

Министерство образования Российской Федерации
Владимирский Государственный Университет
Кафедра Информатики и Вычислительной Техники

ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ

Часть 1. Программирование на языке Паскаль.

Методические указания

Вид обучения : Заочное

Составитель Быков В.И.

Владимир 2001

УДК 681.3

Рецензент
доктор технических наук, профессор
Владимирского государственного университета
И.Е. Жигалов

Печатается по решению редакционно-издательского совета
Владимирского государственного университета

Пособие содержит рабочую программу и методические указания для выполнения контрольных и лабораторных работ по дисциплине «Программирование на языках высокого уровня» – часть 1 - «Программирование на языке Паскаль». Тематика работ соответствует программе дисциплины. Программа и лабораторные работы охватывают все разделы, связанные с программированием на ЯВУ: алгоритмизация задач, основные конструкции и сложные элементы языка программирования Паскаль.

В методических указаниях к лабораторным работам содержатся требования к оформлению отчета и таблицы номеров задач по указанному в литературе задачнику для 2 групп на каждую из 4 работ. В приложении приводятся примеры оформления для каждой лабораторной работы.

Методические указания предназначены для студентов специальностей 220100 – Вычислительные машины, комплексы, системы и сети и 071900 – Информационные системы в технике и технологиях заочной формы обучения, но могут быть полезны студентам других специальностей, изучающим язык Паскаль.

Табл. 1. Ил. 1. Библиогр. 14 назв.

УДК 681.3

ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ, ЕЕ МЕСТО В УЧЕБНОМ ПРОЦЕССЕ

Целью дисциплины является освоение студентами методики постановки, подготовки и решения инженерно-технических задач на современных ЭВМ, создание необходимой фундаментальной основы знаний, необходимой при программировании для вычислительных систем.

В результате изучения дисциплины студенты **ДОЛЖНЫ**:

1. **ИМЕТЬ ПРЕДСТАВЛЕНИЕ**:

- об использовании дополнительных пакетов и библиотек при программировании, о многообразии современных процедурных языков, их областях применения и особенностях.

2. **ЗНАТЬ И УМЕТЬ ИСПОЛЬЗОВАТЬ**:

- современные методы и средства разработки алгоритмов и программ, основные конструкции языка и способы записи алгоритма на языке высокого уровня Паскаль.

3. **ИМЕТЬ ОПЫТ**:

- составления, отладки, испытания и документирования программ с использованием процедурного языка;

- работы в интегрированных средах программирования и с использованием библиотек.

При изучении дисциплины в некоторой степени используются знания, полученные в школьном курсе "Информатика и вычислительная техника" и в дисциплине "Высшая математика".

Знания, полученные при изучении дисциплины, необходимы студентам для изучения дисциплин "Системное программное обеспечение", "Технология программирования", "Операционные системы" и в ряде других дисциплин, связанных с изучением программного обеспечения ЭВМ.

РАБОЧАЯ ПРОГРАММА

ОСНОВНОЕ СОДЕРЖАНИЕ КУРСА

Раздел 1. Введение.

Тема 1. 1. Цели и задачи дисциплины. Назначение ЭВМ. Современный компьютер: супер-, мини-, микро-ЭВМ, перспективы их развития. Использование ЭВМ в научной, инженерной и экономической областях. Основные сведения об операционной системе: состав ОС, функциональное назначение отдельных компонентов. Средства взаимодействия пользователя с ЭВМ. Этапы обработки программ пользователя.

Раздел 2. Алгоритмизация задач.

Тема 2. 1. Этапы решения задачи на ЭВМ. Понятие алгоритма. Определение и свойства алгоритма. Способы описания алгоритмов. Метод схем алгоритмов.

Принцип пошаговой детализации при проектировании алгоритма. Проверка правильности построения алгоритма.

Тема 2. 2. Разновидности структур алгоритмов: организация линейных, вычислительных процессов.

Тема 2. 3. Организация разветвляющихся вычислительных процессов. Приемы алгоритмизации типовых задач. Конструирование конкретных инженерно-технических задач из типовых компонентов (примеры).

Тема 2. 4. Организация циклических вычислительных процессов. Вложенные циклы. Назначение алгоритмического языка и требования к нему. Обзор языков программирования. Области применения различных языков.

Раздел 3. Основные элементы языка Паскаль.

Тема 3. 1. Алгоритмический язык программирования Паскаль. Основные понятия: символы, имена, переменные, константы. Структура программы. Разделы констант, типов, переменных, меток. Стандартные функции, выражения. Операции типа сложения и типа умножения. Оператор присваивания. Запись линейных программ.

Тема 3. 2. Операторы управления - выбора. Организация программ разветвляющейся структуры вычислительного процесса.

Тема 3. 3. Операторы цикла. Организация программ с циклической структурой вычислительного процесса.

Тема 3. 4. Массивы данных. Векторы. Обработка и организация ввода-вывода массивов - векторов.

Тема 3. 5. Двумерные массивы данных - матрицы. Обработка и организация ввода-вывода матриц. Типовые алгоритмы работы с матрицами.

Раздел 4. Сложные типы и элементы языка Паскаль.

Тема 4. 1. Программы, их классификация. Назначение, способы оформления подпрограмм. Глобальные и локальные переменные. Обращение к подпрограммам. Программирование с использованием процедур, функций, рекурсий. Использование внешних подпрограмм. Связь разноязыковых модулей.

Тема 4. 2. Файловые типы. Стандартные файлы ввода-вывода. Обработка файлов прямого и последовательного доступа с различными структурами данных. Обработка текстовых файлов.

Тема 4. 3. Записи. Оператор работы с записями. Программирование с использованием нестандартных скалярных типов данных.

Тема 4. 4. Понятие множества и действия с ними.

Тема 4. 5. Динамические типы данных. Ссылочный тип.

Раздел 5. Методы и алгоритмы решения задач поиска и сортировки.

Тема 5. 1. Методы и алгоритмы выбора – простого и с помощью дерева.

Тема 5. 2. Методы и алгоритмы простого и двоичного включения.

Тема 5. 3. Методы и алгоритмы простого обмена. Шейкер – сортировка. Быстрая сортировка Хоара.

ТЕМЫ ЛАБОРАТОРНЫХ И КОНТРОЛЬНЫХ РАБОТ

1. Разветвляющиеся и циклические вычислительные процессы. Одномерные массивы.
2. Матрицы и вложенные циклы. Процедуры и функции.
3. Записи и файлы.
4. Алгоритмы и программы сортировки.

ТЕХНИЧЕСКИЕ СРЕДСТВА КУРСА

Основными средствами при изучении курса является класс ПЭВМ, совместимых с IBM PC.

ПРИЛОЖЕНИЕ ПО СТАНДАРТИЗАЦИИ К РАБОЧЕЙ ПРОГРАММЕ ПО ДИСЦИПЛИНЕ "Программирование на языке высокого уровня"

1. ГОСТ 19.401-78 ЕСПД. Текст программы. Требования к содержанию и оформлению.
2. ГОСТ 19.101-77 ЕСПД. Виды программ и программных документов.
3. ГОСТ 19.701-90 (ИСО 5807-85) ЕСПД. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения.
4. ГОСТ 19.402-78 ЕСПД. Описание программ.
5. ГОСТ 19.505-79 ЕСПД. Руководство оператора. Требования к содержанию и оформлению.
6. ГОСТ 19.502-78 ЕСПД. Описание применения. Требования к содержанию и оформлению.
7. ГОСТ 19.001-77 ЕСПД. Общие положения.
8. ГОСТ 19.103-77 ЕСПД. Обозначения программ и программных документов.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К КОНТРОЛЬНЫМ И ЛАБОРАТОРНЫМ РАБОТАМ

Работа N1

Разветвляющиеся и циклические вычислительные процессы. Одномерные массивы.

1. Цель и содержание работы.

Изучение структуры алгоритмов и программирование задач, связанных с обработкой одномерных массивов и содержащих выбор и итерации.

2. Задания по подготовке к работе.

