицы**, **диагоналей квадратной мат**-**рицы**.
5. **Какие способы объявления многомерных массивов вы знаете**?
6. **Как обратиться к элементу многомерного массива**?
7. **Приведите пример описания двумерных массивов на языке** TurboPascal.
8. **Поясните порядок использования вложенных циклов при вводе элементов двумерного массива**.
9. **Какие стандартные алгоритмы по работе с двухмерными массива**-**ми Вы знаете**?
10. **Какими способами может быть заполнен двухмерный массив**? **Приведите примеры**.
11. **Как будет выглядеть блок схема для решения задачи поиска эле**-**мента двухмерного массива удовлетворяющего определенному ус**-**ловию**?
12. **Как будет выглядеть блок схема для решения задачи поиска в двухмерном массиве максимального или минимального элемента**?

59

1. **Как будет выглядеть блок схема для решения задачи нахождения суммы элементов строк матрицы**?
2. **В каких случаях допускается обращение к многомерному массиву целиком**?
3. **Какая конструкция применяется для обработки** n–**мерного масси**-**ва**.

60

**Лабораторная работа № 6.**

**Процедуры и функции**

**Цель работы**: **изучение принципов работы с процедурами и функ**-**циями на языке программирования** Turbo Pascal. **Получение навыков по подготовке**, **редактированию**, **компиляции и выполнению про**-**грамм**, **использующих процедуры** Procedure **и функции** Function.

**Теоретические сведения**

**Как известно**, **синтаксически** Turbo Pascal **программа состоит из необязательного заголовка и блока**. **Блоки могут быть**: **глобальные и локальные**.

***Глобальный блок*** –**это основная программа**,**он должен быть в лю**-**бом случае**. ***Локальные блоки*** – **это процедуры и функции**, **их присут**-**ствии необязательно**. **Соответственно объекты программы** (**типы**, **кон**-**станты**, **переменные и т**.**д**.) **глобального блока называются глобальны**-**ми**, **а локального блока локальными**.

**Часто при разработке программ возникает необходимость в реали**-**зации одного и того же алгоритма в разных точках программы и с раз**-**ными исходными данными** . **Чтобы избежать повторения**, **алгоритм составляется пользователем один раз и оформляется в виде процедур или функций**. **Затем этот алгоритм вызывается в соответствующих** (**нужных**) **местах программы**. **Использование процедур и функций по**-**зволяет**:

– **сократить объем и улучшить структуру программы**;

– **уменьшить вероятность ошибок и облегчить процесс отладки** **программы**;

– **дает возможность выполнить разработку программы разным про**-**граммистам**.

**На языке** Turbo Pascal **процедуры и функции записываются заре**-**зервированными словами** Procedure **и** Function **соответственно**. **Опи**-**сание процедур и функций осуществляется в разделе описаний дан**-**ных**.

**Описание процедуры**:

Procedure <**идентификатор процедуры**>

( <**список формальных параметров с объявлением типа**> ); var <**раздел описаний**>; {**локальные данные**}

begin {**начало локального блока**} <**раздел операторов**>

61

end; {**конец локального блока**} **Допускается описание процедур без параметров**. **Описание функции**:

Function <**идентификатор функции**> ( <**список формальных** **параметров с объявлением типа**> ) : <**тип функции**–**результата**>;

var <**раздел описаний**>; {**локальные данные**} begin {**начало локального блока**}

<**раздел операторов**>

end; {**конец локального блока**}

* **разделе операторов должен находиться оператор присваивания**:

<**идентификатор функции**> : = <**значение**–**ответ**>;

**Описание процедуры или функции само по себе никакого действия**

**не вызывает**. **Чтобы использовать процедуру или функцию**, **необхо**-**димо в нужной точке программы поместить обращение к ней**. **Запи**-**сать специальный оператор вызова процедуры в следующем виде**:

<**идентификатор процедуры**> ( <**список фактических пара**-

**метров**> );

**Вызов функции осуществляется по имени функции аналогично об**-

**ращению**, **например к математической функции** sin x:

x:= 0.5;

a:=sin (x);

**или**

**а**:=sin (0.5);

**Вызов** sin x **осуществляется оператором присваивания**.

**Между фактическими и формальными параметрами должно быть соответствие**: **по типу**, **количеству**, **порядку следования**. **В качестве фактических параметров могут выступать константы**, **выражения**, **пе**-**ременные**, **структурированные данные**. **Константы и выражения не могут быть формальными параметрами**.

**Формальные параметры могут быть заданы как**:

– **параметры**–**значения** (**передача параметров по значению**);

– **параметры**–**переменные** (**передача параметров по ссылке**); **Передача параметров по значению обеспечивает сохранность вели**-

**чины передаваемого параметра**. **Изменение их в процедуре никак не отразится на фактических параметрах**.

**Передача параметров по ссылке при изменении формального пара**-**метра обеспечивает изменение фактического параметра**. **Параметры**

62

**переменные записываются в списке формальных параметров после зарезервированного слова** Var.

**Содержание лабораторной работы**

* + **лабораторной работе требуется разработать программу на языке программирования** Turbo Pascal **использующую процедуры и функции**
* **соответствии с Вашим вариантом**. **Если в задаче не указан способ задания переменной**, **то она вводится пользователем с клавиатуры**. **Отчет о проделанной работе должен содержать**: **название и цель рабо**-**ты**; **номер варианта для выполнения задания и условие своего вариан**-**та**; **блок**–**схему решения задачи**; **тексты программ**; **полученные при расчетах численные результаты и выводы по проделанной лаборатор**-**ной работе**.

**Варианты заданий**

1. **Решить уравнение** (*ba*)!*x*2+2(*dc*)!*x*+(*mk*)!=0, **где** *b*,*a*– **сумма модулей и количество элементов**, **расположенных в матрице** *z*(6,6)**ниже побочной диагонали**; *d* , *c* –**сумма модулей и коли**-**чество элементов**, **расположенных в матрице** *w*(11,11) **ниже по**-**бочной диагонали**;*m*,*k*– **сумма модулей и количество элемен**-**тов**, **расположенных в матрице** *v*(12,12) **ниже побочной диагона**-**ли**. **Знакопеременные массивы сформировать случайным образом**.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *d* | 1 | *x* |  | *d* |  |  |  |  |  |
|  | *t* |  |  | 2 |  | *d* , | *t* –**коли**- |  |
| 2. **Решить показательное уравнение** *e* 2 | | | | - |  | = 0 , **где** |  |
|  |  |  |
|  |  |  |  |  | *t*1 | |  | 1 | 1 |  |
|  |  |  |  |  |  |  |  |  |

**чество столбцов**, **не содержащих нулей и сумма модулей членов этих столбцов в матрице** *a*(8,6);*b*2,*t*2– **количество столбцов**, **не содержащих нулей и сумма модулей членов этих столбцов в мат**-**рице** *b*(12,8). **Знакопеременные массивы сформировать случай**-**ным образом**.

