МИНИСТЕРСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ПО СВЯЗИ И ИНФОРМАТИЗАЦИИ

САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ
им. проф. М.А.БОНЧ-БРУЕВИЧА

А.В. Красов

Программирование на языке Си++

Методические указания к лабораторным работам

Для специальности 201800

Часть 1

Санкт-Петербург
2004

УДК 681.3.067

Красов А.В. Программирование на языке Си++. Ч.1.: Методические указания к лабораторным работам / СПбГУТ. - СПб, 2004.

Утверждено в качестве учебного пособия редакционно-издательским советом университета.

Рассматриваются вопросы программирования на языке Си++ и предназначено для студентов, обучающихся по специальности 201800 «Защищенные системы связи» при подготовке к выполнению лабораторных работ.

Ответственный редактор доц. В.С.Качур

Рецензент: д.т.н., проф. С.Е.Душин (СПбГЭТУ «ЛЭТИ»)

Введение

Данные методические указания предназначены для дисциплины программирования на языке Си, читаемой студентам специальности 201800 «Защищенные системы связи». Язык программирования Си был создан в 1972 г. Денисом Ритчи. На сегодняшний день это самый распространенный и популярный язык программирования, позволяющий строить эффективные программы для различных областей программирования, в том числе и решения задач системного программирования и защиты информации. Не случайно самые популярные на сегодняшний день операционные системы Ms Windows и UNIX написаны именно на этом языке программирования.

Материал, рассматриваемый в данном пособии, является основой для изучения последующих специальных дисциплин специальности 201800 «Защищенные системы связи».

Лабораторная работа 1. Вычисление по формулам

1.1. Цель лабораторной работы

1) Освоить создание программ с простейшим линейным алгоритмом.
2) Освоить работу с интегрированной средой Borland C++.
3) Освоить работу с отладчиком.
4) Ознакомится с организацией контролирующих программ.

1.2. Используемое программное обеспечение

Для выполнения лабораторной работы используется Borland C версия 3.1 и программа генерации задания lab1.exe.

1.3. Порядок выполнения лабораторной работы

1. Запустить программу генерации задания lab1 и Borland C++.
2. Зарегистрироваться в контролирующей программе.
3. Получить задание (генерируется контролирующей программой).
4. Составить программу.
5. Проверить правильность работы программы с помощью контролирующей программы и ручного расчета.
6. Проверить работу программы в отладчике.

1.4. Пример выполнения лабораторной работы

Контролирующая программа сгенерировала следующее задание:

$$y = \frac{a}{a + b},\ a = 0.7;\ b = 1.3 \qquad (1.1)$$

Необходимо написать программу, рассчитывающую данное выражение и вывести на экран вид формулы с числовыми значениями переменных.

В начале составляется алгоритм программы, для формулы (1.1) он представлен на рис. 1.1.

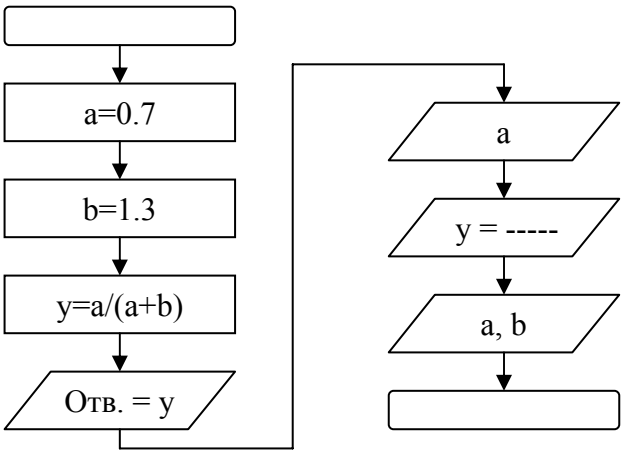


Рис. 1.1. Блок-схема программы вычисления по формуле

По алгоритму составляем программу (листинг 1.1).

Листинг 1.1. Листинг программы вычисления по формуле

```
#include <stdio.h>  // Подключаем прототипы функций ввода-вывода
   void main(void)                  // описание главной функции
  {
        float a, b, y ;         // описание переменных
        a = 0.7 ;      // присвоение значения переменной a
        b = 1.3 ;      // присвоение значений переменной b
        y = a / (a + b) ;     // расчет переменной y
        printf("\n Ответ y = %f" ,y) ;   // вывод на экран ответа
        printf("            %5.2f ", a) ;  // вывод на экран вида формулы
        printf("y = ------------------") ;
        printf("       %5.2f + %5.2f ", a, b) ;
  }             // конец программы
```

В листинге 1.1 при выводе вида формулы для нормальной работы контролирующей программы необходимо использовать формат вывода ***%5.2f*** (5 символов всего, 2 после точки).

Ввести программу в компьютер, сохранить ее в виде файла (в заданном каталоге) (F2) и запускаем на исполнение (Ctrl + F9). Проверить правильность результата программы с результатами ручного расчета по формуле (1.1).

Проверить результаты работы программы с помощью контролирующей программы. Для этого выбрать в контролирующей программе имя исполнимого файла, построенного по листингу (1.1).