2.1. В соответствии с вариантом задания выбрать метод и разработать обобщенный алгоритм решения задачи в виде словесного описания или схемы.

2.2. Написать программу на языке Паскаль [1.6]

2.3. Составить и записать 3-4 тестовых примера для изучения прохождения различных ветвей алгоритма (программы). Тестовый пример должен содержать требуемый набор исходных данных и ожидаемые результаты.

3. Задание по работе на ПЭВМ.

3.1. Изучить требования техники безопасности и противопожарной техники при работе в дисплейном классе.

3.2. Ознакомиться с клавиатурой ПЭВМ.

3.3. Изучить и освоить правила работы в интегрированной среде системы Турбо-Паскаль. [7].

3.4. Набрать текст программы в среде языка Паскаль.

3.5. Транслировать программу, исправляя при этом обнаруживаемые синтаксические ошибки.

3.6. После успешного завершения трансляции и редактирования ввести исходные данные первого тестового примера и сверить полученные результаты с ожидаемыми.

3.7. Прodelать п. 3.6. для 2-го и следующих тестовых примеров.

3.8. В случае несовпадения найти ошибку в алгоритме, исправить ее и повторить решение.

4. Методические указания.

4.1. При программировании задач, связанных с обработкой массивов необходимо предусмотреть именованную константу для описания размерности массива. Величина константы должна быть не менее 20.

4.2. Фактический рабочий размер массива должен определяться переменной, задаваемой в качестве исходных данных.

4.3. Программа должна соответствовать требованиям структурного программирования. [8].

4.4. Для работы в интегрированной среде системы Turbo-Pascal необходимо [7]:

1). Войти в сеть, для чего необходимо в ответ на приглашение F:/LOGIN набрать login сетевое имя. В ответ на запрос пароля необходимо ввести с клавиатуры пароль, который не отображается на экране. Сетевое имя и пароль индивидуальны для каждой группы к определяющиеся преподавателем.

2). Войти в Norton Commander, для чего набрать nc.

- 3). Создать на рабочем диске группы свой каталог.
- 4). Войти в свой каталог.
- 5). Запустить Pascal, для чего войти в меню пользователя (кл.F2).
- 6). Выбрать в меню Pascal (кл.F10) пунктFile, а в нем требуемое подменю: New - для набора нов. программы; Open – для вызова имеющейся программы.
- 7). Набранную программу или внесенные изменения необходимо сохранить (кл. F2) в файле, имя которого необходимо ввести, если система его потребует.
- 8). Для выполнения трансляции, редактирования и вызов программы на решение требуется нажать кл. Ctrl F9.
- 9). Для просмотра результатов решения необходимо кл.Alt F5.
- 10). Для завершения работы и возврата в DOS - кл. Alt-x или File/Guit.
- 11). Для распечатки программы на примере необходимо выйти в DOS (п.10) и обратиться к принтеру командой `prn <номер аудитории>` (например `prn 425`). Далее необходимо выбрать требуемый файл в Norton и скопировать (кл. F5) его на принтер, для чего набрать в появившемся окне имя PRN.

5. Содержание отчета.

- 5.1. Титульный лист
- 5.2. Цель работы
- 5.3. Текст задачи
- 5.4. Метод и алгоритм
- 5.5. Текст программы
- 5.6. Краткая инструкция к программе, в которой должны быть определены порядок, вид, диапазон значений вводимых исходных данных и порядок расположения выводимых результатов.
- 5.7. Тестовые примеры и результаты их решений.
- 5.8. Спецификации процедур и функций.
- 5.9. Выводы, в которых должны быть указаны: соответствие программы требованиям задачи; краткие характеристики программы, прежде всего ее количественные ограничения (например, максимальный размер обрабатываемого массива), а также объем программы в виде количества операторов и объема EXE-файла и время выполнения программы.

Работа N2

Матрицы и вложенные циклы. Процедуры и функции.

1. Цель и содержание работы.

Изучение структуры алгоритмов и программирование задач, связанных с обработкой матриц. Построение и разработка модульных программ, программирование процедур и функций.

2. Задания по подготовке к работе.

2.1. В соответствии с вариантом задания выбрать метод и разработать обобщенный алгоритм решения задачи в виде словесного описания и схемы.

2.2. Выделить части задачи и алгоритма, предполагаемые к реализации в виде процедур и функций.

2.3. Написать программу на языке Паскаль [1.6]

2.4. Составить и записать 3-4 тестовых примера для изучения прохождения различных ветвей алгоритма (программы). Тестовый пример должен содержать требуемый набор исходных данных и ожидаемые результаты.

3. Задание по работе на ПЭВМ.
аналогично п.3 ЛР-1.

4. Методические указания.

4.1 Для выполнения задачи обработки записей использовать не файл (как указано в условиях задачи), а массив записей, учитывая при этом рекомендации п.4.1,4.2 ЛР-1.

4.2 Процесс ввода информации о записи оформить в виде процедуры или функции, результатом работы которой должна быть запись с заполненными полями. Эта функция будет необходима и для ЛР-4.

4.3 В процедурах и функциях передача данных и возвращение и результатов должны быть организованы через формальные параметры. Запрещается использование глобальных переменных и др. объектов программы за исключением некоторых констант.

4.4 Программа должна удовлетворять п.4.3 ЛР-1.

4.5 Фактическое кол-во записей в массиве должно быть не менее 7.

5. Содержание отчета должно соответствовать требованиям п.5 ЛР-1.

5.8 Спецификации процедур и функций включают следующие пункты:

- 1) Назначение (решаемая задача);
- 2) Заголовок и список формальных параметров;
- 3) Описание формальных параметров;
- 4) Используемый метод или его краткое описание (не более двух предложений);
- 5) Используемые процедуры и функции;

Работа N3

Файлы и записи.

1. Цель и содержание работы.

Изучение описания, ввода, вывода записей и работы с ними. Изучение описания файла, его связывания с программой, ввода и вывода информации из файла и других действий с ними.

2 Задание по подготовке к работе- аналогично п.2 ЛР-1.

3.Задание по работе на ПЭВМ - аналогично п.3 ЛР-1.

4. Методические указания.

4.1 Для выполнения задачи обработки записей использовать типизированный файл в соответствии с условиями задачи, размещенный в рабочей директории.

4.2 Процесс ввода информации о записи оформить в виде процедуры или функции, результатом работы которой должна быть запись с заполненными полями. Эта подпрограмма может быть получена путем переработки соответствующей подпрограммы из ЛР-3.

4.3 Необходимо использовать различные способы обращения к полям записи, в т.ч. и через оператор With.

4.4 Программа д. удовлетворять п.4.3 ЛР-1.

4.5 Фактическое кол-во записей в массиве должно быть не менее 7.

5. Содержание отчета должно соответствовать, требованиям п.5 ЛР-1.

Работа N 4.

Алгоритмы и программы сортировки .

1. Цель и содержание работы.

Изучение методов, алгоритмов, процедур и программ поиска и сортировки массивов. Разработка алгоритма и написание процедуры и программы для сортировки массива заданным методом..

2 Задание по подготовке к работе.

- аналогично п.2 ЛР-1.

3.Задание по работе на ПЭВМ.

- аналогично п.3 ЛР-1.

ж4 Методические указания..

.При разработке алгоритма и написании процедуры в ней необходимо ввести операторы, осуществляющие подсчет и вывод количества сравнений и перестановок (отдельно), произведенных. в процессе сортировки

5. Содержание отчета

- должно соответствовать требованиям п.5 ЛР-1.

ПРИМЕРЫ ВЫПОЛНЕНИЯ И ОФОРМЛЕНИЯ
КОНТРОЛЬНЫХ И ЛАБОРАТОРНЫХ РАБОТ

Оформление титульного листа (целый отдельный лист):

Министерство образования Российской Федерации

Владимирский Государственный Университет

Кафедра Информатики и Вычислительной Техники

КОНТРОЛЬНЫЕ И ЛАБОРАТОРНЫЕ РАБОТЫ

по дисциплине:

"Программирование на языке высокого уровня" (часть 1)

Выполнил
студент гр. ЗЭВМ-100
ИВАНОВ В.А

Принял
доцент БЫКОВ В.И.