63

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | 20 |  |  |  |  |  |  |  | 30 |  |  |  |  |  |  |  |  |  |  |
|  |  | *z* = | cos å |  | *ai* |  | + sin å | | | | |  | *bi* |  |  |  |  |  |  | *x* |  |
|  |  |  |  |  |  |  |  |  |  |
| 3. | **Вычислить значения функции** | *i*=1 | | | | |  |  |  | *i*=1 | | | | | | , | **если** | |  |
| æ 20 | | | | | *ai* | | ö |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | lnçå | | | | | ÷*x* | | | | | | |  |  |  |  |  |
|  |  |  | è *i*=1 | | | | |  |  | ø |  |  |  |  |  |  |  |  |  |  |  |
|  | **изменяется от** *xn* **до** *xk* **с шагом** *h*. **Знакопеременные массивы** | | | | | | | | | | | | | | | | | | | |  |
|  | *a*(20)**и** *b*(30)**сформировать случайным образом**. | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |
| 4. | **Вычислить таблицу значений** | **функции** *y*= | | | | | |  | *ne*-*m*!- *le*-*s*! | | | | | | | | |  | *x* , | **где** |  |
|  |  | *kl* | -(*m*!+*c*!) | | | | | |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

*m*, *n* –**количество и сумма отрицательных элементов матрицы** *w*(11,11); *s*, *l* –**количество и сумма отрицательных элементов****матрицы** *v*(10,10);*c*,*k*– **количество и сумма отрицательных эле**-**ментов матрицы** *t*(8,8). **Знакопеременные массивы сформировать случайным образом**.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 5. **Вычислить таблицу значений функции** *z* = |  | *s*1*s*2 |  | , **где** *s*1, *k*1 – |  |
| (*k*1 | + *k*2)*x* | |  |
|  |  |  |
| **сумма и количество отрицательных элементов** | | |  | **в матрице** |  |
| *d*(10,10),**стоящих под главной диагональю**; *s*2, *k*2 | | | – **сумма и ко**- | |  |

**личество отрицательных элементов в матрице** *c*(8,8), **стоящих под главной диагональю**;*x* **изменяется от** *xn* **до** *xk* **с шагом** *h*. **Зна**-**копеременные массивы сформировать случайным образом**.

1. **Вычислить таблицу значений функции** *z*=(*xs*1+*ys*2), **где** *s*1,*k*1–

2*k*1*k*2

**сумма и количество отрицательных элементов массива** *a*(50);*s*2, *k*2–**сумма и количество отрицательных элементов массива** *b*(70); *x* –**фиксированное число**,**вводимое с клавиатуры**,**а** *y* **из**-

**меняется от** 1 **до** 3 **с шагом** 0,5. **Знакопеременные массивы сфор**-**мировать случайным образом**.

1. **Вычислить таблицу значений функции** *z*=*q*1+*q*2*x*, **где** *q*1,*n*1–

2*n*1*n*2

**среднее геометрическое и количество положительных элементов**

64

**массива** *a*(60);*q*2,*n*2– **среднее геометрическое и количество по**-**ложительных элементов массива** *b*(40);*x* **изменяется от** *xn* **до** *xk*

* + **шагом** *h*. **Знакопеременные массивы сформировать случайным образом**.

1. **Вычислить таблицу значений функции** *y*=*k*!*xa*((+()*k*+*c*)*m*)*xb*, **где** *kn* !*x x*

*k*, *a* –**количество и среднее арифметическое положительных эле**-**ментов матрицы** *d*(11,11), **расположенных ниже побочной диаго**-**нали**;*m*,*b*– **количество и среднее арифметическое положитель**-**ных элементов матрицы** *t*(10,10), **расположенных ниже побочной диагонали**;*n*,*c*– **количество и среднее арифметическое положи**-**тельных элементов матрицы** *l*(12,12), **расположенных ниже по**-**бочной диагонали**;*x* **изменяется от** *xn* **до** *xk* **с шагом** *h*. **Знако**-**переменные массивы сформировать случайным образом**.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 9. **Вычислить таблицу значений функции** *z* = | *ax*min *by*min | | , **где** |  |
| *an* | + *bm* |  |
|  | min | min |  |  |

*x*min,*n*min–**минимальный элемент и его номер в массиве** *x*(200); *y*min,*m*min–**минимальный элемент и его номер в массиве** *y*(100); *a* –**фиксированное число**,**вводимое с клавиатуры**; *b* **из**-**меняется от** *bn* **до** *bk*, **с шагом** *h*. **Знакопеременные массивы сформировать случайным образом**.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10. **Вычислить** | | | **таблицу** | | | | | |  |  |  | **значений** |  | **функции** |  |
| *z* = | (*a*2 | + *ab* + *b*2)*x* +(sin2 *c* +sin *c* \* *tgd* + *tg* 2*d* )*x*2 | | | | | | | | | | | , **где** | *a*,*b* –**ко**- |  |
|  | æ |  | *k* | 2 |  | *tk* |  |  | *k* | 2 | ö |  |
|  |  |  |  |  |  |  |  |  |
|  |  | 2ç | |  | + |  | + |  | ÷ |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  | ç | 4 | | | 4 | |  | 4 | | | ÷ |  |  |  |
|  |  | è |  | ø |  |  |  |

**личество и сумма отрицательных элементов в матрице** *q*(8,10);*c*, *d* –**количество и сумма отрицательных элементов в матрице** *m*(12,10); *t*, *k* –**количество и сумма отрицательных элементов в**

65

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **матрице** *l*(15,13);*x* **изменяется от** *xn* **до** *xk* | | |  | **с шагом** *h*. **Знакопе**- | |  |
| **ременные массивы сформировать случайным образом**. | | | | |  |  |
| 11. **Вычислить значение функции** *z* = | *xs*1 + *xs*2 |  | , | **где** *s*,*k*– **среднее** | |  |
|  |  |
|  | *k*1+ *k*2 | |  | 1 | 1 |  |
|  |  |  |  |  |

**арифметическое и количество положительных элементов массива** *a*(100); *s*2, *k*2–**среднее арифметическое и количество положи**-**тельных элементов массива** *b*(80).*x* **изменяется от** *xn* **до** *xk* **с**

**шагом** *h*. **Знакопеременные массивы сформировать случайным образом**.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| æ | *a*1 | ö | *n*1 | |  |  |
| ç | ÷ | = 0 **на ин**- |  |
|  |  | |  |
| 12. **Решить тригонометрическое уравнение** sinç *b* | | ÷*x* - *n* | |  |  |
| è 1 | | ø |  | 2 |  |  |

**тервале** [0;10], **где** *a*1,*n*1 – **сумма и количество отрицательных**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **элементов**, | **стоящих в массиве** *d*(30) | | | **на местах**, **номера которых** | | | |
| **кратны** 4; | *b*1, *n*2–**сумма и количество отрицательных элементов**, | | | | | | |
| **стоящих в массиве** *t*(40) **на местах**, **номера которых кратны** 4. | | | | | | | |
| **Знакопеременные массивы сформировать случайным образом**. | | | | | | | |
|  | - | *d*1 | *x* |  | *d*2 |  |  |
| 13. **Решить показательное уравнение** *e* *b*1 | | | | - *eb*2=0,**где** *d* ,*b* –**сум**- | | | |
|  |  |  |  | 1 | | | 1 |

**ма модулей членов и количество строк в матрице** *a*(12,10), **не со**-**держащих отрицательных значений**;*d*2,*b*2– **сумма модулей чле**-**нов и количество строк в матрице** *t*(8,16), **не содержащих отрица**-**тельных значений**. **Знакопеременные массивы сформировать слу**-

|  |  |  |  |
| --- | --- | --- | --- |
| **чайным образом**. |  |  |  |
| 14. **Вычислить** | **таблицу** | **значений** | **функции** |



|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *y* = |  | *a*2 | - *b*2 | *x* + | | | *tg*2*d* +sin2 *c* | | |  | , **где** *a*,*b* – **сумма и количество** |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  | 2 |  | *t*2- *k* 2 | |  |  |  |
|  |  |  |  |  |  |  | |  |  |