Проверит работу созданной программы в пошаговом режиме в отладчике, контролируем изменение значения переменных (F7, F4, Ctrl + F7).

Оформить отчет по лабораторной работе.

1.5. Требования к отчету по лабораторной работе

Отчет должен содержать:

1) титульный лист;
2) текст задания;
3) блок-схему и листинг программы, желательно размещение операторов программы напротив соответствующих элементов блок-схемы.

Лабораторная работа 2. Разветвляющиеся программы

2.1. Цель лабораторной работы

1) Освоить создание программ с разветвляющимися алгоритмами.
2) Освоить операции ввода значений с клавиатуры.
3) Освоить работу с логическим операциями И, ИЛИ, НЕ.

2.2. Используемое программное обеспечение

Для выполнения лабораторной работы используется Borland C версия 3.1 и программа генерации задания lab2.exe.

2.3. Порядок выполнения лабораторной работы

1. Запустить программу генерации задания lab2.exe и Borland C++.
2. Зарегистрироваться в контролирующей программе.
3. Получить задание (генерируется контролирующей программой).
4. Составить программу.
5. Проверить правильность работы программы с помощью контролирующей программы и ручного расчета.
6. Проверить работу программы в отладчике.

2.4. Пример выполнения лабораторной работы

Контролирующая программа сгенерировала следующее задание (рис. 2.1). Необходимо написать программу, которая должна вводить координаты

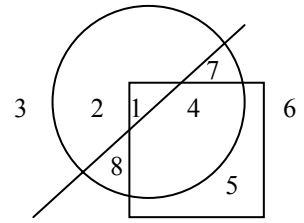


Рис. 2.1. Задание на попадание точки в области

точек и выводить номер области, в которую попала данная точка. Дополнительное условие – в каждом условном операторе допустимо работать только с одной геометрической фигурой, кроме того, необходимо ограничится минимальным кол-вом составных операторов.

Для выполнения задания необходимо по заданному Вам рисунку составить уравнения всех фигур.

Проверка, того что точка лежит ниже прямой

$$y < kx + b. \tag{2.1}$$

Проверка нахождения точки ниже параболы

$$y < ax^2 + bx + c. \tag{2.2}$$

Проверка нахождения точки в круге

$$(x - x_{\text{ц}})^2 + (y - y_{\text{ц}})^2 < r^2. \tag{2.3}$$

Проверка нахождения точки в эллипсе

$$\left(\frac{x - x_{\text{ц}}}{r_x}\right)^2 + \left(\frac{y - y_{\text{ц}}}{r_y}\right)^2 < 1. \tag{2.4}$$

Проверка нахождения точки в прямоугольнике

$$(x_1 < x) \text{ И } (x < x_2) \text{ И } (y_1 < y) \text{ И } (y < y_2). \tag{2.5}$$

Проверка нахождения точки в треугольнике

$$(y < k_1 x + b_1) \text{ И } (y < k_2 x + b_2) \text{ И } (y > k_3 x + b_3). \tag{2.6}$$

Коэффициенты уравнений (2.1-6) определяются по заданному рисунку, значения переменных *x, y* вводятся пользователем с клавиатуры.

Составить алгоритм и программу (рис.2.2). Проверить правильность составления алгоритма, для этого необходимо проверить работу алгоритма при вводе координат точек, попадающих во все области.

Вести программу в компьютер, сохранить на диске (F2), проверить правильность работы (Ctrl+F9). При проверке необходимо рассмотреть попадание точек во все области на рисунке.

Проверить результаты работы программы с помощью контролирующей программы. Для этого выбираем в контролирующей программе имя исполнимого файла, построенного по листингу (2.1).

Проверить работу созданной программы в пошаговом режиме в отладчике, контролируем изменение значения переменных (F7, F4, Ctrl + F7).

Оформить отчет по лабораторной работе.

2.5. Требования к отчету по лабораторной работе

Отчет должен содержать:

1) титульный лист;
2) текст задания (в т.ч. рисунок в масштабе);
3) блок-схему и листинг программы, желательно размещение операторов программы напротив соответствующих элементов блок-схемы.

Лабораторная работа 3. Простейшие программы с использованием циклов

3.1. Цель лабораторной работы

1) Освоить создание циклических программ.
2) Освоить работу с составным операторами.

3.2. Используемое программное обеспечение

Для выполнения лабораторной работы используется Borland C++ версия 3.1 и программа генерации задания lab3.exe.

3.3. Порядок выполнения лабораторной работы

1. Запустить программу генерации задания lab3 и Borland C++.
2. Зарегистрироваться в контролирующей программе.
3. Получить задание (генерируется контролирующей программой).
4. Составить программу.
5. Проверить правильность работы программы с помощью контролирующей программы и ручного расчета.
6. Проверить работу программы в отладчике.

3.4. Пример выполнения лабораторной работы

Контролирующая программа сгенерировала следующее задание (рис. 3.1). Необходимо построить на экране аналогичную фигуру, используя циклы и функцию printf(). В функции printf() можно выводить только один символ. Дополнительное условие: ограничиться минимальным количеством составных операторов.