Лабораторная работа № 1

Разветвляющиеся и циклические вычислительные процессы. Одномерные массивы.

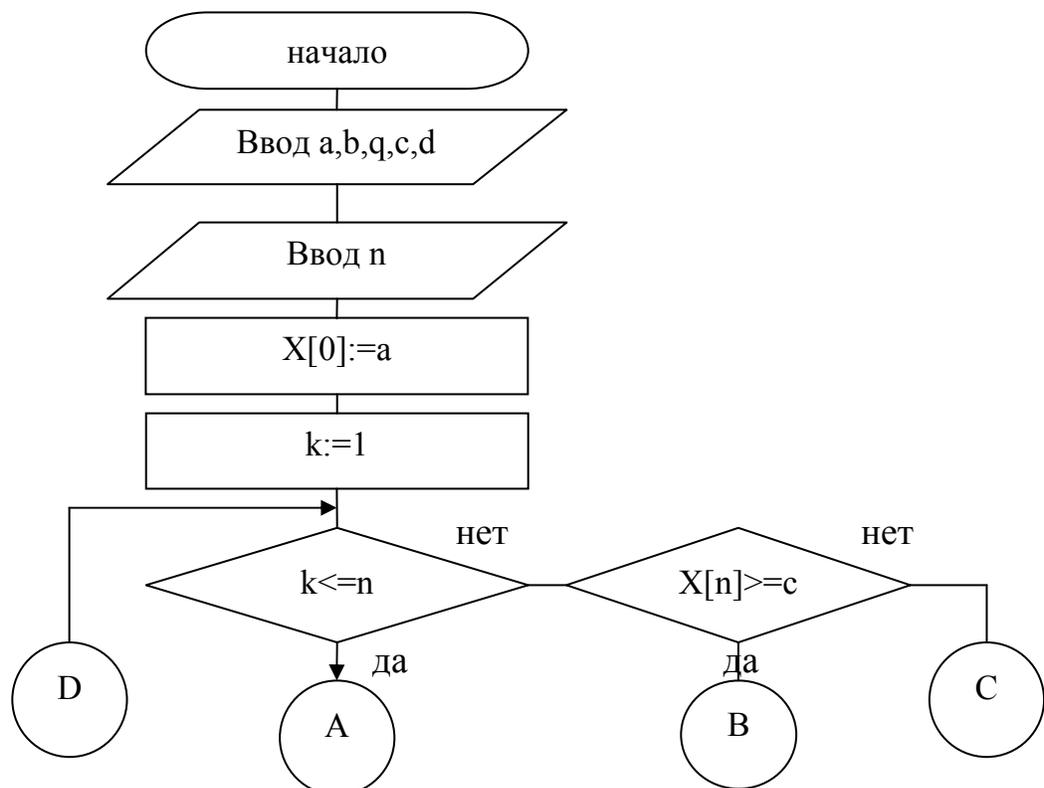
Цель работы : Изучение структуры алгоритмов и программирование задач, связанных с обработкой одномерных массивов и содержащих выбор и итерации.

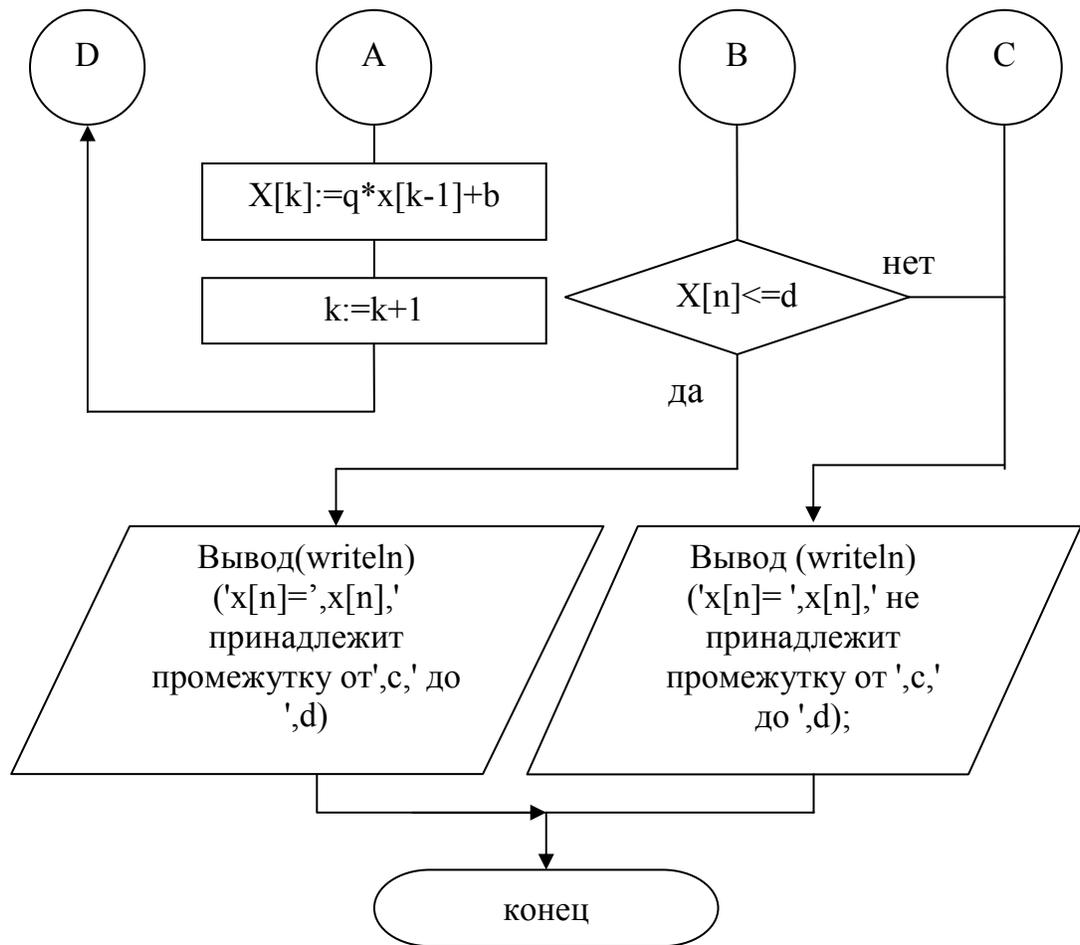
Условие задачи : Пусть $x(0) = a$; $x(k) = q \cdot x(k-1) + b$, ($k=1,2,3, \dots$) Даны неотрицательное целое число n и действительные a, b, c, d, q ($c < d$). Принадлежит ли $x(n)$ промежутку (c, d) .

Метод решения:

В цикле от 0 до n по итерационной формуле производится расчет $x(n)$ и полученное значение сравнивается с заданными границами интервала.

Схема алгоритма :





Текст основной программы:

```

program n1;
const
  m=100;
type
  aaa=array[0..m] of real;
var
  x      :aaa;
  a,b,c,d,q:real;
  k,n :integer;
begin
  writeln('Введите a-значение x[0],b и q-для нахождения x(k),c и d-
  границы интервала.');
```

```

readln(a,b,q,c,d);
{Расчет значения элемента вектора X с номером n}
writeln('Введите n');
readln(n);
x[0]:=a;
for k:=1 to n do
  x[k]:=q*x[k-1]+b;
  {Проверка принадлежности X[n] заданному интервалу}
  if (x[n]>=c) and (x[n]<=d) then
    writeln
      (x[n]=',x[n],' принадлежит интервалу от',c:9:3, ' до ',d:9:3)
  else
    writeln
      (x[n]=',x[n],' не принадлежит интервалу от ',c:9:3,' до ',d:9:3);
readkey;
end.