**положительных элементов в матрице** *q*(6,6), **расположенных на главной диагонали**;*d*,*c*– **сумма и количество положительных элементов в матрице** *m*(11,11), **расположенных на главной диаго**-

66

**нали**;*t*,*k*– **сумма и количество положительных элементов в мат**-

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **рице** *n*(20,20). **Расположенных на главной диагонали**; | | | | | | | |  | *x* **изменя**- | | | | | | |  |
| **ется от** *xn* | **до** *xk* **с шагом** *h*. **Знакопеременные массивы сформи**- | | | | | | | | | | | | | | |  |
| **ровать случайным образом**. | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 15. **Вычислить** | **таблицу значений функции** *y*= | 2 |  | *a* |  | !*xk* + 2 | | |  | *a* - *b* | | |  | !*xm* | , |  |
|  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  | *c* - *a* | - *b* | | |  | !*xn* | | |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

**где** *a*,*k*– **целая часть модуля суммы и количество отрицательных элементов матрицы** *d*(11,11), **расположенных выше побочной диагонали**;*b*,*m*– **целая часть модуля суммы и количество отрица**-**тельных элементов матрицы** *t*(12,12), **расположенных выше по**-**бочной диагонали**; ***с***,*n*– **целая часть модуля суммы и количество отрицательных элементов матрицы** *l*(14,14), **расположенных вы**-**ше побочной диагонали**;*x* **изменяется от** *xn* **до** *xk* **с шагом** *h*. **Знакопеременные массивы сформировать случайным образом**.

1. **Решить уравнение** *an**x*2+*bmx*+*am*=0, **где** *a*,*n*– **сумма и коли**-**чество положительных элементов в матрице** *d*(4,4), **находящихся на главной диагонали**;*b*,*m*– **сумма и количество положительных элементов в матрице** *t*(8,8), **находящихся на главной диагонали**. **Знакопеременные массивы сформировать случайным образом**.
2. **Решить уравнение** *ma**x*2+*bn**x*=0, **где** *m*,*a*– **количество и сумма отрицательных элементов матрицы** *c*(7,7), **расположенных выше главной диагонали**;*n*,*b*– **количество и сумма отрицательных элементов матрицы** *d*(8,8), **расположенных выше главной диаго**-**нали**. **Знакопеременные массивы сформировать случайным обра**-**зом**.
3. **Решить уравнение** *ma**x*2+*db**x*+*nc*=0, **где** *a*,*n*– **среднее арифме**-**тическое и количество положительных элементов массива** *t*(15), **удовлетворяющих условию** *k*<*ti*<*z*;*b*,*m*– **среднее арифмети**-

67

*s*2, *k*2

*t*(6,6).

*d*(12),

**ческое и количество положительных элементов массива**

**удовлетворяющих условию** *k*<*di*<*I*;*c*,*d*– **среднее арифметиче**-**ское и количество положительных элементов массива** *r*(10), **удов**-**летворяющих условию** *k*<*li*<*s*;*k*,*s*,*z*,*I*– **числа**, **вводимые с кла**-

**виатуры**. **Знакопеременные массивы сформировать случайным об**-**разом**.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| æ *a* ö | | |  | 2 | æ | *b* ö | | æ *c* ö | | |  |  |  |  |
| 19. **Решить квадратное уравнение** ç |  | ÷ | *x* |  | + ç |  | ÷ | *x* + ç |  | ÷ | = 0 , **где** | *a*, *n* | – |  |
|  |  |  |  |  |
| è *n* ø | | |  |  | è *m* ø | | | è *l* ø | | |  |  |  |  |

**сумма и количество положительных элементов в матрице** *d*(10,10); *b*, *m* –**сумма и количество положительных элементов в****матрице** *x*(8,8);*c*,*l*– **сумма и количество положительных эле**-**ментов в матрице** *y*(6,6). **Знакопеременные массивы сформиро**-**вать случайным образом**.

1. **Решить уравнение** *cn*+*bl*=*ma*, **где** *l*,*c*– **количество и сумма от**-**рицательных элементов матрицы** *d*(8,8), **расположенных выше главной диагонали**;*n*,*a*– **количество и сумма отрицательных элементов матрицы** *t*(10,10), **расположенных выше главной диа**-**гонали**;*m*,*b*– **количество и сумма отрицательных элементов мат**-**рицы** *q*(7,7), **расположенных выше главной диагонали**. **Знакопе**-**ременные массивы сформировать случайным образом**.
2. **Вычислить таблицу значений функции** *z*=*xS*1+*xS*2, **где** *s*1,*k*1–*k*1- *k*2

**сумма и количество положительных элементов матрицы**

**расположенных ниже главной диагонали**; **сумма и количе**-

**ство положительных элементов матрицы** *t*(6,6), **расположенных ниже главной диагонали**;*x* **изменяется от** *xn* **до** *xk* **с шагом** *h*. **Знакопеременные массивы сформировать случайным образом**.

68

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 22. **Вычислить таблицу значений функции** *z* = | *ax*max + *by*max | | , **где** |  |
|  | |  |
|  | *n* | + *m* | |  |
|  | max | max | |  |

*x*max,*n*max–**максимальный элемент и его номер в массиве** *x*(70); *y*max,*m*max–**максимальный элемент и его номер в массиве** *y*(100);

*a*, *b* –**фиксированное число**,**вводимое с клавиатуры**.**Знакопере**-**менные массивы сформировать случайным образом**.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| æ | 2*d* | 1 | ö |  | *d*2 |  |  |  |  |
| ç |  | ÷ *x* |  |  |  |  |  |
|  |  |  |  |  |  |  |
| ç | *c*1 | | ÷ |  |  | *d* ,*c* – | |  |
| 23. **Решить показательное уравнение** *e*è | ø | - | = 0 , **где** |  |
|  |  |
|  |  |  |  |  | 2*c*2 | | 1 | 1 |  |
|  |  |  |  |  |  |  |  |

**сумма и количество положительных элементов в матрице** *a*(7,6);*d*2,*c*2–**сумма и количество отрицательных элементов в матрице** *a*(7,6).**Знакопеременные массивы сформировать случайным об**-**разом**.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| æ | *a*1 | ö | *a*2 |  |  |
| ç | ÷ | = 0 **на** |  |
|  | *n* |  |
| 24. **Решить тригонометрическое уравнение** *n*1 sinç *n* | | ÷*x* - |  |
| è 2 | | ø | 1 |  |  |

**интервале** [0;15], **где** *n*1,*a*1– **количество и среднее геометрическое положительных элементов массива** *t*(32);*n*2,*a*2– **количество и среднее геометрическое положительных элементов массива** *w*(20).**Знакопеременные массивы сформировать случайным обра**-

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **зом**. |  |  |  |  |  |  |  |
|  | æ | *x* | ö |  | *n*3 | |  |
| 25. **Решить тригонометрическое уравнение** 2*n*2*a*1 | ç | ÷ |  |  |
| *n* |  | |  |
| - *n*2*tg*ç*a*2 | ÷ | = *a* | |  |  |
|  | è | 1 | ø |  |  | 2 |  |

**на интервале** [5;25], **где** *n*1,*a*1– **количество и среднее арифметиче**-**ское отрицательных элементов массива** *t*(30), **стоящих на чётных местах**;*n*2,*a*2– **количество и среднее арифметическое отрица**-**тельных элементов массива** *d*(25), **стоящих на чётных местах**;*n*3,*a*3**количество и среднее арифметическое отрицательных эле**-

**ментов массива** *b*(28), **стоящих на чётных местах**. **Знакоперемен**-**ные массивы сформировать случайным образом**.