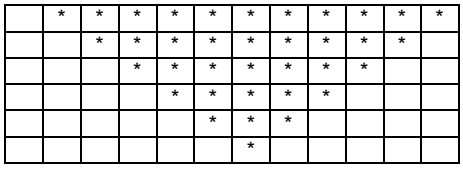


Рис. 3.1. Требуемый результат работы программы

По заданию составить алгоритм (рис. 3.2).

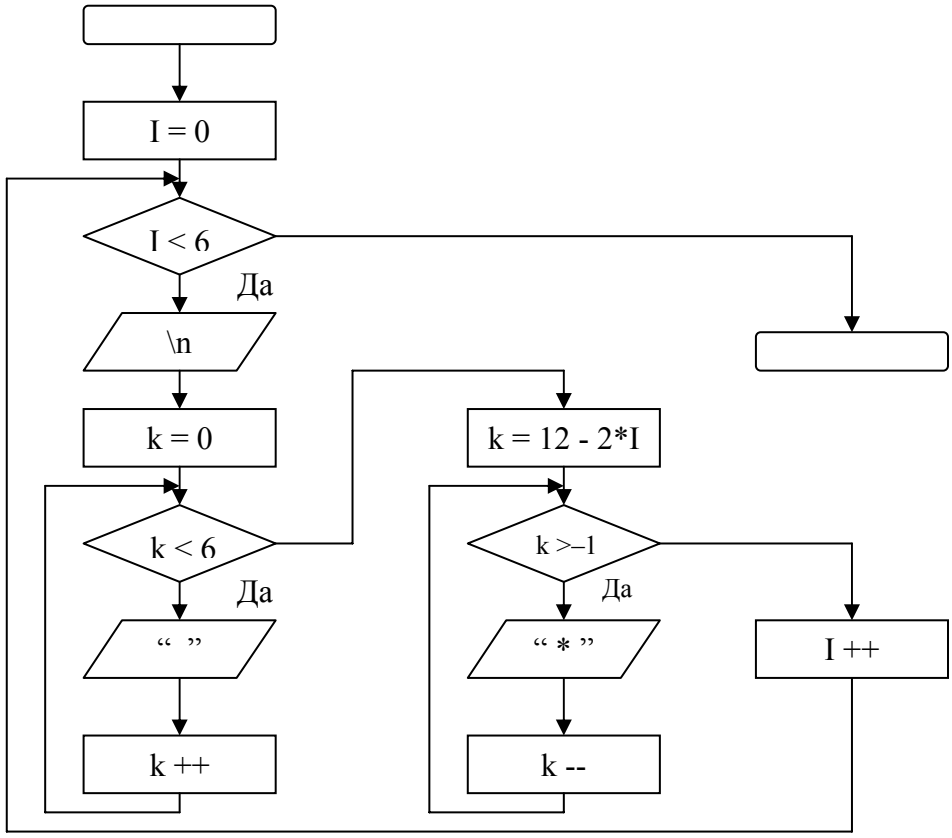


Рис. 3.2. Блок-схема алгоритма вывода на экран фигуры

Листинг программы, реализующий данный алгоритм приведен на листинге 3.1. Программа содержит три цикла. Первый цикл организует перемещение по строкам. В нем расположены: переход на следующую строку – printf(“\n”) и два вложенных цикла. Один из них отвечает за сдвиг с помощью пробелов на нужную позицию на экране, второй выводит указанное количество символов “*”.

Листинг 3.1. Программа вывода на экран фигуры

```
#include <stdio.h>
void main( void )
  {
        int I, k ;      // описание переменных
        for(I = 0; I < 6; I++)      // цикл по строкам
```

```
                {
                        printf("\n") ; // перевод на новую строку
                        for(k = 0;  k < 6 ; k++)   // цикл для сдвига
                                printf(" ");
                        for(k = 12 - 2*I ;  k >-1 ; k++) // цикл вывода
                                printf("* ");                  // символов '*'
                }       // конец цикла for
        }       // конец программы
```

Ввести программу в компьютер, сохранить на диске (F2), проверить правильность работы (Ctrl+F9).

Проверяем результаты работы программы с помощью контролирующей программы. Для этого выбираем в контролирующей программе имя исполнимого файла, построенного по листингу (3.1).

Проверяем работу созданной программы в пошаговом режиме в отладчике, контролируем изменение значения переменных (F7, F4, Ctrl + F7).

Оформляем отчет по лабораторной работе.

3.5. Требования к отчету по лабораторной работе

Отчет должен содержать:

1) титульный лист;
2) текст задания (в том числе рисунок заданной фигуры);
3) блок-схему и листинг программы, желательно размещение операторов программы напротив соответствующих элементов блок-схемы.

Лабораторная работа 4.
Программы на циклы

4.1. Цель лабораторной работы

1) Освоить создание циклических программ.
2) Освоить работу с двух мерными массивами и текстовыми строками.

4.2. Используемое программное обеспечение

Для выполнения лабораторной работы используется Borland C версия 3.1 и программа генерации задания lab4.exe.