```

Тестовые примеры :

1)

a=1 b=1 c=0 d=33 q=0 n=3
 x[n]= 1 принадлежит интервалу от 0 до 33

2)

a=5 b=6 c=2 d=35 q=0 n=7
 x[n]= 6 принадлежит интервалу от 2 до 35

3)

a=2 b=3 c=9 d=93 q=8 n=6
 x[n]= -18725 не принадлежит интервалу от 9 до 93

Инструкция к программе:

Для её запуска необходимо запустить файл n1.exe и далее следовать по инструкциям, которые появятся по ходу работы программы. Входными данными являются числа a,b,q ,а также границы отрезка c и d (c<d). Выходными данными является сообщение о том, что входит или не входит высчитанное число X в отрезок от c до d.

Вывод : Разработаны алгоритм и программа определения принадлежности полученного элемента вектора с заданным номером заданному интервалу. Характеристики полученной программы:

Количество операторов: 17
Объем pas файла: 2К
Объем exe файла: 1.5К
Время работы при размерности: $n = 20$ менее 1 с

Лабораторная работа № 2

Матрицы и вложенные циклы. Процедуры и функции.

Цель: Изучение структуры алгоритмов и программирование задач, связанных с обработкой матриц. Построение и разработка модульных программ, программирование процедур и функций.

Условие задачи: Дана действительная матрица размера $m \times n$. Определить числа b_1, \dots, b_m , равные соответственно произведениям элементов строк.

Алгоритм :

А) Словесное описание

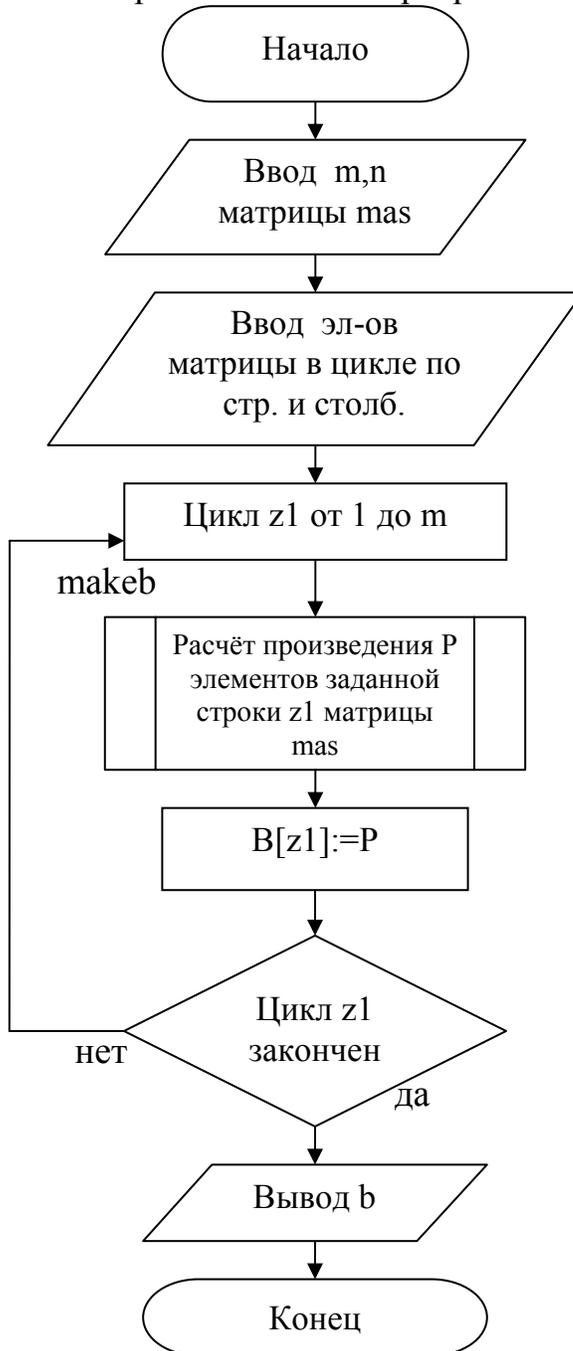
1. Ввод размеров матрицы - m и n
2. Ввод элементов матрицы в цикле по строкам и столбцам
 - 2.1. Цикл по строкам - i присваиваем значение 1
 - 2.2. Сравниваем $i \leq n$ если да, то переход к п.2.1.1, иначе переход к п, 3.
 - 2.1.1. Цикл по столбцам - j присваиваем значение 1
 - 2.1.2. Сравниваем $j \leq m$ если да, то переход к п.2.1.3, иначе переход к п, 2.2.
 - 2.1.3.. Вводим $A[i;j]$
 - 2.1.4. Увеличение номера столбца $j:=j+1$
 - 2.1.5. Возврат к пункту 2.1.2.
 - 2.3. Увеличение номера строки $i:=i+1$
 - 2.4. Возврат к пункту 2.2.
3. Расчет суммы элементов каждой строки, для чего организуется цикл по строкам –
 - 3.1. i присваиваем значение 1
 - 3.2. Сравниваем $i \leq n$ если да, то переход к п.3.3, иначе переход к п, 4.
 - 3.3. $b(i)=0$
 - 3.4. Расчет суммы элементов строки i , для чего производится обращение к функции Sumstr и присвоение $b[i]$ полученного значения.

3.5.Переход к следующей строке $i=i+1$ и возврат к п.3.2.

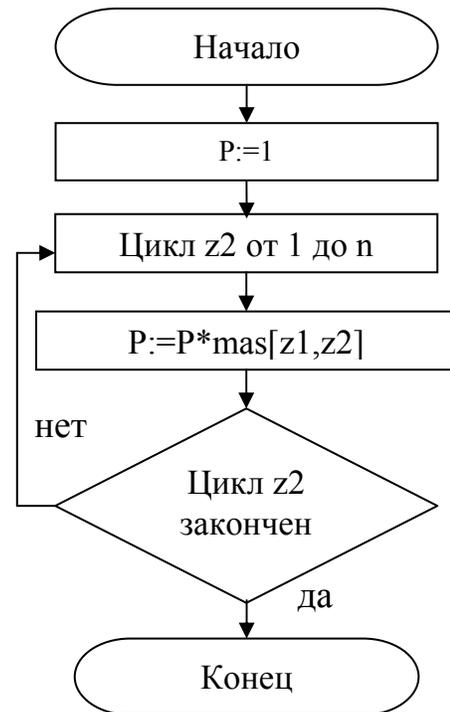
4. Вывод вектора b в цикле.

5. Конец

Алгоритм основной программы



Алгоритм процедуры makeb



Текст основной программы:

=====текст программы=====

```
program n2;
```

```
uses crt;
```

```
const maxmas=20;
```

```
type matrix = array[1..maxmas,1..maxmas]of real;  
      vector = array[1..maxmas]of real;
```

```
var m,n : byte;  
    mas : matrix;  
    b   : vector;  
    c   : char;
```

```
procedure genmatr(var n,m:integer;var matr:matrix);
```

```
var x : real;
```

```
begin
```

```
  writeln('Введите кол-во строк матрицы m и столбцов n');
```

```
  read(m,n);
```

```
  randomize;
```

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

```
    begin
```

```
      writeln('Строка N= ',i);
```

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

```
        begin
```

```
          x := random;
```

```
          matr[i,j]:=x*100-50;
```

```
          write(matr[i,j]:9:3);
```

```
        end; {for j}
```

```
      writeln;
```

```
    end; {for i}
```

```
end; {proc. genmatr}
```

```
procedure inmatr(var n,m:integer;var matr:matrix);
```

```
var z1,z2 : integer;
```

```

begin
  writeln('Введите m: ');readln(m); {ввод m и n}
  writeln('Введите n: ');readln(n);
  writeln('Введите матрицу построчно, разделяя значения пробелом: ');
  for z1:=1 to m do          {ввод}
    begin
      writeln(z1,': ');
      for z2:=1 to n do
        read(matr[z1,z2]);
      end;  {for z1}      {матрицы}
    end;
  end; {proc. inmatr}

```

```

function makeb(L,n:byte;var mas: matrix):real;
{ Расчёт произведения P элементов заданной строки L матрицы mas размера
n}

```

```

var z : byte;
    P : real;

```

```

begin
  P:=1; {первоначальное значение}
  for z := 1 to n do {цикл от 1 до конца строки}
    P:=P*mas[L,z]; {подсчитываем b в итоговую переменную}
  makeb:=P; {присваиваем итог функции}
end; {func.makeb}

```

```

begin
{Текст основной программы}
repeat
  clrscr;
  writeln('Произвести произвольный ввод матрицы(Y/N)?');
  c := readkey;
  if (c='n') or (c='N') then inmatr(n,m,mas);
  else genmatr(n,m,mas);
  writeln('Значения b: ');
  for z1:=1 to m do {цикл от 1 до m}
    b[z1]:=makeb(z1,n,mas); {вычисление b}
  for z1:=1 to m do {цикл от 1 до m}
    write(b[z1]:9:3, ' '); {печать b}
  writeln('Повторить расчёт (Y/N)?');

```

```
c := readkey;  
until (c='n') or (c='N');  
end.
```

=====конец текста программы=====

Тестовые примеры.

