



Кафедра информатики и
информационных технологий

Б1.О.17 АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ

Методические указания к лабораторным работам и самостоятельной работе
обучающихся

ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ ПАСКАЛЬ

для направления подготовки 09.03.03 Прикладная информатика
профиль подготовки Прикладная информатика цифровой экономики

Квалификация выпускника
бакалавр

Уфа 2021

Рекомендовано к изданию методической комиссией экономического факультета 25 марта 2021 г. (протокол №8).

Составитель: доцент Шамсутдинова Т.М.

Ответственный за выпуск:

Заведующий кафедрой информатики и информационных технологий, д.т.н.
доцент Беляева А.С.

г. Уфа, ФГБОУ ВО Башкирский ГАУ, кафедра информатики и информационных технологий

ОГЛАВЛЕНИЕ

Введение	4
Лабораторная работа №1 Основы языка программирования Pascal.	5
Операторы ветвления	
Лабораторная работа №2 Операторы циклов языка Pascal	20
Лабораторная работа №3 Вложенные циклы	28
Лабораторная работа №4 Использование одномерных массивов в среде программирования Pascal	31
Лабораторная работа №5 Использование двумерных массивов в среде Pascal	38
Лабораторная работа №6 Работа со строковыми переменными	43
Лабораторная работа №7 Подпрограммы языка Pascal	49
Библиографический список	57
Приложение А Команды редактора среды Pascal	58
Приложение Б Сообщения и коды некоторых ошибок	63
Приложение В Варианты задач по теме «Линейные структуры»	65
Приложение Г Задачи для самостоятельного решения по теме «Линейные структуры»	67
Приложение Д Варианты задач по теме «Операторы ветвления»	68
Приложение Е Варианты задач по теме «Цикл с параметром»	71
Приложение Ж Варианты задач по теме «Итерационные циклы»	72
Приложение З Задачи для самостоятельного решения по теме «Операторы циклов»	73
Приложение И Варианты задач по теме «Вложенные циклы. Вычисление функции»	75
Приложение К Варианты задач по теме «Вычисление суммы ряда с использованием вложенных циклов»	76
Приложение Л Задачи для самостоятельного решения по теме «Вложенные циклы»	78
Приложение М Варианты заданий по теме «Одномерные массивы»	79
Приложение Н Задачи для самостоятельного решения по теме «Одномерные массивы»	80
Приложение О Варианты заданий по теме «Нахождение итоговых значений в матрицах»	82
Приложение П Варианты заданий по теме «Замена и перестановка элементов матриц»	83
Приложение Р Задачи для самостоятельного решения по теме «Двумерные массивы»	84
Приложение С Варианты заданий по теме «Обработка строки символов»	86
Приложение Т Варианты заданий по теме «Обработка списка строк»	87
Приложение У Задачи для самостоятельного решения по теме «Строковые переменные»	88
Приложение Ф Варианты заданий по теме «Подпрограммы функции»	90
Приложение Х Задачи для самостоятельного решения по теме «Подпрограммы»	91

Введение

Очевидно, что не так уж сложно обучиться некоторым конструкциям какого-либо языка программирования, правилам использования его операторов. Гораздо сложнее приобрести навыки алгоритмического мышления, научиться различным типовым алгоритмам решения задач и особенностям их программной реализации.

Цель данного лабораторного практикума - попытаться всесторонне осветить различные разделы теории программирования на языке Pascal 7.0. Данные указания содержат большое количество примеров, призванных проиллюстрировать типовые приемы решения задач на ЭВМ - работу с циклами, массивами данных, строковыми переменными, подпрограммами и т.д. В лабораторном практикуме в виде приложений предлагаются команды работы в среде Pascal, разнообразные задачи, коды и сообщения некоторых ошибок компиляции программ и др.

В лабораторном практикуме также предусматриваются основные формы контроля полученных знаний: контрольные вопросы в конце каждой из лабораторных работ и задачи для самостоятельного выполнения.

Лабораторная работа «Основы языка программирования Pascal. Операторы ветвления» написана ст. преподавателем Саитовой Э.С., лабораторные работы «Операторы циклов языка Pascal», «Вложенные циклы» - доцентом Дидык Т.Г., лабораторные работы «Использование одномерных массивов в среде программирования Pascal», «Использование двумерных массивов среде Pascal», «Работа со строковыми переменными» - доцентом Шамсутдиновой Т.М. Работа «Подпрограммы языка Pascal» написана Шамсутдиновой Т.М. с использованием материалов Дергач Ю.Г.

Надеемся, что данные методические указания помогут студентам в приобретении и дальнейшем совершенствовании навыков программирования.

Требования к организации рабочего места: Все лабораторные работы должны проводиться в компьютерном классе с установленной средой программирования Pascal ABC.

Лабораторная работа №1

Основы языка программирования Pascal. Операторы ветвления

Цель работы

Ознакомиться со средой программирования Pascal. Изучить основные типы данных, структуру программы, стандартные функции, линейные и разветвляющиеся алгоритмы.

Задачи работы

Научиться создавать программы на языке Pascal с использованием стандартных функций и операторов линейной и разветвляющейся структуры.

1 Общие положения

1.1 Основные элементы языка Pascal

1.1.1 Основные символы

Алфавит языка Pascal состоит из основных символов, которые подразделяются на буквы, цифры и специальные символы:

- Буквы: от **A** до **Z** (26 букв латинского алфавита, причем транслятор языка Pascal не различает прописные и строчные буквы); знак **_** (подчеркивание);
- Цифры: 0 1 2 3 4 5 6 7 8 9;
- Специальные символы: + - * / = < > () { } [] . , ; : # \$ ^ @ ' ,

Неделимые последовательности символов алфавита образуют слова – **идентификаторы**, которые используются для обозначения констант, переменных, процедур, функций и т.д. Имя идентификатора должно начинаться с буквы либо символа подчеркивания. Например: name_1, _summation и т. д.

1.1.2 Зарезервированные слова

Зарезервированные слова являются составной частью языка Pascal, которые нельзя переопределять и использовать в качестве идентификаторов, определяемых пользователем.

Основные зарезервированные слова языка Pascal:

Absolute	And	Array	Assembler
Begin	Break	Case	Const
Constructor	Continue	Destructor	Div
Do	Down To	Else	End
External	Far	File	For
Function	Goto	If	Implementation
In	Inline	Interface	Interrupt
Label	Mod	Near	Nil
Not	Object	Of	Or
Paced	Private	Procedure	Program
Public	Record	Repeat	Set
Shl	Shr	String	Then
To	Type	Unit	Until
Uses	Var	Virtual	While
With	Xor		

Слово в языке Pascal имеет определенный смысл. Элементы языка должны разделяться друг от друга пробелами или другими спецсимволами (разграничителями).

1.2 Стандартные типы данных

Любые данные (переменные, константы, значения функций и т.д.) характеризуются в Pascal своими типами. Тип данных определяет:

- формат представления данных в памяти компьютера, включая объем памяти, отводимой под хранение значений переменных;
- множество допустимых значений, которые может принимать принадлежащая к выбранному типу переменная или константа;
- множество допустимых операций, применимых к этому типу.

Типы данных на Pascal делятся на пять классов:

- простые;
- структурированные;
- ссылочные;
- процедурные;
- объектные.

Каждой переменной в программе необходимо задать свой тип данных. Все сложные типы данных в Pascal составляются из простых (неструктурированных) типов.

Простой тип данных может определяться самим программистом (будет называться описанным скалярным типом – *declared scalar type*). Можно использовать стандартные скалярные типы:

- целочисленный;
- вещественный;
- булевский или логический;
- символьный или литеральный;
- перечисляемый;
- интервальный.

1.2.1 Целочисленный тип

Целочисленные типы данных – это пять предопределяемых типов (таблица 1), используемых для работы с целыми числами.

Таблица 1 Таблица целочисленных типов данных

Тип	Допустимые значения	Длина в байтах
ShortInt (короткое целое)	-128...127	1 байт со знаком
Integer (целое)	-32768...32767	2 байта со знаком
LongInt (длинное целое)	-2147483648...2147483647	4 байта со знаком
Byte (длиной в байт)	0...255	1 байт без знака
Word (длиной в слово)	0...65535	2 байта без знака

1.2.2 Вещественный тип

Для чисел, в записи которых присутствует десятичная точка, в языке Pascal существует особый тип данных – вещественный. К этому типу данных относится подмножество вещественных чисел, которые могут быть представлены в формате с плавающей десятичной точкой.

В Pascal определено пять стандартных вещественных типов (таблица 2):

Таблица 2 Таблица вещественных типов данных

Тип	Точность	Длина в байтах
Real (вещественный)	11-12 знаков	6 байт
Single (с одинарной точностью)	7-8 знаков	4 байта
Double (с двойной точностью)	15-16 знаков	8 байт
Extended (с повышенной точностью)	19-20 знаков	10 байт
Comp (сложный)	19-20 знаков	8 байт

Переполнение, возникающее при выполнении арифметических операций над данными вещественного типа, вызывает остановку программы и на экран выводится сообщение об ошибке. В случае выхода за нижнюю границу интервала результату присваивается нулевое значение.

1.2.3 Булевский или логический тип

Переменные типа **BOOLEAN** являются логическими переменными и могут принимать только два значения **TRUE** (истина) и **FALSE** (ложь).

Значения булевского типа **TRUE** (истина) и **FALSE** (ложь) являются по сути идентификаторами констант. Логические переменные могут использоваться для хранения результатов каких-либо логических вычислений. Для булевых переменных, принимающих одно из двух допустимых значений, разрешаются только две операции сравнения “=” (равно) и “<” (не равно).

Допустимые для логических переменных операции:

AND - Логическое **И**

OR - Логическое **ИЛИ**

XOR - Логическое исключающее **ИЛИ**

NOT - Логическое **НЕ**

3.2.4 Символьный тип

Символьный тип **CHAR** – тип данных, предназначенный для хранения одного символа (буквы, знака или кода). Переменные такого типа могут представить буквы от “A” до “Z” (‘A’...’Z’, ‘a’...’z’), цифры от “0” до “9”, знаки препинания и специальные символы.

Переменные типа **CHAR** в памяти занимают 1 байт, так же как и переменные типа **BYTE**. Значения для переменных типа **CHAR** задаются, как и строки символов (‘ ’) – в апострофах.

1.2.5 Перечисляемый тип

Определяется этот тип как упорядоченный набор идентификаторов, заданных путем их перечисления (список идентификаторов, разделенных запятой, указывается в круглых скобках).

Значения переменных перечисляемых типов не могут вводиться оператором ввода (Readln) и выводиться оператором вывода (Writeln).

1.2.6 Интервальный тип

При описании переменных в программе, как правило, известно, что они будут использованы для представления подмножества значений некоторого типа.

Интервальный тип данных определяется посредством задания подмножества значений одного из ранее определенных типов. В Pascal диапазон значений переменной интервального типа задается с помощью любого простого типа данных за исключением вещественного.

При задании диапазона указывается наименьшее и наибольшее значения, которые может принимать переменная соответствующего типа (обе константы должны быть одного типа).

Например, с помощью объявления

Type znach=0..100

можно создать новый интервальный тип данных, значения которого являются целыми числами в интервале от 0 до 100.

1.3 Переменные

Переменная – величина, к которой обращаются с помощью идентификатора (символического имени) и которая может принимать различные значения в процессе выполнения программы. Имя переменной всегда должно начинаться с буквы или символа подчеркивания, далее можно записывать буквы, цифры и специальные символы. Не допускается использование пробела в имени переменной. Имя переменной не должно совпадать со служебным словом. Все переменные, используемые в программе, процедуре или функции, описываются в соответствующем разделе описания переменных. При описании указывается имя переменной и ее тип.

Различают **простые** переменные и **структурированные**. **Простая переменная** – это величина, имеющая одно текущее значение, обозначенная определенным именем. **Структурированная переменная** – группа величин, обозначенная символическим именем.

1.4 Константы

Константа – величина, которая не изменяется в процессе вычислений.

Различают **именованные** и **обычные** константы. **Именованная константа** – константа, которой в начале программы в разделе описаний дается имя и имеет следующий вид:

Const <Имя константы> = <Значение константы>;

Например,

Const Razmer = 5;

Обычные записываются непосредственно в тексте программы. Тип константы определяется формой ее записи.

- Константа целого типа записывается в виде последовательности десятичных цифр: 123 -56 +457 и т. д.
- Вещественная константа – последовательность десятичных цифры и точки, которая отделяет целую часть от дробной, или последовательность десятичных чисел с порядком десятичного числа: 236.0 2.36E-02 0.0001 -10.0 -10.0E05 и т. д., где E – основание десятичного числа. Наличие десятичной точки в записи вещественной константы обязательно.
- Логическая константа может принимать только два значения TRUE (истина) и FALSE (ложь)
- Текстовая константа (строка) – любая последовательность символов, заключенная в апострофы.

Существуют также так называемые *типизированные константы*, однозначные переменным с заранее заданным значением. Название вызвано тем, что при описании указывается тип:

Const <Имя константы : Тип> = <Значение константы>;

Например,

Ocenka :byte = 5;

Predmet : string = ' Информатика';

Использование типизированных констант позволяет поменять сразу много значений ко всему тексту (столько значений, сколько раз встречается в тексте данные константы), внося изменения только в одном месте программы – разделе Const.

1.5 Выражения

Арифметические выражения – это константы, переменные и функции, соединенные знаками арифметических операций. Для указания порядка вычислений используются круглые скобки. Дробное выражение записывается в одну строку, верхние и нижние индексы не допускаются. Выражения вычисляются слева направо, с учетом приоритета операций.

Некоторые, используемые в Pascal операции:

+ Сложение

- Вычитание

* Умножение

/ Деление

div Целая часть от деления

mod Остаток от деления

Примеры: $7 \text{ div } 2 = 3$ $(-7) \text{ div } 2 = -3$ $3 \text{ div } 5 = 0$ $(-7) \text{ div } (-2) = 3$

$7 \text{ mod } 2 = 1$ $(-14) \text{ mod } 3 = -2$ $3 \text{ mod } 5 = 3$ $(-10) \text{ mod } 5 = 0$

Приоритет операций в порядке убывания следующий:

- Вычисление выражения в скобках; вычисление функций;
- Умножение, деление операндов вещественного или вещественно – целого типов, деление операндов целого типа “div” и “mod”;
- Сложение, вычитание.

Таблица 3 Стандартные функции языка Pascal

Abs (x)	x – целое, веществ.	Абсолютная величина X (модуль X)
Sqr (x)	x – целое, веществ.	Возведение X в квадрат

Sqrt (x)	x – целое, веществ.	Корень квадратный числа X
Sin (x)	x – целое, веществ.	Синус числа X
Cos (x)	x – целое, веществ.	Косинус числа X
Arctan (x)	x – целое, веществ.	Арктангенс числа X
Ln (x)	x – целое, веществ.	Натуральный логарифм числа X
Exp (x)	x – целое, веществ.	Экспонента числа X (e^x)
Frac (x)	x – веществ.	Дробная часть числа X
Trunc (x)	x – веществ.	Выделение целой части числа X
Round (x)	x – веществ.	Округление X до целого числа
Succ (x)	x – целое	Следующее за X число
Pred (x)	x – целое	Предыдущее перед X число
Int (x)	x – веществ.	Выделяет целую часть вещ. числа X
Inc (x, n)	x – целое	Увеличение значения X на n
Dec (x, n)	x – целое	Уменьшение значения X на n
Odd (x)	x – целое	Проверка на нечетность числа (результат – логический)
Random (x)	x – целое	Случайное число от 0 до (x-1). Если функция не содержит аргумента, то генерируется случайное число от 0 до 1
Randomize;	Оператор, позволяющий генерировать новую последовательность случайных чисел при новом запуске программы на выполнение	
Pi	$\pi = 3.14\dots$	

Пример: $\text{trunc}(7.354) = 7$ $\text{round}(7.354) = 7$
 $\text{trunc}(-7.354) = -7$ $\text{round}(-7.354) = -7$
 $\text{trunc}(-1.8) = -1$ $\text{round}(-1.8) = -2$

Логические выражения – это арифметические константы, переменные и выражения, соединенные знаками операций отношения, а также логические константы, переменные, соединенные знаками логических операций. Возможны также и их комбинации.

Примеры: $a \diamond (x + \ln(x))$;

$\text{Abs}(f) \leq \text{eps}$;

$(\text{not } x1 \text{ or } x2) \text{ and } ((x3 \text{ or } x4) \text{ and } x1)$;

$(\text{abs}(f) \leq \text{eps}) \text{ or } (\text{abs}(x1-x2) \leq \text{eps})$.

Результатом выполнения логического выражения является одно из двух значений: true или false.

Логические операции «not» (отрицание), «and» (логическое умножение), «or» (логическое сложение) перечислены в порядке убывания приоритета.

1.6 Структура программы

Любую программу, написанную на Pascal можно условно разделить на следующие основные части:

- заголовок программы
- раздел описаний (описательная часть)
- раздел основного блока (исполнительная часть), включающая: раздел инструкций (операторов) программы, заключаемые в слова *begin* и *end*. В конце программы ставится признак остановки – точка (.)

В самом общем виде структура программы имеет вид:

PROGRAM	Заголовок программы;
USES	Подключение библиотеки;
Раздел описаний (описательная часть)	
LABEL	Подраздел объявления глобальных меток;
CONST	Подраздел объявления глобальных констант;
TYPE	Подраздел объявления глобальных типов;
VAR	Подраздел объявления глобальных переменных;
PROCEDURE	Объявление подпрограммы процедуры;
FUNCTION	Объявление подпрограммы функции;
Раздел основного блока программы (исполнительная часть)	
BEGIN	Начало основного блока программы
Раздел операторов (инструкций) программы;	
END.	Конец.

1.6.1 Заголовок программы

В Pascal заголовок играет часто вспомогательную функцию. Если заголовок присутствует, то он задает имя программы.

Примеры: **Program** primer1;
Program zadacha_22;

Имя программе присваивается пользователем для удобства работы. Оно позволяет отличать одну программу от другой, и никак не связано с именем файла, содержащим ее текст. Имя программы не должно совпадать с именами переменных, используемых в программе.

1.6.2 Раздел описания

В разделе описания, входящего в блок программы, описываются все идентификаторы, которые будут использоваться в разделе операторов данного блока программы (и, возможно, других блоков, находящихся внутри него). Раздел описания состоит из пяти различных разделов:

- раздел описания меток
- раздел определения констант
- раздел определения типов
- раздел описания переменных
- раздел описания процедур и функций

Разделы описания в программе могут располагаться в любом порядке, некоторые могут и отсутствовать.

Раздел описания меток. Перед любым оператором программы можно поставить метку, что позволяет выполнить безусловный переход на этот оператор при выполнении команды *goto*. Метка состоит из имени метки и следующего за ним двоеточия. Раздел описания меток имеет следующую структуру: **Label <метка>;**

Пример:

Label 10, M5, 999;

В качестве меток могут использоваться как числа, так и идентификаторы.

Раздел определения констант. Описываются все именованные константы, значения которых входе выполнения программы постоянны.

Пример:

Const Max=1024; A='студент'; Sbol= True;

Раздел определения типов. Раздел описания типов данных предназначен для описания типов, определяемых пользователем. Раздел начинается с зарезервированного слова **type** и имеет вид:

type

<ИмяТипа1> = <Описание Типа1>;

<ИмяТипа2 >= <Описание Типа2>;

Пример:

Type

Days =1..31;

Month = (jan, feb, ..., dec);

Number = Integer;

Matr = array[1..10, 1..12] of real;

Раздел описания переменных. Все используемые в программе переменные должны быть перечислены в соответствующем разделе. Описание переменных производится после служебного слова **Var** (от англ. **Variable** – переменная).

Инструкция объявления переменной выглядит так:

<Имя Переменной1, ..., Имя ПеременнойN> : <Тип Переменной>;

Пример:

Var I, j, k : integer;

C, f, min : real;

z : Boolean;

chess : char;

A, B : array [0..8] of real;

Раздел описания процедур и функций. В практике программирования часто возникает необходимость выполнения одной и той же последовательности операторов в различных частях программы. Наиболее простым способом реализации программы является запись последовательности операторов в тех участках программы, где это необходимо. Но целесообразней оформить эти операторы в виде подпрограммы – процедуры или подпрограммы – функции. Процедуры и функции позволяют записать повторяющиеся участки один раз, а в нужных местах программы приводить лишь обращение к ним. Использование процедур позволяет сократить объем, улучшить структуру программы и уменьшить вероятность наличия ошибок.

1.6.3 Раздел операторов

Раздел операторов является последним в блоке программы и задает действия, которые должна выполнить программа. Этот раздел имеет вид

сложного составного оператора, который начинается со слова **begin** и заканчивается словом **end**, после которого ставится точка.

1.7 Основные операторы языка Pascal

1.7.1 Оператор присваивания

Используется этот оператор для присвоения значения переменной и имеет следующий вид:

<ИмяПеременной> := <выражение>;

Знак **:=** читается как «присвоить значение». Частным случаем выражения, стоящего в правой части, являются переменные и константы. Переменная и выражение должны быть совместимы по типу.

Пример: **A := B;**

I := I + 1;

Y := sqrt(x) * 121 / (25 + x);

1.7.2 Операторы ввода данных

Ввод данных – это передача исходных данных программы в оперативную память компьютера для обработки.

Для ввода данных в Pascal предусмотрены следующие операторы:

- **READ (a1, a2, a3, ..., aN);**

где a1, ..., aN - переменные, которые последовательно принимают значения, вводимые с клавиатуры;

- **READLN (a1, a2, a3, ..., aN);**

где a1, ..., aN - переменные, которые последовательно принимают значения, вводимые с клавиатуры; после ввода последнего из них происходит переход на новую строку.

1.7.3 Операторы вывода данных

Вывод данных – это передача данных после обработки из оперативной памяти на внешнее устройство (экран, принтер, файл на диске). Для этого используются операторы:

- **WRITE (b1, b2, b3, ..., bN);**

где b1, ..., bN – список выражений, значения которых выводятся на экран;

- **WRITE LN (b1, b2, b3, ..., bN);**

где b1, ..., bN - выводимый на экран список выражений, после вывода последнего из которых происходит переход на новую строку.

Кроме значение выражений можно выводить на экран и произвольный набор символов, заключенный в апострофы. Например, **WRITE ('Сумма=', s);**

Пример 1. Написать программу для вычисления площади круга (S), объема (V) шара и длины окружности (L) с одним и тем же радиусом (R).

```
program okrug;
```

```
var R, S, L, V : real;
```

```
begin
```

```
  write ('Введите радиус круга R ');
```

```
  readln (R);
```

```
  S:= Pi*sqr(R);
```

```
  writeln ('Площадь окружности S= ', S);
```

```
  L:= 2*Pi*R ;
```

```
writeln ('Длина окружности L = ', L);
V:= 4/3*Pi*sqr(R)*R;
writeln ('Объем окружности V= ', V);
end.
```

При использовании оператора вывода для представления на экране значения целого типа выделяется столько позиций экрана (разрядов), сколько требуется для записи числа. Вывод значения вещественного типа осуществляется в экспоненциальном виде (с использованием степени десяти, например, 1.3E+01). Такой вид результатов на экране часто не устраивает пользователей, поэтому необходимо позаботиться о приемлемом **формате** вывода.

В операторах вывода имеется возможность записи выражения, определяющего **ширину поля** вывода для каждой выводимой переменной или константы:

```
: a – формат вывода переменных целого типа;
: a : d - формат вывода переменных вещественного типа,
```

где a задает общую ширину поля (общее количество позиций, отводимых под число, в том числе для знака числа и десятичной точки), d – место под дробную часть. Разделителем в формате вывода является знак “ : ” (двоеточие). Количество символов в дробной части числа будет равен d, а общее количество символов вывода числа будет равно a. Если в операторе вывода формат не указан, то вывод будет осуществляться в нормализованной форме представления вещественного числа.