69

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | - | | *d*1 | |  |  | - | *d* 2 |  |  |  |
| 26. **Решить показательное уравнение** | *e* |  | *t*2 | | - *e t*1=0,**где** *t* ,*d* –**ко**- | | | | | |  |
|  |  |  |  |  |  |  | 1 | | | 1 |  |
| **личество и сумма положительных элементов массива** *a*(50), **удов**- | | | | | | | | | | |  |
| **летворяющих условию** 0<*a*<*m*;*t* | | 2 | | ,*d* | | 2 | – **количество и сумма по**- | | | |  |
| *i* |  |  |  |  |  |  |  |  |
| **ложительных элементов массива** | *b*(40), | | | | | | **удовлетворяющих усло**- | | | |  |

**вию** 0<*bi*<*n*;*m* **и** *n*– **положительные числа**, **вводимые с кла**-**виатуры**. **Знакопеременные массивы сформировать случайным об**-**разом**.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| æ *n* | | ö |  |  | *n* |  |  |  |  |  |
| ç | 1 | ÷ *x* |  |  |  |  |  |  |  |
| ç *a* | | ÷ |  |  | 2 |  | *n* , *a* – | |  |
| 27. **Решить показательное уравнение** 5*e*è 1 | | ø | + 2 | |  | = 0 , **где** |  |
|  |  |  |
|  |  |  |  |  | *a*2 | | | 1 | 1 |  |
|  |  |  |  |  | *d*(8,8), | |  |
| **количество и сумма положительных элементов матрицы** | | | | | | | |  |
| **стоящих ниже побочной диагонали**;*n*2,*a*2 | | | | – **количество и сумма** | | | | | |  |

**положительных элементов матрицы** *t*(10,10), **стоящих ниже по**-

**бочной диагонали**. **Знакопеременные массивы сформировать слу**-**чайным образом**.

1. **Вычислить значение функции** *y*=*bn*+2*nma*, **где** *b*,*n*– **сумма и количество отрицательных элементов массива** *c*(30);*a*,*m*– **сум**-**ма и количество отрицательных элементов массива** *d*(10). **Знако**-**переменные массивы сформировать случайным образом**.
2. **Решить уравнение** *ma**x*2+*nc**x*+*dt*=0, **где** *a*,*m*– **сумма и коли**-**чество положительных элементов массива** *f*(30);*c*,*n*– **сумма**, **и количество положительных элементов массива** *l*(20);*d*,*t*– **сумма и количество положительных элементов массива** *k*(10). **Знакопе**-**ременные**, **массивы сформировать случайным образом**.

|  |  |  |  |
| --- | --- | --- | --- |
| 30. **Вычислить таблицу значений функции** *y* = | *a*!*xn* +(*a* + *b*)!*xk* | , **где** |  |
| (*ac*)!*xm* |  |
|  |  |  |

*a*, *n* –**количество и среднее арифметическое положительных эле**-**ментов матрицы** *d*(10,6);*b*,*k*– **количество и среднее арифмети**-**ческое положительных элементов матрицы** *t*(8,7); ***с***,*m*– **количе**-

70

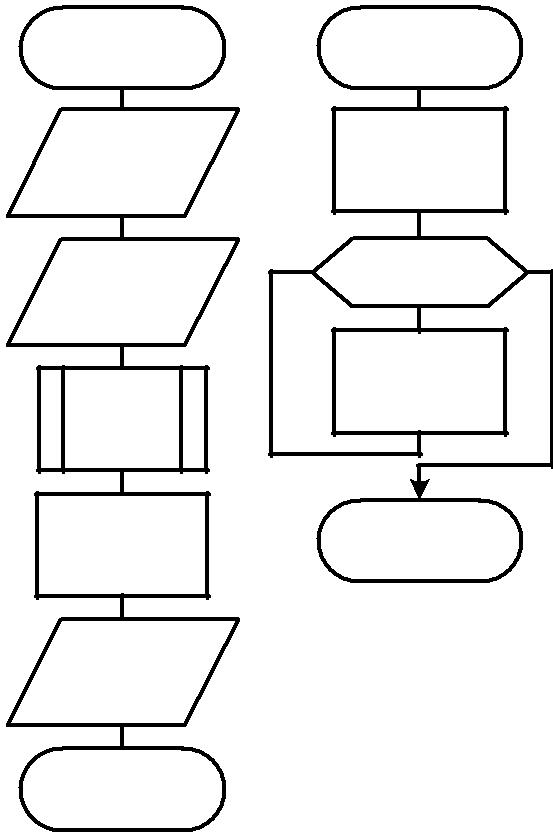
**ство и среднее арифметическое положительных элементов матри**-**цы** *l*(12,10);*x* **изменяется от** *xn* **до** *xk* **с шагом** *h*. **Знакоперемен**-**ные массивы сформировать случайным образом**.

|  |  |  |  |
| --- | --- | --- | --- |
| **Пример программы на языке Turbo Pascal** | |  |  |
| **При известных значениях переменных** *a*,*b*,*c* **и натуральных чи**- | | |  |
| **сел** *n*,*m* **вычислить значения функции**:*y*= | *an* + *bm*+1 | . |  |
| *c*10 |  |
|  |  |  |

**Для вычисления** *y* **необходимо три раза определить функцию за**-

**данной степени при заданном основании**, **причем основания и степени разные**. **Чтобы не выполнять один и тот же алгоритм**, **оформим его процедурой или функцией**.

C**хема алгоритма работы программы представлена на рисунке** 12.



**Начало**

**Вывод**

'**Введите**

a, b, c,

n, m'

**Ввод** a, b,

c, n, m'

y1=STEP,

y2=STEP,

y3=STEP

STEP

Y=1

I = 1 .. n

y=y\*x

y=(y1+y2)/y3 **Конец**

**Вывод** y

**Конец**

**Рис**. 12 – C**хема алгоритма работы программы**

**а**) **программа с процедурой** Procedure

71

Program Primer6a;

var a, b, c, y, y1, y2, y3:real;

n, m: byte; {**Глобальное описание данных**} procedure step (x:real; n:byte; var y:real); {**Объявление процедуры**} var i:byte; {**Локальное описание данных**}

begin {**Начало локального блока**} y:=1;

for i:=1 to n do y:=y\*x;

end; {**Конец локального блока**} begin {**Начало глобального блока**}

writeln ('**Введите исходные данные** a, b, c, n, m'); readln (a, b, c, n, m);