1. Введите m: 3

Введите n: 3

Введите матрицу построчно, разделяя значения пробелом:

1: 2 1 1

2: 2 3 1

3: 1 4 2

Значения b:

2.000 6.000 8.000

Повторить расчёт (Y/N)?n

2. Введите m: 2

Введите n: 4

Введите матрицу построчно, разделяя значения пробелом:

1: 9 1 0 4

2: 2 6 1 -1

Значения b:

0.000 6.000 -12.000

Повторить расчёт (Y/N)?n

Описание процедур и функций:

В функции makeb производится расчёт произведения P элементов заданной строки L матрицы mas размера n. Входными для неё данными являются L – номер строки, n – длина строки, а также mas – матрица. В процедурах genmatr и inmatr производится ввод размеров матрицы m и n, а также элементов матрицы, в первой процедуре автоматически, во второй с клавиатуры.

Инструкция к программе:

Для её запуска необходимо запустить файл n2.exe и далее следовать по инструкциям, которые появятся по ходу работы программы. Входными данными являются количество строк m и количество столбцов n с ограничением не более 20. Затем вводятся действительные элементы

матрицы. Выходными данными являются числа соответствующие произведениям строк.

Вывод: Разработаны алгоритм и программа, осуществляющие расчёт произведений элементов строк матрицы.

Характеристики полученной программы:

Количество операторов: 45

Объем рас файла: ~6К

Объем exe файла: ~4К

Время работы при размерности матрицы: 4X4 менее 1 с

Лабораторная работа № 3

Файлы и записи.

Цель: Изучение описания, ввода, вывода записей и работы с ними. Изучение описания файла, его связывания с программой, ввода и вывода информации из файла и других действий с ними.

Условие задачи: Дан файл, который содержит следующие сведения о студентах: № по списку, Ф.И.О., массив оценок в сессии по 5 и пустое символьное поле для отметки о назначении стипендии и её размерах. Проанализировать оценки каждого студента и заслать сведения о назначении стипендии и её размерах в нужное поле в следующем виде:

V – повышенная стипендия (+50% при всех 5);

S – обычная стипендия (при всех 4 и 5);

P – повышенная стипендия (+25% при двух 4 и остальных 5);

N – нет стипендии (при наличии хотя бы одной 3);

O – отчислить (при трёх 2).

Алгоритм:

Схема алгоритма главной программы приведена на рис.1

Алгоритмы подпрограмм очевидны из комментариев к программе.

Рис.1

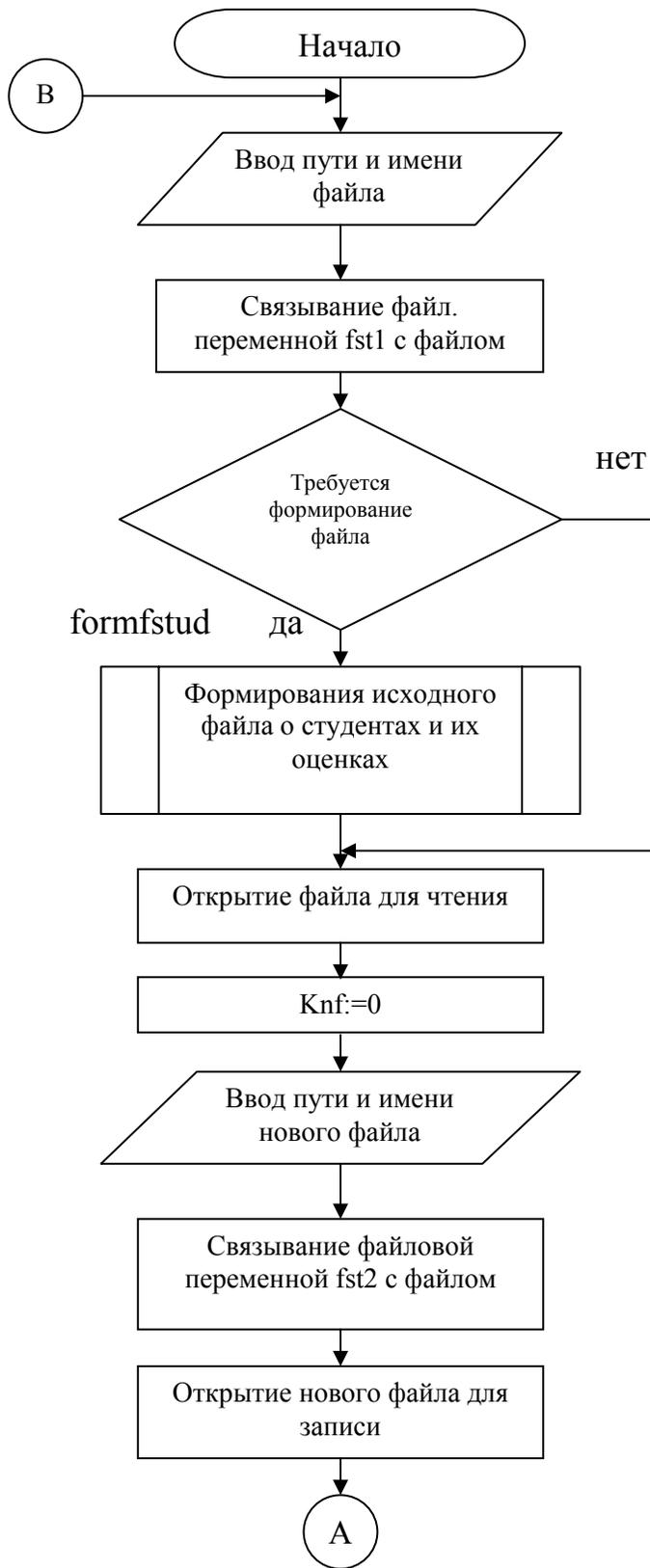
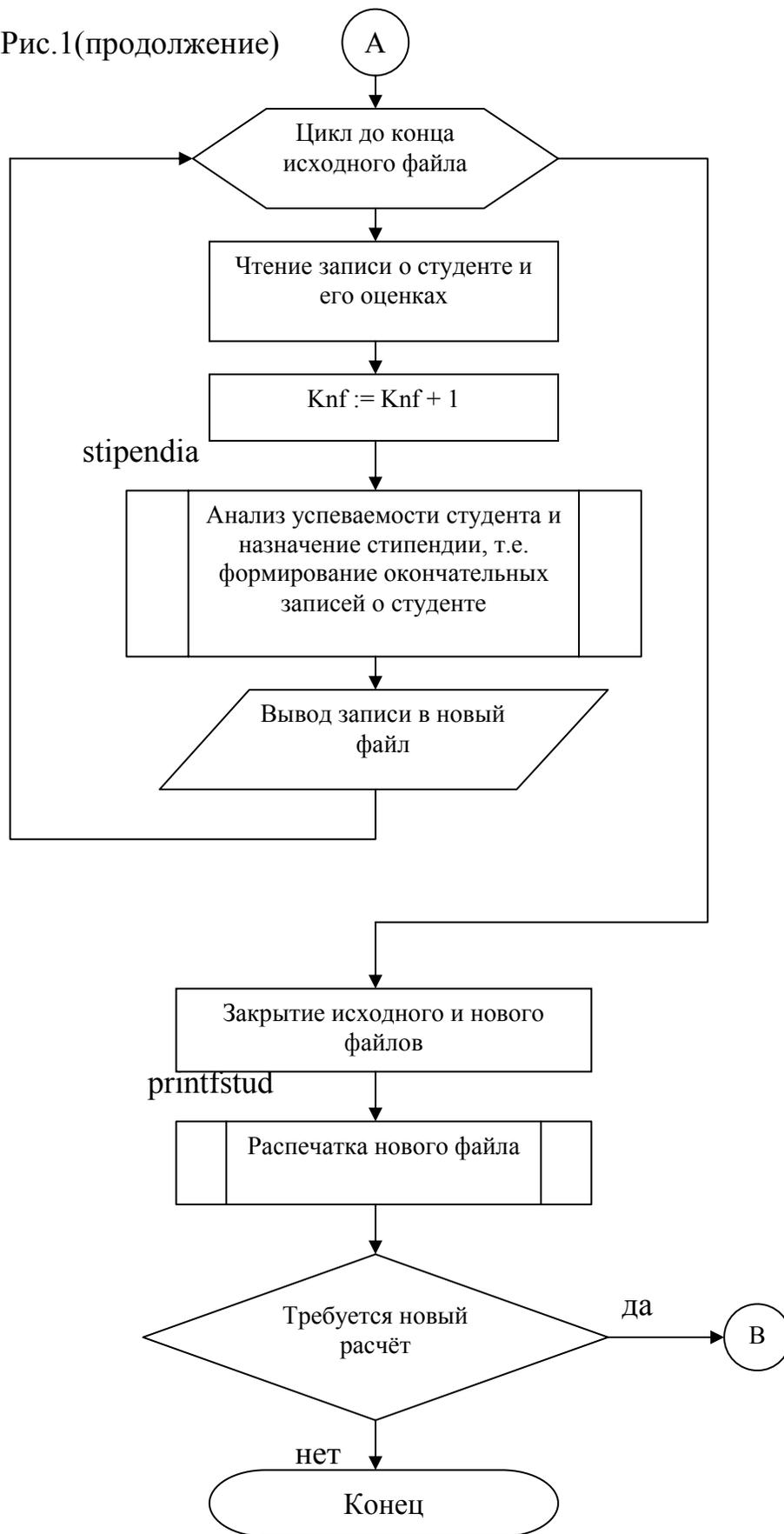


