

Министерство образования Республики Беларусь  
Белорусская государственная политехническая  
академия

---

Кафедра "Организация автомобильных перевозок и  
дорожного движения"

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ  
И КОНТРОЛЬНЫЕ ЗАДАНИЯ**

**по дисциплине "Информатика"**

**для студентов специальности Т04.03 *"Организация движения  
и управление на транспорте"***

**В 2-х частях**

**Часть 2. Программирование на языке ПАСКАЛЬ**

УДК 681.3.06

"Методические указания и контрольные задания по дисциплине "Информатика" для студентов специальности Т.04.03 "Организация движения и управление на транспорте" составлены в соответствии с образовательным стандартом РД РБ 02100.5.009-98.

Приведены программа дисциплины "Информатика", тематика индивидуальной работы, методические указания по программированию на языке Паскаль и контрольные задания.

Пособие может быть использовано студентами очной и заочной форм обучения при выполнении контрольных, курсовых и дипломных работ.

Составители: Д.В.Рожанский  
В.В.Мочалов

Рецензент О.В.Бугай

© Д.В.Рожанский, В.В.Мочалов, составление

## **СОДЕРЖАНИЕ**

1. Программа дисциплины "Информатика".....	3
2. Тематика индивидуальной работы.....	7
3. Методические указания по составлению программ на алгоритмическом языке Паскаль.....	8
4. Контрольные задания.....	14
Литература.....	23

### **1. ПРОГРАММА ДИСЦИПЛИНЫ "ИНФОРМАТИКА"**

#### **Цель преподавания дисциплины**

Цель дисциплины - обучить студентов умению работы с современными ЭВМ и использованию программных и технических средств ЭВМ для решения научных и прикладных задач, характерных для будущей работы по специальности.

#### **Задачи изучения дисциплины**

В результате изучения дисциплины студент должен знать современные технические средства взаимодействия с ЭВМ, программное обеспечение ЭВМ, освоить методы разработки алгоритмов и изучить алгоритмические языки программирования, получить практические навыки работы с прикладными программными средствами для решения задач и обработки информации.

На основе полученных знаний студент должен овладеть навыками подготовки задач из областей будущей работы и решения их с использованием современных алгоритмических языков программирования и программного обеспечения ЭВМ, навыками отладки разработанных программ и применения пакетов прикладных программ.

Полученные знания и навыки при изучении данной дисциплины являются базовыми и далее используются и совершенствуются в дисциплинах "Математические модели в транспортных системах", "Электронная автоматика и техника", "АСУ на транспорте" и др.

Программа разработана в соответствии с Образовательным стандартом РФ РФ 02100.5.009-98, которым на изучение дисциплины предусмотрено 300 ч, в том числе аудиторных - 200 ч.

### **ЛЕКЦИОННЫЙ КУРС**

#### **Введение**

Информатика и научно-технический прогресс. Цель и задачи дисциплины.

Вычислительная техника - составная часть научно-технического прогресса. История создания и развития средств вычислительной техники, тенденции и перспективы развития. Классификация ЭВМ. Представление информации в ЭВМ. Краткая характеристика цифровых и аналоговых машин, персональные ЭВМ (ПЭВМ).

Применение вычислительной техники в организации перевозок и управления движением транспортных средств.

Цель и задачи курса. Рекомендуемая литература.

## **Архитектура и принципы работы ЭВМ**

Основные понятия теории информации. Арифметические и логические операции. Системы счисления, арифметика двоичной и шестнадцатеричной системы счисления. Система команд.

Устройство ЭВМ: структурная схема ЭВМ, назначение, принцип действия и характеристики блоков и устройств. Средства взаимодействия пользователя с ЭВМ.

## **Операционные системы и системы обеспечения**

Общие сведения о программном обеспечении ЭВМ. Операционные системы современных ЭВМ: системы 8-миразрядных ЭВМ (DOS, CP/M), системы более производительных ПЭВМ (MS DOS, OS\2, Windows, Unix, Linux и др.).

Операционная систем MS DOS, основные понятия и команды. Программы-оболочки: Norton Commander, Dos Navigator и др., особенности и основные функции.

Операционная система Windows. Особенности мультизадачного режима.

Языки программирования: назначение, классификация, машинно-зависимые и машинно-независимые, интерпретаторы и компиляторы. Краткий сравнительный обзор языков программирования высокого уровня. Требования к программированию. Отладка и контрольные примеры для тестирования.

Понятие об объектно-ориентированном программировании и графическом пользовательском интерфейсе.

Текстовые редакторы. Назначение и возможности. Основные правила работы. Правила набора и корректировки текста. Сохранение и загрузка текстовых файлов, работа с фрагментами и окнами.

## **Алгоритмизация вычислений**

Алгоритмизация обработки данных. Способы описания и схемы алгоритмов и программ. Символы схем алгоритмов и правила построения схем.

Разновидности структур алгоритмов. Организация алгоритмов линейной, разветвляющейся, циклической и вложенной структур. Схема решения задачи. Этапы подготовки задачи к решению на ЭВМ.

Приемы разработки типовых алгоритмов: линейный алгоритм, алгоритм с разветвлениями, цикл, цикл с несколькими одновременно изменяющимися параметрами, накопление суммы и произведения, вычисление суммы бесконечного ряда с заданной точностью, вычисление полинома, нахождение наибольшего и наименьшего значений и др.

## **Программирование на языках высокого уровня**

### **(Бейсик, Паскаль и др.)**

Общие сведения и основные элементы языка Бейсик. Алфавит и основные конструкции языка: символы, константы, переменные, массивы, стандартные функции, выражения, строки. Операции (арифметические, отношения и логические) и запись выражений на языке Бейсик. Примеры. Операторы и команды. Назначение и классификация. Описание и синтаксис основных операторов и команд языка. Структурное программирование, структурирование программ. Отладка программ.