Например, если для переменной S:=66.25183 оператор вывода имеет следующий вид:

```
Writeln (' S= ', S:5:2),
```

то результат в приведенном формате будет выглядеть как S=66.25 (для вывода числа отведено 5 позиций всего, а для дробной части отведено 2 позиции).

1.8 Операторы ветвления

1.8.1 Условный оператор

Условный оператор позволяет выполнить один из входящих в его состав операторов в зависимости от выполнения какого-либо условия.

Оператор IF определяет, что тот или иной оператор должен выполняться в том случае, если справедливо заданное условие, т.е. выражение принимает значение true (истина). Если условие не выполняется, то выполняется оператор, следующий за служебным словом **else** (иначе).

Существует полная и краткая форма оператора. **Полная форма** условного оператора **IF** имеет следующий вид:

```
IF <условие> THEN <оператор1> ELSE <оператор2>;
```

где **IF** – в переводе с англ. «если», <условие> - некоторое логическое выражение, **THEN** – в переводе с англ. «тогда», **ELSE** – «иначе».

Если условие истинно, то выполняется <оператор1>. Если условие ложно, то выполняется <оператор2>.

Любая встретившаяся часть **ELSE** соответствует ближайшей к ней «сверху» части **THEN** условного оператора.

Действие условного оператора **IF** можно пояснить с помощью блок – схемы (рисунок 1):

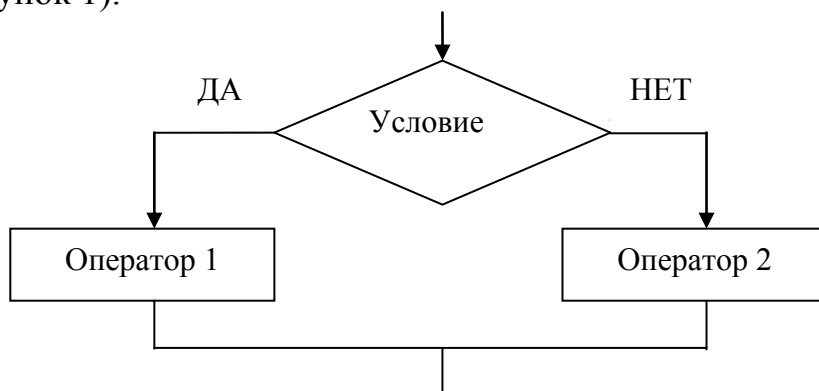


Рисунок 1- Блок-схема алгоритма выполнения полной формы оператора IF

Если за словами **ELSE** и **THEN** необходимо выполнить группу операторов, то тогда используются операторные скобки **begin ... end**. Такая структура называется **вложенной**:

```

if <условие > then
  begin
    <оператор1>;
    <оператор2>;
    ....
    <оператор N>
  end
else
  begin
    <оператор1>;
    <оператор2>;
    ....
    <оператор M>
  end ;

```

Пример 2. Дано число x . Определить является ли число x четным числом.

```

program chislo;
var x : integer;
begin
  write ('Введите число x ');
  readln (x);
  if x mod 2 =0 then writeln ('четное ') else writeln ('нечетное');
end.

```

Краткая форма оператора **IF** имеет следующий вид:

IF <условие> THEN <оператор>;

Если логическое выражение истинно, то выполняется **<оператор>**, в противном случае программа переходит к выполнению операторов, следующих

за оператором **IF**. Если условие ложно, то выполняются следующие за **IF** операторы.

Блок-схема алгоритма, соответствующая краткой форме условного оператора показана на рисунке 2:

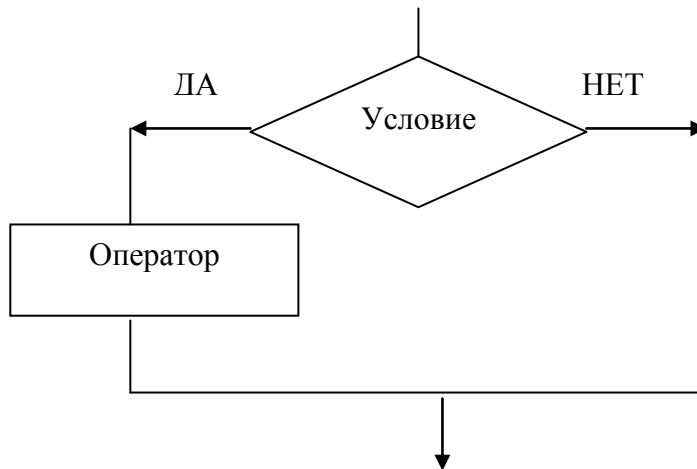


Рисунок 2 - Блок-схема алгоритма оператора IF в краткой форме

1.8.2 Оператор выбора CASE

Условий, по которым организуются ветвления в программе, может быть не два, а больше. Для задания множественного выбора служит оператор CASE. В общем виде он записывается следующим образом:

```

Case <выражение> of
    <список 1>: <оператор 1>;
    <список 2>: <оператор 2>;
    ...
    <список n>: <оператор n>
    else    <оператор n+1>
end;
  
```

Здесь <выражение> - это выражение, от значения которого зависит дальнейший ход выполнения программы, <список > - это список констант, разделенных запятыми. Если константа представляет диапазон чисел, то вместо списка можно указать первую и последнюю константы диапазона, разделив их двумя точками. Обратите внимание – в конце оператора case стоит ключевое слово end, для которого нет парного слова begin.

Пример: **Case a of**

```

    1,5,7..10: c:=b+a;
    2,4: c:=b/a;
    20..100: c:=b-a;
end;
  
```

Выполняется оператор case следующим образом:

1. Вычисляется значение выражения, следующего за словом **case**.
2. Полученные значения последовательно сравниваются с константами из списков констант перед двоеточием. После этого возможен один из следующих вариантов действий:

- если значение выражения совпадает с константой из списка, то выполняется соответствующая этому списку последовательность инструкций и выполнение инструкции **case** завершается;
- если значение выражения не совпадает ни с одной константой из всех списков, то выполняется последовательность инструкций, следующая за словом **else**, и выполнение инструкции **case** завершается;
- если блока **else** в инструкции **case** нет, то ее выполнение завершается.

Пример 3. Написать программу для вывода названий дней недели по их порядковому номеру.

```

Program nedeli;
Var number: integer;
Begin
    Write ('Введите номер дня недели');
    Readln (number);
    Case number of
        1: writeln ('понедельник');
        2: writeln ('вторник');
        3: writeln ('среда');
        4: writeln ('четверг');
        5: writeln ('пятница');
        6: writeln ('суббота');
        7: writeln ('воскресенье');
    end
end.

```

1.9 Среда программирования Pascal

Для запуска среды программирования Pascal необходимо загрузить файл turbo.exe (для среды Turbo Pascal) или bp.exe (для среды Borland Pascal). Создание нового файла в среде Pascal осуществляется командой File►New. По умолчанию, первому из созданных файлов присваивается имя NONAME00.PAS, следующему - NONAME01.PAS и т.д.

Для сохранения файла на текущем месте диска используется команда File►Save (либо функциональная клавиша [F2]).

После подготовки текста программы ее необходимо откомпилировать, связать с библиотекой стандартных программ и функций, загрузить в оперативную память и выполнить. Эта последовательность действий – прогон программы – осуществляется с помощью команды Run►Run главного меню Pascal или комбинацией клавиш [Ctrl]+[F9]

Прервать выполнение программы можно, нажав комбинацию клавиш [Ctrl]+[Break]. Результаты вывода на экран можно посмотреть командой [Alt]+[F5].

Выход из среды Pascal – команда File►Exit либо комбинация клавиш [ALT]+[X].

Более подробное описание команд создания, отладки, запуска программ приводится в приложении А. Описание кодов и сообщений некоторых ошибок компиляции программ находится в приложении Б.

2 Содержание работы

2.1 Вычислить расстояние между двумя точками с координатами (x_1, y_1) и (x_2, y_2) .

Возможный вид программы расчета:

```
Program primer;
var x1, x2, y1, y2, d: real;
begin
    write ('Введите координаты 1-ой точки (x1,y1)');
    readln (x1,y1);
    write ('Введите координаты 2-ой точки (x2,y2)');
    readln (x2,y2);
    d:= sqrt (sqr (x2-x1) + sqr (y2-y1));
    write (' Расстояние между точками = ', d:6:4)
end.
```

2.2 Составить и отладить на компьютере программы решения задач, предложенных в приложении В. Варианты заданий указываются преподавателем.

2.3 Выполнить задания для самостоятельной работы из приложения Г по указанию преподавателя.

2.4 Составить и отладить программу для решения следующей задачи: выбрать наименьшее из трех различных чисел.

Использовать фрагмент программы вида:

```
readln (a, b, c);
min:= a;
if b < min then min:=b;
if c < min then min:=c;
write ('Наименьшее число = ', min);
```

2.5 Отладить программу для решения задачи: дано вещественное число x . Вычислить $F(x)$, где

$$F = \begin{cases} 0, & \text{при } x \leq 0 \\ x^2 - x, & \text{при } 0 < x \leq 1 \\ x^2 - \sin \pi x^2, & \text{при } x > 1 \end{cases}$$

Фрагмент программы будет выглядеть следующим образом:

```
if x <=0 then f:=0 else
    if x <=1 then f:=sqr(x)-x else f:=sqr(x)-sin(pi*x*x);
```

2.6 Составить и отладить на компьютере задачу определения значения Y в зависимости от значения текущей переменной N для некоторого заданного действительного числа x :

$$Y = \begin{cases} \sin(x), & N = 1 \\ \cos(x), & N = 2 \\ \ln(x), & N = 3 \\ |x|, & N = 4 \end{cases}$$

Для других значений N выдать сообщение «Функция не определена».

Фрагмент программы с использованием оператора case:

```
...
Case N of 1 : writeln ('y=' , sin(x));
          2 : writeln ('y=' , cos(x));
          3 : writeln ('y=' , ln(x));
          4 : writeln ('y=' , abs (x))
else writeln ('Функция не определена');
end;
```

2.7 Решить задачи, указанные преподавателем из Приложения Д.

3 Требование к отчету

Отчет о проделанной работе должен содержать:

- Название работы, ее цель;
- Номер и условие своего варианта для выполнения задания;
- Блок-схемы и тексты программ;
- Полученные результаты выполнения программы;
- Ответы на контрольные вопросы из пункта 6 по указанию преподавателя.

4 Контрольные вопросы и задания

- 1) Что определяет тип данных языка Pascal?
- 2) Какие бывают стандартные типы данных?
- 3) Как описываются переменные на языке Pascal?
- 4) Какие требования предъявляются к имени программы языка Pascal?
- 5) Какой оператор служит для вывода информации на печать?
- 6) Какой оператор служит для ввода значений с клавиатуры?
- 7) Для чего необходим формат вывода данных и что он предполагает?
- 8) Из каких условных основных частей состоит программа (ее структура)?
- 9) Какие основные разделы могут быть включены в раздел описания в структуре программы?
- 10) Что такое идентификатор языка Pascal?
- 11) Что такое константа? Каких она бывает видов?
- 12) Что такое переменная?
- 13) Какие стандартные функции существуют в языке Pascal?
- 14) Что понимают под алгоритмом ветвления?
- 15) Какие бывают формы условного оператора? Приведите их вид.
- 16) Что такое арифметическое выражение и логическое выражение?
- 17) Назовите основные логические операции.
- 18) Назовите оператор выбора данных. Приведите его общий вид.

Лабораторная работа №2 Операторы циклов языка Pascal

Цель работы

Изучить операторы циклов языка Pascal.

Задачи работы

Научиться работать с алгоритмами циклической структуры на языке Pascal.

1 Общие положения

1.1 Алгоритмы циклической структуры

Цикл - это любая многократно выполняемая часть программы, состоящая из группы операторов. При этом повторяющиеся действия составляют тело цикла.

Различают циклы с известным числом повторений (цикл с параметром) и итерационные (с пред- и постусловием). Операторы циклов с постусловием и предусловием используются, как правило, когда заранее неизвестно число повторений цикла.

1.2 Оператор цикла с параметром

Для организации цикла с известным числом повторений в языке Pascal используется оператор For. В этом операторе обязательно указываются следующие параметры:

- имя переменной, в которой хранится число повторений цикла. Ее называют параметром (или счетчиком) цикла;
- начальное значение для переменной цикла, которое она получает при первом выполнении цикла;
- конечное значение для переменной цикла, достигнув которого повторение цикла прекращается (условие завершения цикла).

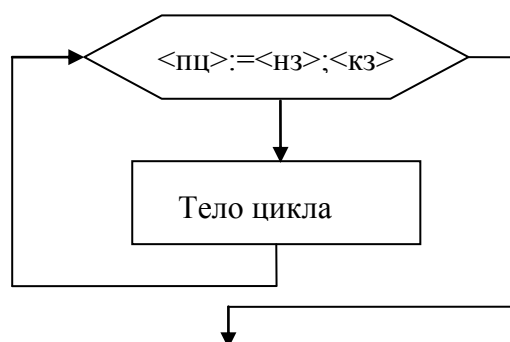
Блок-схема цикла с параметром представлена на рисунке 3.

Рисунок 3 Блок-схема цикла с параметром

Оператор цикла с параметром имеет следующий вид:

FOR<пц>:=<нз> TO <кз> DO <оператор>. Здесь:

- FOR, TO, DO – зарезервированные слова (англ. для, до, выполнить);
- <пц> - параметр_цикла – переменная типа integer (точнее любого



порядкового типа);

- <нз> - начальное_значение - константа или выражение того же типа;
- <кз> - конечное_значение - константа или выражение того же типа;
- <оператор> - тело цикла (произвольный оператор языка Pascal).

В вышеприведенной записи цикла переменная цикла изменяется, увеличиваясь на 1 до конечного значения.

Порядок выполнения цикла следующий: вначале определяются начальное и конечное значения параметра цикла. Параметр цикла принимает начальное значение. После этого циклически повторяется:

- проверка условия; если условие не выполнено, оператор For завершает свою работу;
- выполнение тела цикла;
- наращивание переменной цикла на 1¹.

Выход из цикла осуществляется, когда оператор цикла становится больше конечного значения.

Если условие не выполняется в самом начале работы оператора For, то цикл не исполняется ни разу.

Существует другая форма оператора For:

FOR <пц>:=<нз> DOWNT0 <кз> DO <оператор>.

Замена зарезервированного слова to на downto означает, что шаг приращения переменной цикла равен -1.

Если в операторе цикла с параметром начальное или конечное значение параметра заданы переменными или выражениями, то значения этих переменных должны быть определены в программе до оператора цикла. Не следует внутри цикла изменять параметр цикла, его начальное и конечное значения с помощью операторов присваивания или ввода.

<оператор> может быть любым оператором Pascal, в том числе оператором цикла.

Если тело цикла содержит больше одного оператора, то он оформляется как составной оператор с помощью операторных скобок:

FOR <пц>:=<нз> TO <кз> DO

BEGIN

<оператор_1>;

<оператор_2>;

.

.

<оператор_N>

END;

Пример 1. Дано действительное число x . Вычислить сумму ряда $\sin 2x + \sin 3x + \dots + \sin 10x$.

```
program summa;
```

```
var s, x : real;
```

```
    i : integer;
```

```
begin
```

```
    write('Введите x');
```

¹ После выполнения тела цикла параметр цикла автоматически получает значение, следующее за текущим значением параметра в соответствии с его типом. Если параметр целый, то значение параметра увеличивается на единицу. Если параметр символьный, то выбирается следующий символ кодовой таблицы. Если логический, то выбирается следующее логическое значение. В любом случае параметр должен быть порядкового типа. Отсюда следует, что данные вещественного типа не могут быть параметрами цикла.

```

readln(x);
s:=0; { начальное значение суммы }
for i:=2 to 10 do s:=s+sin(i*x); { накопление суммы }
writeln('Сумма равна ', s:5:2)
end.

```

Пример 2. Написать программу для нахождения среднего арифметического нечетных натуральных чисел на отрезке от а до b. Блок-схема решения задачи представлена на рисунке 4.

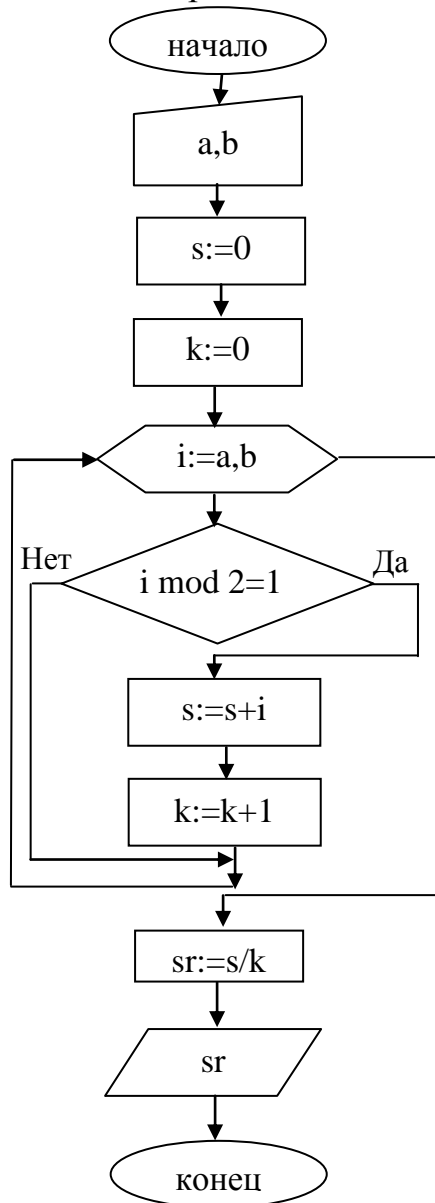


Рисунок 4

```

program summa;
var a, b, s, i, k : integer;
    sr : real;
begin
    write('Введите границы отрезка [a,b] ');
    readln(a,b);
    s:=0; {начальное значение суммы}
    k:=0; {начальное значение количества}
    for i:=a to b do
        if i mod 2 =1 then begin s:=s+i; k:=k+1 end; {
            если число нечетное, то добавить его к сумме и
            считать количество таких чисел }
    sr:=s/k;
    writeln('Среднее арифметическое =',sr:5:2)
end.

```

Пример 3. Вычислить $S = \underbrace{\sqrt{2 + \sqrt{2 + \sqrt{2 + \dots + \sqrt{2}}}}}_{n \text{ раз}}$.

```

program summa;
var n, i : integer;
    s : real;
begin
    write('Введите количество слагаемых');

```

```

readln(n);
s:=0;
for i:=n downto 1 do s:=sqrt(2+s); { накопление суммы }
writeln('Сумма =',s:7:3)
end.

```

1.3 Оператор цикла с предусловием

Общий вид оператора:

WHILE <условие> DO <оператор>. Здесь:

- WHILE, DO – зарезервированные слова (англ. пока, делать);
- <оператор> - тело цикла (произвольный оператор языка Pascal).

Оператор цикла с предусловием действует таким образом: проверяется условие и если оно истинно, то выполняется оператор, указанный после DO. Если ложно, то оператор WHILE прекращает свою работу.

Блок-схема цикла с предусловием представлена на рисунке 5.

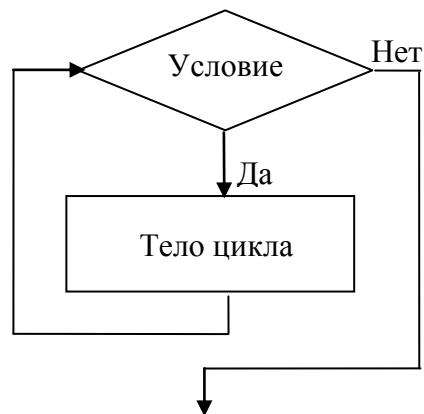


Рисунок 5- Блок-схема цикла с предусловием

Если тело цикла содержит больше одного оператора, то он оформляется как составной оператор:

```
WHILE <условие> DO
```

```
BEGIN
```

```
  Оператор_1;
```

```
  Оператор_2;
```

```
  .
```

```
  .
```

```
  Оператор_N
```

```
END;
```

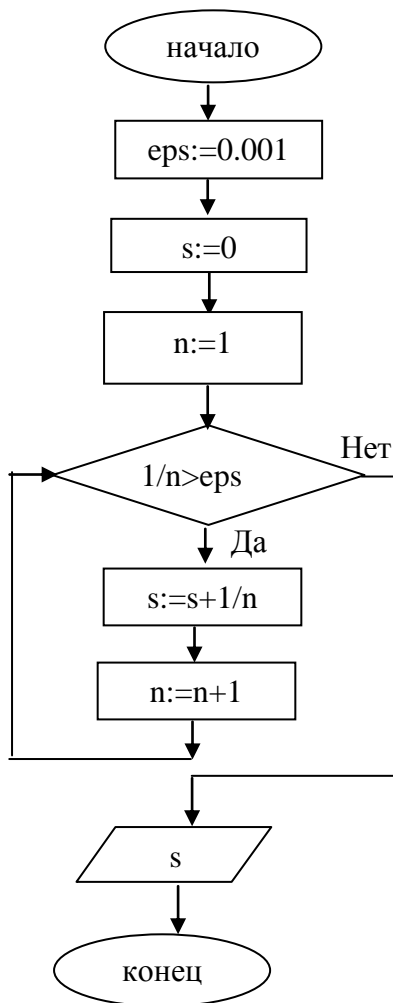
Пример 4. Вычислить сумму обратных величин натуральных чисел. Суммирование вести до тех пор, пока очередное обратное число $1/N$ станет меньше заданной положительной константы ε .

Блок-схема решения задачи представлена на рисунке 6.

```

program sum;
const eps=0.001;

```



```

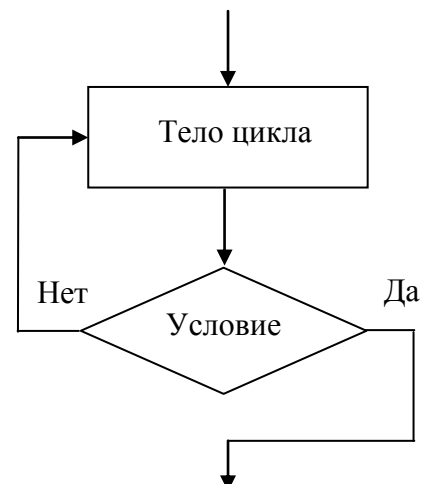
var n : integer;
    s : real;
begin
    s:=0; {начальное значение суммы}
    n:=1; {номер шага цикла}
    while 1/n>=eps do
        begin
            s:=s+1/n; {вычисление суммы на
                        следующем шаге}
            n:=n+1 {переход к следующему шагу
                  цикла}
        end;
    writeln('Сумма ряда =',s:6:2)
end.

```

Рисунок 6

1.4 Оператор цикла с постусловием

Блок-схема цикла с постусловием приведена



на рисунке 7.

Рисунок 7 Блок-схема цикла с постусловием

Общий вид оператора:

```

REPEAT
    <тело цикла>
UNTIL <условие>.

```


• REPEAT, UNTIL – зарезервированные слова (англ. повторять, до тех пор [пока не будет выполнено]);

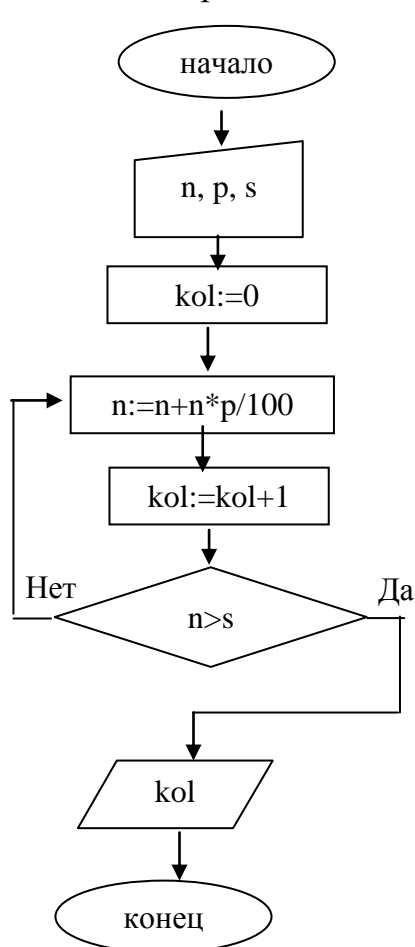
• тело цикла - произвольный оператор (или операторы) языка Pascal.