4.3. Порядок выполнения лабораторной работы

Запустить программу генерации задания lab4 и Borland C.

1. Зарегистрироваться в контролирующей программе.
2. Получить задание (генерируется контролирующей программой).

3. Составить программу.
4. Проверить правильность работы программы с помощью контролирующей программы и ручного расчета.
5. Проверить работу программы в отладчике.

4.4. Пример выполнения лабораторной работы

Контролирующая программа сгенерировала следующее задание: «Необходимо ввести с клавиатуры размеры массива вещественных чисел (не больше 10), ввести элементы массива с клавиатуры, найти максимальный элемент массива, записать вместо него значение нуль и вывести полученный массив на экран».

По полученному заданию составить алгоритм (рис. 4.1).

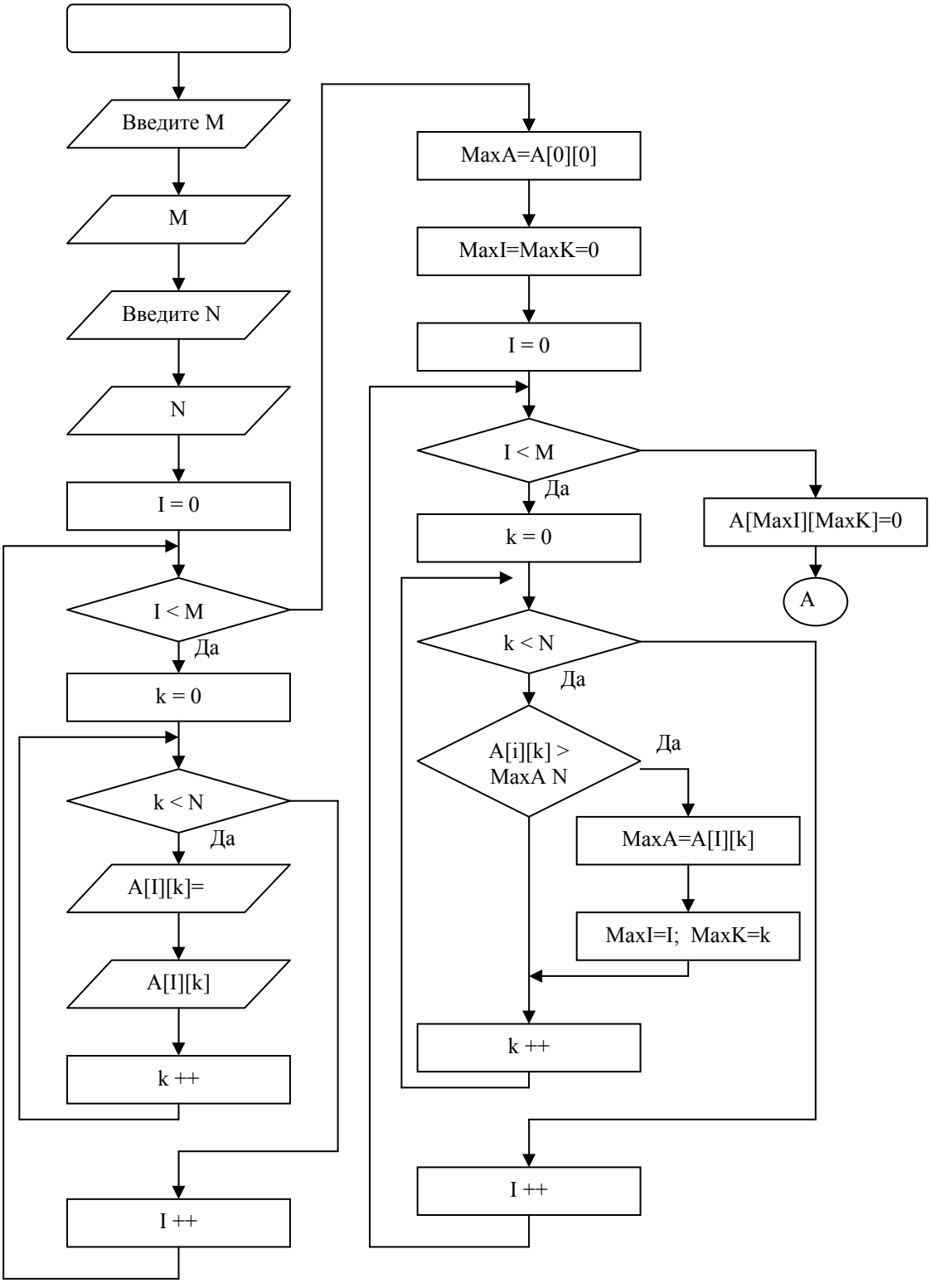


Рис. 4.1. Блок-схема алгоритма задания по циклам (начало)