Программирование алгоритмов линейной структуры. Операторы присваивания, ввода-вывода информации, Запись программ на Бейсике. Этапы подготовки для составления алгоритма и программы. Программирование алгоритмов с разветвлениями. Операторы безусловного и условного переходов.

Программирование алгоритмов с циклами. Организация цикла с использованием оператора условного перехода, с использованием оператора цикла. Цикл с заданным числом повторений и итерационный (с предусловием и с постусловием). Вложенные циклы.

Программирование и обработка массивов данных. Элементы статистического анализа. Программирование с использованием подпрограмм и функций.

Приемы программирования типовых алгоритмов: накопление суммы и произведения, нахождение наибольшего и наименьшего значений, построение графиков, цикл с несколькими одновременно изменяющимися параметрами, вычисление суммы бесконечного ряда с заданной точностью, вычисление полинома.

Программирование с использованием функций и процедур. Формальные и фактические параметры. Структура исполняемых программ (оверлейность, модульность, межмодульный обмен).

Файлы. Операторы работы с файлами. Текстовые данные, способы их хранения и обработки.

Дополнительные возможности языка Бейсик. Ввод, отладка, редактирование и компиляция программ. Разновидности языков Бейсик. Дополнительные возможности языков Бейсик для современных компьютеров. Составление модульных подпрограмм. Графические и звуковые возможности.

Особенности программирования на языках Паскаль, Фортран и др.

### **Программирование на объектно-ориентированных языках**

Основы современной технологии программирования. Объектно-ориентированное и визуальное программирование. Обзор, основные особенности и дополнительные возможности объектно-ориентированных языков. Виды объектов (проект, формы, модули, свойства). Описание и изучение основных объектов.

Общие правила создания программ. Изучение возможностей отладчиков программных продуктов.

Реализация типовых алгоритмов с учетом особенностей элементов объектно-ориентированного и визуального программирования. Принципы и формы организации меню. Организация различных способов ввода-вывода данных.

Работа с графическими объектами.

Взаимодействие с другими программными пакетами (экспорт-импорт данных).

Библиотеки программных объектов.

### **Системы программного обеспечения**

Основные системы программного обеспечения ЭВМ. Инструментальные, интегрированные, функционально и проблемно ориентированные пакеты.

Работа в среде Windows.

Современная система прикладных программ Microsoft Office. Назначение, состав, программные особенности и правила работы. Требования к техническому оборудованию.

Текстовые редакторы. Шрифты, страничное разбиение. Подготовка оглавления. Печать. Обзор редакторов WordStar, Lexicon, Фотон, Слово и дело, Word и др.

Электронные таблицы: понятие, способы записи данных (в клетки, колонки, строки). Правила работы с меню команд электронной таблицы. Стандартные функции: арифметические, статистические, логические. Корректировка, сортировка и печать таблиц. Электронная таблица Excel.

Системы управления базами данных: понятие, назначение, классификация. Структура данных: поле, тип поля, запись. Создание и корректировка базы данных: изменение полей, добавление данных. Выбор данных, статистика и вычисления в базе. Вывод отчетов. Типовые СУБД (Paradox, FoxPro, Access).

Обучающие и контролируемые программы: назначение, режимы контроля и обучения, правила работы с меню. Экспертные системы.

Графические редакторы. Назначение, обзор. Настройка редактора на используемое оборудование. Правила вычерчивания геометрических фигур. Редактирование графических изображений, сохранение и вывод на принтер, графопостроитель.

Пакеты прикладных программ - автоматизированные рабочие места (АРМ) инженера-конструктора, математика: понятие, правила работы с меню, примеры основных команд (AutoCad, MathCad); системы графического и голосового ввода и распознавания информации; обработка звуковой и видеoinформации; компьютерные переводчики и словари.

Компьютерные сети. Обеспечение обмена информацией между ЭВМ. Техническое и программное обеспечение: модемы, их технические характеристики и особенности программирования, Hayes-совместимые команды, обзор терминальных и сетевых программных пакетов, основные правила работы в сетях. Глобальные сети. INTERNET. Защита информации.

### **Заключение**

Перспективы развития и применения компьютерных технологий. Использование ЭВМ и микропроцессоров в различных отраслях, в частности, при организации перевозок грузов и пассажиров, управлении на транспорте и организации дорожного движения. Особенности программирования микропроцессоров. Перспективы развития и применения средств вычислительной техники.

### **ЛАБОРАТОРНЫЕ ЗАНЯТИЯ**

#### **(примерный перечень)**

Изучение устройства ЭВМ, правил техники безопасности.

Операционные системы. Системные оболочки.

Алгоритмизация задач, ввод и редактирование текстов в системах программирования.

Подготовка компьютерных программ с линейной структурой вычислений.

Подготовка компьютерных программ с разветвляющейся структурой.

Подготовка компьютерных программ с циклической структурой вычислений, табулирование значений функции, вычисление сумм, произведения, нахождение наибольшего и наименьшего значений.

Подготовка компьютерных программ с использованием структуры вложенных циклов: табулирование функций нескольких переменных, действия с матрицами, вычисление полиномов.

Подготовка компьютерных программ с обработкой символьной информации.

Подготовка компьютерных программ с использованием подпрограмм пользователя.

Подготовка компьютерных программ с использованием библиотечных подпрограмм.

Подготовка компьютерных программ организации дорожного движения и автомобильных перевозок.

Подготовка в среде Windows и решение задач прикладного программирования (обработка текстовой информации, работа с базами данных и электронными таблицами).

Разработка общего проекта на объектно-ориентированном языке с включением в него ранее разработанных программ-модулей и библиотечных объектов.

Программирование модемов, работа компьютеров в терминальном режиме и в сетях.

## **2. ТЕМАТИКА ИНДИВИДУАЛЬНОЙ РАБОТЫ**

Разработка и отладка вопросов (по индивидуальной тематике) и 4-х вариантов ответов к каждому вопросу с оценками по каждому варианту для системы программированного обучения и контроля знаний по информатике.