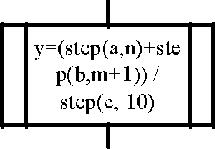
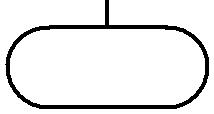
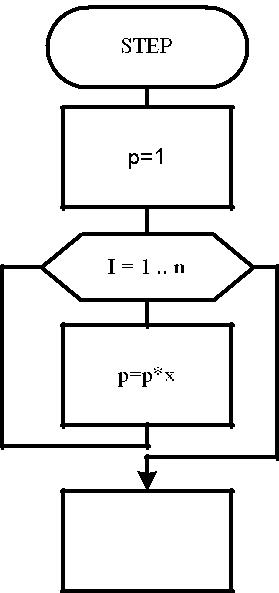
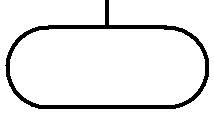
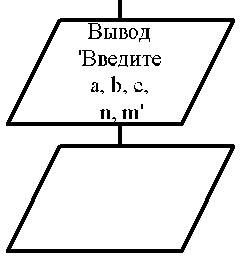
step (a, n, y1); {**Обращение к процедуре**} step (b, m+1, y2); {**Обращение к процедуре**} step (c, 10, y3); {**Обращение к процедуре**} y:= (y1+y2)/y3;

writeln ('y=', y:8:4);

end. {**Конец глобального блока**}

C**хема алгоритма работы программы представлена на рисунке** 13.

72



**Рис**. 13 – C**хема алгоритма работы программы**

**б**) **программа с процедурой** FunctionProgram primer6a;

var a, b, c, y, y1, y2, y3:real;

n, m: byte; {**Глобальное описание данных**}

function step (x:real; n:byte) : real; {**Объявление функции**} var p:real; i:byte; {**Локальное описание данных**}

begin {**Начало локального блока**} p:=1;

for i:=1 to n do p:=p\*x;

step:=p; {**Имени функции присваивается**

**значение результата**}

end; {**Конец локального блока**} begin {**Начало глобального блока**}

writeln ('**введите исходные данные** a, b, c, n, m');

readln (a, b, c, n, m);

y:=(step(a,n)+step(b,m+1))/step(c, 10);

writeln ('y=', y:8:4);

end. {**Конец глобального блока**}

73

**Контрольные вопросы**

1. **Какие алгоритмы называют вспомогательными**?
2. **Какое количество вспомогательных алгоритмов может присутст**-**вовать в основном алгоритме**?
3. **Можно ли вспомогательные алгоритмы**, **написанные для решения данной задачи**, **использовать при решении других задач**?
4. **Для чего нужны процедуры и функции пользователя**?
5. **Что такое** “**глобальный блок**” **и** “**глобальные объекты**” **и** “**локаль**-**ный блок**”, “**локальные объекты**”?
6. **Как описываются в программе процедуры**?
7. **Как описываются в программе функции**?
8. **Чем отличается описание процедур от описания функций**?
9. **В каких случаях целесообразно использовать функции**?
10. **Какого типа может быть значение функции**?
11. **Чем отличаются фактические параметры от формальных**?
12. **Почему**, **если в функции используются параметры**–**переменные**, **необходимо преобразовать её в процедуру**?
13. **Как вызываются процедуры**, **функции**?
14. **Как задаются** (**передаются**) **формальные параметры**?
15. **Какой вариант**, **представленной в качестве примера программы**, **более рационален а**) **или б**) **и почему**?

74

**Лабораторная работа № 7. Файлы и их применение в Turbo Pascal**

**Цель работы**: **ознакомление с разновидностями файлов в** Pascal. **Принципы организации и порядок работы с типизированными и нети**-**пизированными файлами**. **Организация текстовых файлов**, **операции чтения и записи текстовых файлов**.

**Теоретические сведения**

* Pascal **под файлом понимается именованная область памяти на** **долгосрочном запоминающем устройстве**, **например**, **на жестком дис**-**ке или дискете**. **Выделяют три типа файлов**: **текстовые**, **типизирован**-**ные**, **нетипизированные**.

***Текстовый файл*** –**содержит строки текста различной длины**,**чте**-**ние или запись осуществляется** (**как правило**) **построчно**.

***Типизированный файл*** –**с точки зрения**Turbo Pascal**содержит эле**-**менты одного типа**, **единицей обмена информацией является один элемент**.

***Нетипизированный файл*** –**содержит любые данные**, Turbo Pascal

**никак не интерпретируемые**.

**Типизированные файлы**

**Это файл**, **содержащий компоненты одного конкретного типа**. **Это могут быть простые или сложные типы данных** Turbo Pascal (**числа**, **строки**, **символы**, **массивы**, **записи и т**.**д**.). **Объявление типизированно**-**го файла выглядит следующим образом**:

Type <**имя**\_**типа**> = file of <**тип**>;

Var

<**имя**> : <**имя**\_**типа**>;

**или**

Var

<**имя**> : file of <**тип**>;

**Здесь**:

<**имя**\_**типа**> – **имя файлового типа**; <**тип**> – **любой тип данных** Pascal, **кроме файлов**; <**имя**> – **имя файловой переменной**.

**После объявления файловой переменной для работы с файлом не**-**обходимо связать ее с реальным файлом на диске**. **Это выполняется процедурой** Assign:

Assign ( <**имя**>, ’ <**путь**\_**к**\_**файлу**> ’);

75

**Первый параметр** – **имя файловой переменной** , **используемой в дальнейшем для обращения к файлу**. **Может иметь любой файловый тип**. **Второй параметр** – **строка**, **указывающая на путь и имя файла**, **с которым** Turbo Pascal **далее будет оперировать**.

**Для создания нового файла на диске после выполнения процедуры** Assign **необходимо открыть файл в режиме записи** : ReWrite (<**имя**>); **Если файл с таким именем не существовал**, **он создается**. **Если он существовал**, **вся информация из этого файла стирается и файл стано**-

**вится пустой**. **При этом никаких предупреждений не выдается**.

**Для открытия файла в режиме чтения он должен быть открыть процедурой**:Reset(<**имя**>);

**После этого можно выполнять чтение или запись**, **используя соот**-**ветствующую файловую переменную**:

Read(F, i, j);

Write(F, n);

* **первом примере из файла**, **связанного с переменной** F, **считыва**-**ется два значения**, **во втором** – **записывается одно**.

**Для определения позиции указателя** – **порядкового номера компо**-**нента файла**, **который будет обрабатываться следующей операцией ввода**-**вывода**, **применяется функция** FilePos. **Например**:

N := FilePos(F);

**Эта функция возвращает переменной** N **положение указателя отно**-**сительно начала файла**. **Самая первая позиция имеет номер** 0. **Пере**-**местить указатель в требуемую позицию можно**, **используя функцию** Seek. **Например**:

Seek(F, 5);

Seek(F, FilePos(F)-1);

**Первый оператор устанавливает позицию указателя в файле после** 5 **элемента** (**и перед** 6-**м**). **Следующий оператор вызывает смещение** **указателя файла на** 1 **элемент назад относительно текущего положе**-**ния указателя**.