Зарезервированные слова REPEAT и UNTIL ограничивают тело цикла, поэтому операторные скобки BEGIN ... END не нужны, хотя их использование не запрещено.

Операторы тела цикла всегда выполняются хотя бы один раз, после чего проверяется <условие>. Если его значение ложно, операторы тела цикла повторяются, в противном случае цикл завершается.

Пример 5. Сколько лет нужно ожидать вкладчику суммы S , если начальный вклад равен N при процентах прироста вклада за год P ?

Блок-схема решения задачи представлена на рисунке 8.



```

program vklad;
var kol : integer;
    n,p,s : real;
begin
  writeln('Введите начальный вклад');
  readln(n);
  writeln('Введите процент за год');
  readln(p);
  writeln('Введите требуемую сумму');
  readln(s);
  kol:=0;
  repeat
    n:= n +n*p/100; { накопление суммы с
                    процентами }
    kol:= kol+1     { подсчет количества лет }
  until n > s;
  writeln('Через ', kol, ' лет будет сумма', s:8:1)
end.
  
```

Рисунок 8

2 Содержание работы

2.1 Составить программу, определяющую сумму $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{9}$ с точностью 3 знаков после десятичной точки.

Для решения поставленной задачи сначала необходимо выявить закономерность построения ряда, т.е. зависимость вида слагаемого от его порядкового номера. В данной задаче общий вид слагаемого с номером i будет иметь вид: $1/i$, где i изменяется от 1 до 9.

Накопление суммы производится по шагам: на каждом шаге цикла к имеющемуся уже значению суммы должно прибавляться очередное слагаемое. Фрагмент программы с использованием оператора цикла с параметром:

```
...
s:=0;
for i:=1 to 9 do s:=s+1/i;
writeln('Сумма=', s:7:3)
...
```

2.2 Составить и отладить программу с использованием цикла с параметром с заданной точностью в соответствии с указанным преподавателем вариантом из приложения Е.

2.3 Вычислить функцию $y=x^2+3$, если x изменяется от 0 до 1 с шагом 0.1. Для разных значений аргумента предстоит вычислить значение функции по одной и той же формуле, поэтому для решения задачи необходимо использовать цикл. Так как x изменяется с шагом 0.1, то для решения задачи следует воспользоваться циклом с предусловием или циклом с постусловием. Фрагмент программы с использованием цикла с предусловием:

```
...
x:=0;
while x<=1 do
begin
  y:=sqr(x)+3;
  writeln('x=',x:3:1,' y=',y:4:2);
  x:=x+0.1;
end;
...
```

2.4 Составить и отладить программу с использованием цикла с предусловием.

Вычислить значение функции $y=f(x)$, начиная от значения $x=x_{\text{нач}}$ до значения $x=x_{\text{кон}}$ с шагом s в соответствии с указанным преподавателем вариантом из приложения Ж. Результаты вычислений вывести в виде таблицы.

2.5 Решить поставленную в пункте 2.4 задачу с использованием цикла с постусловием.

2.6 Выполнить задания для самостоятельной работы из приложения 3 по указанию преподавателя.

3 Требования к отчету

Отчет о проделанной работе должен содержать:

- название работы, ее цель;
- номер и условие своего варианта для выполнения задания;
- блок-схемы и тексты программ;

- полученные результаты выполнения программы;
- ответы на контрольные вопросы из пункта 4 по указанию преподавателя.

4 Контрольные вопросы и задания

- 1) Что называется циклом, телом цикла?
- 2) Назовите отличия итерационных циклов и циклов с параметром.
- 3) Какова структура оператора цикла с параметром? Как выполняется цикл с параметром?
- 4) Какого типа должны быть параметр цикла, его начальное и конечное значения в цикле с параметром в языке Pascal?
- 5) Могут ли параметр цикла, его начальное и конечное значения в цикле с параметром в языке Pascal быть разных типов? Обоснуйте ответ.
- 6) Какова структура цикла с предусловием? Как выполняется этот цикл?
- 7) Какова структура цикла с постусловием? Как выполняется этот цикл?
- 8) Каково минимальное количество исполнений циклов с предусловием? С чем это связано?
- 9) Каково минимальное количество исполнений циклов с постусловием? С чем это связано?
- 10) Можно ли в теле цикла изменять значение параметра цикла?
- 11) Можно ли в теле цикла использовать значение параметра цикла?
- 12) Выполнится ли цикл следующего вида: `for i := 1 to -1 do k:=k*i` ?
- 13) Найти ошибки в следующих фрагментах программ:
`var k,i : integer; x,y : real;`
 а) `y:=0;`
 `for x:=1 to 0.9 do y:=y+sin(x);`
 б) `k:=64; y:=3;`
 `for i:=1 to sqrt(k) do y:=2*y;`
 в) `k:=1;`
 `for i:=1 to 64 do`
 `begin`
 `i:=2*i; k:=k+i`
 `end.`
- 14) Сколько раз выполнится цикл: `m :=123; while m<>0 do m := m mod 10` ?
- 15) Сколько раз выполнится цикл: `For i :=1 downto -1 do k:=k*i` ? Запишите данный цикл, используя оператор While.
- 16) Сколько раз выполнится цикл: `for i := -1 to 1 do k:=k*i` ? Запишите данный цикл, используя оператор Repeat.

Лабораторная работа №3 Вложенные циклы языка Pascal

Цель работы

Изучить вложенные циклы языка Pascal.

Задачи работы

Научиться работать с алгоритмами вложенных циклических структур.

1 Общие положения

В теле любого оператора цикла могут находиться другие операторы цикла. При этом цикл, содержащий в себе другой, называется **внешним**, а цикл, находящийся в теле первого – **вложенным**. Внешний и внутренний циклы могут быть трех видов: циклами с параметром, с предусловием и с постусловием. Правила организации внешнего и вложенного циклов такие же, как и для простого цикла. Но при программировании вложенных циклов необходимо соблюдать следующее дополнительное условие: все операторы внутреннего цикла должны полностью располагаться в теле внешнего цикла.

Пример 1. Найти цифровой корень целого числа.

Цифровой корень ищется суммированием цифр числа, а затем суммированием цифр получающихся чисел до тех пор, пока эта сумма сама не станет цифрой. Например, цифровой корень числа $5412=3$, то есть $5+4+1+2=12$, $1+2=3$.

```
program zifr_kor;
var n : longint;
    k,s : integer;
begin
  writeln('Введите целое число');
  readln(n);
  while n>9 do
  begin
    s:=0;
    repeat
      s:=s+n mod 10;
      n:=n div 10
    until n=0;
    n:=s
  end;
  writeln('Цифровой корень =',n)
end.
```

Пример 2. Вычислить $\sum_{i=1}^n \frac{1}{(2i)!}$.

Блок-схема решения задачи приведена на рисунке 9.

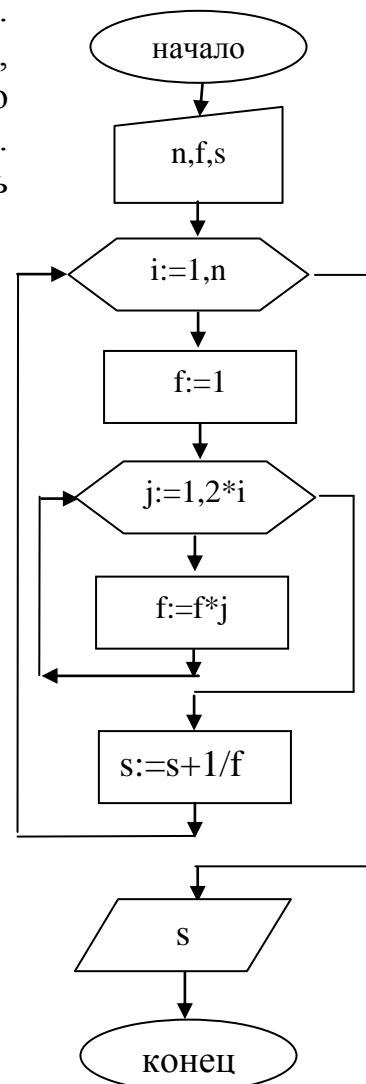


Рисунок 9

```

program sum;
var i, j, n, f : integer; s : real;
begin
  writeln('Введите n');
  readln(n);
  for i:=1 to n do
    begin
      f:=1;
      for j:=1 to 2*i do f:=f*j; {Подсчет факториала}
      s:=s+1/f
    end;
  writeln('Сумма = ', s:8:4)
end.

```

Пример 3. Для натурального n вычислить сумму степеней:

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

```

program summa;
var n, i, j : integer;
    a, s, p : real;
begin
  writeln('Введите показатель степени');
  readln(n);
  s:=0; { Обнуление начального значения суммы }
  for i:=1 to n do { Начало внешнего цикла }
    begin
      a:=1/i; p:=a;
      for j:=2 to n do p:=p*a; { Вычисление слагаемого }
      s:=s+p { Накопление суммы степеней }
    end;
  writeln('Сумма степеней ', s:8:2)
end.

```

2 Содержание работы

2.1 Составить и отладить программу с использованием вложенного цикла. В хозяйстве выработка продукции за одну неделю месяца составляла n единиц. Еженедельно она увеличивалась на 10%. Каждый месяц цена продукции увеличивалась на 2%. Рассчитать стоимость продукции, выработанной хозяйством, за каждую неделю трех месяцев. Считать, что в месяце 4 недели.

```

program stoim;
var i, j : integer;
    n, c, s : real;
begin
  writeln('Введите выработку продукции за одну неделю');
  readln(n);

```

```
writeln('Введите первоначальную цену продукции');
readln(c);
for i:=1 to 3 do {Цикл по месяцам}
begin
  for j:=1 to 4 do {Цикл по неделям}
  begin
    n:=n+n*0.1; {Увеличение выработки продукции еженедельно}
    s:=c*n; {Расчет стоимости еженедельно}
    writeln('Стоимость продукции за ',i,' месяц ',j,' неделю ',s:6:2)
  end;
  c:=c+c*0.02 { Увеличение цены продукции ежемесячно}
end
end.
```

2.2 Вычислить значение функции в соответствии с указанным преподавателем вариантом из приложения И.

2.3 Выполнить задания по вариантам из приложения К по указанию преподавателя.

2.4 Выполнить задания для самостоятельной работы из приложения Л по указанию преподавателя.

3 Требования к отчету

Отчет о проделанной работе должен содержать:

- название работы, ее цель;
- номер и условие своего варианта для выполнения задания;
- блок-схемы и тексты программ;
- полученные результаты выполнения программы;
- ответы на контрольные вопросы из пункта 4 по указанию преподавателя.

4 Контрольные вопросы и задания

- 1) Какой цикл называется вложенным?
- 2) Какой цикл называется внешним?
- 3) Могут ли операторы внутреннего цикла располагаться вне тела внешнего цикла?
- 4) Какими видами циклов могут быть внешний и внутренний циклы?
- 5) Какие дополнительные условия необходимо соблюдать при организации вложенных циклов?
- 6) Задан фрагмент программы вида:

```
a:=2; b:=2; s:=0;
while a+b<8 do
begin
  for i:=1 to 2 do s:=s+a+b; a:=a+2; b := b+2
end;
```

Сколько раз выполнится внешний цикл? Чему будут равны значения a, b, s?

Лабораторная работа №4

Использование одномерных массивов

в среде программирования Pascal

Цель работы

Ознакомиться с понятием массивов данных; научиться описывать одномерные массивы на языке Pascal; ознакомиться с некоторыми алгоритмами обработки массивов данных.

Задачи работы

Научиться обрабатывать одномерные массивы данных на языке программирования Pascal.

1 Общие положения

1.1 Основные сведения о массивах

Массивом называется совокупность элементов одинакового типа, объединенных общим именем. Число элементов в массиве называется **размерностью (размером)** массива; каждый элемент массива задается своим порядковым номером в массиве - **индексом**.

Массивы, элементы которых однозначно определяются одним индексом, называются **одномерными**. В виде одномерного массива можно представить, например, список фамилий студентов одной группы, где каждый студент однозначно определяется своим порядковым номером в списке.

В языке Pascal нет ограничения на число элементов в массиве (т.е. на его размерность), однако, суммарная длина любого из массивов во внутреннем представлении не должна превышать 65520 байт.

1.2 Описание одномерных массивов на языке Pascal

Язык Pascal допускает описание в программах одномерных массивов в следующем виде:

TYPE <Имя_типа_массива>=**ARRAY** [<Диапазон_индексов>] **OF** <Тип>;

где **TYPE** - служебное слово, используемое для создания пользовательского типа данных;

ARRAY ... OF - служебные слова для описания массивов («Массив ... из»);

<Имя_типа_массива> - задаваемое пользователем имя типа массива (требования к именам типов совпадают с требованиями к именам переменных);

<Диапазон_индексов> — диапазон изменения индексов массива. В качестве данного диапазона может выступить любой порядковый тип, кроме LONGINT. Чаще всего используют тип-диапазон, в котором задают границы изменения индексов;

<Тип> - тип элементов массива, например, REAL, INTEGER, CHAR и др.

Примеры описания типа массивов:

TYPE M=ARRAY [1..10] OF INTEGER;

TYPE AR=ARRAY [0..15] OF REAL;

Переменные, относящиеся к тому или иному типу массивов, должны быть соответственно объявлены в разделе описания переменных, например:

VAR REZ:M;

VAR P:AR;

Можно также описать переменную как непосредственно имеющую тип массива, например:

VAR REZ: ARRAY [1..10] OF INTEGER;

При обращении к элементам массива в тексте программы обязательно требуется указывать их порядковый номер в массиве, например, REZ[1], P[5] и т.д.

Одномерные массивы также иногда называются **векторами**. Как правило, предполагается, что диапазон индексов вектора начинается с 1. Например, если задан вектор V из 5 чисел (4, 2, 1, -7, 8), то первый элемент вектора равен 4 (т.е. V[1]=4), второй элемент вектора равен 2, третий элемент вектора равен 1 и т.д.

Пример 1. Векторные операции.

Найти скалярное произведение двух заданных векторов из 5 элементов.

Возможный вид решения задачи:

```
PROGRAM scal_pr;
  TYPE Vec=array[1..5] of real;
  var a,b:Vec; i:integer; s:real;
begin
  writeln('Введите элементы вектора a');
  for i:=1 to 5 do readln(a[i]);
  writeln('Введите элементы вектора b');
  for i:=1 to 5 do readln(b[i]);
  s:=0;
  for i:=1 to 5 do s:=s+a[i]*b[i];
  writeln('s=',s:6:2)
end.
```

Для работы с массивом как с единым целым используется идентификатор массива без указания индексов. В данном случае допустимыми являются только операции отношения =, <> и оператор присваивания :=, например A=B; A<>B; A:=B; где A и B – массивы идентичной структуры.

1.3 Типовые задачи обработки массивов данных

К стандартным типам задач на обработку массивов данных можно отнести следующие задачи:

- нахождение суммы элементов массива;
- нахождение количества элементов массива, удовлетворяющих определенным условиям;
- задачи на поиск элементов массива, удовлетворяющих некоторым заданным условиям;
- преобразование массивов путем замены и перестановки их элементов;
- упорядочение элементов массива и др.

Рассмотрим некоторые типовые алгоритмы обработки массивов данных.

1.3.1 Нахождение суммы элементов массива

К данному классу относятся разнообразные задачи, связанные с вычислением итоговых сумм для заданного множества значений. Например, нахождение суммарного объема перевозок автохозяйства при известных показателях грузоперевозок отдельных машин, суммарного объема сбора урожая по району при известном урожае отдельных хозяйств и т.д. Суммирование при этом может проводиться как по всем элементам массива, так и только по тем элементам, которые удовлетворяют заданным условиям.

Для нахождения суммы элементов массива используется следующий алгоритм. Сначала предполагается, что значение искомой суммы равно нулю. Далее на каждом шаге цикла проводится накопление данной суммы путем добавления к уже имеющемуся значению суммы значения очередного элемента массива. Если суммирование проводится не по всем элементам массива, то в программу добавляется условный оператор проверки выполнения наложенного на элементы массива условия.

Пример 2. Нахождение суммы элементов массива, удовлетворяющих некоторому заданному условию.

Найти сумму четных элементов заданного одномерного массива из 10 целых чисел.

```
PROGRAM Sum_mas;
Type Mas=array[1..10] of integer;
Var a: Mas;
    i,s: integer;
begin
s:=0;
for i:=1 to 10 do
begin
writeln('Введите элемент массива');
readln(a[i]);
if a[i] mod 2=0 then s:=s+a[i]  {Проверка четности элемента и накопление
суммы}
end;
writeln('Сумма s=', s)
end.
```

1.3.2 Поиск данных в массивах

Поиск информации, удовлетворяющей определенным условиям, является одной из наиболее распространенных задач обработки массивов. В частности, к данному классу задач можно отнести поиск наибольших и наименьших элементов массива, поиск элементов, удовлетворяющих разнообразным операциям отношений (больших заданного числа, равных ему и т.д.) и другие.

Пример 3. Поиск наибольшего элемента в массиве.

Задан одномерный массив из 7 действительных элементов. Найти наибольший элемент данного массива.

```
PROGRAM Max_mas;
Type Mas=array[1..7] of real;
```

```

Var a: Mas;
    max : real;
    i: integer;
begin
    for i:=1 to 7 do
begin
    writeln('Введите элемент a[' ,i,']');
    readln(a[i])
end;
    max:=a[1];
    for i:=2 to 7 do
        if max<a[i] then max:=a[i];
        writeln('max=', max)
    end.

```

В данной программе реализован следующий алгоритм. В начальный момент предполагается, что наибольшим является первый из элементов массива. Далее этот элемент сравнивается с остальными элементами массива, и, если найдется элемент, больший, чем предполагаемый максимум, то переменной max присваивается это значение.

1.3.3 Сортировка массивов данных

Сортировкой массива называют упорядочение его элементов в соответствии с определенными правилами (как правило, по возрастанию или убыванию элементов).

Рассмотрим следующие алгоритмы сортировки массивов данных: линейная сортировка (сортировка отбором) и сортировка методом «пузырька». Алгоритм **линейной** сортировки массивов по возрастанию заключается в следующем. Сначала первый элемент массива сравнивается по очереди со всеми оставшимися элементами. Если очередной элемент массива меньше по величине, чем первый, то эти элементы переставляются местами. Сравнение продолжается далее уже для обновленного первого элемента, в результате чего будет найден и установлен на первое место самый наименьший элемент массива. Далее продолжается аналогичный процесс уже для оставшихся элементов массива, т.е. второй элемент сравнивается со всеми остальными и, при необходимости, переставляется с ними местами. Алгоритм завершается, когда сравниваются и упорядочиваются предпоследний и последний из оставшихся элементов массива.

Пример 4. Линейная сортировка массива по возрастанию.

Известны результаты тестирования 9 учащихся. Расположить данные результаты в порядке возрастания баллов.

Программа реализации изложенного выше алгоритма линейной сортировки отбором может иметь следующий вид:

```

PROGRAM Sort;
TYPE M=array[1..9]of integer;

```

```

Var a: M;
    i,j,m, k, x: integer;
Begin
    For i:=1 to 9 do
        begin
            writeln('Введите результаты ',i,'-го учащегося');
            readln(a[i])
        end;
        {Для всех элементов массива, кроме последнего, сделать}
        for j:=1 to 8 do
            begin
                m := j;                {Ищем минимальный элемент (в начале j-й) }
                for k:=j+1 to 9 do
                    if a[k]<a[m] then m:=k; {Текущий минимальный элемент - k-й}
                x:=a[j];                {Обменять минимальный элемент с j-ым}
                a[j]:=a[m];
                a[m]:=x
            end;
        writeln('Отсортированные по возрастанию результаты:');
        for i:=1 to 9 do write (a[i], ' ')
    end.

```

Особенностью сортировки **методом «пузырька»** является не сравнение каждого элемента со всеми, а сравнение в парах соседних элементов. Данный алгоритм состоит в последовательных просмотрах от начала к концу элементов массива. Если для пары соседних элементов выполняется условие, что элемент справа больше элемента слева, то производится обмен значениями этих элементов, т.е. в процессе выполнения алгоритма постепенно «всплывают» более «легкие» элементы массива.

Фрагмент программы на Pascal выглядит так:

```

...
for j:=n downto 2 do
    {Сравнить все пары соседних элементов от первого до (j-1)-го}
    for k:=1 to j-1 do
        if a[k]>a[k+1] then
            begin
                x:=a[k];                {Обменять k-й элемент с (k+1)-ым}
                a[k]:=a[k+1];
                a[k+1]:=x
            end;....

```

Дополнительный ряд алгоритмов решения типовых задач обработки массивов приводится также в [4-7].

2 Содержание работы

2.1 Составьте и отладьте программу решения следующей задачи.

Задан массив из 6 целых чисел. Найдите сумму его элементов.

Указание: Соответствующий фрагмент программы имеет вид:

```
...
s:=0;
for i:=1 to 6 do
begin
  writeln('Введите элемент массива');
  readln(a[i]);
  s:=s+a[i]
end;
writeln('сумма s=', s);
...
```

2.2 Дополните созданную программу. Добавьте фрагмент, позволяющий найти количество неотрицательных элементов в описанном выше массиве.

Возможный вид решения:

```
k:=0;
for i:=1 to 6 do
if a[i]>=0 then k:=k+1;
  writeln('количество k=', k);
...
```

2.3 Дополните программу. Найдите наименьший элемент в заданном массиве.

Пример возможного решения:

```
...
min:=a[1];
for i:=2 to 6 do if a[i]<min then min:=a[i];
writeln('min=', min);
...
```

2.4 Дополните созданную программу. Поменяйте местами первый и последний элемент исходного массива.

Указание: Используйте алгоритм следующего вида:

```
...
c:=a[1];
a[1]:=a[6];
a[6]:=c;
for i:= 1 to 6 do write (a[i]:5);
...
```

2.5 Выполните задание в соответствии с указанным преподавателем вариантом из приложения М.

2.6 Составьте и отладьте программу решения следующей задачи:

Упорядочите по возрастанию произвольный целочисленный массив из 8 элементов:

а) методом линейной сортировки;

б) методом «пузырька».

Указание: Воспользуйтесь алгоритмами, приведенными в соответствующем разделе теоретических положений.

2.7 Решите указанные преподавателем задачи из приложения Н.

3 Требования к отчету

Отчет о проделанной работе должен содержать:

- название и цель работы;
- номер и условие своего варианта;
- тексты разработанных программ;
- полученные при расчетах результаты;
- анализ полученных результатов;
- ответы на контрольные вопросы в соответствии с указаниями преподавателя.

4 Контрольные вопросы

- 1) Что понимают под массивом данных?
- 2) Поясните понятия размерности массива, индекса элемента массива.
- 3) Какой массив называют одномерным? Приведите примеры одномерных массивов.
- 4) Как создать тип «массив» на языке Pascal?
- 5) Как описать переменную типа «массив» на языке Pascal?
- 6) Сколько элементов содержится в массиве `a: array[-5..5] of integer;`?
- 7) Какие ограничения на размер массива накладывает язык Pascal?
- 8) Как осуществляется обращение к отдельному элементу одномерного массива?
- 9) Каким образом выводятся элементы массива на экран? Приведите пример вывода на экран одномерного массива из 8 действительных чисел.
- 10) Что понимают под сортировкой массива?
- 11) Какие методы сортировки массивов Вы знаете?
- 12) Какие задачи можно отнести к задачам на поиск данных в массивах? Приведите примеры.
- 13) Охарактеризуйте алгоритм нахождения наибольшего (наименьшего) элемента массива.
- 14) Приведите алгоритм нахождения суммы положительных элементов массива.
- 15) Приведите алгоритм нахождения количества нулевых элементов массива.