### **Курсовая работа**

Разработка алгоритма и компьютерной программы численного расчета многопараметрической функции (по вариантам) с обработкой данных с внешних устройств (с элементами статистического анализа, выводом данных на магнитные диски и бумагу, с построением графиков) или разработка алгоритма и программы обработки символьной информации в базе данных (по вариантам) с элементами ввода, удаления, замены, печати, поиска элемента базы и составление диалогового меню.

### **Контрольные работы**

#### **(для заочной формы обучения)**

Разработка алгоритма и программы численного расчета многопараметрической функции на языке Бейсик (повариантно).

Разработка алгоритма и программы численного расчета многопараметрической функции на алгоритмическом языке высокого уровня (Visual Basic или др.), повариантно.

### 3. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО СОСТАВЛЕНИЮ ПРОГРАММ НА АЛГОРИТМИЧЕСКОМ ЯЗЫКЕ ПАСКАЛЬ

Программа состоит из строк. Максимальная длина строки - 127 символов. Строка может начинаться с любой позиции от левой границы экрана дисплея. Количество операторов в строке не регламентировано. Операторы отделяются друг от друга символом ";".

Программа начинается с заголовка, который записывается в следующем виде:

```
PROGRAM <имя программы>;
```

Под заголовком располагаются блок описаний и блок операторов. Программу рекомендуется снабжать комментарием, в котором указываются номер группы, вариант задания и фамилия студента. Текст комментария заключается в { } или (\* \*) и помещается перед блоком описаний.

#### Блок описаний

В блоке описываются все переменные, константы, метки, типы данных, процедуры и функции, используемые в программе.

1. Раздел описания переменных:

```
VAR <имя переменной, ...> : <тип переменной>;
```

При составлении программ, предназначенных для выполнения вычислений, чаще всего используют типы переменных **REAL** (вещественный) и **INTEGER** (целый). Логические переменные описываются идентификатором **BOOLEAN**, а литерные - идентификатором **CHAR**.

Пример.

```
VAR  
SUMMA, RESULT : REAL;  
I, J, NOMER : INTEGER;  
LG : BOOLEAN;  
BUKVA : CHAR;
```

Массив данных можно описать в разделе описания переменных следующим образом:

```
VAR  
<имя массива, ...> : ARRAY [тип индексов] OF <тип компонентов>;
```

Тип индексов определяет количество индексов массива (его размерность), их тип (индекс не может иметь вещественный тип) и границы изменения значений индексов. Тип компонентов определяет тип данных, записанных в массив.

Пример.

```
VAR  
X : ARRAY [1..10] OF REAL;  
NM, SVT : ARRAY [1..4, 1..20] OF INTEGER;
```

Массив **X** может содержать до 10 чисел вещественного типа, а каждый из двумерных массивов **NM** и **SVT** - до 80 целых чисел.



## 2. Раздел описания констант

**CONST** <Имя константы>=<значение константы>;

После того, как константа определена, ей нельзя присвоить другое значение.

Пример.

**CONST**

**A=15.0; B=5; T='ПРИМЕР';**

Константа **A** вещественного типа, **B** - целого типа, **T** - символьного (строкового) типа.

В Паскале имеется ряд констант, которые определены стандартно, к ним можно обращаться без предварительного описания. Например, число "пи" - **PI** =3.1415926536.

## 3. Раздел описания меток

Перед любым оператором языка Паскаль можно поставить метку, что позволяет перейти к выполнению этого оператора с помощью оператора безусловного перехода.

Формат описания меток:

**LABEL** <Имя метки, ...>;

Имя метки может состоять из цифр и (или) букв.

Пример.

**LABEL 1, M2;**

В блоке операторов оператор с меткой записывается следующим образом:

**1:<оператор>;**

**M2:<оператор>;**

### Блок операторов

Блок начинается ключевым словом **BEGIN**, за которым следуют все операторы программы, и заканчивается ключевым словом **END**, после которого ставится точка.

#### 1. Операторы ввода-вывода данных

Оператор чтения **READ** служит для ввода числовых и символьных данных с помощью клавиатуры. Записывается по формату:

**READ (X1, X2, ..., XN);**

Значения переменных **X1, X2, ..., XN** набираются минимум через один пробел (без запятых).

Оператор чтения **READLN** отличается от оператора **READ** тем, что после считывания последнего из вводимых значений обеспечивает перевод строки.

Оператор записи **WRITE** производит вывод на экран дисплея числовых, символьных и логических значений. Формат записи:

**WRITE (Y1, Y2, ..., YN);**

Для вывода данных на печатающее устройство перед списком переменных записывается идентификатор **LST**, например,

**WRITE (LST, A, B);**

При работе в Турбо-Паскале, кроме того, необходимо подключить стандартный модуль **PRINTER**. Для этого сразу после заголовка программы следует записать:

### **USES PRINTER;**

Оператор записи **WRITELN** отличается от оператора **WRITE** тем, что после вывода последнего значения обеспечивает перевод строки.

В операторах **WRITE** и **WRITELN** можно задавать формат вывода данных. В частности, существуют следующие форматы вывода вещественных чисел:

**WRITE (R)** - значение **R** выводится в форме с порядком в поле шириной 18 позиций;

**WRITE (R:P)** - значение **R** выводится в форме с порядком в поле шириной **P** позиций, причем после десятичной точки выводится по крайней мере одна цифра (поэтому при выводе положительных чисел минимальное значение равно 7, а при выводе отрицательных чисел - 8);

**WRITE (R:P:q)** - значение **R** выводится в форме с фиксированной точкой в крайние правые позиции поля шириной **P**, причем после десятичной точки выводится **q** цифр.