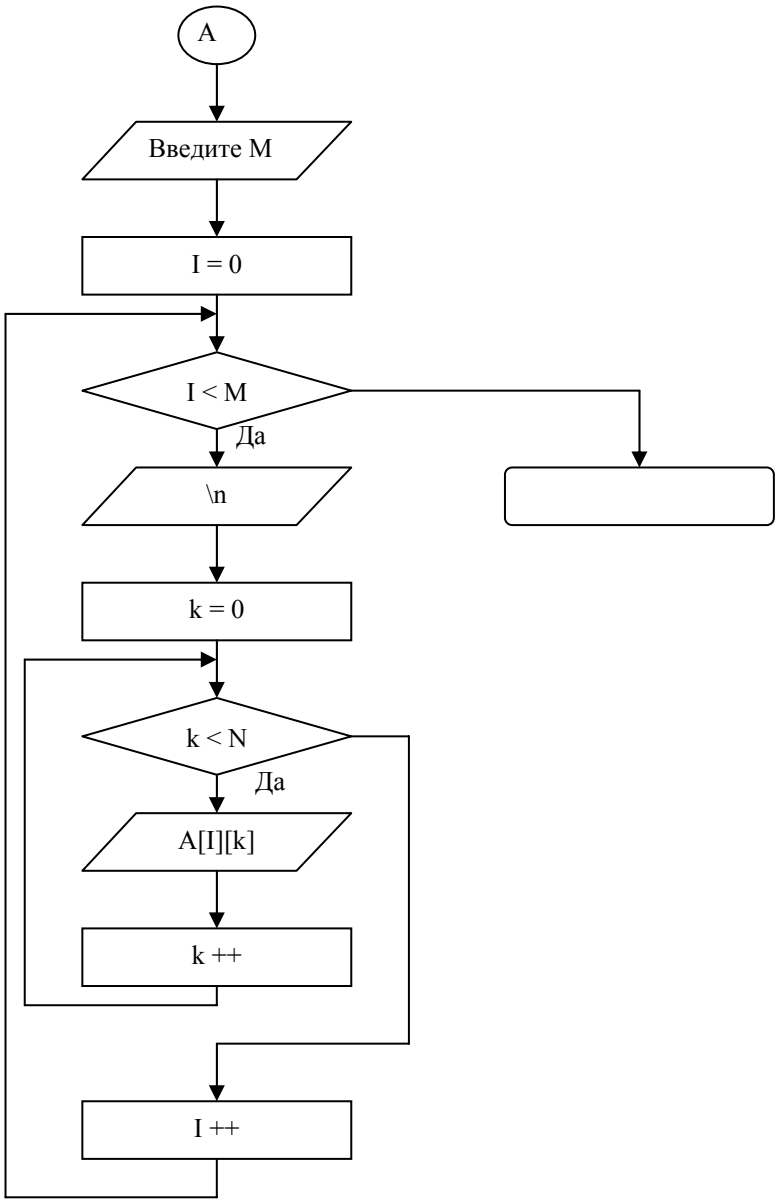


Рис. 4.1. Блок-схема алгоритма программы на циклы (продолжение)

По алгоритму построить программу. Ввод массива построен с помощью двух вложенных циклов. Используются M строк и N столбцов. Предполагаем, что пользователь вводит размер массива правильно, проверка ввода не производится. После чего присваиваем переменной MaxA значение первого элемента, а переменным MaxI и MaxK координаты этого элемента в массиве. После этого в двух вложенных циклах последовательно просмотреть все элементы массива и сравнить их со значением MaxA. Если элемент массива оказывается больше, чем значение MaxA, то присвоить переменным новые значения (MaxA, MaxI, MaxK). После завершения циклов обнуляем найденный максимальный элемент. Вывод массива на экран содержит два вложенных цикла (Рис. 4.1). Листинг программы показан на листинге 4.1.

Листинг 4.1. Листинг программы обработки массива

```
#include<stdio.h>
void main(void)
  {
        float A[10][10], MaxA ;
        int I, k, MaxI, MaxK, N, M ;
        printf("\n Введите количество строк:") ;
        scanf("%d", &M) ;
        printf("\n Введите количество столбцов:") ;
        scanf("%d", &N) ;
        MaxA = A[ 0 ][ 0 ] ;
        MaxI=MaxK = 0 ;
        for(I=0; I<M; I++) //циклы поиска максимального значения
              for(k=0; k<N; k++)
                    if(MaxA<A[ I ][ k ])
                          {
                                MaxA = A[ I ][ k ] ;
                                MaxI = I ;
                                MaxK = k ;
                          }
        A[MaxI][MaxK]=0 ;//Обнуление максимального значения
        for(I=0; I<M; I++) // Циклы вывода на экран массива
              {
                    printf("\n") ;
                    for(k=0; k<N; k++)
                          printf("%5.2f", A[ I ][ k ]) ;
              }

  }     // Конец программы
```

Ввести программу в компьютер, сохранить на диске (F2), проверить правильность работы (Ctrl+F9). При проверке необходимо проверить попадание точек во все области на рисунке.

Проверить результаты работы программы с помощью контролирующей программы. Для этого выбирать в контролирующей программе имя исполнимого файла, построенного по листингу 2.1.

Проверить работу созданной программы в пошаговом режиме в отладчике, при этом контролировать изменение значения переменных (F7, F4, Ctrl + F7).