**Дополнительно для проверки положения указателя есть функция** EOF. **Эта функция возвращает логическое значение**, **показывающее**, **находится ли указатель файла после последнего элемента файла**. **Чаще всего эта функция применяется в задачах полного перебора элементов файла**:

While not EOF(F) do Read(F, i);

76

**Этот оператор считывает элементы файла**, **пока не доберется до конца файла**.

**Общее число компонент файла можно определить**, **вызвав функ**-**цию** FileSize. **Например**:

Count := FileSize(F);

**Эта функция возвращает целое число**, **равное количеству компо**-**нент в указанном файле**.

**Для закрытие файла следует воспользоваться следующей процеду**-**рой**:

Close ( <**имя**> );

**При этом связь файловой переменной**, **с именем файла установлен**-**ная процедурой** Assign **сохраняется**.

**Текстовые файлы**

**Текстовый файл содержит строки текста различной длины**. **Каждая строка завершается символами** «**Возврат каретки**» (**код** 13) **и** «**Перевод строки**» (**код** 10). **При обычном просмотре эти символы на экране не отображаются**, **а интерпретируются компьютером как признак окон**-**чания одной строки и начала следующей**.

**Объявление текстового файла выглядит следующим образом**:

Var

<**имя**> : Text;

**Как и в случае других типов файлов**, **перед работой с текстовым файлом**, **предварительно необходимо связать файловую переменную с файлом на диске процедурой** Assign.

**При работе с текстовым файлом разрешается либо чтение**, **либо за**-**пись**. **Открытие файла для чтения или записи выполняется аналогично типизированным файлам процедурами** ReSet **или** ReWrite. **Для тек**-**стовых файлов**, **помимо процедур** ReSet/ReWrite, **определена еще одна процедура открытия**:

Append(<**имя**>);

**Данная процедура открывает файл для записи**, **при этом новые данные добавляются в конец файла**, **прежнее содержимое файла со**-**храняется** (**в отличие от процедуры** ReWrite).

**Далее в зависимости от способа открытия файла можно либо чи**-**тать из него данные**:

Read(F, i, j); ReadLn(F, s);

**Либо записывать в него данные**:

Write(F, a, b, c); WriteLn(F, Result);

77

**При чтении**/ **записи данных применяются те же правила**, **что и при вводе**/**выводе с клавиатуры**/**на экран**, **т**.**е**. **можно использовать форма**-**тирование**:

WriteLn(F, x:5:2);

* **данном примере** F– **имя файловой переменной**;x– **выводимое выражение**;5– **параметр**, **указывающий минимальную ширину поля**, **в которое будет записываться символьное представление**;2– **задает ко**-**личество десятичных знаков в дробной части вещественного числа**.

**Нетипизированные файлы**

**Нетипизированные файлы очень похожи на типизированные**, **за ис**-**ключением того**, **что при работе с ними можно сразу считать или за**-**писать несколько элементов файла**.

**Объявление нетипизированного файла**:

Var

<**имя**> : File;

**Открытие нетипизированного файла выполняется процедурами** ReSet **или** ReWrite. **При этом может указываться дополнительный** **параметр**, **который указывает размер одного блока данных в байтах**:

ReSet(F, 25);

**или**

ReWrite(F, SizeOf(Integer)\*12);

* **первом примере открывается файл для чтения**/**записи с размером блока** 25 **байт**. **Во втором случае размер блока указывается через раз**-**мер типа** Integer. **Функция** SizeOf **возвращает количество байт**, **отво**-**димых под указанный тип или переменную**. **Если размер блока дан**-**ных опущен**, **принимается размер по умолчанию**, **равный** 128 **байт**.

**Чтение данных выполняется следующей процедурой**:

BlockRead(F, Biffer, Count); BlockRead(F, Biffer, Count, Result);

* **примерах из файла** F **в область памяти**, **задаваемую переменной** Buffer, **считывается** Count **блоков данных**. **Во втором случае в пере**-**менную** Result **дополнительно заносится число реально считанных блоков данных**. **Такая ситуация возможна в тех случаях**, **когда конец файла встречается раньше**, **чем будет считано заданное число блоков**. **Если значения** Count **и** Result **различаются**, **это говорит о неполном чтении**.

**Аналогичные правила действуют для записи данных в файл**:

BlockRead(F, Biffer, Count);

78

BlockRead(F, Biffer, Count, Result);

**Частичная запись возможна в том случае**, **если на диске заканчива**-**ется свободное место**.

**Содержание лабораторной работы**

* + **лабораторной работе требуется разработать программу на языке программирования** Pascal 7.0 **использующую принципы организации и порядка работы с файлами в соответствии с Вашим вариантом**. **Отчет**
* **проделанной работе должен содержать**: **название и цель работы**; **но**-**мер варианта для выполнения задания и условие своего варианта**; **блок**–**схему решения задачи**; **тексты программ**; **полученные при рас**-

**четах численные результаты и выводы по проделанной лабораторной работе**.

**Варианты заданий**

1. **Организовать простейшую базу данных по студентам группы**. **Сведения о студенте включают**: **ФИО**, **год рождения**, **пол**, **средний балл**. **Обеспечить ввод данных**, **редактирование**, **вывод на экран**. **Информацию хранить в типизированном файле**.
2. **Запись содержит данные о студенте**: **фамилию и инициалы**, **номер курса**, **номер группы**, **оценки**, **полученные в сессию**. **Получить списки студентов заданной группы**, **претендующих на повышен**-**ную стипендию и студентов**-**задолжников**.
3. **Дан текстовый файл**. **Считая**, **что количество букв в одном слове не превосходит** 20, **определить**, **сколько в файле имеется слов**, **со**-**стоящих из одного**, **двух**, **трех и т**.**д**. **символов**. **Результат вывести в другой текстовый файл**.
4. **Запись содержит данные о студенте**: **фамилию и инициалы**, **номер курса**, **номер группы**, **оценки**, **полученные в сессию**. **Для каждого студента заданной группы определить средний балл**, **а для всей группы вычислить средний балл по каждому предмету**.
5. **Сведения об автомобиле состоят из его марки**, **номера и фамилии владельца**. **Создать файл**, **содержащий сведения о нескольких ав**-**томобилях**, **после чего определить фамилии владельцев и номера автомобилей заданной марки**. **Марка автомобиля вводится пользо**-**вателем**.
6. **Запись содержит данные о студенте**: **фамилию и инициалы**, **номер курса**, **номер группы**, **аттестацию** (0 **или** 1) **по каждому предмету**.

79

**Получить списки неаттестованных студентов заданной группы по каждому предмету**.

1. **Сведения об автомобиле состоят из его марки**, **номера и фамилии владельца**. **Создать файл**, **содержащий сведения о нескольких ав**-**томобилях**, **после чего определить количество автомобилей каж**-**дой марки**.
2. **Запись содержит данные о студенте**: **фамилию и инициалы**, **номер курса**, **номер группы**, **количество часов**, **пропущенных по уважи**-**тельной и неуважительной причине**. **Получить списки студентов заданной группы**, **пропустивших занятия отдельно по уважитель**-**ной и неуважительной причинам с указанием часов пропуска**.
3. **Дан текстовый файл**, **содержащий программу на языке Паскаль**.

**Проверить эту программу на соответствие числа открывающих и закрывающих круглых скобок**.

1. **Запись содержит данные о студенте**: **фамилию и инициалы**, **номер курса**, **номер группы**, **количество часов**, **пропущенных по уважи**-**тельной и неуважительной причине**. **Для каждого студента задан**-

**ной группы определить суммарное число пропущенных часов и выдать список студентов этой группы**, **пропустивших более** 30 **ча**-**сов**.