Лабораторная работа №5

Использование двумерных массивов в среде Pascal

Цель работы

Ознакомиться с понятием двумерных массивов данных; научиться описывать и обрабатывать двумерные массивы на языке Pascal.

Задачи работы

Научиться решать задачи по обработке двумерных массивов данных на языке программирования Pascal.

1 Общие положения

1.1 Двумерные массивы

Массив, для задания элементов которого требуется использовать два индекса, называется **двумерным**. Примерами двумерных массивов являются разнообразные таблицы, в которых каждый элемент характеризуется своим номером строки и номером столбца.

Например, в качестве двумерного массива можно представить оценки из аттестационной ведомости студентов, где каждая оценка показывает знания некоторого студента по определенному предмету:

Ф.И.О.	Математика	Информатика	Физика	Химия
Андреев А.Л.	5	5	4	5
Биктимиров С.Р.	3	4	3	2
Симонова В.П.	4	3	4	3

Двумерные массивы, в которых диапазоны индексов начинаются с 1, также называются иногда **матрицами**. Размерность каждой матрицы определяется как $M \times N$, где M – число строк в матрице, N – число столбцов.

Например, в предложенной выше матрице оценок $A = \begin{pmatrix} 5 & 5 & 4 & 5 \\ 3 & 4 & 3 & 2 \\ 4 & 3 & 4 & 3 \end{pmatrix}$ из

трех строк и четырех столбцов элемент $A[3, 1]=4$, $A[1, 2]=5$, $A[3, 2]=3$, $A[2, 4]=2$. Первый индекс элемента матрицы характеризует номер строки матрицы, в которой расположен элемент, второй индекс – номер столбца.

Если число строк матрицы равняется числу столбцов, то матрицы данного типа называются **квадратными**. Элементы квадратной матрицы вида $B[1,1]$, $B[2,2]$, $B[3,3]$... составляют главную **диагональ** матрицы.

1.2 Описание двумерных массивов на языке Pascal

Описание типов двумерных массивов в языке Pascal осуществляется аналогично описанию типов одномерных массивов с добавлением диапазона изменения второго индекса, например:

```
TYPE MATR=ARRAY[1..4, 1..5] OF INTEGER;
```

```
TYPE T=ARRAY[0..10, 2000..2004] OF REAL;
```

Также допускается указание имени другого типа массива в качестве типа элементов массива, например:

```
TYPE VEC=ARRAY[1..4] OF REAL;
```

```
MAS=ARRAY[1..5] OF VEC;
```

Переменные, относящиеся к типу массива, должны быть соответствующим образом объявлены в разделе описания переменных, например:

```
VAR A: MATR;
```

```
VAR B, C, D: T;
```

```
VAR R: ARRAY [1..5, 1..10] OF INTEGER;
```

1.3 Обработка двумерных массивов данных

При решении задач на использование двумерных массивов применяются те же алгоритмы, что и при работе с одномерными массивами. При вводе и выводе элементов двумерных массивов используются вложенные циклы, в которых внешний оператор цикла, как правило, задает изменение номеров строк массива, внутренний оператор цикла – изменение номеров столбцов.

Пример 1. Обработка элементов матрицы.

Найти количество положительных элементов в заданной целочисленной матрице размерности 5x6.

Возможное решение задачи:

```
PROGRAM Mas_count;
```

```
Type Matr=array[1..5,1..6] of integer;
```

```
Var A: Matr;
```

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

```
begin
```

```
  k:=0;
```

```
  for i:=1 to 5 do
```

```
    for j:=1 to 6 do
```

```
      begin
```

```
        writeln('Введите элемент массива');
```

```
        readln(a[i,j]);
```

```
        if a[i,j]>0 then k:=k+1 {Подсчет количества положительных  
элементов}
```

```
      end;
```

```
      writeln('k=', k)
```

```
end.
```

Пример 2. Обработка заданного столбца матрицы.

Найти наибольший элемент в первом столбце заданной действительной матрицы размерности 4x5.

Пример возможного решения задачи:

```
PROGRAM Mas_max;
```

```
Type Matr=array[1..4,1..5] of real;
```

```
Var A: Matr;
```

```
  i, j: integer; max: real;
```

```
begin
```

```
  for i:=1 to 4 do
```

```
    for j:=1 to 5 do
```

```
      begin
```

```
writeln('Введите элемент массива');
readln(a[i,j])
end;
max:=a[1,1];
for i:=2 to 4 do
if a[i,1]>max then max:=a[i,1];    {Проверка элементов первого столбца}
writeln('max=', max)
end.
```

Пример 3. Обработка строк матрицы.

Известны экзаменационные оценки 10 студентов по 5 предметам. Вычислить средний балл каждого студента по итогам экзаменационной сессии.

При решении задачи используем матрицу размерности 10x5. Поставим в соответствие первому индексу матрицы порядковый номер студента в списке, а второму индексу - номер предмета, по которому получена оценка.

Матрицу оценок зададим с использованием функции генератора случайных чисел Random.

```
PROGRAM Obr_mas;
Type Mas=array[1..10,1..5] of byte;
Var A: Mas;
    i,j: integer; s, sr:real;
begin
  for i:= 1 to 10 do
    for j:= 1 to 5 do A[i, j]:=2+random(4); {генерирование матрицы оценок}
    writeln('Матрица оценок ');
  for i:= 1 to 10 do                      {начало вывода матрицы оценок на экран}
    begin
      for j:= 1 to 5 do write(A[i,j]:3); {вывод в строку оценок i-го студента}
      writeln                          {переход к следующей строке}
    end;
    writeln('Средний балл ');
  for i:= 1 to 10 do
    begin
      s:= 0;
      for j:= 1 to 5 do S:= S + A[i,j]; {подсчет суммы оценок i-го студента}
      sr:=s/5;                          {подсчет его среднего балла}
      writeln(i,' студент - ',sr:4:2)   {вывод на экран результата}
    end
  end.
```

2 Содержание работы

2.1 Составьте и отладьте программу решения следующей задачи.

Задана целочисленная матрица из 3 строк и 4 столбцов. Найти сумму всех элементов данной матрицы.

Указание: Используйте фрагмент программы следующего вида:


```

s:=0;
for i:=1 to 3 do
  for j:=1 to 4 do
    begin
      writeln('Введите элемент массива');
      readln(a[i,j]);
      s:=s+a[i,j]
    end;
  writeln('сумма s=', s);

```

2.2 Дополните полученную программу. Найдите сумму элементов первой строки данной матрицы.

Вид возможного решения:

```

s1:=0;
for j:=1 to 4 do s1:=s1+a[1,j];
writeln('сумма первой строки s1=', s1);

```

2.3 Организуйте вывод на экран исходной матрицы. Учтите, что печать элементов матрицы должна осуществляться в три строки и четыре столбца.

Указание: Дополните программу фрагментом следующего вида:

```

for i:=1 to 3 do
begin
  for j:=1 to 4 do write (a[i,j]:5);
  writeln
end;

```

2.4 Выполните задание из указанного преподавателем варианта приложения О.

2.5 Составьте и отладьте программу решения следующей задачи:

Задана целочисленная матрица A размерности 5x7. Переставьте местами первую и последнюю строку данной матрицы.

Указание: Для задания матрицы A используйте случайные числа из диапазона [-10,10]:

```

for i:=1 to 5 do
  for j:=1 to 7 do
    a[i,j]:=-10+random(21);

```

Замечание: Функция random(n) служит для задания случайных целых чисел из диапазона $0 \leq x \leq n-1$.

После заполнения матрицы выведите ее на экран (см. п.4.3).

Для перестановки местами первой и последней (в данном случае пятой) строки используйте цикл следующего вида:

```

for j:=1 to 7 do
begin
  c:=a[1,j];
  a[1,j]:=a[5,j];
  a[5,j]:=c
end;

```

В завершении программы выведите на экран полученную в результате преобразований матрицу (см. п.4.3).

2.6 Выполните задание на перестановку или замену элементов матрицы в соответствии с указанным преподавателем вариантом приложения П.

Указание: Для заполнения исходной матрицы используйте случайные числа из диапазона $[-5,5]$. При этом выведите на экран исходный и результирующий двумерный массив в матричной форме (с предварительным указанием соответствующего названия).

2.7 Решите указанные преподавателем задачи из приложения Р.

3 Требования к отчету

Отчет о проделанной работе должен содержать:

- название и цель работы;
- номер и условие своего варианта;
- тексты разработанных программ;
- полученные при расчетах результаты;
- анализ полученных результатов;
- ответы на контрольные вопросы в соответствии с указаниями преподавателя.

4 Контрольные вопросы

- 1) Какой массив называют двумерным?
- 2) Приведите примеры двумерных массивов.
- 3) Поясните понятие матрицы, размерности матрицы.
- 4) Какие матрицы называют квадратными?
- 5) Как описываются двумерные массивы на языке Pascal?
- 6) Приведите пример описания на языке Pascal целочисленной матрицы из 5 строк и 8 столбцов.
- 7) Сколько элементов содержится в двумерном массиве $A: \text{Array}[0..1, -1..1] \text{ of integer}$? Объясните ответ.
- 8) Как Вы думаете, сколько чисел можно записать в трехмерный массив $X: \text{Array}[0..1, 0..1, 0..1] \text{ of integer}$? Почему?
- 9) Как осуществить обращение к отдельному элементу двумерного массива?
- 10) Как можно организовать автоматическое заполнение элементов матрицы случайными числами? Приведите пример.
- 11) Приведите пример фрагмента программы, который выводит на экран двумерный массив в виде матрицы из N строк и M столбцов.
- 12) Какие индексы соответствуют элементам, расположенным на главной диагонали квадратной матрицы размерности 5×5 ?
- 13) Как найти сумму элементов матрицы, расположенных выше главной диагонали? Приведите соответствующий фрагмент программы.
- 14) Приведите алгоритм нахождения наибольшего элемента в первой строке матрицы из N строк и M столбцов.
- 15) Приведите алгоритм нахождения суммы элементов второго столбца заданной матрицы размерности $N \times M$.

Лабораторная работа №6

Работа со строковыми переменными

Цель работы

Научиться принципам работы со строковыми переменными в среде программирования Pascal.

Задачи работы

Изучить приемы решения задач с использованием строковых переменных.

1 Общие положения

1.1 Тип STRING

Для обработки текстов в среде программирования Pascal используется тип **STRING** (строка). Количество символов в строке-переменной задается объявлением типа **STRING[N]**, где значение **N** может быть любой константой порядкового типа, не превышающей 255.

Объявление типа **STRING[N]** указывает, что число символов в строке-переменной может изменяться в диапазоне от 1 до **N**. Если при описании типа количество символов в строке не указано, то по умолчанию длина строки принимается максимально возможной, т.е. 255 символов.

Пример 1. Ввод и проверка пароля.

Составить программу, запрашивающую пароль и определяющую, является ли введенный с клавиатуры пароль верным.

Предположим, что для входа в программу установлен пароль Admin (Администратор). Возможный в этом случае вид решения задачи:

```
PROGRAM password;  
var w:string;  
begin  
  writeln('Введите пароль');  
  readln(w);  
  if w='Admin' then writeln ('Пароль верен') else writeln ('Пароль неверен')  
end.
```

Строка в Pascal трактуется как цепочка символов. К любому символу в строке можно обратиться так же, как к элементу символьного массива типа **ARRAY [1..N] OF CHAR**, например, к первому символу строковой переменной **s** можно обратиться как **s[1]**. Над символами строки можно осуществлять необходимые действия и, тем самым, изменять содержание строки, ее длину и т.д.

Пример 2. Перестановка символов.

Поменять местами первый и второй символ в заданном слове произвольной длины.

Возможное решение:

```
PROGRAM Str;  
var t: string; c: char;
```

```

begin
  writeln('Введите слово');
  readln(t);           { ввод слова }
  c:=t[1];             { перестановка первого и второго символа }
  t[1]:=t[2];
  t[2]:=c;
  writeln('Ответ: ',t)  { печать строки, полученной в результате
перестановки }
end.

```

Пример 3. Обработка списка строк.

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

Одно из возможных решений данной задачи имеет следующий вид:

```

PROGRAM Spisok;
Var   s: string[20];
      i, k: integer;
begin
  k:=0;
  for i:=1 to 6 do
    begin
      writeln('Введите слово');
      readln(s);
      if s[1]='A' then k:=k+1 {Сравнение первой буквы слова с буквой А}
    end;
    writeln('k=',k)
  end.

```

К строкам можно применять операцию сцепления, например,
 fio:=fam+' '+imya+' '+otch;

Допускается сравнение строковых переменных. Операции отношений =, <, >, <=, >= выполняются над двумя строками посимвольно, слева направо, с учетом внутренней кодировки символов.

1.2 Строковые процедуры и функции

Для обработки строковых и символьных переменных в языке Pascal используется ряд стандартных функций и процедур (таблицы 4-5).

Таблица 4 Функции обработки строковых и символьных переменных

Общий вид функции	Назначение
<i>Length</i> (S: string): byte	Находит длину (количество символов) строки S
<i>Concat</i> (S1,S2,...,SN: string): string	Возвращает объединенную строку S1+S2+...+SN
<i>Copy</i> (S: string; p, k: integer): string	Возвращает фрагмент строки S, который начинается в позиции p и имеет длину k

Окончание таблицы 4

Общий вид функции	Назначение
<i>Pos</i> (S1, S: string): byte	Ищет первое вхождение подстроки S1 в строку S и возвращает номер первого символа S1 в строке S или 0, если вхождение не найдено
<i>UpCase</i> (C: char):char	Преобразует строчную латинскую букву в прописную. Любые другие символы возвращаются без преобразования
<i>Ord</i> (Ch: char): longint	Возвращает код ASCII символа Ch
<i>Chr</i> (x: byte):char	Возвращает символ с заданным кодом ASCII x
<i>Succ</i> (ch: char):char	Возвращает следующий за ch символ таблицы ASCII. В общем случае, данная функция применима ко всем порядковым типам данных, возвращает элемент, следующий в списке значений.
<i>Pred</i> (ch: char):char	Возвращает предшествующий ch символ таблицы ASCII. Данная функция применима и ко всем порядковым типам данных; возвращает элемент, предшествующий в списке значений.

Таблица 5 Процедуры обработки строковых переменных

Общий вид процедуры	Назначение
<i>Delete</i> (var S: string; p, k: integer)	Удаляет из строки S фрагмент, который начинается в позиции p и имеет длину k
<i>Insert</i> (S1: string; var S:string; p:integer)	Вставляет подстроку S1 в строку S, начиная с заданной позиции p
<i>Str</i> (X [:n:d]; Var S: string)	Преобразует число X типа Real или Integer в строку символов S. Параметры n и d не являются обязательными и задают формат преобразования (общую ширину поля и количество символов в дробной части соответственно)
<i>Val</i> (S: String; var X; var Code: integer)	Преобразует строковое значение S в его численное представление X типа Real или Integer. Параметр Code содержит признак ошибки преобразования (0 – нет ошибки)

Использование данных функций и процедур позволяет решать разнообразные задачи обработки данных строкового типа.

Пример 4. Нахождение наиболее короткого слова списка.

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

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

Пример возможного решения поставленной задачи:

```
PROGRAM Str_min;
type M_st=array[1..8] of string[20];
    M_int=array[1..8] of integer;
var s: M_st;
    n: M_int;
    i, min: integer;
begin
  for i:=1 to 8 do
    begin
      writeln('Введите слово');
      readln(s[i]);
      n[i]:=length(s[i])          {Вычисление количества букв в i-ом слове }
    end;
  min:=n[1];
  for i:=2 to 8 do
    if min>n[i] then min:=n[i];    {Нахождение наименьшего из количеств
букв}
  writeln('Наиболее короткое слово');
  for i:=1 to 8 do
    if n[i]=min then writeln(s[i]) {Печать слов, состоящих из наименьшего
количества букв}
  end.
```

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

Дополнительные сведения по обработке строк приводятся в [2, 5-7].

2 Содержание работы

2.1 Составьте и отладьте программу решения следующей задачи.

Задана строка произвольной длины. Найдите количество символов в данной строке.

Указание: Используйте фрагмент программы следующего вида:

```

...
writeln('Введите слово');
readln(s);
n:=length(s);
writeln('n=',n)

```

2.2 Дополните полученную программу. Определите, начинается ли слово со слога «Не».

Указание: Воспользуйтесь условным оператором вида:

```
if (s[1]='Н') and (s[2]='е') then writeln('да, начинается') else writeln('нет');
```

2.3 Дополните полученную программу. Определите, сколько раз встречается буква «а» в данном слове.

Пример возможного решения:

```

...
k:=0;
for i:=1 to length(s) do
  if s[i]='а' then k:=k+1;
writeln('k=',k);

```

2.4 Выполните задание из указанного преподавателем варианта приложения С.

2.5 Составьте и отладьте программу решения следующей задачи.

Задан список из 7 слов произвольной длины. Найдите все слова списка, заканчивающиеся на букву «и».

Указание: Опишите переменную s как переменную типа array[1..7] of string;

Далее воспользуйтесь следующим программным фрагментом:

```

...
for i:=1 to 7 do
begin
  writeln('Введите слово');
  readln(s[i])
end;
writeln('Заканчиваются на букву -и');
for i:=1 to 7 do
begin
  n:=length(s[i]);
  if s[i,n]='и' then writeln(s[i])
end;

```

2.6 Выполните задание из указанного преподавателем варианта приложения Т.

2.7 Решите указанные преподавателем задачи из приложения У.

3 Требования к отчету

Отчет о проделанной работе должен содержать:

- название и цель работы;
- номер и условие своего варианта;
- тексты разработанных программ;
- полученные при расчетах результаты;
- анализ полученных результатов;
- ответы на контрольные вопросы в соответствии с указаниями преподавателя.

4 Контрольные вопросы

- 1) Как описываются строковые переменные?
- 2) Какая максимальная длина строки допустима в языке Pascal?
- 3) Какие операции допустимы над строковыми данными?
- 4) Какие стандартные процедуры и функции для работы со строками Вы знаете?
- 5) Как можно обратиться в программе к первому символу строковой переменной *s*?
- 6) Чему равно значение *x[3]* после присваивания *x:='012345'*, где переменная *x* имеет тип *string*?
- 7) Как происходит сравнение строковых переменных?
- 8) Какими способами можно сцепить строковые переменные?
- 9) Поясните назначение и формат оператора *Length*.
- 10) Как удалить несколько символов из строковой переменной?
- 11) Как добавить несколько символов в строковую переменную?
- 12) Поясните назначение операторов *Str* и *Val*.
- 13) Для чего служит оператор *UpCase*?
- 14) Чему равно значение выражения *S:=Copy(S, 2, 3)*, если *S:='ABCDEF'* ?
- 15) Чему равно значение процедуры *Delete(T,8,4)*, если *T:='Иванов Петр'*?

Лабораторная работа №7

Подпрограммы языка Pascal

Цель работы

Усвоить принцип конструирования процедур и функций и их использование в программах.

Задачи работы

Научиться оформлять задачи в виде подпрограммы, использовать процедуры и функции на языке Pascal, составлять рекурсивные подпрограммы.

1 Общие положения

1.1 Общие сведения о подпрограммах

Решение сложной задачи довольно часто сводится к решению нескольких более простых подзадач, которые удобно выделить в отдельные подпрограммы. Под **подпрограммой** принято понимать выделенный в самостоятельную единицу участок программы, имеющий собственное имя, позволяющий вызвать подпрограмму из основной программы. Разработка подпрограмм необходима при многократном использовании в разных местах программы группы операторов, выполняющих однотипные действия. В языке Pascal имеется также возможность создания библиотек (модулей), состоящие из подпрограмм.

В языке Pascal имеется две разновидности подпрограмм – процедуры и функции. Структура подпрограммы аналогична структуре программы (рисунок 10).

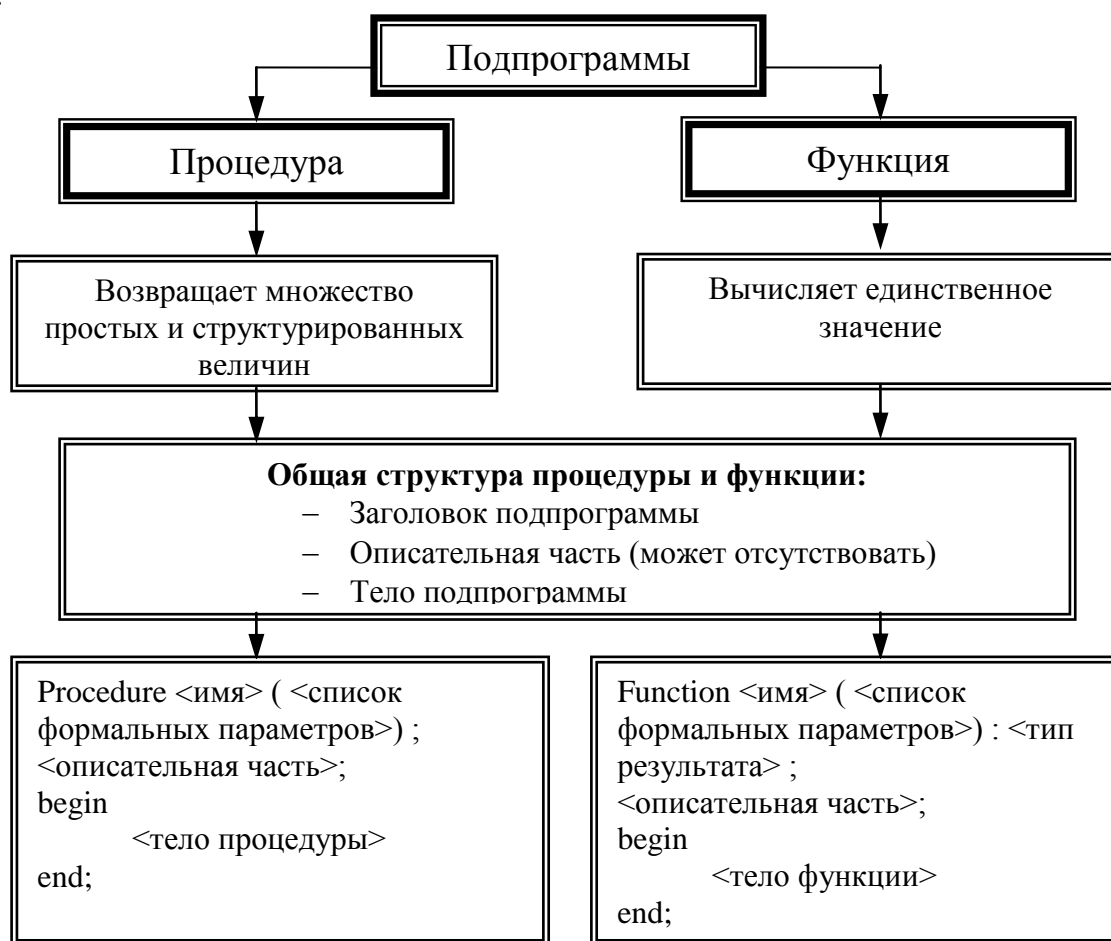


Рисунок 10 - Структура подпрограмм

1.2 Локальные и глобальные переменные, формальные параметры

Все переменные, которые использует подпрограмма, можно разбить на две категории: **локальные** переменные, объявленные внутри подпрограммы и доступные только ей самой, и **глобальные** - объявленные в основной программе и доступные как программе, так и всем ее подпрограммам. Обмен информацией между основной программой и подпрограммой может осуществляться только с помощью глобальных переменных двумя способами: непосредственно обращаясь к глобальной переменной по имени или используя механизм **формальных** параметров.