Оформить отчет по лабораторной работе.

4.5. Требования к отчету по лабораторной работе

Отчет должен содержать:

1) титульный лист;
2) текст задания;
3) блок-схему и листинг программы, желательно размещение операторов программы напротив соответствующих элементов блок-схемы.

Дополнительное задание

Организовать контроль правильности ввода размеров массива.

Лабораторная работа 5. Движение символа по экрану

5.1. Цель лабораторной работы

1) Освоить использование с функций работы с экраном.
2) Освоить работу с оператором switch.
3) Освоить алгоритм построения рамки.

5.2. Используемое программное обеспечение

Для выполнения лабораторной работы используется Borland C++ версия 3.1.

5.3. Порядок выполнения лабораторной работы

1. Запустить программу Borland C++.
2. Зарегистрироваться в контролирующей программе.
3. Получить задание у преподавателя.
4. Составить программу.
5. Проверить правильность работы.

5.4. Варианты задания на лабораторную работу

1. Написать программу движения символа только по клавишам управления курсора.

2. Написать программу движения символа в рамке с указанными координатами (с проверкой координат).

3. Написать программу движения символа только в пределах эллипса вписанного в рамку.

4. Написать программу движения символа в треугольнике образуемого двумя сторонами рамки и диагональю (по выбору преподавателя).

5. Написать программу движения указанного количества звездочек по экрану.

6. Ввести с клавиатуры строку и написать программу ее движения по экрану.

7. Задания п.1–6 реализовать в режиме непрерывного движения по экрану.

Общие требования к программам:

- программы должны содержать рамку, координаты которой должны вводится с клавиатуры;
- двигающиеся символы не должны выходить за границу рамки (или появляться с другой стороны);
- программа должна содержать строку подсказки, выведенную с центрированием под рамкой.