1. **Дан текстовый файл**. **Записать все строки файла в новый файл**, **изменив порядок следования букв в каждой строке на противопо**-**ложный**.
2. **Запись содержит данные о студенте**: **фамилию и инициалы**, **номер курса**, **номер группы**, **сведения о месте проживания** (**дома**, **в об**-**щежитии или на квартире**). **Получить списки студентов заданной группы отдельно по каждому месту проживания**.
3. **Создать файл**, **содержащий не более** 100 **случайных целых чисел**. **Выполнить сортировку чисел по возрастанию**.
4. **Запись содержит данные о студенте**: **фамилию и инициалы**, **номер курса**, **номер группы**, **экзаменационную оценку и дату сдачи экза**-**мена**. **Получить списки студентов заданной группы**, **сдавших экза**-**мен на отлично** (5), **хорошо** (4), **удовлетворительно** (3), **не сдавших экзамен** (2) **и не явившихся на экзамен** (0).
5. **Создать файл**, **содержащий не более** 100 **случайных целых чисел**. **Создать новый файл**, **разместив все нечетные числа в начале фай**-**ла**, **а четные** – **в конце**, **при этом порядок следования чисел сохра**-**няется**.

80

1. **Запись содержит сведения о результатах медосмотра**: **фамилия и инициалы**, **рост**, **вес**. **Для каждого человека определить индекс Кетле и выдать списки людей с нормальным**, **малым и избыточ**-**ным весом**.

***Указание к решению***

|  |  |  |  |
| --- | --- | --- | --- |
| **Индекс Кетле определяется по формуле** | | ***вес*** (***кг***) |  |
| ***рост***2(***м***) |  |
|  |  |  |
| **Индекс Кетле** | **Показатели веса** | |  |
| **до** 19,5 | **малый вес** | |  |
| **от** 19,5 **до** 24,9 | **нормальный вес** | |  |
| **от** 25 **и выше** | **избыточный вес** | |  |

1. **Запись содержит данные о работнике**: **фамилия и инициалы**, **та**-**бельный номер**, **номер подразделения**, **должность**, **стаж работы**, **величина зарплаты**. **Получить списки работников**, **заданного под**-**разделения**, **стаж которых не менее** 10, 20, 25 **лет**.
2. **Дан текстовый файл**, **содержащий строки произвольной длины**. **Отформатировать текст и записать его в новый файл так**, **чтобы все строки имели одинаковую длину**, **равную длине самой длин**-**ной строки**. **Форматирование выполняется добавлением пробелов между словами**.
3. **Запись содержит данные о работнике**: **фамилия и инициалы**, **та**-**бельный номер**, **номер подразделения**, **должность**, **стаж работы**, **величина зарплаты**. **Определить для каждого работника заданного подразделения сумму к выдаче** (**в простейшем случае это** 87 **про**-**центов от зарплаты**) **и общую сумму заработка работников этого подразделения**.
4. **Дан текстовый файл**, **содержащий сведения о студентах группы**. **Сведения включают ФИО**, **год рождения**, **оценки по** 4 **экзаменам за последний семестр**. **Сведения об одном студенте находятся в од**-**ной строке**, **отделены друг от друга точкой с запятой**. **Считать све**-**дения**, **записать в типизированный файл соответствующего типа**, **вывести на экран ФИО студента с максимальным средним баллом**.
5. **Запись содержит данные о работнике**: **фамилия и инициалы**, **та**-**бельный номер**, **номер подразделения**, **количество отработанных часов за месяц**, **стоимость одного часа работы**. **Определить для каждого работника заданного подразделения сумму заработка и общую сумму для этого подразделения**.

81

1. **Запись содержит данные о товаре**: **наименование**, **артикул** (**штрих**-**код**), **цена за единицу**, **количество**. **Для каждого товара определить общую сумму и получить сумму всех товаров**. **Выдать списки то**-**варов**, **стоимость** (**цена**) **которых выше и ниже средней**.
2. **Запись содержит данные о товаре**: **наименование**, **артикул** (**штрих**-**код**), **количество единиц товара**, **количество проданного товара за день**. **Для каждого товара получить остаток на конец дня**. **Выдать списки распроданных товаров и товаров**, **не пользующихся спро**-**сом**.
3. **Запись содержит сведения о перевозках авиапассажиров на рейсах аэропорта**: **номер рейса**, **маршрут**, **марка самолета**, **общие затраты на рейс**, **количество пассажиров**. **Подсчитать стоимость перевозки одного пассажира на рейсе**. **Для каждой марки самолета выдать список с указанием затрат**, **количества пассажиров и стоимости перевозки одного пассажира**.
4. **Запись содержит сведения о результатах медосмотра**: **фамилия и инициалы**, **рост**, **вес**. **Для каждого человека определить идеальный вес и выдать списки людей**, **чей вес близок к идеальному** (±3 **кг**), **и тех**, **у кого он избыточен**.

***Указание к решению***

**Идеальный вес определяется в зависимости от роста**:

|  |  |
| --- | --- |
| **рост** (**см**) | **идеальный вес** (**кг**) |
| **до** 155 | **вес** – 50 |
| **до** 165 | **вес** – 60 |
| **до** 175 | **вес** – 70 |
| **свыше** 175 | **вес** – 85 |

1. **Запись содержит сведения о перевозках авиапассажиров на рейсах аэропорта**: **номер рейса**, **маршрут**, **марка самолета**, **общие затраты на рейс**, **количество пассажиров**. **Подсчитать стоимость перевозки одного пассажира на рейсе и среднюю стоимость перевозки одно**-**го пассажира по аэропорту**. **Выдать списки номеров рейсов**, **для которых стоимость перевозки одного пассажира ниже и выше средней по аэропорту**.
2. **Запись содержит сведения о расходе топлива на автопредприятиях города**: **название предприятия**, **количество израсходованного топ**-**лива и количество автомашин на предприятии**. **Подсчитать сред**-

**ний расход топлива на одну машину на каждом предприятии и в**

82

**целом по городу**. **Выдать списки предприятий**, **у которых расход топлива ниже и выше среднего по городу**.

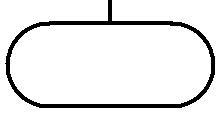
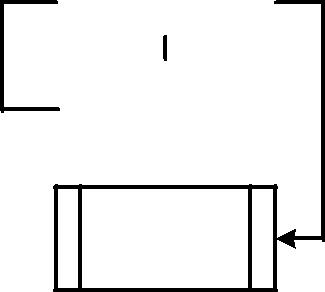
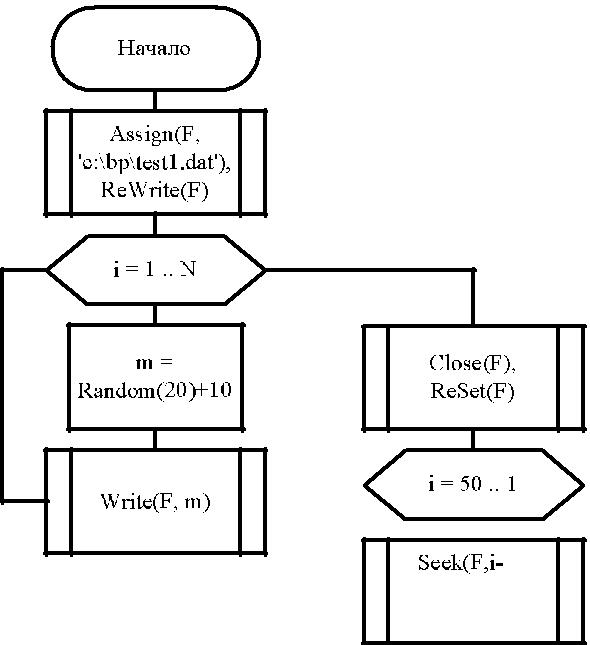
1. **Запись содержит данные о книгах**: **автор**, **название**, **издательство**, **год издания**, **цена**. **Получить список книг**, **изданны**x **в указанный год**, **и список книг заданного автора**.
2. **Запись содержит сведения о погоде в городах России**: **город**, **днев**-**ная и ночная температуры**. **Для каждого города определить раз**-**ность температур и выдать списки городов**, **в которых дневная температура выше нуля**, **а ночная** – **ниже и в которых разность температур превышает** 5**оС**.
3. **Запись содержит сведения о странах мира**: **название страны**, **сто**-**лица**, **территория** (**тыс**.**кв**. **км**), **население** (**тыс**. **чел**.). **Для каждой страны определить плотность населения и выдать списки стран с наименьшей** (**меньше минимума** +20 **тыс**. **чел**.) **и наибольшей** (**больше максимума** –50 **тыс**. **чел**.) **численностью населения**.