Рис.1(продолжение)



Текст основной программы:

```
Program studstip;  
{программа анализа успеваемости студентов и назначения стипендии}  
{автор Быков В.И. дата 10.04.01}  
Uses crt,dos;  
Const ko =5;  
Type stud = record  
    nom : integer;  
    Fio : string[20];  
    Bal : array[1..ko] of integer;  
    Stip : char;  
End;  
fstud = title of stud;  
  
Var  i,j,knf  : integer;  
    Fst1,fst2 : fstud;  
    Str1,str2 : string; {строки, определяющие пути к файлам – иск. и нов.}  
    Sqt,tmax,tmin,tn : real;  
    Rst : stud;  
    Ch : char;  
    str : char; {назначенная стипендия}  
  
Procedure formfstud(var fst:fstud);  
{формирование файла о студентах}  
var  i,j,kst : integer;  
    rst  : stud;  
begin  
    rewrite(fst);  
    ch:='y';  
    while (ch<>'n') and (ch<>'t') do {цикл ввода по принципу пока ненадоест}  
        begin  
{ввод данных о ст-те в запись}  
            writeln('Введите № ст-та, ФИО и 5 оценок от 2 до 5');  
            with rst do  
                begin  
                    read(nom);  
                    readln(fio);  
                    for j:=1 to ko do  
                        read(ball[i]);  
                    end; {with}  
            write(fst,rst); {занесение записи в файл}
```

```

        writeln('будете ли вы вводить новые данные y/n');
        ch:=readkey;
    end; {while}
    close (fst);
    {Вывод результатов}
end; {proc.formfstud}

```

```

procedure printfstud(var fst : tstud);
{распечатка файла о студентах}
var    i,knf : integer;
        rst : stud;
begin
    reset(fst);
    knf:=0;
    while not eof (fst) do
        begin
            read(fst,rst); {чтение записи из файла}
            knf:=knf+1;
        {Вывод (распечатка) записи}
            writeln;
            with rst do
                begin
                    write ('ст-т ',nom:2,' ',fio);
                    for i :=1 to ko do
                        write(bal[i]:3);
                    write(stip:3);
                end; {with}
            end; { while}
        close (fst);
        {Вывод результатов}
        writeln;
        writeln('кол-во ст-тов knt= ',knt:5);
    end; {proc. printfstud}

```

```

procedure stipendia(var rst : stud;var stp : char);
{Анализ успеваемости студента и назначение стипендии}
var    i,j : integer;
        kbal : array [2..5] of integer; {массив кол-ва оценок каждого вида}
begin
    {обнуление кол-ва оценок}
    for j:= 2 to 5 do

```

```

    kbal[j]:=0;
{анализ записи}
    with rst do
        begin
            for i:=1 to ko do
{подсчёт кол-ва оценок различного вида}
                kbal[bal[i]:=kbal[bal[i]]+1;
{анализ кол-ва оценок различного типа}
                if kbal[5]=5 then
                    stp:='V'
                else
                    if(kbal[5]>1) and ((kbal[5]+kbal[4])=5) then
                        stp:='P'
                    else
                        if ((kbal[3]+kbal[2])=0) then
                            stp:='S'
                        else
                            if kbal[2]>2 then
                                stp:='O'
                            else
                                stp:='N';
            end; {with}
        end; {proc. stipendia}

```

```

begin
    write('Введите путь и имя исх. Файла :');
    readln(str1);
    {str1:='d:\vib\fdst1';}    {для отладки}
    assign (fst1,str1); {связывание файловой переменной с файлом}
    writeln('требуется формирования файла(Y\N)?');
    readln(ch);
    if (ch='Y') or (ch='y') then
        formfstud(fst1);
{анализ каждого студента и назначении стипендии}
    reset (fst1); {открытие исходного файла для чтения}
    knt:=0; {кол-во записей в исходном файле}
    write('Введите путь и имя нового файла : ');
    readln(str2);
    {str2:='d:\vib\fdst2';}    {для отладки}
    assign (fst2,str2); {связывание файловой переменной с новым файлом}
    rewrite(fst2);

```

```

{цикл для чтения записей из исх.файла, их анализа, заполнения поля (графы)
о стипендии и запись в новый файл}
while not eof(fst1) do
begin
    read (fst1,rst); {чтение записи о студенте}
    knt:=knt+1;
{анализ успеваемости студента}
    stipendia(rst,stp);
    write (fst2,rst); {запись в новый файл}
{подсчёт кол-ва стипендий каждого вида}
end; {while}
close(fst1); {закрытие исх.файла}
close(fst2); {закрытие нов.файла}
{распечатка нов.файла о студентах}
printfstud(fst2);
ch:=readkey;
end.

```

Тестовые примеры:

Исходные данные:

```

1 Анисимов С.А.
  3 4 3 5 4
2 Бобров В.И.
  4 5 4 5 4
>3 Герасимов А.Н. <
  5 5 5 5 5
4 Иванов В.И.
  2 3 3 4 3
5 Кубатко И.В.
  4 5 5 5 4

```

Файл после анализа:

```

1 Анисимов С.А.
  3 4 3 5 4 N
2 Бобров В.И.
  4 5 4 5 4 S
>3 Герасимов А.Н. <
  5 5 5 5 5 V
4 Иванов В.И.
  2 3 3 4 3 O
5 Кубатко И.В.
  4 5 5 5 4 P

```

Описание процедур и функций:

Процедура Formfstud(var fst:fstud): формирование файла о студентах;
входные данные:

файловая переменная fst, которая связана с исходным файлом;

выходные:

файл заполненный сведениями о студентах, введенными с клавиатуры.