Пример программы ввода-вывода значений переменной **A**.

```
PROGRAM PR1;
```

```
  VAR A:REAL;
```

```
  BEGIN
```

```
    WRITE ('Введите значение A  ');
```

```
    READLN (A);
```

```
    WRITELN ('A=', A);
```

```
    WRITELN ('A=', A:10);
```

```
    WRITELN ('A=', A:10:3)
```

```
  END.
```

В результате выполнения программы на экране появится сообщение:

**Введите значение A**

После ввода числа, например, -248.12, на экран будет выведено значение переменной **A** по трем различным форматам:

```
A=-2.4812000000E+02
```

```
A=-2.481E+02
```

```
A= -248.120
```

## 2. Арифметические выражения и оператор присваивания

Арифметические выражения записываются при помощи следующих знаков арифметических операций: **+, -, \*, /**. Операция возведения в степень отсутствует.

Основные арифметические функции и процедуры:

$x^2$ - <b>SQR (X)</b>	$ x $ - <b>ABS (X)</b>
$\sqrt{x}$ - <b>SQRT (X)</b>	$\sin x$ - <b>SIN (X)</b>
$e^x$ - <b>EXP (X)</b>	$\cos x$ - <b>COS (X)</b>
$\ln x$ - <b>LN (X)</b>	$\arctg x$ - <b>ARCTAN (X)</b>

Для возведения в степень (например,  $x^a$ ) используется следующее выражение **EXP (A\*LN (X))**. Чтобы возвести в целую степень отрицательное число, необходимо организовать цикл.

Присваивание вычисленного значения выражения какой-либо переменной осуществляется с помощью оператора присваивания **:=**, например:

```
A:=X*C*COS (Y)+SQRT (Z) ;
```

### 3. Операторы передачи управления

Условный оператор **IF-THEN-ELSE** может записываться двумя способами:

```
IF <условие> THEN <оператор 1>  
ELSE <оператор 2>;
```

или

```
IF <условие> THEN <оператор>;
```

Условие - это выражение логического типа или логическая переменная. Для записи условий используются операции отношения **=**, **<>** (не равно), **<**, **>**, **<=**, **>=** и логические операции **NOT**, **AND**, **OR**, **XOR**.

Поскольку приоритет логических операций выше, чем операций отношения, при записи сложных логических условий необходимо использовать скобки. Например, неравенство  $1 < x \leq 5$  следует записать так:

```
(X>1.0) AND (X<=5.0)
```

При записи условного оператора **IF** первым способом, если условие истинно, то выполняется **<оператор 1>**, а если ложно - **<оператор 2>**. В случае записи условного оператора вторым способом, если условие истинно, то выполняется оператор после слова **THEN**, в противном случае - следующий оператор программы.

После ключевых слов **THEN** и **ELSE** можно записать несколько операторов, но их необходимо заключить в операторные скобки **BEGIN** - **END**, такая структура называется составным оператором, например:

```
IF X>Y THEN BEGIN  
WRITELN('X больше Y');  
Z:=X-Y  
END
```

**ELSE**

**BEGIN**

**WRITELN('X меньше или равно Y');**

**Z:=Y-X**

**END;**

Один оператор **IF** может входить в состав другого оператора **IF**, т.е. допускается вложенность операторов.

Оператор выбора **CASE-OF-ELSE-END** позволяет сделать выбор из произвольного числа имеющихся вариантов.

Формат записи оператора:

**CASE** <выражение-селектор> **OF**

<список 1>:<оператор 1>;

<список 2>:<оператор 2>;

...

<список N>:<оператор N>

**ELSE** <оператор>

**END;**

<список 1>, <список 2>, ... - списки констант, которые называются константами выбора.

Сначала вычисляется значение выражения-селектора, а затем выполняется тот оператор, в списке констант выбора которого имеется константа, совпадающая с полученным значением селектора. Если такая константа отсутствует, то выполняется оператор после слова **ELSE**. Константы выбора и выражение-селектор должны иметь одинаковый тип (не допускается вещественный тип). Строка со словом **ELSE** может отсутствовать.

Пример.

**CASE I OF**

1,2:X:=10\*I;

3..10:X:=100\*I

**ELSE X:=I**

**END;**

Оператор безусловного перехода **GOTO** предназначен для передачи управления оператору, помеченному меткой, например:

**GOTO M1;**

#### 4. Операторы повтора

Оператор повтора **FOR-TO-DO** может быть записан двумя способами:

**FOR** <параметр цикла>:=<S1> **TO** <S2> **DO** <оператор>;

**FOR** <параметр цикла>:=<S1> **DOWNTO** <S2> **DO** <оператор>;

S1 и S2 - выражения, определяющие соответственно начальное и конечное значения параметра цикла (не допускается вещественный тип). Если параметр цикла целого типа, то при каждом повторе тела цикла параметр либо увеличивается на 1 (при использовании формата **FOR-TO-DO**), либо уменьшается на 1 (при использовании формата **FOR-DOWNTO-DO**). Выполнение цикла прекращается по достижении

параметром цикла конечного значения. После слова **DO** может быть записан простой или составной оператор.

Оператор повтора **WHILE-DO** используется для создания циклов в тех случаях, когда число повторов заранее неизвестно, либо если шаг изменения параметра цикла отличен от 1.

Формат оператора:

**WHILE** <условие> **DO** <оператор>;

Условие - выражение логического типа. Оператор может быть простым или составным. Оператор выполняется до тех пор, пока условие истинно.

Оператор повтора **REPEAT-UNTIL** записывается по формату:

**REPEAT**

<оператор 1>;

<оператор 2>;

...

<оператор N>

**UNTIL** <условие>;

Условие - выражение логического типа. Операторы, записанные между **REPEAT** и **UNTIL**, выполняются до тех пор, пока условие ложно. Как только условие примет значение "истина", выполнение цикла прекращается.

В теле цикла операторов повтора **WHILE-DO** и **REPEAT-UNTIL** необходимо обеспечить изменение переменных, входящих в условие окончания цикла. Иначе цикл будет выполняться бесконечно.