**Пример программы на языке Turbo Pascal**

**Создать файл**, **содержащий** 50 **случайных целых чисел в диапазоне** [10..30], **вывести числа на экран в обратном порядке**.

C**хема алгоритма работы программы представлена на рисунке** 14.

83



**Рис**. 14 – C**хема алгоритма работы программы**

Program Primer7;

const

N=50;

var

1. file of integer; {**Объявление файла с целыми числами**} i, m : integer;

begin

Assign(F,'c:\bp\test1.dat');

{**Связь файловой переменной с файлом**} ReWrite(F); {**Открыть файл для записи**} for i:=1 to N do

begin

m := Random(20)+10; {**Получение случайных чисел**} Write(F, m); {**Запись числа в файл**}

84

end;

Close(F); {**Закрытие файла**}

ReSet(F); {**Открыть тот же файл для чтения**} for i:=50 downto 1 do

begin

Seek(F,i-1); {**Установка указателя файла**}

Read(F,m); {**Считывание числа из файла**}

Write(m:4); {**Вывод числа на экран**}

end;

Close(F); {**Закрытие файла**}

WriteLn;

end.

**Контрольные вопросы**

1. **Что такое** «**файл**» **и** «**файловая переменная**»?
2. **Что такое** «**типизированный файл**»?
3. **Что такое** «**текстовый файл**»?
4. **Что такое** «**нетипизированный файл**»?
5. **Что такое** «**указатель файла**»?
6. **Как установить указатель файла в требуемую позицию**?
7. **Для каких типов файлов можно устанавливать позицию указателя файла**?
8. **Какие процедуры предназначены для открытия типизированного файла**?
9. **Какие процедуры предназначены для открытия текстового файла**?
10. **Какие процедуры и в какой последовательности надо вызывать для чтения данных из типизированного файла**?
11. **Какими процедурами выполняется чтение и запись для нетипизи**-**рованных файлов**?
12. **Какими процедурами выполняется чтение и запись для текстовых файлов**?
13. **Как задать размер блока для нетипизированного файла**?
14. **Какие функции позволяют узнать текущее положение указателя в файле**?
15. **Как определить возникновение какой**-**либо ошибки при работе с файлами**?

85

**Рекомендованная литература**

1. **Рапаков**, **Г**. **Г**. **Программирование на языке** Pascal / **Г**. **Г**. **Рапаков**, **С**. **Ю**. **Ржеуцкая**. – **СПб**. : **БХВ**-**Петербург**, 2005. – 470 **с**.

2. **Потопахин**, **В**. **В**. Turbo Pascal. **Освой на примерах** / **В**. **В**. **Потопахин**. – **СПб**. : BHV, 2005. – 240 **с**.

1. **Малыхина**, **М**. **П**. **Программирование на языке высокого уровня**

Turbo Pascal / **М**. **П**. **Малыхина**. – **СПб**. : **БХВ**-**Петербург**, 2006. – 523 **с**.

1. **Потопахин**, **В**. **В**. Turbo Pascal. **Решение сложных задач** /

**В**. **В**. **Потопахин**. – **СПб**. : BHV, 2006. – 208 **с**.

1. **Марченко**, **А**. **И**. **Программирование в среде** Turbo Pascal 7.0 /

**А**. **И**. **Марченко**, **Л**. **А**. **Марченко**. – **М**. : **Век**, 2007. – 464 **с**.

6. **Фаронов**, **В**. **В**. Turbo Pascal 7.0. **Начальный курс**. / **В**. **В**. **Фаронов**. – **М**. : **КноРус**, 2007. – 576 **с**.

1. **Немнюгин**, **С**. **А**. Turbo Pascal: **практикум** / **С**. **А**. **Немнюгин**. –

**СПб**. : **Питер**, 2007. – 267 **с**.

1. **Фаронов**, **В**. **В**. Turbo Pascal / **В**. **В**. **Фаронов**. – **СПб**. : **БХВ**-

**Петербург**, 2007. – 1037 **с**.

1. **Вольский**, **С**. **В**. Turbo Pascal 7.0 **для студентов и школьников** / **С**. **В**. **Вольский**, **П**. **А**. **Дмитриев**. – **М**. : **Наука и техника**, 2007. –214 **с**.

10. **Культин**, **Н**. **Б**. Turbo Pascal **в задачах и примерах** / **Н**. **Б**. **Культин**. – **СПб**. : **БХВ**-**Петербург**, 2008. – 256 **с**.

1. **Фаронов**, **В**. **В**. Turbo Pascal 7.0. **Практика программирования** /

|  |  |  |
| --- | --- | --- |
|  | **В**. **В**. **Фаронов**. – **М**. : **КноРус**, 2008. – 576 **с**. |  |
| 12. | **Немнюгин**, **С**. **А**. **Изучаем** Turbo Pascal / | **С**. **А**. **Немнюгин**, |
|  | **Л**. **В**. **Перколаб**. – **СПб**. : **Питер**, 2008. – 312 **с**. |  |
| 13. | **Шпак**, **Ю**. **А**. Turbo Pascal. **Просто как** | **дважды два** / |
|  | **Ю**. **А**. **Шпак**. – **М**. : **ЭКСМО**, 2008. – 396 **с**. |  |

1. **Меженный**, **О**. **А**. Turbo Pascal: **самоучитель** / **О**. **А**. **Меженный**. –

**М**. : **Диалектика**, 2008. – 335 **с**.

1. **Немнюгин**, **С**. **А**. **Программирование на языке высокого уровня** / **С**. **А**. **Немнюгин**. – **СПб**. : **Питер**, 2008. – 543 **с**.
2. **Культин**, **Н**. **Б**. **Программирование в** Turbo Pascal 7.0 **и** Delphi / **Н**. **Б**. **Культин**. – **СПб**. : **БХВ**-**Петербург**, 2008. – 380 **с**.
3. **Ковтанюк**, **Ю**. **С Программирование на** Turbo Pascal: **шаг за ша**-

**гом** / **Ю**. **С**. **Ковтанюк**. – **М**. : **ЭКСМО**, 2008. – 587 **с**.

86