Процедура printfstud(var fst : tstud): распечатка файла о студентах;
входные данные:

файловая переменная fst2, которая связана с файлом о студентах;

выходные:

вывод на экран содержимого файла о студентах.

Процедура stipendia(var rst : stud; var stp : char): анализ успеваемости студента и назначение стипендии:

входные данные:

rst - запись о студенте;

выходные:

stp – символ определяющий назначенную стипендию в соответствии с условием задачи.

Инструкция к программе:

Для её запуска необходимо запустить файл n3.exe и далее следовать по инструкциям, которые появятся по ходу работы программы. Входными данными являются пути и имена входного и выходного файлов и в случае необходимости формирование нового файла с записями, путём ввода их с клавиатуры. А также, данные из входного файла. Выходными данными являются данные о студентах из исходного файла и оценки о стипендиях.

Вывод: Изучение описания, ввода, вывода записей, описания файла, его связывания с программой, ввода и вывода информации из файла.

Характеристики полученной программы:

Количество операторов: 90

Объем pas файла: ~9К

Объем exe файла: ~6К

Время работы при размерности: 5 записей в файле менее 1 с.

Лабораторная работа № 4

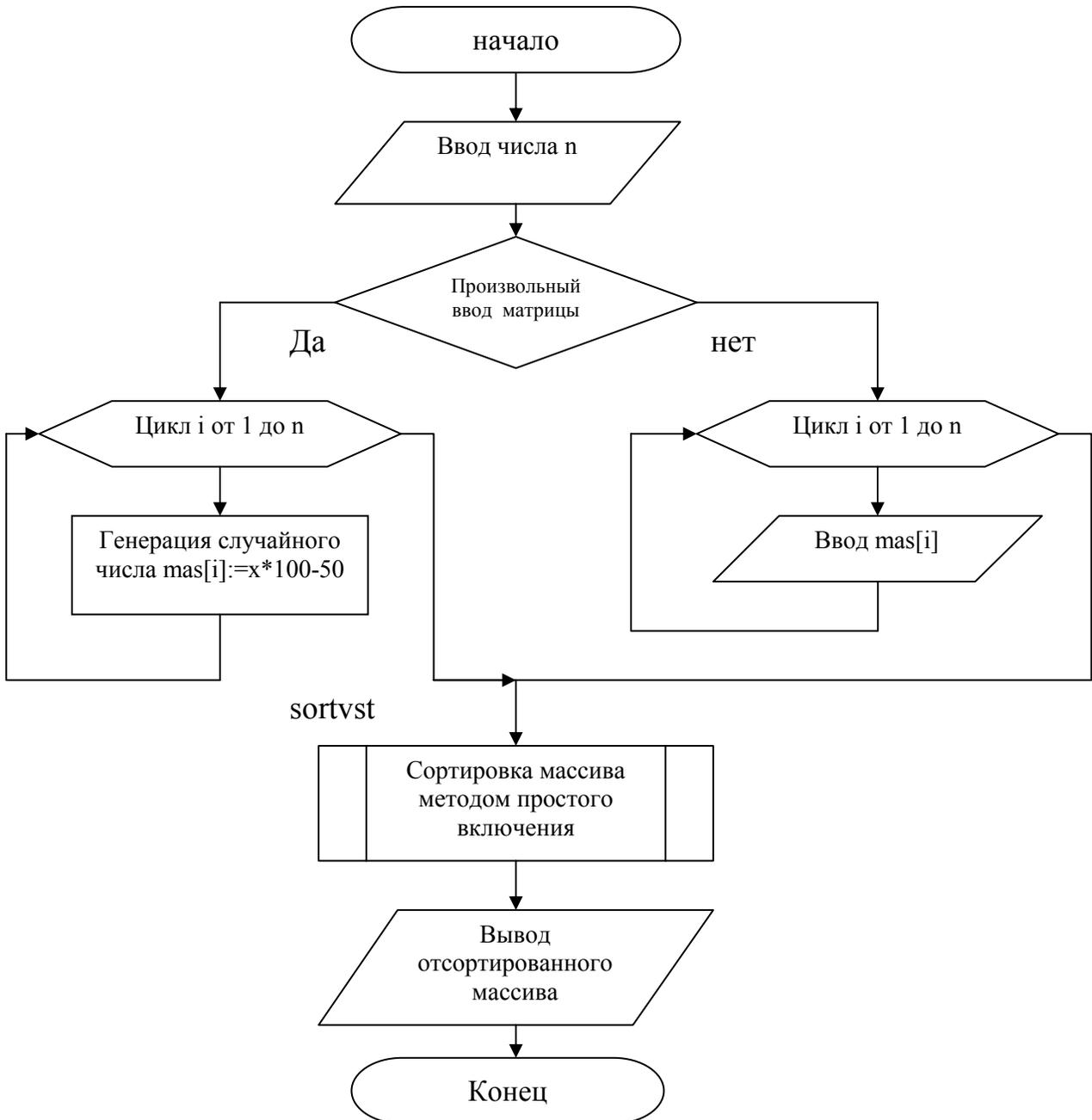
Алгоритмы и программы сортировки массива.

Цель: Изучение структуры алгоритмов и программирование задач, связанных с сортировкой массивов.

Условие задачи: выполнить сортировку массива действительных чисел заданного размера методом простого включения.

Метод решения: сортировка производится путём включения элемента в соответствующее место ранее отсортированного массива, находящегося в начале заданного массива.

Алгоритм основной программы :



Текст основной программы:

Program lab4;

Uses crt;

Const

 m=50;

Type

 vector = array[1..m] of real;

Var mas : vector;

 i,n : integer;

Procedure sortvst(n : integer; var mas: vector);

{Процедура сортировки массива mas размера n простым включением}

var b,x : real;

 i,j,k : integer;

begin

 for i:= 2 to n do

 begin

 b:= mas[i];

 j:= 1;

 while (b > mas[j]) do

 j:= j+1;

 for k:= (i -1) downto j do

 mas[k +1]:= mas[k];

 mas[j]:= b;

 end; {for i}

end; {proc.sortvst}

Begin {текст основной программы}

 Clrscr;

 writeln('Введите n: ');readln(n);

 writeln('Произвести произвольный ввод массива(Y/N)?');

 c := readkey;

 if (c='n') or (c='N') then

 begin

 writeln('Введите массив : ');

 for i:=1 to n do {ввод массива вручную}

 begin

 write(i,' ');

 read(mas[i]);

 end; {for i}

 end

```

else
begin
    Randomize;
    For i:= 1 to n do
    Begin
        X:= random(99);
        mas[i] := x*100 + 50; {формирование произвольного
массива}
    end;
    End;
    Writeln('Не отсортированный массив ');
    For i:= 1 to n do
        Write(mas[i]:9:3,' '); {вывод не отсортированного массива }
    Writeln;
    Sortvkl(n, mas);
    Writeln('Отсортированный массив ');
    For i:= 1 to n do
        Write(mas[i]:9:3,' '); {вывод отсортированного массива }
    End.

```

Тестовые примеры:

Введите n: 4

Произвести произвольный ввод массива(Y/N)?n

Введите массив :

1: 2

2: 4

3: 1

4: 8

Не отсортированный массив

2.000 4.000 1.000 8.000

Отсортированный массив

1.000 2.000 4.000 8.000

Введите n: 3

Произвести произвольный ввод массива(Y/N)?y

Не отсортированный массив

2.454 4.122 7.435 8.765

Отсортированный массив

2.454 4.122 7.435 8.765

Введите n: 6

Произвести произвольный ввод массива(Y/N)?n

Введите массив :

1: 7

2: 6

3: 5

4: 4

5: 3

6: 2

Не отсортированный массив

7.000 6.000 5.000 4.000 3.000 2.000

Отсортированный массив

2.000 3.000 4.000 5.000 6.000 7.000

Инструкция к программе:

Для её запуска необходимо запустить файл n4.exe и далее следовать по инструкциям, которые появятся по ходу работы программы. Входными данными являются количество элементов массива n. Затем вводятся действительные элементы массива. Выходными данными является упорядоченный массив путём по- элементного вывода.