5.5. Пример выполнения лабораторной работы

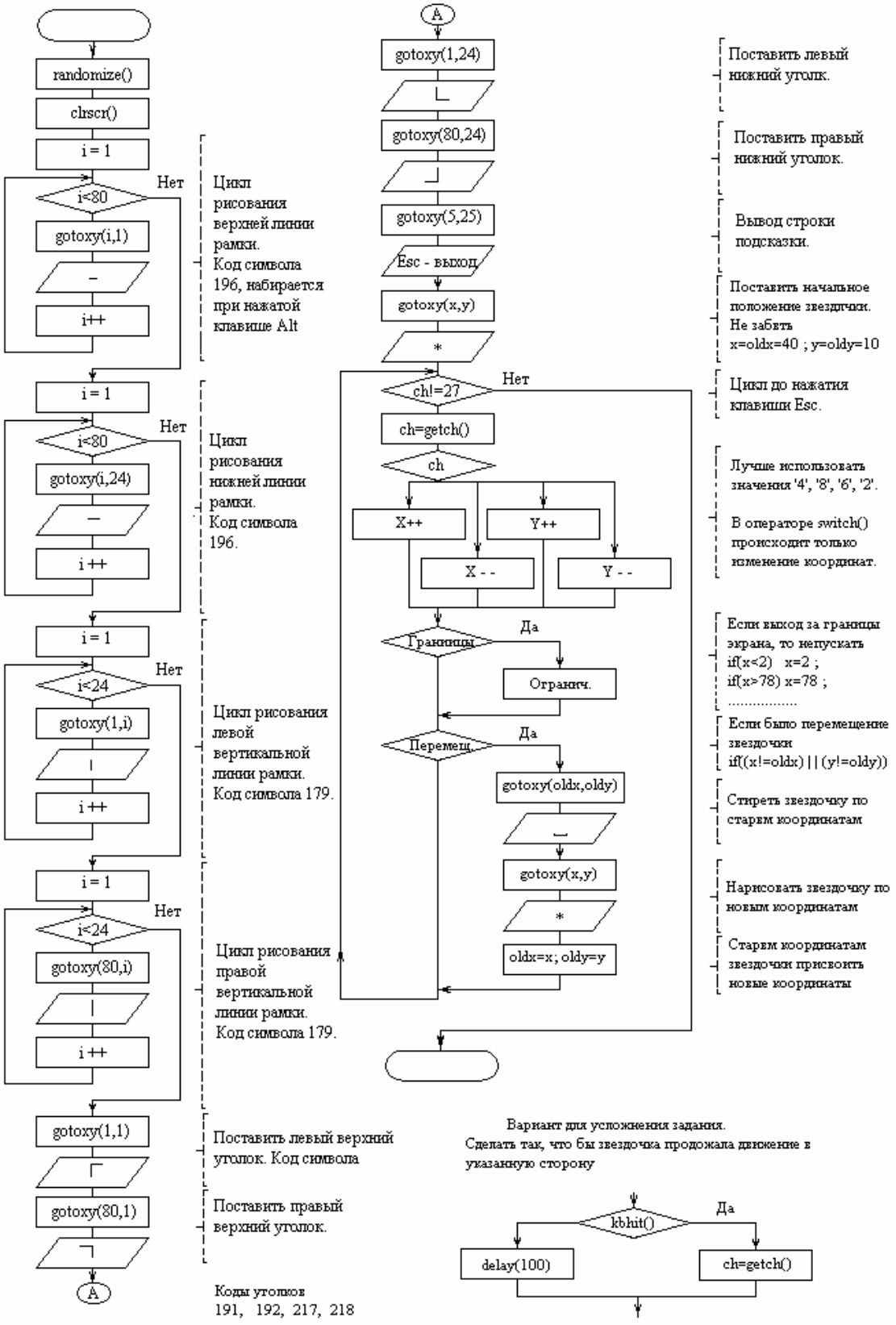


Рис. 5.1. Алгоритм программы движения символа по экрану

Для определения кодов клавиш управления курсором и номеров символов для построения рамки рекомендуется использовать программы (листинги 5.1 и 5.2).

Листинг 5.1. Программа вывода символов из таблицы ASCII

```
#include<conio.h>
#include<stdio.h>
void main(void)
  {
        int i ;
        for(i=0;  i<256 ; i++)
              {
                  printf("\n Символ \'%c\' имеет номер %d", i, i ) ;
                  if(i%20 == 0)  // Задержка — каждый 20-й символ
                     getch() ;
              }

  }
```

Листинг 5.2. Программа определения кодов клавиш

```
#include<conio.h>
#include<stdio.h>
void main(void)
  {
        char ch ;
        while( ch!=27)
              {
                    ch=getch() ;
                    printf("\n Нажат символ \'%c\' с кодом %d",ch,
ch) ;              }
  }
```

Во время работы программы определения клавиш, обратите внимание на то, что при нажатии на клавиши управления курсором в буфер клавиатуры помещается сразу несколько символов.

Написанная программа проверятся вручную преподавателем. По программе оформляется отчет.

5.6. Требование к отчету по лабораторной работе

Отчет должен содержать:

1) титульный лист;
2) текст задания;

3) блок-схему и листинг программы, желательно размещение операторов программы напротив соответствующих элементов блок-схемы.

Лабораторная работа 6. Программа меню

6.1. Цель лабораторной работы

1) Освоить создание программ содержащих меню.
2) Закрепить работу с циклами.
3) Освоить работу с массивами строк.

6.2. Используемое программное обеспечение

Для выполнения лабораторной работы используется Borland C++ версия 3.1 и программа генерации задания lab6.exe.

6.3. Порядок выполнения лабораторной работы

1. Запустить программу генерации задания lab6.exe и Borland C++.
2. Зарегистрироваться в контролирующей программе.
3. Получить задание (генерируется контролирующей программой).
4. Составить программу.
5. Проверить правильность работы программы с помощью контролирующей программы и ручного расчета.
6. Проверить работу программы в отладчике.

6.4. Пример выполнения лабораторной работы

Необходимо написать программу, содержащую меню. После выбора пункта меню программа должна сообщить номер и название выбранного пункта меню. Название и количество пунктов меню генерируется контролирующей программой, способ выбора пунктов (только клавишами управления курсора, допускается использование «горячих» клавиш).

Для описания меню предлагается ввести массив указателей на строки:

char *Menu[]={“File”, ”Save”, “Test”, “Help”, “Quit”, NULL }, значения вводятся при инициализации переменной. Кроме того, понадобится переменная, отвечающая за количество пунктов.

Укрупненная блок-схема приведена на рис. 6.1. В общих чертах программа повторяет программу, представленную на рис. 5.1.

В первом блоке строится рамка, задаются начальные значения переменных (см. рис. 5.1), выводятся на экран пункты меню. Для вывода на экран пунктов меню рекомендуется воспользоваться функ-

циями gotoxy() и cprintf(), предварительно задав цвета вывода символов. При выводе пунктов меню, рекомендуется использовать фиксированный формат вывода строки, например “%20s”.

Далее организован основной цикл программы, до нажатия клавиши Enter (Ch==13). Переменная y отвечает за номер текущего выбранного пункта меню, oldy за старое положение. При измерении значения переменной y, необходимо контролировать выход за пределы массива (эти блоки не показаны на рис. 6.1, их реализация аналогична предыдущей программе).

Работа по «горячим» клавишам строится аналогично (фактически сводится к выбору значения переменой y, соответствующего этой горячей клавише). По нажатию клавиши Esc программа выдает сообщение и завершает свою работу.

Если было обнаружено несовпадение значений переменных y и oldy, то выполняется перерисовка соответствующих пунктов меню.

После завершения цикла необходимо вывести на экран номер выбранного пункта меню и его название. **Внимание!** Для правильной работы контролирующей программы вывод необходимо осуществить с помощью функции printf().