Формальные параметры задаются в заголовке подпрограммы в виде списка, разбитого на группы, разделенного точками с запятыми. Для каждой группы переменных указывается ее тип, например:

```
Procedure treug(x,y,z: Integer; Var ss: Real; Var pp: Integer);
```

По своему назначению формальные параметры могут быть **параметрами-значениями** и **параметрами-переменными**. Параметр-значение в подпрограмму передается, но обратно не возвращается; параметр-переменная в подпрограмме вычисляется, и в программу передается его новое значение. Перед **параметром-переменной** ставится служебное слово **Var**. Действие слова **Var** распространяется до ближайшей точки с запятой, т. е. в пределах одной группы.

Имена параметров могут быть любыми, в том числе и совпадать с именами глобальных переменных. В этом случае они хранятся в различных ячейках памяти, а глобальная переменная с таким именем становится недоступной для использования подпрограммой.

Тип формального параметра может быть практически любым за исключением файлового, однако в заголовке подпрограммы нельзя вводить новый тип. Например, чтобы из процедуры передать массив, следует в основной программе ввести тип-массив, а затем использовать его в заголовке:

```
type Mas = array[1..100] of Real;
procedure ZapolnMas (k: integer, var A: Mas);
```

здесь A - параметр-переменная, k - параметр значение.

Для вызова подпрограммы следует указать ее имя и список **фактических параметров**, значения которых передаются формальным параметрам. Все фактические параметры при этом разделяются запятыми. Между формальными и фактическими параметрами должно быть **взаимнооднозначное соответствие**, т.е. они должны совпадать по количеству, типу и порядку следования.

При обращении к подпрограмме формальные параметры заменяются соответствующими фактическими параметрами. Заметим, что параметры-значения могут быть заменены не только переменными и числами, но и выражениями соответствующего типа, тогда как параметры-переменные могут быть заменены только переменными.

1.3 Подпрограмма-процедура

Любая процедура начинается с заголовка. В отличие от основной программы заголовок в процедуре обязателен. Общий вид заголовка:

Procedure <имя процедуры>(<формальные параметры> : <тип>);

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

В качестве результата процедура может возвращать в вызывающую программу множество простых или структурированных величин. Параметры, передающие результаты работы процедуры, должны быть обязательно параметрами-переменными.

Разрешается описывать процедуры без формальных параметров, тогда организация передачи информации из программы в процедуру и обратно проводится с помощью глобальных переменных, а заголовок процедуры будет иметь более простой вид: **Procedure** <имя процедуры>;

Для вызова процедуры из основной программы или другой подпрограммы следует записать имя процедуры и указать список **фактических параметров**. Например:

```
Streug(a,b,c,s);
Streug(a/b,c+5,7,s1); .
```

Вызов процедуры без формальных параметров осуществляется по имени.

Приведем пример использования процедуры, служащей для организации вывода на экран двумерного массива в матричном виде.

Пример 1. Задана целочисленная матрица размера 3x4. Требуется заменить в ней все отрицательные элементы числом 0.

Ниже представлен пример решения данной задачи с использованием процедуры `wr_matr`, предназначенной для распечатывания элементов массива в виде матрицы из трех строк и четырех столбцов. Данная процедура используется в программе дважды - для вывода на экран исходной матрицы и для вывода матрицы, полученной после требуемых преобразований.

Program m_m;

Type Matr=array[1..3,1..4] of integer;

Var a,b : Matr; i,j: integer; {Описание глобальных переменных}

Procedure wr_matr(a: Matr);

begin

for i:=1 to 3 do

begin

for j:=1 to 4 do write(a[i,j]:5);

writeln;

end;

end;

begin

for i:=1 to 3 do

for j:=1 to 4 do

begin

writeln('Введите элемент a['i','j,']'); readln(a[i,j]);

```

end;
writeln('Исходная матрица');
wr_matr(a); {Вызов подпрограммы с указанием фактического параметра}
for i:=1 to 3 do
  for j:=1 to 4 do if a[i,j]<0 then b[i,j]:=0 else b[i,j]:=a[i,j];
writeln('Итоговая матрица');
wr_matr(b);
end.

```

Пример 2.

Даны два массива из N целых элементов из интервала [-20, 30]. В каждом массиве найти максимальный элемент и затем сравнить их.

В программе будем использовать три подпрограммы процедуры: заполнения массивов случайными числами, вывода массивов и нахождения максимального элемента.

Program SravMas;

Const n=15;

Type massiv=Array [1..n] of Integer;

Var m1, m2 : massiv; max1, max2 : Integer;

Procedure RandomMas(Var x:massiv);

Var i:Integer;

Begin

For i:=1 to n do x[i] := -20+Random(51);

End;

Procedure VivodMas(Var x:massiv);

Var i:Integer;

Begin

For i:=1 to n do write(x[i]:5);

End;

Procedure MaxMas(Var x:massiv; Var MX:Integer);

Var i:integer;

Begin

MX:=x[1];

For i:=2 to n do if x[i]>MX then MX:=x[i];

End;

Begin {основная программа}

Randomize;

RandomMas(m1);

Writeln('Первый массив:');

VivodMas(m1);

RandomMas(m2);

Writeln('Второй массив:');

VivodMas(m2);

MaxMas(m1,max1); MaxMas(m2, max2);

if max1>max2 then

```

        writeln('В первом массиве max элемент больше и равен', max1)
else if max1<max2 then
        writeln('Во втором массиве max элемент больше и равен', max2)
else writeln('В обоих массивах max равны ', max1);

```

End.

1.4 Подпрограмма-функция

Подпрограмма-функция имеет общий вид:

```

Function <имя> ( <список формальных параметров> ) : <тип результата>;
<раздел описаний локальных переменных>;
begin
    <тело функции>
end;

```

Подпрограмма-функция вычисляет **одно единственное значение**, которое передается через ее **имя**. Функция может возвращать параметры следующих типов: любого порядкового, любого вещественного, строкового типа string, указателя.

В отличие от подпрограммы-процедуры, заголовок подпрограммы-функции состоит из слова **Function**, за которым следует имя функции со списком формальных параметров и типом возвращаемого параметра; в теле подпрограммы-функции обязательно должно быть присвоено значение
 <имя функции> := <выражение>.

Пример 3. Составить программу вычисления (табулирования) функции $F(x)=x \cdot \sin(x)$ а отрезке $[a, b]$ с шагом h .

Вычисление значения функции оформим в виде функции.

```

Program Primer;
Var a,b,h,x : real;
Function f(t : real) : real;
Begin
    f:=t*sin(t);
End;
Begin
    Write('Введите начало и конец отрезка: '); Readln(a,b);
    Writeln('Введите шаг табуляции: '); Readln(h);
    Writeln(' X      F(X)');
    x:=a;
    While x<=b do
        begin
            writeln(x:6:3, f(x):10:3);
            x:=x+h;
        end
End.

```

Пример 4. Составить программу нахождения корня уравнения $F(x)=0$ с использованием метода половинного деления.

Метод половинного деления (дихотомии) нахождения корня уравнения $F(x)=0$ заключается в следующем. Выбирается некоторый отрезок $[A, B]$, на котором предположительно находится один из корней уравнения. Далее исходный отрезок делится пополам. Затем анализируется изменение знака функции на половинах отрезка, и та граница отрезка, со стороны которой функция не меняет своего знака, переносится в его середину. Далее процесс повторяется. Итерации прекращаются при выполнении одного из условий: либо длина интервала $[A, B]$ становится меньше заданной погрешности нахождения корня E , либо значение функции становится сравнимо с погрешностью расчетов.

Соответственно, программа расчетов может иметь следующий вид:

Program Dihotom;

Var A, B, E, Y, YA, x: real;

Function F(x: real): real;

Begin

F:=cos(x); {пример задания конкретной функции для расчетов}

End;

begin

writeln('Ведите отрезок A, B');

readln(A, B);

writeln('Введите погрешность E');

readln(E);

Repeat {начало итерации}

 x:= (A + B)/2; {x - середина отрезка [A, B]}

 Y:= F(x);

 YA:= F(A); {значения функции в середине и на конце отрезка}

 if Y*YA > 0 {если знаки функции в точках A и x совпадают}

 then A:= x else B:= x {то перенос границы A, иначе - B}

Until (ABS(B-A)< E) OR (ABS(F(x))< E);

writeln('x=',x: 7: 3)

end.

Как видим, описание функции $F(x)$ осуществляется в отдельной подпрограмме-функции, что позволяет легко изменять программу для решения других видов уравнений.

1.5 Рекурсивная подпрограмма

Рекурсивной называется подпрограмма, в которой содержится обращение к самой себе. Возможность рекурсивного обращения связана с тем, что при каждом новом обращении к подпрограмме параметры, которые она использует, заносятся в память, причем параметры предыдущего обращения также сохраняются. Ясно, что для завершения вычислений каждая рекурсия должна содержать хотя бы одну нерекурсивную ветвь алгоритма, заканчивающуюся возвращением в вызывающую программу. В ряде случаев

рекурсивное оформление подпрограммы может быть более компактным и эффективным, но не следует забывать о расходе времени на повторные вызовы функции и об опасности переполнения ячейки памяти.

Простой пример рекурсивной функции – вычисление факториала:

$$n! = \begin{cases} 1 & \text{при } n = 0, 1, \\ n * (n-1)! & \text{при } n > 1. \end{cases}$$

Пример 5. Пример использования функции, рекурсивно вычисляющей факториал.

```
Program rec;
var n:byte;
rez: longint;
Function fact(k: byte): longint;
begin
  if k > 0 then fact:= k*fact(k-1)
  else fact:=1;
end;
begin
  writeln ('Ведите n');
  readln(n);
  rez:=fact(n);
  writeln('rez=', rez)
end.
```

2 Содержание работы

2.1. С использованием подпрограммы-процедуры составить программу решения следующей задачи. Даны два массива из N целых чисел из интервала [-20, 20]. В каждом массиве найти наименьший элемент и затем сравнить их между собой. При выполнении задания использовать соответствующий пример из текста методических указаний.

2.2. С использованием подпрограммы-функции выполнить задание для вычисления на заданном отрезке функции своего варианта из приложения Ф.

2.3. Используя метод половинного деления найти корень уравнения $F(x) = -x^2 - 2x + 1$ на отрезке [0, 1] с точностью $E = 0.001$. Для описания функции $F(x)$ использовать подпрограмму-функцию.

2.4. Составить программу вычисления числа сочетаний $C_n^k = \frac{n!}{k!(n-k)!}$, используя рекурсивную функцию вычисления факториала числа.

2.5. Выполнить индивидуальное задание по вариантам из приложения Х, оформив основную подпрограмму в виде функции, а подпрограммы заполнения и вывода массива – в виде отдельных процедур.

Указания. Размерности массивов следует задать именованными константами. Все необходимые данные должны передаваться подпрограммам в качестве параметров; все величины, используемые только внутри подпрограмм, должны быть описаны как локальные. Вывод результатов работы подпрограмм

должен выполняться в главной программе. Заполнение массивов следует производить с помощью генератора случайных чисел Random.

2.6. Сохранить все результаты в файлах и оформить отчет.

3 Требования к отчету

Отчет по работе должен содержать:

- название, цель, задачи работы;
- номер и условие своего варианта;
- текст программ;
- полученные численные результаты;
- ответы на контрольные вопросы по указанию преподавателя.

4 Контрольные вопросы

- 1) Дайте определение подпрограммы.
- 2) В каких случаях целесообразно использование подпрограмм?
- 3) Какие разновидности подпрограмм имеются в языке программирования Pascal?
- 4) Поясните отличия между локальными и глобальными переменными.
- 5) Какими способами может осуществляться обмен информацией между основной программой и подпрограммой?
- 6) Где задаются формальные параметры подпрограммы?
- 7) Чем отличаются формальные параметры-значения и параметры-переменные?
- 8) Могут ли имена формальных параметров совпадать с именами глобальных переменных?
- 9) Приведите общий вид заголовка подпрограммы-процедуры.
- 10) Можно ли описать подпрограмму-процедуру без формальных параметров?
- 11) Как вызвать процедуру из основной программы?
- 12) Приведите общий вид подпрограммы-функции.
- 13) Назовите основные отличия подпрограмм-функций от подпрограмм-процедур.
- 14) Какая подпрограмма называется рекурсивной?

Библиографический список

- 1) Аляев Ю.А. Алгоритмизация и языки программирования Pascal, C++, Visual Basic/ Ю.А. Аляев. – М.: Финансы и статистика, 2007. – 320 с.
- 2) Информатика. Базовый курс: учеб. пособие для студентов ВТУЗов/ Ред. С.В.Симонович. – СПб. и др.: Питер, 2008. - 639 с.
- 3) Культин Н. Turbo Pascal в задачах и примерах.- СПб.: BHV, 2004. – 256 с.
- 4) Немнюгин С.А. Turbo Pascal. Программирование на языке высокого уровня. Учебник для ВУЗов. - СПб.: Питер, 2005. – 544 с.
- 5) Павловская Т.А. Паскаль. Программирование на языке высокого уровня. Учебник для ВУЗов. - СПб.: Питер, 2003. – 393 с.
- 6) Практикум по алгоритмизации на языке Паскаль/ Ю.А. Аляев, В.П. Гладков, О.А. Козлов. – М.: Финансы и статистика, 2004. – 527 с.
- 7) Программирование на языке Паскаль: задачник/ Под ред. О.Ф.Усковой. – СПб.: Питер, 2003. – 336 с.
- 8) Фаронов В.В. Турбо Паскаль. – СПб.: Питер, 2006. - 368 с.
- 9) Шамсутдинова, Т. М. Электронный учебник по основам алгоритмизации и программирования на языке Паскаль 7 [Электронный ресурс]: [электрон. учеб.]: подгот. по свидетельству об отраслевой регистрации разработки № 2366 от 20.02.2003; ИнРФ 50200300141 от 28.02.2003. – Электрон. дан. и прогр. – Уфа: БГАУ, 2003. – 1 электрон. опт. диск (CD-ROM); 12 см. – Систем. требования: ПК Pentium; Windows 95/98/2000/XP; ОЗУ 32 Мб. – Загл. с экрана.

Команды редактора среды Pascal

Работа в среде программирования Pascal начинается с загрузки файлов turbo.exe (для среды Turbo Pascal) или bp.exe (для среды Borland Pascal). Затем необходимо ввести текст программы. Взаимодействие пользователя со средой осуществляется с помощью основного и контекстного меню. Основное меню вызывается во время работы клавишей [F10] и имеет команды, приведенные в таблице А1.

Таблица А1 Команды основного меню среды Pascal

File	Режим работы с файлами. Позволяет выполнять все основные операции с файлами: NEW – создать новый файл OPEN – открыть ранее созданный файл SAVE - сохранить на диске под текущим именем SAVE AS – сохранить на диске под любым именем, с указанием места расположения файла
Edit	Редактирование файла (копирование, вставка, удаление фрагмента текста, возможность восстановить первоначальный вариант редактируемого текста)
Search	Поиск файла (поиск фрагментов текста и при необходимости производить замену найденного фрагмента новым)
Run	Запуск программы на выполнение
Compile	Компиляция программы
Debug	Режим отладки (содержатся команды, облегчающие процесс поиска ошибок в программе: Breakpoints – точки остановки, окно отладки Watch, окно используемых подпрограмм, окно регистров, окно выходных результатов и др.)
Tools	Инструменты (позволяет выполнять некоторые программы, не выходя из интегрированной инструментальной оболочки Turbo Pascal)
Options	Опции (позволяет установить необходимые для работы параметры компилятора в Turbo Pascal)
Window	Окно (позволяет выполнять все основные операции с окнами: открывать, закрывать, перемещать, изменять размер)
Help	Помощь (позволяет получить справочную информацию)

После ввода и редактирования текста программы необходимо ее сохранить с помощью клавиши [F2] или выбрать соответствующую команду сохранения из команды File основного меню.

Продолжение приложения А

Компиляция программы - это следующий этап работы. Для этого необходимо нажать клавишу [F9]. При этом компилятор может обнаружить синтаксические ошибки в программе (выдается соответствующее сообщение (Error...) на экран, с указанием места ошибки), которые нужно исправить (смотри приложение Б). При возникновении неясных ситуаций, относящихся как к характеру ошибки, так и к правилам языка Pascal или правилам использования библиотечных подпрограмм можно смело использовать встроенную контекстную справочную систему.

Затем, откорректировав программу, сохранить ее последнюю версию с помощью [F2] и перейти к процессу запуска на решение. Запуск программы на решение осуществляется комбинацией клавиш [Ctrl]+[F9] (или вызовом команды Run из меню Run). Если в программе не предусмотрена приостановка выполнения для просмотра результатов работы, выведенные на экран результаты, перекрываются окном интегрированной среды. Временно убрать это окно и просмотреть результаты работы можно, нажав комбинацию клавиш [Alt]+[F5].

Таблица А2 Специальные комбинации клавиш для запуска и отладки программ

Клавиши	Команда меню	Функция
Alt+F9	Compile ► Compile	Компиляция файла
Ctrl+F2	Run ► Program Reset	Сброс выполняемой программы в исходное состояние
Ctrl+F4	Debug ► Evaluate/Modify	Вычислить выражение или модифицировать значение переменной
Ctrl+F7	Debug ► Add Watch	Добавить выражение для просмотра
Ctrl+F8	Debug ► Toggle Breakpoint	Установить или удалить точку останова
Ctrl+F9	Run ► Run	Запустить программу на выполнение

Таблица А3 Специальные комбинации клавиш для редактирования

Клавиши	Команда меню	Функция
Ctrl +Del	Edit ► Clear	Удаление выбранного текста
Ctrl +Ins	Edit ► Copy	Копирование выбранного текста в буфер
Shift +Del	Edit ► Cut	Перемещение выбранного текста в буфер
Shift +Ins	Edit ► Paste	Запись текста из буфера в активное окно
Ctrl +L	Search ► Search Again	Повторяет последнюю команду Find или Replace

Продолжение приложения А

Таблица А4 Назначение функциональных клавиш

Клавиши	Команда меню	Функция
F1	Help	Показывает экран справки
F2	File ► Save	Сохранение файла
F3	File ► Open	Открытие файла
F4	Run ► Goto Cursor	Выполнение фрагмента программы подсвеченной строки до строки, на которой стоит курсор
F5	Window ► Zoom	Изменение размера активного окна
F6	Window ► Next	Переход к следующему окну редактирования
F7	Run ► Trace Into	Запуск программы в режиме отладки с заходом внутрь процедур
F8	Run ► Step Over	Запуск программы в режиме отладки, минуя вызовы процедур
F9	Compile ► Make	Компиляция программы из текущего окна
F10		Активизация строки меню

Таблица А5 Специальные комбинации клавиш редактирования

Клавиши	Функции
Ctrl +K, затем R	Вставить в текущей позиции курсора файл с диска
Ctrl +K, затем W	Записать выделенный блок в файл
Ctrl +K, затем любая цифра	Установить маркер с номером, соответствующим использованной цифре
Ctrl +K, затем T	Выделить слово
Ctrl +Q, затем B	Перейти в начало выделенного блока
Ctrl +Q, затем K	Перейти в конец выделенного блока
Ctrl +Q, затем любая цифра	Перейти к маркеру с номером, соответствующим использованной цифре
Ctrl +Q, затем A	Найти и заменить
Ctrl +Q, затем P	Перейти к предыдущей позиции курсора
Ctrl +Q, затем L	Установить строку
Ctrl +Q, затем Y	Удалить текст до конца строки
Ctrl +Q, затем W	Восстановить последнее сообщение об ошибке компиляции
Ctrl +Q, затем [Найти парную скобку в направлении вперед
Ctrl +Q, затем]	Найти парную скобку в направлении назад
Ctrl +Y	Удалить строку
Ctrl +N	Вставить строку
Ctrl +Home	Перейти в начало экрана
Ctrl +End	Перейти в конец экрана
Ctrl +PgUp	Перейти к началу файла
Ctrl +PgDn	Перейти к концу файла
Shift + клавиши управления курсором	Выделение блока

Таблица А6 Специальные комбинации клавиш для управления окнами

Клавиши	Команда меню	Функция
Alt+ цифра 1-9		Переход к окну с заданным номером
Alt +O	Window ► List	Показать список открытых окон
Alt +F3	Window ► Close	Заккрыть активное окно
Alt +F5	Window ► User Screen	Показать экран пользователя
Shift +F6	Window ► Previous	Переход к предыдущему открытому окну
Ctrl +F5	Window ► Size/Move	Изменение размера (Shift + клавиши со стрелками) или положения (клавиши со стрелками) открытого окна

Таблица А7 Специальные клавиши и комбинации клавиш встроенной справочной системы

Клавиши	Команда меню	Функция
F1	Help ► Contents	Открывает контекстно – ориентированный экран справочной информации
F1 F1	Help ► Help on Help	Вызов справки по справочной системе
Shift + F1	Help ► Index	Вызов оглавления справочной системы
Alt +F1	Help ► Previous Topic	Показать предыдущий экран справочной информации
Ctrl +F1	Help ► Topic Search	Вызов контекстной информации по языку Pascal

Для поиска ошибок времени выполнения можно воспользоваться встроенным отладчиком. Сообщение о такой ошибке имеет вид

Run - time error <errnum> at <segment>:<offset>, где <errnum> - это код ошибки, а <segment>:<offset> - адрес в памяти, где произошла ошибка.

Интегрированный отладчик Turbo Pascal дает возможность пошагового выполнения программы. При этом можно рассматривать значения различных переменных, что иногда дает ценную информацию о реальной работе программы. Для запуска сеанса отладки нужно выбрать команду Run ► Trace Into или нажать клавишу [F7]. При этом программа вначале компилируется, а затем начинается ее пошаговое выполнение. Каждый шаг заключается в выполнении очередной строки операторов, и происходит он при очередном нажатии на клавишу [F7].

Во время отладки можно использовать окно просмотра (Watch). Если нажать клавиши [Alt]+[D] для появления меню Debug и выбрать команду Add Watch (или [Ctrl]+[F7]), далее в появившемся окне можно выбрать имя переменной, текущее значение которой необходимо узнать. Для добавления в окно просмотра других переменных можно повторно использовать команду Add Watch. Таким образом, используя возможность пошагового выполнения в сочетании с просмотром текущих значений переменных программы, можно провести достаточно подробный анализ ее работы. Это позволяет в случае неправильного поведения программы поставить ей достаточно точный «диагноз».

Контекстное меню среды Pascal вызывается с помощью правой клавиши мыши и содержит команды:

Cut (Shift +Del)-перемещение выделенного фрагмента в буфер обмена;

Copy (Ctrl +Ins) – копирование выделенного фрагмента в буфер обмена;

Paste (Shift +Ins) – вставка содержимого буфера обмена в текст программы;

Clear (Ctrl +Del) – удаление выделенного фрагмента.

Завершить работу «зациклившейся» программы можно нажатием клавиш [Ctrl]+[Break], а завершение работы интегрированной среды производится одновременным нажатием [Alt]+[X] или с помощью команды File ► Exit.

Сообщения и коды некоторых ошибок

2 Identifier expected (Не указан идентификатор). В этом месте должен находиться идентификатор.

3 Unknown identifier (Неизвестный идентификатор). Этот идентификатор не был описан.

4 Duplicate identifier (Повторяющийся идентификатор). Попытка дважды описать один и тот же идентификатор.

5 Syntax error (Синтаксическая ошибка). В исходном тексте найден недопустимый символ.

10 Unexpected end of file (Не найден конец файла). Причины этого сообщения могут быть следующие:

- исходный файл закончился перед последним **END** основного раздела операторов; вероятно в программе неодинаковое количество операторов **BEGIN** и **END**;

- не закончен комментарий.