Примеры использования операторов повтора для обнуления элементов двумерного массива **A** размером 10x20, предназначенного для записи вещественных чисел.

**PROGRAM PR2;**

**VAR A:ARRAY [1..10, 1..20] OF REAL;**

**I, J: INTEGER;**

**BEGIN**

**FOR I:=1 TO 10 DO**

**FOR J:=1 TO 20 DO**

**A[I, J]:=0.0**

**END.**

**PROGRAM PR3;**

**VAR A:ARRAY [1..10, 1..20] OF REAL;**

**I, J: INTEGER;**

**BEGIN**

**I:=1;**

**WHILE I<=10 DO**

**BEGIN**

**J:=1;**

**WHILE J<=20 DO**

**BEGIN**

**A[I, J]:=0.0;**

```

                J:=J+1
            END;
        I:=I+1
    END
END.

PROGRAM PR4;
    VAR A:ARRAY [1..10, 1..20] OF REAL;
        I, J: INTEGER;
BEGIN
    I:=1;
    REPEAT
        J:=1;
        REPEAT
            A[I, J]:=0.0;
            J:=J+1
        UNTIL J>20;
        I:=I+1
    UNTIL I>10
END.

```

5. Основные правила использования точки с запятой при написании программы

Точка с запятой ставится между операторами, т.к. является их разграничителем.

Точка с запятой не ставится после **BEGIN** и перед **END**.

В условных операторах точка с запятой не ставится после **THEN**, **CASE** и перед **ELSE**.

В операторах повтора точка с запятой не ставится после **WHILE**, **REPEAT**, **DO** и перед **UNTIL**.

#### 4. КОНТРОЛЬНЫЕ ЗАДАНИЯ

Алгоритм решения задачи представляется в виде блок-схемы с использованием блоков стандартной конфигурации. В заданиях разработанная программа должна обеспечивать возможность ввода с помощью клавиатуры значений коэффициентов, входящих в выражения заданных функций, начального, конечного значений и шага изменения аргументов. Ввод и печать векторов и матриц оформлять в виде таблиц.

### Задание 4.1

Составить алгоритм и программу расчета значений функции

$$Y = \begin{cases} FA(X), & \text{если условие } U \text{ выполняется} \\ FB(X), & \text{если условие } U \text{ не выполняется.} \end{cases}$$

Функции **FA(X)**, **FB(X)** принять по табл.4.1, подставив в обозначения функций вместо букв **A** и **B** соответственно предпоследнюю и последнюю цифры шифра зачетной книжки (если цифры одинаковы, то к последней прибавить **1**). Условие **U** выбрать из табл.4.2 в соответствии с номером варианта, который определяется последней цифрой шифра зачетной книжки.

Таблица 4.1

Обозначение	Функция
<b>F0(X)</b>	$a \cdot x^4 + b \cdot x^3 + c$
<b>F1(X)</b>	$a \cdot x^3 + b \cdot x^2 + c$
<b>F2(X)</b>	$e^{-ax} + b \cdot x^2$
<b>F3(X)</b>	$a \cdot \sin x + \cos^2 x$
<b>F4(X)</b>	$b \cdot \cos x - \sin^3 x$
<b>F5(X)</b>	$\sqrt{x^2 + \sin^2 x}$
<b>F6(X)</b>	$a \cdot x^3 - b \cdot \sin x$
<b>F7(X)</b>	$b \cdot x^{3/2} + a \cdot \cos x$
<b>F8(X)</b>	$x^{a+b} - c \cdot \sin^2 x$
<b>F9(X)</b>	$a \cdot \ln x - b \cdot \cos^3 x$

При составлении программы предусмотреть ввод значений переменных  $P$  и  $Q$  с помощью клавиатуры.

Таблица 4.2

Номер варианта	Условие	Номер варианта	Условие
0	$P \geq 0$ и $Q > 0$	5	$Q > 1$ и $P < 1$
1	$0 < P < 1$ и $Q < 0$	6	$Q - P > 0$ или $Q \leq 0$
2	$Q < P$ или $P \leq 1$	7	$Q^2 \cdot P > 4$ или $P < 0$
3	$P^2 > 1$ или $Q > 0$	8	$0 < Q < P$ и $P > 2$
4	$P < 0$ и $Q > 1$	9	$P - Q < 0$ или $2 \cdot Q > 5$

### Задание 4.2

Составить алгоритм и программу расчета значений функции  $Y = FC(X)$  и определения ее минимального и максимального значений при изменении  $X$  от  $X_0$  до  $X_N$  с шагом  $DX$ . Функцию  $FC(X)$  принять по табл.4.1, подставив в обозначение функции вместо буквы  $C$  ее значение, найденное по формуле  $C = A + B$ , где  $A$  и  $B$  - предпоследняя и последняя цифры шифра зачетной книжки. Если значение  $C$  больше или равно 10, то из него вычитается 10.

### Задание 4.3

Составить алгоритм и программу определения суммы значений функции  $FD(X)$  и доли каждого из значений в полученной сумме при изменении  $X$  от  $X_0$  до  $X_N$  с шагом  $DX$ . Функцию  $FD(X)$  принять по табл.4.1, подставив в обозначение функции вместо буквы  $D$  ее значение, найденное по формуле  $D = |A - B|$ , где  $A$  и  $B$  - соответственно предпоследняя и последняя цифры шифра зачетной книжки.