Вывод: Разработаны алгоритм и программа, осуществляющие сортировку массива действительных чисел заданного размера методом простого включения. Характеристики полученной программы:

Количество операторов: 35

Объем pas файла:~7К

Объем exe файла:~5К

Время работы при размерности: 6 элементов менее 1 с.

Приложение 2

ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ К КОНТРОЛЬНЫМ И ЛАБОРАТОРНЫМ РАБОТАМ

Таблица 1

Номер студента по журналу	Номер лабораторной работы, шифр группы, номер задачи по [13]							
	Работа № 1		Работа №2		Работа №3		Работа №4	
	ЗЭВМ-1	ЗЭВМ-2	ЗЭВМ-1	ЗЭВМ-2	ЗЭВМ-1	ЗЭВМ-2	ЗЭВМ-1	ЗЭВМ-2
1	189	190	370а	371	507б	506в	650в	650в
2	193	194	376а	378а	508б	507в	646в	630б
3	197	198	379д	383	509б	508в	628в	631б
4	201а	201б	385	386	510б	509в	640в	640б
5	202б	203	390а	393а	506а	510в	630б	646б
6	206	207	397б	398	507а	511в	631б	650б
7	210	211	423д	423е	508а	512	640б	628а
8	214	215а	404	405	509а	513в	646б	629
9	216а	217	410а	379г	510а	514в	650б	630а
10	178в	223е	379б	376б	511а	515	628а	631а
11	179в	219в	412а	390б	512	516	629	633
12	181в	220б	415	413а	513а	517в	630а	634
13	185	215б	394а	367	514а	511б	631а	635
14	221	201е	374а	373а	515	513б	633	636
15	222а	178е	377	380	516	514б	634	637
16	223в	180	403б	412в	517а	506а	635	640а
17	226	182	410в	394б	506в	507а	636	641
18	227	212	382	374б	507в	508а	637	643
19	219б	186	373	410е	508в	509а	640а	644
20	201д	222г	423а	423в	509в	510а	641	645
21	220в	228	401	402а	510в	511а	643	646а
22	229	229	394г	413б	511в	512	644	647
23	230	230	393г	423д	512	513а	645	648
24	231	231	410д	423г	513в	514а	646а	649
25	232	232	423б	394д	514в	515	647	650а
26	233	233	379в		515	516	648	
27	234	234	423б		516	517д	649	
28	272	272	376а		517в		650а	
29	236	236	405		511б		628а	
30	237	237	390б		514б		630в	

Вопросы к экзамену

1. Классификация ЭВМ.
2. Области и способы применения ЭВМ.
3. Этапы решения задач на ЭВМ и участие в них человека.
4. Понятие об алгоритме и требования к нему.
5. Схема прохождения задачи через ЭВМ
6. Источники ошибок при решении задачи на ЭВМ.
7. Типы вычислительных процессов.
Структура линейного вычислительного процесса.
8. Структуры разветвляющихся вычислительных процессов.
9. Структуры циклических вычислительных процессов.
10. Правила структурного программирования.
11. Вложенные циклы. Работа с матрицами.
12. Алфавит языка Pascal. Комментарии.
13. Понятие синтаксической диаграммы.
14. Структура блока в языке Pascal.
15. Идентификатор в Pascal. Синтаксическая диаграмма, правила образования.
16. Объявления: виды и структурная диаграмма.
17. Константы в Pascal.
18. Стандартные типы данных в Паскале. Скалярный тип.
19. Простые типы данных в Паскале. Структурная диаграмма простого типа.
20. Оператор присваивания, построение и запись выражений в Паскале. Совместимость типов в выражениях.
21. Стандартные функции в Паскале.
22. Операторы ввода и вывода.
23. Управляющие структуры Паскаля. Структуры следования и группировки.
24. Структура и операторы вывода в Паскале.
25. Условные операторы в Паскале. Их вложенность.
26. Оператор цикла с перечислением в Паскале.
27. Оператор цикла с предусловием в Паскале.
28. Оператор цикла с постусловием в Паскале.
29. Логические операторы и выражения в Паскале.
30. Оператор безусловного перехода в Паскале.
31. Массивы в Паскале. Объявления и работа с ними.

32. Типы структур программ в Паскале. Преимущества модульных программ.

33. Процедуры в Паскале.

34. Функции в Паскале.

35. Символьные переменные в Паскале, операции и функции с ними.

36. Строковые переменные в Паскале, операции и функции с ними.

37. Множества в Паскале: описание, присваивание значения и операции.

38. Записи в Паскале: описание (объявления) и обращение к полям.

39. Файлы в Паскале, их преимущества, типы файлов .

40. Описание файлового типа в Паскале и действия с данными из файлов.

41. Операции с файлами в Паскале.

42. Особенности текстовых файлов и операции с ними в Паскале.

43. Операции с записями в Паскале.

44. Операции копирования файлов.

45. Операции и алгоритм сортировки файлов.

46. Алгоритм сортировки массива методом пузырька.

47. Ссылочный пит в Паскале.

48. Типы структурных данных, образуемых с помощью ссылочного типа.

49. Алгоритм и программа быстрой сортировки Хоара.

50. Алгоритм сортировки слиянием(простым, естественным, многопутевым).

51. Алгоритм и программа сортировки методом простого включения.

52. Алгоритм и программа шейкер-сортировки.

—

РЕКОМЕНДАТЕЛЬНЫЙ БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Абрамов С.А., Зима Е.В. Начало программирования на языке Паскаль. - М.: Наука, 1987. – 110 с.
2. Офицеров Д.В., Старых В.А. Программирование в интегрированной среде Турбо-Паскаль. – Минск: Беларусь, 1992. – 239 с.
3. Прайс Д. Программирование на языке Паскаль. Практическое руководство. - М.: Мир, 1987. – 232 с.
4. Форсайт Р. Паскаль для всех. – М.: Машиностроение, 1986. – 288 с.
5. Иенсен К., Вирт Н. Паскаль: Руководство для пользователя. – М.: Компьютер, 1993. – 256 с.
6. Сергиевский М.В., Шалашов А.В. Турбо-Паскаль 7.0. Язык, среда программирования. – М.: Машиностроение, 1994. – 253 с.
7. Фигурнов В.Э. IBM PC для пользователя: - М.: ИНФРА, 1996. – 639с.
8. Дейкстра Э. Дисциплина программирования. – М.: Мир, 1978. – 275с.
9. Перминов О.Н. Программирование на языке Паскаль. - М.: Радио и связь, 1988. – 224 с.
10. Барков В.А., Черникова Л.К. Методические указания к лабораторным работам по теме "Алгоритмизация задач". – Владимир: 1980. – 52 с.
11. Семашко Г.Л., Салтыков А.И. Программирование на языке Паскаль. - М.: Наука, 1988. – 125 с.
12. Боон К. Паскаль для всех. - М.: Энергоатомиздат, 1988. - 190 с.
13. Задачи по программированию. / С.А. Абрамов и др. - М.: Наука, 1988. – 224 с.
14. Ставровский А.Б. Турбо Паскаль 7.0. Учеб. – К.: Изд. группа ВНУ, 2000. – 400 с.

Оглавление

Цели и задачи	
Лабораторная работа №1. Разветвляющиеся и циклические вычислительные процессы. Одномерные массивы.....	
Лабораторная работа №2. Программирование вложенных циклов и матричных задач Процедуры и функции	
Лабораторная работа №3. Файлы и записи.....	
Лабораторная работа №4. Алгоритмы и программы сортировки массивов.....	
Приложения.	
Рекомендательный библиографический список.....	