11 Line too long (Слишком длинная строка). Максимальная длина строки, обрабатываемая компилятором, равна 126 символам.

12 Type identifier expected (Здесь нужен идентификатор типа). Не указан тип идентификатора.

16 Disk full (Диск заполнен). Нужно удалить некоторые файлы или воспользоваться новым диском.

20 Variable identifier expected (Отсутствует идентификатор переменной). На этом месте должен быть идентификатор переменной.

21 Error in type (Ошибка в объявлении типа). Объявление типа не может начинаться с этого символа.

26 Type mismatch (Несоответствие типа). Это сообщение может быть вызвано следующими причинами:

- несовместимые типы переменной и выражения в операторе присваивания;

- тип выражения не совместим с типом индекса при объявлении массива;

- несовместимые типы операндов в выражении.

33 Label identifier expected (Нужен идентификатор метки). Метка не обозначена с помощью идентификатора, как это требуется из контекста программы.

36 BEGIN expected (Нужен BEGIN)

37 END expected (Нужен END)

38 Integer expression expected (Нужно выражение типа Integer).

41 Operand types do not match operator (Типы операндов не соответствуют операции). Данная операция не может быть применена к указанным операндам.

42 Error in expression (Ошибка в выражении). Данный символ не может участвовать в выражении указанным образом. Возможно, не указана операция между двумя операндами.

Окончание приложения Б

50 DO expected (Нужен оператор DO)

57 THEN expected (Требуется THEN)

58 TO or DOWNT0 expected (Требуется TO или DOWNT0)

62 Division by zero (Деление на ноль). Предшествующая операция пытается выполнить деление на ноль.

64 Cannot Read or Write variables of this type (Нет возможности считать или записать переменные данного типа). Нарушены следующие ограничения:

- процедуры **READ** и **READLN** могут считывать переменные символьного, целого, действительного и строкового типов;

- процедуры **WRITE** и **WRITELN** могут выводить переменные символьного, целого, действительного, логического и строкового типов.

76 Constant out of range (Константа нарушает границы). Возможные причины сообщения:

- попытка указать индекс массива, выходящий за его границы;
- попытка присвоить переменной значение, выходящее за границы, допустимые для типа этой переменной.

79 Integer or real expression expected (Нужно выражение вещественного или целого типа).

81 Label already defined (Метка уже определена). Данная метка уже помечает оператор.

85 «;» expected (Нужно указать «;»). 97 Invalid FOR control variable (Неправильный параметр цикла оператора FOR).

98 Integer variable expected (Нужна переменная целого типа). Предшествующая переменная должна иметь целый тип.

103 Integer or real variable expected (Нужна переменная типа INTEGER or REAL).

113 Error in statement (Ошибка в операторе). Данный символ не может быть первым символом в операторе.

207 Invalid floating point operation (Недопустимая операция с плавающей запятой). Возможные причины сообщения:

- отрицательный аргумент функции SQRT;
- аргумент функции LN равен нулю или имеет отрицательное значение.

Варианты задач по теме «Линейные структуры»

Вычислить значение выражения по формуле (все переменные принимают вещественные значения):

1	$d = \frac{b + \sqrt{b^2 + 4ac}}{2a + 1} - a^3 c + b^{-2}$
2	$z = \ln \left \left(y - \sqrt{ x } \right) \left(x - \frac{y}{x + \frac{x^2}{4}} \right) \right $
3	$z = (1 - \operatorname{tg} x)^2 + \cos x (x - y)$
4	$z = \frac{\cos(x)}{\pi - 2x} + 16x \cdot \cos(xy) - 2$
5	$z = e^x - \frac{y^2 + 12xy - 3x^2}{18y - 1}$
6	$y = x^3 - 4x + \left(e^x - \sqrt{ x } \right)$
7	$z = \frac{1 + \sin^2(x - y)}{2 + \left x - \frac{2x}{1 + x^2 y^2} \right } + x$
8	$z = x \ln x + \frac{y}{\cos x - \frac{x}{3}}$
9	$y = 12 \operatorname{ctg}(3x) - \frac{1}{12x^2 + 7x - 5}$
10	$z = e^{-x} - \cos x + \sin(2xy)$
11	$y = \frac{\operatorname{Ln} \cos x }{\operatorname{Ln}(1 + x^2)}$

12	$y = \frac{(1 + \sin \sqrt{x+1})}{\cos(12x-4)}$
13	$y = 2\operatorname{ctg}(3x) - \frac{\operatorname{Ln} \cos(x)}{\operatorname{Ln}(1+x^2)}$
14	$y = x - 10x \sin(x) + x^4 - x^5 $
15	$y = \cos^2\left(\sin \frac{1}{x}\right)$
16	$y = \frac{1}{x} \ln \frac{1+x}{1-x}$
17	$y = \frac{x^2 - 7x + 10}{x^2 - 8x + 12}$
18	$y = 2\operatorname{ctg}(5x) + \sin(x) - \cos\left(\frac{1}{x}\right)$
19	$y = \frac{\sin(x)}{5} + \frac{\operatorname{tg}(x)}{7} + x^3 - x^4 $
20	$y = x^2 - x^3 - \frac{7x}{x^3 - 15x}$

Задачи для самостоятельного решения по теме «Линейные структуры»

- 1) Даны основание и высота равнобедренного треугольника. Найти его периметр.
- 2) Даны два числа. Найти среднее арифметическое и среднее геометрическое их модулей.
- 3) Даны два числа. Найти их сумму, разность, произведение, а также частное от деления первого числа на второе.
- 4) Треугольник задан координатами трех вершин – (x_1, y_1) , (x_2, y_2) , (x_3, y_3) соответственно. Найти периметр и площадь треугольника.
- 5) Составьте программу, складывающую две обыкновенные дроби.
- 6) Найти площадь треугольника, две стороны которого равны a и b , а угол между этими сторонами – α .
- 7) Найти площадь кольца, внутренний радиус которого равен r_1 , а внешний r_2 .
- 8) Дано действительное число A . Получить целую часть числа, затем число A , округленное до ближайшего целого.
- 9) Вычислить объем цилиндра с диаметром основания d и высотой h .
- 10) Дана длина ребра куба. Найти объем куба и площадь его боковой поверхности.
- 11) Даны три стороны треугольника a , b , c . Вычислить его площадь.
- 12) Дано целое двухзначное число. Используя операции `div`, `mod` найти сумму его цифр.
- 13) Составить программу решения линейного уравнения $ax + b = 0$, где $a \neq 0$.
- 14) Даны катеты прямоугольного треугольника. Найти его периметр.
- 15) Дано целое двухзначное число. Используя операции `div`, `mod` найти произведение его цифр.
- 16) Даны длины сторон прямоугольного параллелепипеда. Найти его объем и площадь боковой поверхности.
- 17) Известны плановый объем P выпуска продукции и объем F фактического производства. Найти процент выполнения плана.
- 18) Составьте программу для вычисления величины силы тока на участке электрической цепи сопротивлением R Ом при напряжении U В.
- 19) Скорость первого автомобиля V_1 км/ч, второго – V_2 км/ч, начальное расстояние между ними S км. Определить расстояние между ними через T часов, если автомобили удаляются друг от друга.
- 20) Составьте программу для вычисления величины работы, совершенной при равномерном подъеме груза массой M кг на высоту H метров. Ускорение свободного падения описать как константу $G = 9,81$.
- 21) Всхожесть семян пшеницы равна 98 %. Сколько тонн семян нужно посеять, чтобы 35 тонн из них проросло?
- 22) Масса молекул кислорода в организме человека составляет 65% от его массы. Составьте программу для определения массы молекул кислорода в организме человека.

Варианты задач по теме «Операторы ветвления»

- 1) Заданы числа a и b . Определить, эти числа одного или разных знаков.
- 2) Составьте программу, проверяющую, верно ли утверждение, что введенное вами целое число делится без остатка на 3.
- 3) Даны действительные числа a , b , c . Проверить, выполняются ли неравенства $a < b < c$.
- 4) Даны два действительных числа. Заменить первое число нулем, если оно равно второму, и оставить числа без изменения в противном случае.
- 5) Даны три вещественных числа. Выбрать те из них, которые принадлежат интервалу $[1, 3]$.
- 6) Даны три действительных числа. Возвести в квадрат те из них, значения которых неотрицательны.
- 7) Составьте программу, определяющую, пройдет ли график функции $y = 5x^2 - 7x + 2$ через заданную точку с координатами (a, b) .
- 8) Даны положительные действительные числа x , y , z . Выяснить существует ли треугольник с длинами сторон x , y , z .
- 9) Даны целые числа a , b . Если числа не равны, то заменить каждое из них одним и тем же числом, равным большему из исходных, а если равны, то заменить числа нулями.
- 10) Даны три вещественных числа x , y , z . Вычислить значения выражений $x + y + z$ и $xуz$ и вывести максимальное из них.
- 11) Даны две точки: $A(x_1, y_1)$ и $B(x_2, y_2)$. Составить алгоритм, определяющий, которая точка находится ближе к началу координат.
- 12) Заданы два целых числа. Определить являются ли они оба четными или оба нечетными или какое из них четное, а какое нечетное.
- 13) Проверить имеет ли квадратное уравнение вида $ax^2 + bx + c = 0$ действительные корни. Если да, то найти их.
- 14) Составить программу упорядочивания трех чисел x , y , z по возрастанию.
- 15) Составить программу, проверяющую, верно ли утверждение, что сумма цифр введенного двузначного целого числа делится без остатка на 5.
- 16) Дано некоторое целое двузначное число. Проверить, равна ли сумма его цифр числу 7.
- 17) Определить, находится ли точка с координатами x , y во второй координатной четверти.
- 18) Даны две переменные целого типа A и B . Если их значение не равны, то присвоить каждой переменной сумму этих значений, а если равны, то присвоить переменным нулевые значения.
- 19) Известны площади круга и квадрата. Определить, поместится ли круг в квадрате?
- 20) Определите номер квадранта, в котором находится точка с заданными координатами x, y .
- 21) Заданы координаты вершин треугольника и точки. Определить, лежит ли точка внутри треугольника.

Продолжение приложения Д

- 22) С клавиатуры вводится шестизначный номер трамвайного билета. Определить, является ли он счастливым?
- 23) Составить программу, которая для натурального числа от 1 до 32000 выдает количество цифр в этом числе.
- 24) Даны два вещественных положительных числа x и y . Арифметические действия над числами пронумерованы (сложение – 1, вычитание – 2, умножение – 3, деление – 4). Составить программу, которая по введенному номеру выполнит то или иное действие над числами.
- 25) Написать программу, которая по введенному номеру единицы измерения (миллиграмм – 1, грамм – 2, килограмм – 3, центнер – 4, тонна – 5) и массе M выводит соответствующее значение массы в килограммах.
- 26) Написать программу, которая по введенному номеру месяца (числу от 1 до 12) выводит все приходящиеся на этот месяц праздничные дни (например, если введено число 1, то должно получиться Новый год, Рождество).
- 27) Дано натуральное число $n < 101$, определяющее возраст человека (в годах). Дать для этого числа подпись «год», «года», или «лет». Например, 1 год, 3 года или 55 лет и т.д.
- 28) В кубический, наполненный на 70% аквариум со стороной a метров выпустили рыбу-шар диаметром b см. Определить, выплеснется ли вода из аквариума (хвост и плавники рыбы не учитываются).
- 29) Заданы площади круга и квадрата. Определите, поместится ли квадрат в круге.
- 30) Определите, есть ли среди цифр заданного целого трехзначного числа одинаковые.
- 31) Заданы координаты двух точек. Определите, лежат ли они на одной окружности с центром в начале координат.
- 32) Станции А, В и С расположены на n -ом, m -ом и p -ом километрах железной дороги соответственно. Какие из этих станций расположены наиболее близко друг к другу?
- 33) Проверьте можно ли построить параллелограмм из отрезков с длинами x , y , v , w .
- 34) Требуется составить программу перевода латинских слов на русский язык. База знаний словаря должна включать следующие слова: aqua – вода, astrum – звезда, aurum – золото, caelum – небо, gloria – слава, humus – земля, jus – закон, locus – место, memoria – память, vita – жизнь. Входные данные программы: латинское слово. Выходные данные: соответствующий русский перевод или сообщение «нет в словаре»
- 35) Для каждой введенной цифры (0 - 9) вывести соответствующее название на английском языке (0 – zero и т.д.).
- 36) Составить программу, позволяющую получить словесное описание школьных отметок (1 – очень плохо, 2 – неудовлетворительно и т.д.)
- 37) Написать программу, которая по введенному номеру времени года выдает название его месяцев.

Окончание приложения Д

- 38) Дан номер некоторого года (положительное целое число). Вывести соответствующий ему номер столетия, учитывая, что, к примеру, началом 20 столетия был 1901 год.
- 39) Дан номер некоторого года (положительное целое число). Определить, является ли данный год високосным. Високосным считается год, делящийся на 4, за исключением тех годов, которые делятся на 100 и не делятся на 400 (например, годы 300, 1300 и 1900 не являются високосными, а 1200 и 2000 — являются).
- 40) Дано целое число, лежащее в диапазоне от 1 до 9999. Вывести строку — словесное описание данного числа вида «четное двузначное число», «нечетное четырехзначное число» и т.д.
- 41) Любитель горнолыжного спорта собирается провести свой недельный отпуск на одном из трех курортов. Курорт А открыт с начала ноября по конец апреля, но из-за лавинной опасности его закрывают на весь январь. Курорт В открыт с начала декабря по конец марта. Его закрывают на соревнования с 1 по 15 февраля. Курорт С постоянно открыт с начала октября по конец мая. Стоимость отдыха на каждом из курортов, включая проезд, составляет, соответственно, P1, P2 и P3 рублей. По дате начала отпуска определите, сможет ли он провести свой отпуск в горах, и какой минимальной суммой он должен располагать.
- 42) Стартовый номер участника соревнований по автотоспорту определяется на квалификационных заездах. При этом фиксируется время начала и конца прохождения так называемого "быстрого" круга (часы, минуты, секунды). Проверьте, корректно ли зафиксированы данные участника, и найдите время прохождения им «быстрого» круга.
- 43) Создать программу, определяющую название введенного года согласно древнекитайскому календарю, опирающемуся на двенадцатилетний цикл. Известно, что годы внутри цикла носят названия животных: крысы, коровы, тигра, зайца, дракона, змеи, лошади, овцы, обезьяны, петуха, собаки и свиньи. Название года соответствует остатку от деления номера этого года на 12. Следует учесть, что 2005 год приходится на год петуха.
- 44) Проверить, поместится ли на диске компьютера музыкальная композиция, которая длится m минут и n секунд, если свободное дисковое пространство 6 мегабайт, а для записи одной секунды звука необходимо 16 килобайт.
- 45) Для нормального разведения золотых рыбок необходимо, чтобы на каждую рыбку в аквариуме приходилось не менее 3-х литров воды. По известным объему аквариума и количеству рыбок, в нем содержащихся, определить, является ли аквариум «перенаселенным» или нет, и указать количество рыбок, которых в случае перенаселенности необходимо поместить в другой аквариум.
- 46) Диаметр колеса автомобиля 80 см. Колесо потребует замены через 200 000 оборотов. Определить, доедет ли колесо от города Уфы до города N-ска, если расстояние между ними x километров.

Варианты задач по теме «Цикл с параметром»

№ варианта	Ряд	Количество знаков после десятичной точки
1	$2 + \frac{3}{2} + \frac{4}{3} + \frac{5}{4} + \dots + \frac{11}{10}$	5
2	$1 + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \dots + \frac{1}{100}$	4
3	$3 + \frac{4}{2} + \frac{5}{3} + \dots + \frac{11}{9}$	5
4	$\frac{3}{2} + \frac{4}{3} + \frac{5}{4} + \dots + \frac{10}{9}$	3
5	$\frac{e^{-1}}{3} + \frac{e^{-2}}{4} + \frac{e^{-3}}{5} + \dots + \frac{e^{-9}}{11}$	4
6	$\frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \dots + \frac{1}{81}$	5
7	$\frac{2}{3} + \frac{3}{4} + \frac{4}{5} + \dots + \frac{10}{11}$	3
8	$\frac{e^2}{3} + \frac{e^3}{4} + \frac{e^4}{5} + \dots + \frac{e^9}{10}$	5
9	$2 + \frac{3}{2^3} + \frac{4}{3^3} + \dots + \frac{7}{6^3}$	4
10	$\frac{e^{-3}}{2} + \frac{e^{-4}}{3} + \frac{e^{-5}}{4} + \dots + \frac{e^{-9}}{8}$	3
11	$\frac{1^2}{2} + \frac{2^2}{3} + \dots + \frac{7^2}{8}$	2
12	$\frac{3}{4} + \frac{4}{5} + \frac{5}{6} + \dots + \frac{9}{10}$	5
13	$\frac{3}{4} + \frac{4}{9} + \frac{5}{16} + \dots + \frac{9}{64}$	4
14	$\frac{7}{4} + \frac{8}{5} + \frac{9}{6} + \dots + \frac{17}{14}$	3
15	$\frac{1}{2^2} + \frac{2}{3^2} + \dots + \frac{7}{8^2}$	5
16	$\frac{\ln(1)}{4} + \frac{\ln(2)}{5} + \dots + \frac{\ln(7)}{10}$	5
17	$\frac{\sin 1}{2} + \frac{\sin 2}{3} + \dots + \frac{\sin 8}{9}$	3
18	$\frac{1^3}{2} + \frac{2^3}{3} + \dots + \frac{7^3}{8}$	4
19	$\frac{2}{e^3} + \frac{3}{e^4} + \frac{4}{e^5} + \dots + \frac{9}{e^{10}}$	2
20	$\frac{\cos 2}{3} + \frac{\cos 3}{4} + \dots + \frac{\cos 7}{8}$	4

Варианты задач по теме «Итерационные циклы»

№ варианта	Вид функции $y=f(x)$	$x_{нач}$	$x_{кон}$	s
1	$\frac{e^{-x} + \operatorname{tg} x}{e^x - 1}$	0,2	0,5	0,1
2	$\ln(3x) + \sqrt{5 \sin(\pi x) }$	0,3	3,7	0,4
3	$\frac{\sqrt{2 + \sin x}}{x + \cos x}$	0,2	1	0,2
4	$\frac{\sqrt{\cos x + 2}}{\ln x + x^2}$	0,1	0,6	0,1
5	$\frac{\operatorname{tg} x - \sin x}{3 + x^2}$	1	3	0,5
6	$\frac{\sqrt{\sin x + 4}}{\ln x + x^3}$	0,2	0,8	0,2
7	$\frac{\sqrt{4 + \cos x}}{x^3 + \sin x}$	0,3	0,6	0,1
8	$\frac{\sqrt{ \ln x + x}}{\sqrt{\cos x + 2}}$	0,2	1	0,2
9	$0,05 \operatorname{arctg} x + \frac{2 \operatorname{tg} x}{x^2 + 5}$	-2	0	0,5
10	$\frac{0,2x}{x^2 - \sqrt{1 + 2x}}$	0	0,8	0,2
11	$\frac{45x^3}{ 3 - 10 \sin^2 3x }$	0	2	0,5
12	$30 + 2\sqrt{0,18x - \frac{3x}{1-x}}$	2	4	0,5
13	$2,2x^3 - \frac{3\sqrt{5 - 2 \sin^2 4x}}{ 0,3 \operatorname{tg} x - 2 }$	-0,5	1,5	0,5
14	$-0,8x - 5 \operatorname{tg} 3x^3 + 3\sqrt{2 + 2x}$	0,2	1,4	0,4
15	$\frac{-0,6x}{2 + 5x} - 3\sqrt{2 - \operatorname{arctg} x}$	2	2,4	0,1
16	$2,3 - 0,5\sqrt{2 + \operatorname{tg}^2 2x}$	-5	-3	0,5
17	$25,6x^2 - \frac{10\sqrt{1 + 2 \sin x}}{8x - 3}$	1	3	0,5
18	$4e^{2x} + \frac{6,3}{1 - \sin x - \cos 2x}$	0,2	1	0,2
19	$\frac{\operatorname{tg} x - \sin^2 x}{e^2 + x^2}$	-1	-0,2	0,2
20	$0,2 \operatorname{tg} x + \sqrt{ 2 \sin x }$	-1,5	0,5	0,5

Задачи для самостоятельного решения по теме «Операторы циклов»

- 1) Найти сумму чисел от 100 до 200, кратных 17. Подсчитать количество таких чисел.
- 2) Дано действительное число A , большее 1. Найти среди чисел $1, 1+1/2, 1+1/2+1/3, \dots$ первое, большее A . Вывести его порядковый номер.
- 3) С клавиатуры вводятся действительные числа и последовательно суммируются. Когда их сумма превысит 150, найти количество введенных чисел и их сумму.
- 4) С клавиатуры вводится десять чисел. Если квадрат числа меньше 100, то напечатать число и его квадрат. Подсчитать и вывести количество таких чисел.
- 5) С клавиатуры вводится десять двузначных нечетных чисел. Если число кратно 13, то найти произведение чисел, из которого оно состоит. Результат вывести на экран.
- 6) С клавиатуры вводятся десять двузначных чисел. Напечатать те из них, которые делятся нацело на 4, но не делятся на 6. Подсчитать их количество.
- 7) В бригаде, работающей на уборке сена, имеется N сенокосилок. Первая сенокосилка работала m часов, а каждая следующая на 10 минут больше, чем предыдущая. Сколько часов проработала вся бригада?
- 8) Начав тренировки, спортсмен в первый день пробежал 10 км. Каждый день он увеличивал дневную норму на 10% нормы предыдущего дня. Какой суммарный путь пробежит спортсмен за 7 дней?
- 9) Начав тренировки, спортсмен в первый день пробежал 10 км. Каждый день он увеличивал дневную норму на 10% нормы предыдущего дня. В какой день впервые спортсмен пробежит более 20 км?
- 10) Начав тренировки, спортсмен в первый день пробежал 10 км. Каждый день он увеличивал дневную норму на 10% нормы предыдущего дня. В какой день впервые суммарный путь превысит 60 км?
- 11) Дана последовательность из 10 действительных чисел. Найти количество элементов последовательности, превышающих по модулю число 7.
- 12) Дано n вещественных чисел. Найти сумму тех из них, которые имеют четный остаток от деления на некоторое число t .
- 13) Дано n вещественных чисел. Найти сумму тех из них, которые имеют остаток от деления на некоторое число m , равный 0. Посчитать количество таких элементов.
- 14) Дано n вещественных чисел. Заменить те из них, у которых остаток от деления на m равен 2 числом m . Подсчитать количество таких замен.
- 15) Дано n вещественных чисел. Заменить все ее члены, большие данного числа x , этим числом. Подсчитать количество таких замен.
- 16) Подрабатывая вечерами, студент решил накопить сумму в S рублей для покупки компьютера. В первый месяц он отложил P рублей. Затем его вклад каждый раз был на 3 % больше предыдущего вклада. Через сколько месяцев студент сможет купить компьютер?