### Задание 4.4

Составить алгоритм и программу вычисления значений функции двух переменных:

$$Y = \begin{cases} FC(X_1) + FD(X_2) & \text{при } X_1 \geq X_2 \\ FC(X_1) - FD(X_2) & \text{при } X_1 < X_2. \end{cases}$$



Аргумент  $X1$  изменяется от  $X01$  до  $XN1$  с шагом  $H1$ ; аргумент  $X2$  изменяется от  $X02$  до  $XN2$  с шагом  $H2$ . Функции  $FC(X1)$ ,  $FD(X2)$  принять по табл.4.1, подставив в обозначения функций вместо букв  $C$  и  $D$  - их значения, найденные по формулам  $C=9-A$  и  $D=9-B$ , где  $A$  и  $B$  - соответственно предпоследняя и последняя цифры шифра зачетной книжки (если цифры одинаковы, то к последней прибавить 1).

### Задание 4.5

1. Составить блок-схему алгоритма и программу решения задачи, указанной в задании.
2. Получить распечатку результатов расчета по разработанной программе в соответствии с заданными исходными данными.
3. Описать блок-схему алгоритма решения задачи.
4. Составить программу с применением подпрограмм для вывода на печатающее устройство титульного листа установленной формы.

Вариант 1. Из вектора  $p=(p_1, \dots, p_n)$  ( $n \leq 20$ ) выделить вектор  $r=(r_1, \dots, r_m)$  ( $m \leq n$ ) по правилу: компонент вектора  $p$  является компонентом вектора  $r$ , если квадратное уравнение  $x^2 - 2p_i x + q = 0$  имеет вещественные и различные корни.

Исходные данные:  $n=7$ ;  $q=4,0$ ;  $p=(2,6; 3,3; 1,8; 5,6; 0,5; -2,8; -4,2)$ .

Вариант 2. Вычислить компоненты вектора  $x=(x_1, \dots, x_n)$  ( $n \leq 20$ ) по формуле:

$$x_i = 4 \cdot \arctg(1 + 0,1 \cdot i) + e^{-0,5 \cdot i} - \pi, \quad i=1, \dots, n.$$

Преобразовать далее полученный вектор по правилу: все отрицательные компоненты увеличить на 0,5, а все положительные - заменить на 0,1. Напечатать вычисленный и преобразованный векторы.

Исходные данные:  $n=13$ .

Вариант 3. Найти норму заданной квадратной матрицы порядка  $m$  ( $m \leq 20$ ) по формуле:

$$N = \max_i \sum_{k=1}^m |a_{ik}|$$

Определить наименьший элемент матрицы, номер строки и столбца этого элемента.

Исходные данные:  $m=3$ ;  $A=$

1,1	6,1	-0,4
0,8	3,2	0,2
0,9	0,4	8,7

Вариант 4. Задана матрица  $A=(a_{ij})$  размером  $m \times n$  ( $n, m \leq 20$ ). Получить новую матрицу путем деления всех элементов первоначальной на элемент, наибольший по абсолютной величине. Полученную матрицу напечатать.

Исходные данные:  $m=3; n=2; A=$ 

	1,2	-6,4
2,8	0,5	
-0,8	1,4	

Вариант 5. Образовать вектор  $a=(a_1, \dots, a_{21})$  из членов последовательности  $\cos x, \cos(x+h), \dots, \cos(x+20h)$ .

Найти сумму тех членов последовательности, которые по абсолютной величине больше, чем 0,5.

Исходные данные:  $x=0,2; h=0,3$ .

Вариант 6. Вычислить значения компонентов вектора  $a=(a_1, \dots, a_{30})$  по формуле:

$$a_i = \sin \frac{i^2 + 1}{30}, \quad i=1, \dots, 30$$

и вывести значения  $a_i$  на печать. Вычислить суммы:

$$S_1 = a_1 + a_6 + a_{11} + \dots + a_{26};$$

$$S_2 = a_2 + a_7 + a_{22} + \dots + a_{27};$$

.....

$$S_5 = a_5 + a_{10} + a_{15} + \dots + a_{30}.$$

Полученный вектор  $s=(S_1, \dots, S_5)$  напечатать.

Вариант 7. Компоненты заданного вектора  $a=(a_1, \dots, a_9)$  расположены в порядке убывания. Из вектора  $a$  и переменной  $x$  составить вектор  $b=(b_1, \dots, b_{10})$ , содержащий переменную  $x$ , все компоненты которого расположены в порядке возрастания. Вычислить сумму:

$$s = \lg b_2^2 + \lg b_4^2 + \dots + \lg b_{10}^2.$$

Вектор  $b$  и значение  $s$  вывести на печать.

Исходные данные:  $a=(9,6; 7,4; 4,1; 4,0; 3,8; 2,5; 2,4; 2,2; 1,7); x=2,3$ .

Вариант 8. Заданы логический вектор  $a=(a_1, \dots, a_n)$  и вещественный вектор  $x=(x_1, \dots, x_n)$  ( $n \leq 20$ ). Преобразовать вектор  $x$  по правилу: если  $a_i$  имеет значение TRUE, то  $x_i$  увеличить на 10, в противном случае заменить  $x_i$  нулем. Преобразованный вектор напечатать.

Исходные данные:  $n=4; a=(T, F, T, F); x=(1,2; 6,4; -3,8; 0,6)$ , где T-TRUE, F-FALSE.

Вариант 9. Вычислить компоненты вектора  $\mathbf{x} = (x_1, \dots, x_n)$  ( $n \leq 20$ ) по формуле:

$$x_i = \cos\left(i + \frac{1}{n}\right), \quad i=1, \dots, n.$$

Подсчитать число отрицательных компонентов вектора  $\mathbf{x}$  и число компонентов, принадлежащих отрезку  $[0,5; 1]$ . Все положительные компоненты, не принадлежащие указанному отрезку, заменить на 1. Остальные компоненты оставить без изменения.

Исходный и преобразованный векторы напечатать.

Исходные данные:  $n=15$ .

Вариант 10. Вычислить значения компонентов вектора  $\mathbf{x} = (x_1, \dots, x_{20})$  по формуле:

$$x_i = \arctg\left(\sqrt{i} + \frac{1}{20}\right) + e^{-0,5 \cdot i}, \quad i=1, \dots, 20.$$

Логической переменной присвоить значение TRUE, если компоненты вектора  $\mathbf{x}$  образуют монотонно возрастающую последовательность

$(x_1 < x_2 < \dots < x_{20})$ , и значение FALSE - в противном случае.

Вектор  $\mathbf{x}$  и значение логической переменной вывести на печать.

Вариант 11. Вычислить значения компонентов вектора  $\mathbf{x} = (x_1, \dots, x_{15})$  по формуле:

$$x_i = 1,3 \cdot \cos(0,1 \cdot i) + \sin(2 \cdot i)$$

и вывести вектор на печать. Найти сумму компонентов вектора  $\mathbf{x}$ , принадлежащих отрезку  $[-0,5; 0,5]$ , и число таких компонентов, а также сумму компонентов, принадлежащих отрезку  $[0,5; 1]$ , и число таких компонентов.

Вариант 12. Значения компонентов вектора  $\mathbf{x} = (x_1, \dots, x_{10})$  вычислить по формуле:

$$x_i = 1,5 \cdot \lg(0,5 \cdot i), \quad i=1, \dots, 10$$

и вывести вектор на печать. Вектор  $\mathbf{x}$  и заданный вектор  $\mathbf{y}$  преобразовать по правилу: большее из  $x_i, y_i$  принять в качестве нового значения  $x_i$ , а меньшее - в качестве нового значения  $y_i$ ,  $i=1, \dots, 10$ . Преобразованные векторы  $\mathbf{x}$  и  $\mathbf{y}$  вывести на печать.

Исходные данные:  $\mathbf{y}=(0,4; 0,1; 0,6; 1,2; 4,4; 0,2; -6,8; -1,4; 2,6; 1,5)$ .

Вариант 13. Задан вещественный вектор  $\mathbf{x}=(x_1, \dots, x_{10})$ . Компоненты вектора  $\mathbf{y}=(y_1, \dots, y_{10})$  вычислить по формуле:

$$y_i = 0,26 \cdot \ln(1 + \sqrt{i}).$$

Логической переменной  $\mathbf{a}$  присвоить значение TRUE, если

$$s = (x_1 \cdot y_{10})^2 + (x_2 \cdot y_9)^2 + \dots + (x_{10} \cdot y_1)^2$$

принадлежит отрезку  $[0; 1]$ , и значение FALSE - в противном случае.

Векторы  $\mathbf{x}$ ,  $\mathbf{y}$ , а также значения  $\mathbf{s}$  и  $\mathbf{a}$  вывести на печать.

Исходные данные:  $\mathbf{x}=(0,6; 1,8; 1,2; 0,4; -0,8; 2,6; 3,1; -0,4; -0,2; 1,3)$ .

Вариант 14. Задана матрица  $\mathbf{A}=(a_{ij})$ ,  $i, j=1, 2, \dots, n \leq 20$ . Переменной  $\mathbf{t}$  присвоить значение, равное скалярному произведению векторов  $\mathbf{x}$  и  $\mathbf{y}$ , компоненты которых определяются следующим образом:

$$x_i = \max_j(a_{ij}), \quad y_i = \min_k(a_{ki}), \quad i=1, 2, \dots, n.$$

Напечатать векторы  $\mathbf{x}$ ,  $\mathbf{y}$  и значение переменной  $\mathbf{t}$ .

$$-1,1 \quad -0,6 \quad 6,4$$

Исходные данные:  $n=3$ ;  $\mathbf{A} =$

$$0,4 \quad 9,4 \quad 1,2$$

$$-1,3 \quad 0,8 \quad 1,6$$

Вариант 15. Значения компонентов вектора  $\mathbf{x}=(x_1, \dots, x_{19})$  вычислить по формуле:

$$x_i = 1,2 \cdot \sin(0,1 \cdot i) + 0,4 \cdot \cos(i - 1), \quad i=1, \dots, 19.$$

и вывести на печать. Напечатать номер  $\mathbf{k}$  наибольшего компонента вектора  $\mathbf{x}$ . Все компоненты вектора  $\mathbf{x}$  с четными номерами, которые предшествуют  $X_k$ , домножить на  $X_k$ . Остальные компоненты оставить без изменения. Преобразованный вектор вывести на печать.

Вариант 16. Задан вектор  $\mathbf{x}=(x_1, \dots, x_n)$ ,  $n \leq 20$ . Если хотя бы один компонент этого вектора меньше  $\mathbf{t}$ , то все отрицательные компоненты заменить их квадратами, оставив остальные компоненты без изменения. В противном случае положительные компоненты вектора  $\mathbf{x}$  домножить на  $\mathbf{t}$ . Преобразованный вектор вывести на печать.

Исходные данные:  $n=9$ ;  $\mathbf{x}=( -1,6; 2,4; -0,8; 0,4; 0,2; -2,1; 6,4; 0,0; 0,1)$ ; 1)  $\mathbf{t}=-2$ ; 2)  $\mathbf{t}=-10$ .

Вариант 17. Задана вещественная матрица  $\mathbf{A}=(a_{ij})$ ,  $i, j=1, \dots, n \leq 5$ . Построить логический вектор  $\mathbf{b}=(b_1, \dots, b_n)$  по

следующему правилу: если в строке с номером  $k$  ( $k=1, \dots, n$ ) число положительных элементов больше числа отрицательных, то  $b_k$  присвоить значение TRUE, в противном случае - FALSE.

Исходные данные:  $n=4$ ,  $A=$

2,4	0,8	-0,6	0,2
1,4	1,6	-0,2	-0,4
-0,8	2,6	-6,4	-1,2
1,2	1,4	2,8	4,6

Вариант 18. Задана вещественная матрица  $A=(a_{ij})$ ,  $i, j=1, \dots, n$ ,  $n \leq 10$ . Построить вещественный вектор  $b=(b_1, \dots, b_n)$  по правилу: если  $a_{ii} < 0$  ( $i=1, \dots, n$ ), то в качестве  $b_i$  принять сумму элементов  $i$ -й строки, предшествующих элементу  $a_{ii}$ ; если  $a_{ii} \geq 0$ , то в качестве  $b_i$  принять сумму элементов  $i$ -й строки, следующих за  $a_{ii}$  (включая  $a_{ii}$ ).