Окончание приложения 3

- 17) В водоеме 100 т рыбы. Каждый год рыболовецкая бригада вылавливает 15 т. Воспроизводство рыбы 5 % в год. Для сохранения воспроизводства необходимо прекращать лов, когда в водоеме ее остается менее 5 т. Через сколько лет лов рыбы должен быть прекращен?
- 18) С клавиатуры вводится десять чисел – положительных и отрицательных. Заменить все отрицательные числа их модулями, подсчитать количество таких замен.
- 19) Дана последовательность вещественных чисел $a_1, a_2, a_3, \dots, a_n$. Заменить все ее члены, меньшие данного числа b , квадратом числа b . Подсчитать и вывести количество таких замен.
- 20) Дано действительное число x . Вычислить сумму ряда $\frac{\sin x}{2} + \frac{\sin 2x}{3} + \frac{\sin 3x}{4} + \dots + \frac{\sin 9x}{10}$.
- 21) Дано натуральное число n . Вычислить $\frac{1}{\sin 1} + \frac{1}{\sin 1 + \sin 2} + \dots + \frac{1}{\sin 1 + \dots + \sin n}$.
- 22) Дано натуральное число n . Вычислить $\frac{\cos 1}{\sin 1} + \frac{\cos 1 + \cos 2}{\sin 1 + \sin 2} + \dots + \frac{\cos 1 + \dots + \cos n}{\sin 1 + \dots + \sin n}$.
- 23) Распечатать все трехзначные числа, сумма цифр которых равна данному натуральному числу M .
- 24) Дано n – целое число. Вычислить произведение: $P = 3 * 6 * \dots * (3n)$. Вычислить значения функции $F(x) = \sin^2 x$ на отрезке $[0, 1]$ с шагом 0,2. Результат вывести в два столбца: первый столбец – значения аргумента, второй – соответствующие значения функции.
- 25) Вычислить значения функции $F(x) = \cos^2 x$ на отрезке $[-1, 1]$ с шагом 0,4. Результат вывести в два столбца: первый столбец – значения аргумента, второй – соответствующие значения функции.
- 26) Составить программу, выводящую таблицу синусов угла, изменяющегося от 1 до 360 градусов с шагом 1 градус. Значения выводить парами $(x - \sin(x))$ в четыре колонки, причем угол выводится в целочисленном формате, а его синус – с точностью 4 знака после десятичной точки.
- 27) Вычислить сумму ряда $s = 2 - 4 + 6 - 8 + \dots + (-1)^n * (2n)$.
- 28) Вычислить сумму ряда: $s = 1 - \frac{1}{2} + 3 - \frac{1}{4} + \dots + (2n - 1) - \frac{1}{2n}$
- 29) Дано натуральное число n . Вычислить $\sum_{k=1}^n k^k$.
- 30) Составить программу, определяющую сумму ряда $s = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots + \frac{(-1)^{n-1}}{n}$. Вывести номера членов ряда со значением из диапазона 0.02...0.03.

Варианты задач по теме «Вложенные циклы. Вычисление функции»

- 1) Вычислить функцию $f(x, a) = (x/(a + 2))^2$ при $x=0, 0.2, 0.4, \dots, 1.2$ и $a=1, 2, 3$ с точностью 5 знаков после десятичной точки.
- 2) Вычислить функцию $f(x, a) = \ln(\sin^2(x) + a)$ при $x=0.3, 0.6, 0.9, \dots, 2.4$ и $a=3, 4, 5$ с точностью 6 знаков после десятичной точки.
- 3) Вычислить функцию $f(x, a) = (x/2a)^2 - \sin x$ при $x=2.1, 2.2, 2.3, \dots, 3$ и $a=1, 2, 3$ с точностью 7 знаков после десятичной точки.
- 4) Вычислить функцию $f(x, a) = \cos(\ln^2(x) + a)$ при $x=1.3, 1.6, 1.9, \dots, 2.8$ и $a=1, 3, 5, 7$ с точностью 8 знаков после десятичной точки.
- 5) Вычислить функцию $f(x, b) = b \cos(1 - x/b)$ при $x=1, 2, 3, 4, 5$ и $b=0.1, 0.2, 0.3, 0.4, 0.5$ с точностью 7 знаков после десятичной точки.
- 6) Вычислить функцию $f(x, b) = b \sin(x - 4/b)$ при $x=0.1, 0.2, 0.3, 0.4, 0.5$ и $b=2, 3, 4, 5$ с точностью 8 знаков после десятичной точки.
- 7) Вычислить функцию $f(x, a) = \sqrt{a+1} \sin\left(x^2/(a^3+1)\right)$ при $x=0.1, 0.2, 0.3, 0.4, 0.5$ и $a=1, 2, 3, 4, 5$ с точностью 4 знаков после десятичной точки.
- 8) Вычислить функцию $f(x, a) = 2a|\sin^2 x - 1|$ при $x=0.2, 0.4, 0.6, \dots, 1$ и $a=4, 5, 6, 7$ с точностью 6 знаков после десятичной точки.
- 9) Вычислить функцию $f(x, a) = a\sqrt{x} \cos^2 x$ при $x=1, 3, 5, 7$ и $a=3, 4, 5, 6, 7$ с точностью 7 знаков после десятичной точки.
- 10) Вычислить функцию $f(x, a) = \sin(x+a) + \operatorname{tg} x$ при $x=0, 0.1, 0.2, \dots, 0.5$ и $a=1, 2, 3, 4$ с точностью 6 знаков после десятичной точки.
- 11) Вычислить функцию $f(x, a) = 2\operatorname{tg} x/2 + a - 1$ при $x=1.3, 1.6, 1.9, \dots, 3.1$ и $a=1, 2, 3, 4$ с точностью 7 знаков после десятичной точки.
- 12) Вычислить функцию $f(x, a) = \operatorname{tg} x/2 + 2\cos(a + x)$ при $x=0.4, 0.6, \dots, 1.2$ и $a=2, 3, 4, 5$ с точностью 6 знаков после десятичной точки.
- 13) Вычислить функцию $f(x, a) = \ln(3x) + \sqrt{a^3/2x}$ при $x=0.4, 0.6, \dots, 1.2$ и $a=2, 3, \dots, 6$ с точностью 4 знаков после десятичной точки.
- 14) Вычислить функцию $f(x, a) = \operatorname{ctg} x/3 + a/2\sin x$ при $x=0.3, 0.5, 0.7, \dots, 1.1$ и $a=1, 2, 3, 4, 5$ с точностью 5 знаков после десятичной точки.
- 15) Вычислить функцию $f(x, a) = 1/2\sin(a + x) + 1/a$ при $x=0.1, 0.2, \dots, 0.5$ и $a=1, 2, 3, 4, 5$ с точностью 7 знаков после десятичной точки.
- 16) Вычислить функцию $f(x, a) = ax^2\sin x + 1$ при $x=0.2, 0.3, \dots, 0.6$ и $a=2, 3, 4, 5$ с точностью 4 знаков после десятичной точки.
- 17) Вычислить функцию $f(x, a) = \operatorname{tg}(2ax) - 3$ при $x=2.4, 2.5, \dots, 3$ и $a=2, 3, 4, 5, 6, 7$ с точностью 6 знаков после десятичной точки.
- 18) Вычислить функцию $f(x, b) = -2b\sin(1 - x/b)$ при $x=1, 3, 5, 7$ и $b=1, 2, 3, 4$ с точностью 5 знаков после десятичной точки.
- 19) Вычислить функцию $f(x, a) = ax/\cos x + 4$ при $x=0.1, 0.2, \dots, 0.6$ и $a=1, 2, 3, 4, 5$ с точностью 6 знаков после десятичной точки.
- 20) Вычислить функцию $f(x, a) = a\ln x + \operatorname{arctg} 2x/(a + 1)$ при $x=0.3, 0.6, 0.9, \dots, 2.4$ и $a=3, 4, 5, 6$ с точностью 8 знаков после десятичной точки.

Варианты задач по теме «Вычисление суммы ряда с использованием вложенных циклов»

- 1) Вычислить сумму $1 + \frac{1}{1!} + \frac{1}{2!} + \dots + \frac{1}{n!}$, ($1 < n \leq 10$).
- 2) Даны натуральные числа n , m . Вычислить $\sum_{i=1}^n \sum_{k=1}^m \frac{k+1}{k-i}$.
- 3) Вычислить $\sum_{i=1}^{100} \sum_{j=1}^{50} \frac{1}{i+j^2}$.
- 4) Вычислить $\sum_{i=1}^{100} \sum_{j=1}^{40} \sin(i^3 + j^4)$.
- 5) Вычислить $\sum_{i=1}^{100} \sum_{j=1}^{100} \frac{j-i+1}{i+j}$.
- 6) Вычислить $\sum_{i=1}^{100} \sum_{j=i}^i \frac{1}{2j+i}$.
- 7) Даны натуральное число n , действительное число x . Вычислить $\sum_{k=1}^n \sum_{m=k}^n \frac{x+k}{m}$.
- 8) Вычислить значение функции $z = \sum_{i=1}^{15} a_i \prod_{k=1}^{10} \frac{a_i + b_k}{2}$, где a_i – элементы, вводимые с клавиатуры. Значения b изменяются от 0 с шагом 0,1.
- 9) Вычислить $p = \prod_{i,j=1}^{20} \frac{1}{i+j^2}$.
- 10) Вычислить значения функции $z_i = \sum_{i=1}^{15} \frac{i+x_i}{i!}$, где x_i – элементы, вводимые с клавиатуры.
- 11) Дано натуральное число n . Вычислить $\sum_{k=1}^n \frac{1}{\left(\binom{k}{2}\right)!}$.
- 12) Даны натуральное число n , действительное число x . Вычислить $\sum_{i=1}^n \frac{(2i)+|x|}{(i^2)!}$.
- 13) Даны натуральное число n , действительное число x . Вычислить $\frac{1}{n} \sum_{k=1}^n \frac{x}{k!+1}$.
- 14) Дано натуральное число n . Вычислить значение функции $z = \sum_{i=1}^{20} \sum_{k=1}^n \frac{x_i}{k!}$, где x_i – элементы, вводимые с клавиатуры.
- 15) Для натурального числа n и действительного числа x вычислить сумму степеней $\left(\frac{x}{1}\right)^n + \left(\frac{x}{2}\right)^n + \dots + \left(\frac{x}{n}\right)^n$

Окончание приложения К

16) Даны натуральное число n , действительное число x . Вычислить

$$\sum_{k=1}^n \sum_{x=1}^n k^k x^{2k}.$$

17) Вычислить $s = \sum_{k=1}^{10} a_k \frac{\sum_{n=1}^k \sin kn}{k}.$

18) Вычислить $\sum_{i=1}^n \frac{10 + \sum_{j=1}^m (j+i)^2}{20 + i^2 \sum_{k=1}^m k^2}, x=1.5, m=25, n=7.$

19) Вычислить $\sum_{i=2}^n \frac{x^3 + (\sum_{k=1}^m k^3)^2}{x + \sum_{j=1}^m \sqrt{i+j}}, x=3, m=20, n=12.$

20) Вычислить $\sum_{i=1}^n \frac{5x + \sum_{k=1}^m (k+x)^2}{x + \sum_{j=1}^m \sqrt{2i+3j}}, x=2.5, m=12, n=9.$

21) Вычислить $\sum_{i=1}^n \frac{1+x \sum_{k=1}^m (\frac{1}{k} + k)}{3i + \sum_{j=1}^m (x + \frac{j}{i})^2}, x=4.5, m=15, n=10.$

22) Вычислить сумму элементов ряда $1 - \frac{2x}{1!} + \frac{x^3}{3!} - \frac{2x^5}{5!} + \frac{x^7}{7!} - \dots$ с точностью 10^{-5} .

23) Дано вещественное число $x \neq 0$. С заданной точностью ε вычислить сумму членов бесконечного ряда.

24) Вычислить сумму членов бесконечного ряда:

$$1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots + (-1)^n \frac{x^{2n}}{(2n)!} + \dots = 1 + \sum_{i=1}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!}$$

с точностью до члена ряда, меньшего ε .

25) Вычислить сумму бесконечного ряда:

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

с точностью 10^{-8} .

Задачи для самостоятельного решения по теме «Вложенные циклы»

- 1) Составить программу вывода на экран таблицы умножения.
- 2) Составить программу, которая выводит таблицу сложения n натуральных чисел ($n=9$) в восьмеричной системе счисления.
- 3) Среди чисел $1 < n < 100$ найти все пары чисел, для которых их сумма равна их произведению.
- 4) Среди чисел $1 < n < 100$ найти все пары чисел, произведение которых в 10 раз больше их суммы. Сколько таких пар?
- 5) Найти все двузначные числа, сумма квадратов цифр которых кратна 15.
- 6) Найти все четырехзначные числа, у которых все цифры различны.
- 7) Натуральное число из n цифр является числом Армстронга, если сумма его цифр, возведенных в n -ую степень, равна самому числу. Получить эти числа, состоящие из трех цифр.
- 8) Найти в промежутке от 1 до 1000 числа, у которых пять делителей.
- 9) Среди чисел $1 < n < 100$ найти все пары чисел, для которых их сумма равна их произведению. Подсчитать их количество.
- 10) Даны целые числа K , N и K наборов целых чисел по N элементов в каждом наборе. Вывести общую сумму всех элементов, входящих в данные наборы.
- 11) Даны целые числа K , N , а также K наборов целых чисел по N элементов в каждом наборе. Для каждого набора вывести сумму его элементов.
- 12) Даны целые числа K , N , а также K наборов целых чисел по N элементов в каждом наборе. Найти количество наборов, содержащих число 2. Если таких наборов нет, то вывести 0.
- 13) Дано натуральное число M . Можно ли представить его в виде суммы двух квадратов натуральных чисел. Если можно, то указать все пары x , y таких натуральных чисел, что $M = x^2 + y^2$, $x \geq y$.
- 14) Распечатать все трехзначные числа, в которых вторая цифра больше первой, а третья больше второй. Условные операторы не использовать.
- 15) Автобусный билет считается счастливым, если в его шестизначном номере сумма первых трех цифр равна сумме трех последних. Подсчитать число счастливых билетов с различными номерами (от 000001 до 999999).
- 16) Разложить натуральное число на простые множители.
- 17) Распечатать все трехзначные числа, в которых есть две одинаковые цифры.
- 18) Для заданного натурального числа n найти все тройки натуральных чисел a , b , c таких, что $a+b+c=n$.
- 19) Купили 100 голов скота на 100 рублей. Бык стоит 10 рублей, корова - 5 рублей, теленок - 0.5 рублей. Сколько купили каждой голов скота?
- 20) Имеется товар в ящиках по 16, 17, 21 кг. Как получить 185 кг этого товара, не вскрывая ящики?
- 21) Дано натуральное число $n < 99$. Получить все способы выплаты суммы n с помощью монет достоинством 1, 2, 5, 10 руб.
- 22) Подсчитать число точек с целочисленными координатами, находящимися внутри круга с центром в начале координат и заданным радиусом $r \leq 1000000$.

Варианты заданий по теме «Одномерные массивы»

- 1) Найдите среднее арифметическое элементов заданного массива из 8 целых чисел и определите, сколько элементов данного массива больше среднего.
- 2) Задан массив из 7 целых чисел. Замените все положительные элементы данного массива его максимальным элементом.
- 3) Задан массив из 8 действительных чисел. Замените все отрицательные элементы данного массива его минимальным элементом.
- 4) Найдите сумму максимального и минимального элементов заданного массива из 7 действительных чисел.
- 5) В заданном массиве из 8 действительных чисел вычислите сумму и среднее арифметическое значение положительных элементов.
- 6) Задан массив из 10 целых чисел. Найдите среднее арифметическое всех четных элементов массива.
- 7) Подсчитайте количество нулевых элементов в массиве из 7 целых чисел и распечатайте исходный массив в обратном порядке.
- 8) Дан целочисленный массив из 8 целых чисел. Преобразуйте его, прибавив к четным числам значение первого элемента. Первый элемент массива оставьте без изменений.
- 9) Дан целочисленный массив из 7 целых чисел. Преобразуйте его, прибавив к нечетным числам значение последнего элемента. Последний элемент массива оставьте без изменений.
- 10) Найдите наибольший по модулю элемент заданного массива из 9 действительных чисел.
- 11) В заданном массиве из 5 действительных чисел определите количество чисел, не попадающих в интервал от -1 до 1.
- 12) Замените все четные элементы заданного целочисленного массива из 7 элементов числом 0, а все нечетные элементы – числом 1.
- 13) Замените все положительные элементы заданного целочисленного массива из 8 элементов числом 1, а все отрицательные элементы – числом 0.
- 14) Задан массив из 6 действительных чисел. Определите, каких элементов в массиве больше - положительных или отрицательных.
- 15) Задан массив из 8 целых чисел. Определите, каких элементов в массиве больше – четных или нечетных.
- 16) Вычислите среднее арифметическое всех отрицательных элементов заданного массива из 7 действительных чисел.
- 17) Поменяйте местами максимальный и минимальный элемент заданного массива из 7 целых чисел.
- 18) Найдите минимальный среди положительных элементов заданного массива из 6 целых чисел.
- 19) Найдите максимальный среди отрицательных элементов заданного массива из 8 действительных чисел.
- 20) Задан массив из 7 целых чисел. Определите, является ли данный массив упорядоченным по возрастанию.

Задачи для самостоятельного решения по теме «Одномерные массивы»

- 1) Дан целочисленный массив из 7 чисел. Вывести его элементы в обратном порядке.
- 2) Дан целочисленный массив A из 10 чисел. Вывести номер тех его элементов, которые удовлетворяют двойному неравенству: $A[1] < A[i] < A[10]$.
- 3) Дан целочисленный массив из 10 чисел. Переставить в обратном порядке элементы массива, расположенные между его минимальным и максимальным элементами.
- 4) Дан массив из 8 действительных чисел. Вывести вначале его элементы с нечетными индексами, а затем — с четными.
- 5) Дан целочисленный массив размера N . Вывести вначале все его четные элементы, а затем — нечетные.
- 6) Дан действительный массив размера N . Осуществить циклический сдвиг элементов массива: а) влево на одну позицию; б) вправо на одну позицию.
- 7) Дан целочисленный массив размера N и натуральное число k ($0 < k < 5$, $k < N$). Осуществить циклический сдвиг элементов массива: а) влево на k позиций; б) вправо на k позиций.
- 8) Проверить, образуют ли элементы целочисленного массива размера N арифметическую прогрессию. Если да, то вывести разность прогрессии, если нет — вывести 0.
- 9) Дан массив ненулевых целых чисел размера N . Проверить, чередуются ли в нем четные и нечетные числа.
- 10) Дан массив ненулевых целых чисел размера N . Проверить, чередуются ли в нем положительные и отрицательные числа.
- 11) Дан целочисленный массив размера N . Определить количество участков, на которых его элементы монотонно: а) возрастают; б) убывают.
- 12) Дан действительный массив размера N . Определить количество его промежутков монотонности (то есть участков, на которых его элементы возрастают или убывают).
- 13) Дано действительное число R и массив из N действительных чисел. Найти элемент массива, который: а) наиболее; б) наименее близок к данному числу R .
- 14) Дано вещественное число R и массив из N действительных чисел. Найти два элемента массива, сумма которых: а) наиболее; б) наименее близка к данному числу R .
- 15) Дан целочисленный массив размера N . Определить максимальное количество его одинаковых элементов.
- 16) Дан целочисленный массив размера N . Удалить из массива все элементы, встречающиеся менее двух раз.
- 17) Дан целочисленный массив размера N . Если он содержит все числа от 1 до N , то вывести 0, в противном случае вывести номер первого недопустимого элемента.

- 18) Дан действительный массив размера N . Преобразовать его, вставив после каждого отрицательного элемента нулевой элемент.
- 19) Даны два целочисленных массива A и B размера 5, элементы которых упорядочены по возрастанию. Объединить эти массивы так, чтобы результирующий массив остался упорядоченным.
- 20) Упорядочить действительный массив размера N : а) по возрастанию; б) по убыванию.
- 21) Дан целочисленный массив размера N . Вывести индексы массива в том порядке, в котором соответствующие им элементы образуют: а) возрастающую; б) убывающую последовательность.
- 22) Дано множество из N точек с координатами вида $(x_1, y_1), \dots, (x_N, y_N)$. Среди всех точек этого множества, лежащих в первой координатной четверти, найти точку, наиболее близкую к началу координат. Если таких точек нет, то вывести точку с нулевыми координатами.
- 23) Дано множество из N точек с действительными координатами вида $(x_1, y_1), \dots, (x_N, y_N)$. Найти такую точку из данного множества, сумма расстояний от которой до остальных его точек минимальна.
- 24) Дано множество из N точек с целочисленными координатами. Порядок на координатной плоскости определим следующим образом: $(x_1, y_1) < (x_2, y_2)$, если либо $x_1 < x_2$, либо $x_1 = x_2$ и $y_1 < y_2$. Расположить точки данного множества: а) по возрастанию; б) по убыванию в соответствии с указанным порядком.
- 25) Даны два массива A и B , состоящие из 8 целых чисел. Получить массив C , состоящий из тех элементов массива A , которых нет в массиве B .
- 26) Даны два массива A и B , состоящие из 6 целых чисел. Вычислить сумму и произведение элементов, содержащихся как в A , так и в B .
- 27) Даны два массива A и B , состоящие из 7 действительных элементов. Вычислить сумму квадратов элементов, содержащихся только в A , и произведение элементов, содержащихся только в B (элементы, содержащиеся как в A , так и в B , не учитываются).
- 28) Расположить элементы заданного массива из 8 целых чисел в обратном порядке. Первый элемент меняется с последним, второй - с предпоследним и т.д. до середины. Если массив содержит нечетное количество элементов, то средний остается без изменения.
- 29) В данном массиве из 8 целых чисел поменять местами элементы, стоящие на нечетных местах, с элементами, стоящими на четных местах (первый элемент меняется со вторым, третий с четвертым и т.д.)
- 30) Даны две упорядоченные по возрастанию последовательности чисел X и Y размером m и n элементов соответственно. Получить из них упорядоченную по убыванию последовательность Z размером $m+n$ элементов.

Варианты заданий по теме «Нахождение итоговых значений в матрицах»

- 1) Дана действительная матрица размера 3×4 . Найти сумму элементов последнего столбца данной матрицы.
- 2) Дана целочисленная матрица размера 4×3 . Найти произведение элементов первого столбца данной матрицы.
- 3) Дана действительная матрица размера 3×4 . Найти минимальное значение в первой строке данной матрицы.
- 4) Задана целочисленная квадратная матрица размера 3×3 . Найти наименьший элемент главной диагонали данной матрицы.
- 5) Дана целочисленная матрица размера 4×4 . Найти минимальное значение во втором столбце данной матрицы.
- 6) Дана действительная матрица размера 4×3 . Найти максимальное значение в третьей строке данной матрицы.
- 7) Дана действительная матрица размера 3×5 . Найти сумму элементов второй строки данной матрицы.
- 8) Дана целочисленная матрица размера 3×5 . Найти максимальное значение в четвертом столбце данной матрицы.
- 9) Дана действительная матрица размера 3×4 . Определить, сколько раз встречается число 5 во второй строке данной матрицы.
- 10) Дана целочисленная матрица размера 5×3 . Найти произведение элементов последней строки данной матрицы.
- 11) Дана целочисленная матрица размера 4×4 . Определить, сколько раз встречается число 7 в первом столбце данной матрицы.
- 12) Задана действительная квадратная матрица размера 4×4 . Найти сумму элементов главной диагонали данной матрицы.
- 13) Задана целочисленная квадратная матрица размера 3×3 . Найти произведение элементов главной диагонали данной матрицы.
- 14) Задана действительная квадратная матрица размера 4×4 . Определить, сколько раз встречается число 0 среди элементов главной диагонали данной матрицы.
- 15) Задана действительная квадратная матрица размера 4×4 . Найти наибольший элемент главной диагонали данной матрицы.
- 16) Дана целочисленная матрица размера 3×4 . Найти количество нулевых элементов, расположенных в последнем столбце данной матрицы.
- 17) Дана квадратная действительная матрица размерности 4×4 . Найти количество нулевых элементов, стоящих на главной диагонали.
- 18) Дана целочисленная матрица размерности 3×4 . Найти количество нулевых элементов, расположенных во второй строке данной матрицы.
- 19) Дана действительная матрица размерности 3×5 . Найти количество положительных элементов, расположенных в третьей строке данной матрицы.
- 20) Дана целочисленная матрица размерности 4×4 . Найти количество отрицательных элементов, расположенных во втором столбце данной матрицы.