Исходные данные:  $n=4$ ;  $A=$

0,8	0,4	6,2	2,6
-1,1	-0,4	0,2	3,4
1,2	-1,6	-1,8	4,4
4,8	1,4	2,9	0,6

Вариант 19. Вычислить значения компонентов вектора  $x = (x_1, \dots, x_n)$ ,  $n \leq 30$  по формуле:

$$\arctg \frac{\sqrt{i+2}}{n+3}, \quad \text{при } i \leq 13$$

$x_i =$

$$\cos(i \cdot e^{i+\cos n}), \quad \text{при } i > 13, i=1, \dots, n$$

Напечатать полученный вектор. Подсчитать и вывести на печать количество положительных и отрицательных компонентов вектора  $x$ .

Исходные данные:  $n=18$ .

Вариант 20. Заданы логический вектор  $a=(a_1, \dots, a_n)$ ,  $n \leq 20$  и вещественный вектор  $x=(x_1, \dots, x_n)$ . Преобразовать вектор  $x$  по правилу: если  $a_i$  имеет значение TRUE, то  $x_i$  умножить на 10, в противном случае изменить знак  $x_i$  на противоположный. Напечатать преобразованный вектор  $x$ .

Исходные данные:  $n=8$ ;  $a=(T, F, F, T, F, T, T, F)$  (T - TRUE, F - FALSE);  $x=(0,8; 4,1; -2,6; 0,1; 2,1; 8,9; -0,5; -8,1)$ .

Вариант 21. Значения компонентов векторов  $a=(a_1, \dots, a_{15})$  и  $b=(b_1, \dots, b_{15})$  вычислить по формулам:

$$a_i = \arctg \frac{i+2}{15}, \quad b_i = e^{i+\cos \sqrt{15}}$$

Полученные векторы вывести на печать. Из векторов  $\mathbf{a}$  и  $\mathbf{b}$  получить вектор  $\mathbf{c} = (a_1, b_1, a_2, b_2, \dots, a_{15}, b_{15})$  и напечатать его компоненты. Определить наибольший и наименьший компоненты вектора  $\mathbf{c}$ .

Вариант 22. Компоненты вектора  $\mathbf{a} = (a_1, \dots, a_{10})$  расположить в порядке возрастания по абсолютной величине. Получить вектор  $\mathbf{b} = (b_1, \dots, b_{10})$  путем исключения из вектора  $\mathbf{a}$   $k$ -го компонента и дополнения полученного вектора значением  $r$  так, чтобы компоненты вектора  $\mathbf{b}$  располагались в порядке возрастания по абсолютной величине.

Исходные данные  $\mathbf{a} = (0,8; 0,1; -0,4; 12,6; 5,2; -8,9; 9,1; -1,8; 4,6; 1,5)$ ;  $k=6$ ;  $r=-1,2$ .

Вариант 23. Вектор  $\mathbf{x} = (x_1, \dots, x_6)$  преобразовать по правилу: если  $x_1 \leq x_2 \leq \dots \leq x_6$ , то всем компонентам присвоить значение наибольшего из них; если  $x_1 > x_2 > \dots > x_6$ , то вектор оставить без изменения; в противном случае все компоненты возвести в квадрат. Компоненты вектора  $\mathbf{x}$  вычислить по формуле:

$$x_i = \arctg\left(\sqrt{i} + \frac{2}{9}\right), \quad i=1, \dots, 6.$$

Исходный и преобразованный векторы вывести на печать.

Вариант 24. Задана вещественная матрица  $\mathbf{A} = (a_{ij})$ ,  $i, j=1, \dots, n$ ,  $n \leq 15$ . Построить логический вектор  $\mathbf{l} = (l_1, \dots, l_n)$  по правилу: если  $i$ -я строка матрицы  $\mathbf{A}$  образует неубывающую последовательность, то  $l_i$  присвоить значение TRUE, в противном случае  $l_i$  - FALSE.

Исходные данные:  $n=4$ ,

$$\mathbf{A} = \begin{matrix} & \begin{matrix} 1,4 & 1,4 & 2,2 & 3,6 \end{matrix} \\ \begin{matrix} -0,1 & 0,2 & 8,4 & 1,2 \\ -2,4 & -1,2 & 0,0 & 1,1 \\ 8,6 & 2,2 & 4,1 & 1,8 \end{matrix} & \end{matrix}$$

## Литература

1. Абрамов В.Г., Трифонов Н.П., Трифонова Г.Н. Введение в язык Паскаль. - М.: Наука, 1988. - 224 с.
2. Воронич В.С., Вальвачев А.Н., Кузьмич А.И. Паскаль для персональных компьютеров. - Мн.: Выш.школа, 1991. - 365 с.
3. Вальвачев А.Н., Крисевич В.С. Программирование на языке Паскаль для персональных ЭВМ ЕС. - Мн.: Выш.школа, 1989. - 223 с.
4. Ошицков Д.В., Старых В.А. Программирование в интегрированной среде Турбо-Паскаль. - Мн.: Беларусь, 1992. - 240 с.
5. Саваньев А.А. и др. Электронные вычислительные машины // Кн.2. Основы информатики. - М.: Высш.школа, 1991. - 159 с.

Подписано к печати 4.06.2001 Лицензия №113 от 19.12.1997г.  
Формат 60x84/16. Объем 14 печ.л., 1,0 уч.-изд.л. Заказ 67. Тираж 300

Отпечатано на ризографе БЕЛМАПО.

г. Минск, ул. П.Бровки, 3.