Варианты заданий по теме «Замена и перестановка элементов матриц»

- 1) Дана квадратная целочисленная матрица размера 4×4 . Заменить нулями элементы матрицы, лежащие на главной диагонали.
- 2) Дана целочисленная матрица размера 5×6 . Поменять местами 1 и 2 столбцы.
- 3) Дана целочисленная матрица размера 4×6 . Переставить местами 1 и 2 строку матрицы.
- 4) Дана целочисленная матрица размера 5×7 . Заменить в данной матрице все отрицательные элементы числом 0, положительные элементы – числом 1.
- 5) Дана целочисленная матрица размера 5×6 . Поменять местами первый и последний из столбцов.
- 6) Дана целочисленная матрица размера 5×5 . Заменить в данной матрице все отрицательные элементы первой строки числом 0.
- 7) Дана целочисленная матрица размера 5×7 . Переставить местами 1 и 5 строку.
- 8) Дана целочисленная матрица размера 5×6 . Заменить в данной матрице все элементы второй строки числом 10.
- 9) Дана целочисленная матрица размера 5×7 . Переставить местами 1 и 4 столбец.
- 10) Дана целочисленная матрица размера 6×6 . Заменить в данной матрице все положительные элементы первого столбца числом 1.
- 11) Дана целочисленная матрица размера 4×5 . Переставить местами 1 и 3 строку.
- 12) Дана целочисленная матрица размера 5×6 . Заменить в данной матрице все положительные элементы числом 5.
- 13) В целочисленной матрице чисел размером 4×6 поменять местами 2 и 4 столбцы.
- 14) Дана целочисленная матрица размера 5×5 . Переставить местами 4 и 5 строку.
- 15) Дана целочисленная матрица размера 5×7 . Заменить в данной матрице все элементы второго столбца числом 100.
- 16) Дана целочисленная матрица размера 4×6 . Переставить местами 5 и 6 столбец.
- 17) Дана квадратная целочисленная матрица размера 5×5 . Заменить нулями элементы матрицы, лежащие ниже главной диагонали.
- 18) Дана квадратная целочисленная матрица размера 6×6 . Заменить нулями элементы матрицы, лежащие выше главной диагонали.
- 19) Дана целочисленная матрица размера 5×6 . Заменить в данной матрице все элементы первой строки числом 7.
- 20) Дана целочисленная матрица размера 5×7 . Переставить местами 2 и 3 строку.

Задачи для самостоятельного решения по теме «Двумерные массивы»

- 1) Получить целочисленную матрицу A размерности 10×12 , для которой $A_{ij} = i + 2*j$, для всех $i = 1..10$; $j = 1..12$.
- 2) Дана вещественная матрица размерности 5×8 . Сформировать вектор b , в котором элементы вычисляются как произведение элементов соответствующих строк.
- 3) Дана действительная матрица размера 5×6 . В каждой строке найти количество элементов, больших среднего арифметического всех элементов этой строки.
- 4) В матрице целых чисел размера 10×15 определить количество и сумму элементов, значения которых находятся вне интервала $[k_1, k_2]$.
- 5) Дана целочисленная матрица размера 7×8 . Преобразовать матрицу, поменяв местами минимальный и максимальный элемент в каждой строке.
- 6) Дана целочисленная матрица размера 5×10 . Найти минимальное значение среди сумм элементов всех ее строк.
- 7) Дана действительная матрица размера 5×10 . Вывести номер ее первой строки, содержащей только положительные элементы. Если таких строк нет, то вывести 0.
- 8) Дана целочисленная матрица размера $M \times N$. Найти количество ее строк, все элементы которых различны.
- 9) Дана целочисленная матрица размера $M \times N$. Вывести номер столбца, содержащего максимальное количество одинаковых элементов.
- 10) Дана квадратная целочисленная матрица порядка M . Найти сумму элементов ее побочной диагонали.
- 11) Дана квадратная целочисленная матрица порядка M . Заменить нулями элементы, лежащие одновременно выше главной диагонали (включая эту диагональ) и выше побочной диагонали (также включая эту диагональ).
- 12) Дана квадратная целочисленная матрица порядка M . Повернуть ее на 90 градусов в положительном направлении.
- 13) Дана действительная матрица размера 5×10 . Вывести количество строк, элементы которых монотонно возрастают.
- 14) Дано число k ($k \leq 4$) и целочисленная матрица размера 4×10 . Удалить строку матрицы с номером k .
- 15) Дана целочисленная матрица размера 5×10 . Удалить столбец, содержащий минимальный элемент матрицы.
- 16) Дана целочисленная матрица размера 5×10 . Удалить все столбцы, содержащие только положительные элементы.
- 17) Дана целочисленная матрица размера 4×9 . Продублировать строку матрицы, содержащей ее максимальный элемент.
- 18) Дана целочисленная матрица размера 5×9 . После каждого столбца, содержащего только положительные элементы, добавить столбец, состоящий из единиц.

- 19) Дана целочисленная матрица размера $M \times N$. Найти элемент, являющийся максимальным в своей строке и минимальным в своем столбце. Если такой элемент отсутствует, то вывести 0.
- 20) Дана целочисленная матрица размера $M \times N$. Элемент называется локальным минимумом (максимумом), если он меньше (больше) всех окружающих его элементов. Заменить все локальные минимумы данной матрицы на 0.
- 21) Дана действительная матрица размера $M \times N$. Поменять местами ее строки так, чтобы их минимальные элементы образовывали возрастающую последовательность.
- 22) Для целочисленной матрицы размера $N \times N$ вычислить разность элементов главной и побочной диагоналей, расположенных на одной строке.
- 23) В целочисленной матрице размера $M \times N$ найти отношение суммы элементов, попадающих в интервал от k_1 до k_2 , к максимальному элементу матрицы.
- 24) Для действительной матрицы размера $N \times N$ вычислить суммы элементов главной и побочной диагоналей, расположенных в одном столбце.
- 25) В действительной матрице размера $M \times N$ обнулить элементы, не попадающие в интервал от k_1 до k_2 , и выдать транспонированную матрицу.
- 26) Для матрицы A целых чисел размера $M \times N$ вычислить сумму и разность максимального числа, расположенного выше побочной диагонали, и минимального числа, расположенного ниже главной диагонали.
- 27) В квадратной матрице действительных чисел размером $N \times N$ вычислить сумму положительных и произведение отрицательных чисел, расположенных на главной и побочной диагоналях.
- 28) Даны три последовательности действительных чисел: A – 10 элементов; B – 5 элементов; C – 10 элементов. Необходимо заполнить квадратную матрицу размером 5×5 элементов так, чтобы элементы из A размещались выше побочной диагонали, элементы из B – на побочной диагонали, а элементы из C – ниже побочной диагонали.
- 29) Даны две последовательности действительных чисел: A – 16 элементов; B – 9 элементов. Необходимо заполнить квадратную матрицу размером 5×5 элементов так, чтобы элементы из A размещались в крайних строках и столбцах матрицы по часовой стрелке, начиная с элемента с индексами $[1,1]$, а элементы из B – внутри матрицы.
- 30) Из матрицы целых чисел A размерности $N \times M$ получить матрицу B размерности $(N-1) \times (M-1)$ путем вычеркивания строки с минимальным элементом и столбца с максимальным элементом. Считать, что все элементы исходной матрицы A различны.

Варианты заданий по теме «Обработка строки символов»

- 1) Задано слово произвольной длины. Определить, сколько раз в нем встречается буква «и».
- 2) Определить, содержит ли заданное слово букву «д».
- 3) Задана строка произвольной длины. Вывести данную строку в обратном порядке.
- 4) Задано слово произвольной длины. Поменять в нем местами первую и последнюю букву.
- 5) Дана строка. Определить, заканчивается ли она точкой.
- 6) Задано произвольное слово. Определить, состоит ли оно из четного количества букв.
- 7) Задано слово произвольной длины. Проверить, начинается и заканчивается ли данное слово на одну и ту же букву.
- 8) Задана строка. Удалить из нее два последних символа.
- 9) Задано произвольное слово. Определить, состоит ли оно из 7 букв.
- 10) Дана строка. Подсчитать количество содержащихся в строке запятых.
- 11) Задано произвольное слово. Определить, начинается ли оно со слога «ка».
- 12) Задано слово произвольной длины. Поменять в нем местами первую и третью букву.
- 13) Задано слово произвольной длины. Поменять в нем местами последнюю и предпоследнюю букву.
- 14) Задано произвольное слово. Определить, заканчивается ли оно на слог «ва».
- 15) Задано слово произвольной длины. Определить, содержится ли в нем три буквы «а»
- 16) Задано произвольное слово. Определить, состоит ли оно из 5 либо 6 букв.
- 17) Задано слово произвольной длины. Удалить из него последний символ.
- 18) Заменить в заданной строке все точки восклицательными знаками.
- 19) Найти количество пробелов в заданной строке произвольной длины.
- 20) Задано произвольное слово. Определить, заканчивается ли оно на букву «а» либо «о».

Варианты заданий по теме «Обработка списка строк»

- 1) Задан список из 7 фамилий. Определить, имеется ли в списке фамилия, начинающаяся с буквы «А» и состоящая из 6 букв. Если да, то вывести ее на экран
- 2) Найти в списке из 6 фамилий все фамилии, имеющие окончание «ов».
- 3) Найти в списке из 8 фамилий все фамилии, начинающиеся со слога «Ра».
- 4) Задан список из 7 слов. Определить, сколько слов в списке состоят из 6,7 или 8 букв.
- 5) Задан список из 8 фамилий. Определить, сколько фамилий в списке начинаются с букв «А», «Б» или «В»
- 6) Найти в списке из 7 фамилий все фамилии, состоящие из 5 или 6 букв.
- 7) Задан список из 6 слов. Определить, сколько букв в самом длинном слове списка.
- 8) Найти в списке из 5 фамилий все фамилии, начинающиеся со слога «Ни».
- 9) Найти в списке из 7 фамилий все фамилии, состоящие из 4 и менее букв.
- 10) Задан список из 8 фамилий. Определить, сколько фамилий в списке имеют окончание «ова».
- 11) Задан список из 6 слов. Определить, сколько букв в самом длинном слове списка.
- 12) Задан список из 8 слов. Определить, сколько букв в самом коротком слове списка.
- 13) Найти самую длинную фамилию в списке из 7 фамилий. Если таких фамилий несколько, то распечатать их в одну строку.
- 14) Найти в списке из 6 фамилий все фамилии, начинающиеся с букв «В» или «Г».
- 15) Найти в списке из 7 фамилий все фамилии, имеющие окончание «ев».
- 16) Найти в списке из 5 фамилий все фамилии, начинающиеся со слога «Ма».
- 17) Задан список из 7 фамилий. Определить, сколько фамилий в списке состоят менее чем из 6 букв.
- 18) Задан список из 6 слов. Определить, сколько слов в списке состоят из 8 или 9 букв.
- 19) Задан список из 8 фамилий. Найти в списке все фамилии, состоящие из 6 или 7 букв.
- 20) Упорядочить заданный список из 7 слов в алфавитном порядке.

Задачи для самостоятельного решения по теме «Строковые переменные»

- 1) Вывести строку длины N (N — четное), которая состоит из чередующихся букв А и Б, начиная с А.
- 2) Дана строка. Преобразовать все строчные латинские буквы в прописные.
- 3) Дана строка, состоящая из русских слов, разделенных пробелами (одним или несколькими). Определить количество слов в строке.
- 4) Дана строка, состоящая из русских слов, разделенных пробелами (одним или несколькими). Определить количество слов, которые начинаются и заканчиваются одной и той же буквой.
- 5) Дана строка. Подсчитать количество содержащихся в ней цифр.
- 6) Дана строка, состоящая из русских слов, разделенных пробелами (одним или несколькими). Определить количество слов, которые содержат ровно три буквы "А".
- 7) Дана строка, состоящая из русских слов, разделенных пробелами (одним или несколькими). Определить длину самого короткого слова.
- 8) Дана строка-предложение на латинском языке. Преобразовать строку так, чтобы каждое слово начиналось с заглавной буквы.
- 9) Дана строка-предложение, содержащая избыточные пробелы. Преобразовать ее так, чтобы между словами был ровно один пробел.
- 10) Дана строка-предложение на русском языке. Зашифровать ее, выполнив циклическую замену каждой буквы на следующую за ней в алфавите и сохраняя при этом регистр букв ("А" перейдет в "Б", "а" — в "б", "Б" — в "В", "я" — в "а" и т.д.). Букву "ё" в алфавите не учитывать ("е" должна переходить в "ж"). Знаки препинания и пробелы не изменять.
- 11) Дана строка-предложение на русском языке и число k ($0 < k < 10$). Зашифровать строку, выполнив циклическую замену каждой буквы на букву того же регистра, расположенную в алфавите на k -й позиции после шифруемой буквы (например, для $k = 2$ "А" перейдет в "В", "а" — в "в", "Б" — в "Г", "я" — в "б" и т.д.). Букву "ё" в алфавите не учитывать, знаки препинания и пробелы не изменять.
- 12) Дана строка-предложение. Зашифровать ее, поместив вначале все символы, расположенные на четных местах, а затем, в обратном порядке, все символы, расположенные на нечетных местах (например, строка "Программа" превратится в "ргамамроП").
- 13) Дана строка, содержащая несколько круглых скобок. Если скобки расставлены правильно (то есть каждой открывающей соответствует одна закрывающая), то вывести число 0. В противном случае вывести или номер позиции, в которой расположена первая ошибочная закрывающая скобка, или, если закрывающих скобок не хватает, число -1 .
- 14) Дано целое число. Вывести набор символов, содержащий цифры этого числа в обратном порядке.

- 15) В каждом слове текста замените "а" на букву "е", если "а" стоит на четном месте, и заменить букву "б" на сочетание "ак", если "б" стоит на нечетном месте.
- 16) Дан текст, содержащий от 2 до 30 слов, в каждом из которых от 2 до 10 латинских букв; между соседними словами - не менее одного пробела. Напечатать все слова, отличные от последнего слова, предварительно преобразовав каждое из них по следующему правилу: 1) перенести первую букву в конец слова; 2) перенести последнюю букву в начало слова.
- 17) Составить таблицу слов данного текста, начинающихся с буквы "А", с указанием числа повторений каждого слова.
- 18) Составить программу для вычеркивания из слов текста всех букв, стоящих на нечетных местах после буквы "а".
- 19) Составить программы для перевода арабских чисел в римские и для обратной операции. Например, $255 = \text{CCLV} = \text{сто} + \text{сто} + \text{пятьдесят} + \text{пять}$
- 20) Задано конкретное сочетание букв (например, «НЕТ»). Определить, сколько таких групп символов содержится в тексте, вводимом с клавиатуры.
- 21) Записать программу, выясняющую, можно ли из букв некоторого введенного с клавиатуры слова составить слово «окно».
- 22) В последовательности из N вводимых символов подсчитать количество цифровых символов (0..9), количество заглавных и количество прописных русских букв.
- 23) Для заданного предложения выдать эквивалентное ему предложение, полученное путем замены нескольких подряд следующих пробелов одним пробелом и удаления знаков "+" перед положительными числами.
- 24) Из строки символов А получить строку В путем удаления наиболее часто встречающихся символов.
- 25) Из заданной строки символов удалить символы, отличные от русских букв и пробелов.
- 26) Из двух заданных строк А и В по n слов в каждой сформировать строку С, записав в нее поочередно слова из строк А и В.
- 27) Определить, встречается ли некоторая подстрока в строке, и если да, то выдать номер символа, с которого она начинается.
- 28) В заданной строке символов найти слова минимальной и максимальной длины и выдать позиции, с которых они начинаются.
- 29) Составить программу «Переводчик», выдающую для 10 английских слов их перевод на русском языке.
- 30) Определить, является ли заданная последовательность символов арифметическим выражением, состоящим из целых чисел и четырех основных арифметических операций.

Варианты задач по теме «Подпрограммы функции»

Составить программу табулирования функции $y = y(x)$ на данном отрезке с данным шагом.

- 1) $y = \sqrt{x^2 + 5x + 8}$ на интервале $[0, 2]$ с шагом 0.1.
- 2) $y = (3x^2 - 4)/(x + 1)$ на интервале $[0.5, 2.5]$ с шагом 0.1.
- 3) $y = x - \sqrt{x + 2}$ на интервале $[-2, 2]$ с шагом 0.2.
- 4) $y = (7x + 4)/(3x + 5) + \cos x$ на интервале $[-1.7, 1]$ с шагом 0.1.
- 5) $y = x^3 + 4x^2 + x - 2$ на интервале $[-1, 1]$ с шагом 0.1.
- 6) $y = x + \sin 3x$ на интервале $[-1.5, 1.5]$ с шагом 0.12.
- 7) $y = \cos x + \sqrt{x}$ на интервале $[0, 3]$ с шагом 0.11.
- 8) $y = \sin 2x + \operatorname{tg} 3x$ на интервале $[0, 1]$ с шагом 0.05.
- 9) $y = \sqrt{x + 2} - \sqrt{x + 3}$ на интервале $[-1, 2]$ с шагом 0.25.
- 10) $y = 2x + \sin 2x$ на интервале $[-3, 3]$ с шагом 0.25.
- 11) $y = 5 - \cos 3x$ на интервале $[0, 2]$ с шагом 0.1.
- 12) $y = \sqrt{x} - \sqrt{x + 1}$ на интервале $[0, 5]$ с шагом 0.5.
- 13) $y = |x - 5| + x^2$ на интервале $[-3, 3]$ с шагом 0.25.
- 14) $y = e^x + 1$ на интервале $[-2, 2]$ с шагом 0.2.
- 15) $y = \sqrt{|\sin x|}$ на интервале $[0, 2]$ с шагом 0.1.
- 16) $y = |x + 3| - x^2$ на интервале $[-6, 3]$ с шагом 0.5.
- 17) $y = (\sin x + \cos x) \cdot \operatorname{tg}(x)$ на интервале $[-1.2, 1.2]$ с шагом 0.1.
- 18) $y = e^x - x - 2$ на интервале $[-2, 2]$ с шагом 0.2.
- 19) $y = (1 - \operatorname{tg} x)^2 - 2$ на интервале $[-1.5, 1.5]$ с шагом 0.1.
- 20) $y = \sin \sqrt{x + 1} + \sqrt{\sin x + 1} - 3$ на интервале $[-1, 2]$ с шагом 0.1.
- 21) $y = 3^x - 4x - 1$ на интервале $[-5, 5]$ с шагом 0.5.
- 22) $y = 2x - \sin 2x$ на интервале $[-\pi, \pi]$ с шагом $\pi/8$.
- 23) $y = \sqrt{|\operatorname{tg} 3x|} - 3x$ на интервале $[-0.5, 0.5]$ с шагом 0.05.
- 24) $y = x - 10^{2x}$ на интервале $[-1.5, 1.5]$ с шагом 0.05.
- 25) $y = x \cdot \cos x - 1$ на интервале $[-\pi, \pi]$ с шагом $\pi/8$.
- 26) $y = \sqrt{3 + x} - \operatorname{arctg} x$ на интервале $[-3, 3]$ с шагом 0.3.
- 27) $y = 2x - \sqrt{3 + |x|}$ на интервале $[-5, 5]$ с шагом 0.5.
- 28) $y = 3 + x \cdot \operatorname{tg} x$ на интервале $[-\pi, \pi]$ с шагом $\pi/8$.
- 29) $y = \ln \left| \sqrt{1 + x^2} - 2 \right| - 1$ на интервале $[-5, 5]$ с шагом 0.5.
- 30) $y = 3 + \cos x^3$ на интервале $[-\pi, 3\pi]$ с шагом $\pi/4$.

Задачи для самостоятельного решения по теме «Подпрограммы»

- 1) Для заданной матрицы размером 8×8 найти такие k , что k -я строка матрицы совпадает с k -м столбцом.
- 2) Даны три квадратных матрицы A, B, C n -го порядка. Вывести на печать ту из них, норма которой наименьшая. Нормой матрицы считать максимум из абсолютных величин ее элементов.
- 3) Даны три квадратных матрицы A, B, C n -го порядка. Вывести на печать ту из них, норма которой наибольшая. Нормой матрицы считать максимум из абсолютных величин ее элементов.
- 4) Подсчитать количество локальных минимумов заданной квадратной матрицы. Элемент матрицы называется локальным минимумом, если он строго меньше всех имеющихся у него соседей. Соседями элемента A_{ij} в матрице назовём элементы A_{km} , где $i - 1 \leq k \leq i + 1, j - 1 \leq m \leq j + 1, (k, m) \neq (i, j)$.
- 5) Составить программу перемножения двух матриц произвольной размерности, предусмотреть проверку возможности перемножения.
- 6) Составить программу сложения двух матриц произвольной размерности, предусмотреть проверку возможности сложения.
- 7) Дана целочисленная прямоугольная матрица. Определить количество строк, не содержащих ни одного нулевого элемента.
- 8) Дана целочисленная прямоугольная матрица. Определить количество столбцов, не содержащих ни одного нулевого элемента.
- 9) Дана целочисленная прямоугольная матрица. Определить количество столбцов, содержащих хотя бы один нулевой элемент.
- 10) Дана целочисленная прямоугольная матрица. Определить произведение элементов в тех строках, которые не содержат отрицательных элементов.
- 11) Дана целочисленная прямоугольная матрица. Определить сумму элементов в тех столбцах, которые не содержат отрицательных элементов.
- 12) Дана целочисленная прямоугольная матрица. Определить сумму элементов в тех строках, которые содержат хотя бы один отрицательный элемент.
- 13) Дана целочисленная прямоугольная матрица. Определить сумму элементов в тех столбцах, которые содержат хотя бы один отрицательный элемент.
- 14) Дана квадратная матрица действительных чисел. Найти сумму модулей элементов, расположенных ниже главной диагонали.
- 15) Дана квадратная матрица действительных чисел. Найти сумму модулей элементов, расположенных выше главной диагонали.
- 16) Дана прямоугольная матрица действительных чисел. Найти количество строк, среднее арифметическое элементов которых меньше заданной величины.
- 17) Дана прямоугольная матрица действительных чисел. Найти номер первой из строк, содержащих хотя бы один положительный элемент.
- 18) Дана квадратная матрица действительных чисел. Найти сумму модулей элементов, расположенных ниже побочной диагонали.

Окончание приложения X

- 19) Дана квадратная матрица действительных чисел. Найти сумму модулей элементов, расположенных выше побочной диагонали.
- 20) Дана прямоугольная матрица действительных чисел. Найти номер первого из столбцов, содержащих хотя бы один положительный элемент.
- 21) Найти номер первого из столбцов прямоугольной матрицы, не содержащих ни одного отрицательного элемента.
- 22) Найти номер первой из строк прямоугольной матрицы, не содержащих ни одного отрицательного элемента.
- 23) Дана целочисленная прямоугольная матрица. Определить номер первого из столбцов, содержащих хотя бы один нулевой элемент.
- 24) Дана целочисленная прямоугольная матрица. Определить номер первой из строк, содержащих хотя бы один нулевой элемент.
- 25) Дан двумерный массив. Определить: номер столбца, в котором расположен минимальный элемент каждой из строк массива. Если элементов с минимальным значением в этой строке несколько, то должен быть найден номер столбца самого левого из них.
- 26) Дан двумерный массив. Определить: номер строки, в которой расположен максимальный элемент каждого из столбцов массива. Если элементов с максимальным значением в этом столбце несколько, то должен быть найден номер строки самого верхнего из них.
- 27) Дан двумерный массив целых чисел. Составить программу, которая определяет, есть ли в каждой из строк массива хотя бы один элемент, оканчивающейся цифрой 3. В случае положительного ответа должны быть напечатаны координаты любого из таких элементов
- 28) Дан двумерный массив. Составить программу, которая определяет, является ли последовательность элементов каждой из строк массива неубывающей. В случае отрицательного ответа должны быть напечатаны координаты первого элемента, "нарушающего" указанную упорядоченность.
- 29) Дан двумерный массив. Составить программу, которая определяет, является ли последовательность элементов каждого из столбцов массива невозрастающей. В случае отрицательного ответа должны быть напечатаны координаты первого элемента, "нарушающего" указанную упорядоченность.
- 30) Дана целочисленная прямоугольная матрица. Найти произведение элементов всех диагоналей, параллельных главной диагонали матрицы.