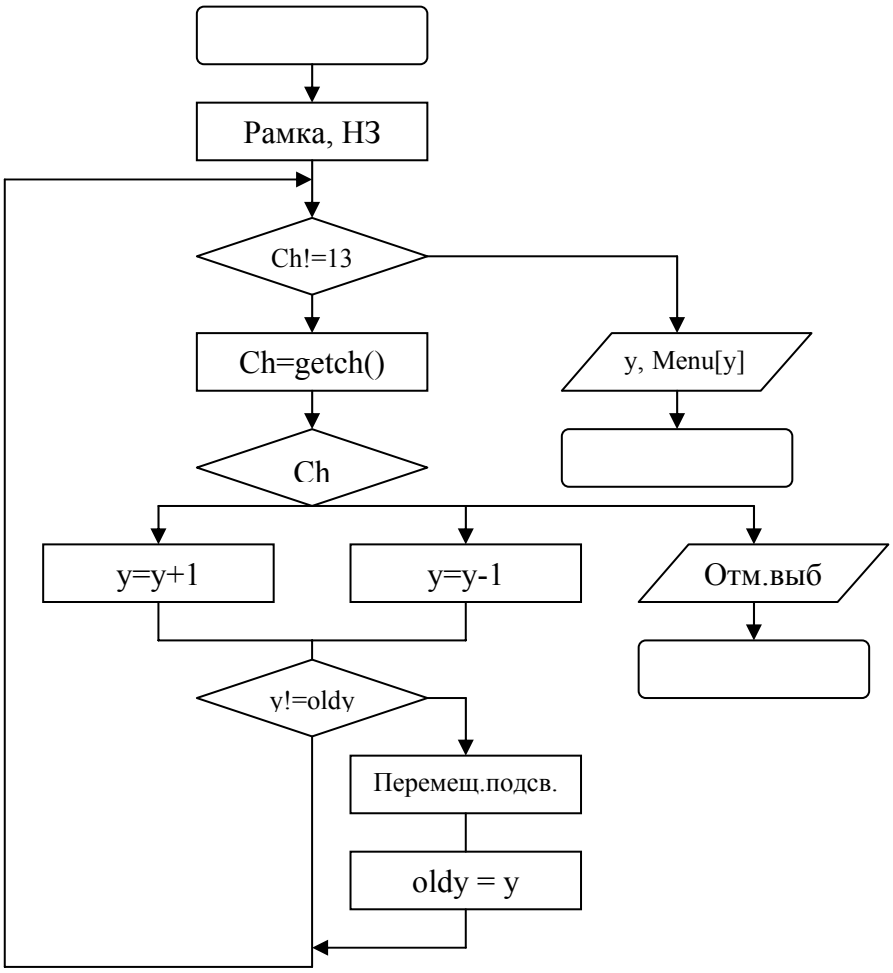


Рис. 6.1. Укрупненная блок-схема программы вывода меню

Ввести программу в компьютер, сохранить на диске (F2), проверить правильность работы (Ctrl+F9).

Проверить результаты работы программы с помощью контролирующей программы. Для этого выбирать в контролирующей программе имя исполнимого файла, построенного по блок-схеме, представленной на рис. 6.1.

Проверить работу созданной программы в пошаговом режиме в отладчике, при этом контролировать изменение значения переменных (F7, F4, Ctrl + F7).

Оформить отчет по лабораторной работе.

6.5. Требования к отчету по лабораторной работе

1) Титульный лист;
2) Текст задания (в т.ч. рисунок заданной фигуры);
3) Блок-схему и листинг программы, желательно размещение операторов программы напротив соответствующих элементов блок-схемы.

Лабораторная работа 7.
Простейшая программа телефонного справочника

7.1. Цель лабораторной работы

1) Освоить работу со структурами.
2) Освоить методы сортировки структурированных данных.

7.2. Используемое программное обеспечение

Для выполнения лабораторной работы используется Borland C++ версия 3.1 и контролирующая программа lab7.exe.

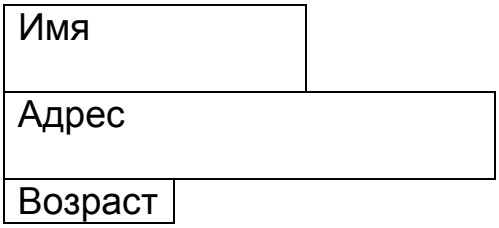
7.3. Порядок выполнения лабораторной работы

1. Запустить программу генерации задания lab7.exe и Borland C.
2. Зарегистрироваться в контролирующей программе.
3. Получить задание (генерируется контролирующей программой).
4. Составить программу.
5. Проверить правильность работы программы с помощью контролирующей программы и ручного расчета.
6. Проверить работу программы в отладчике.

7.4. Пример выполнения лабораторной работы

Программа генерации заданий придумывает формат структуры и критерии сортировки. На основании задания описывается структура, например:

```
struct BOOK     /* Описание своего типа struct BOOK */
{
       char Name[20];    /* Поле имени */
       char Adres[40];   /* Поле адреса */
       int Year;         /* Возраст */
};  /* Кон. описания структуры BOOK */
struct BOOK Book[50]; /* Описание переменной массива структур */
```



После этого составляем блок-схему программы (рис. 7.1 или 7.3).

На рис. 7.1 представлена программа, организующая ввод указанного числа записей и удаления одного из них. На рис. 7.3 представлена программа, организующая сортировку по возрасту.

В том случае, если сравнение необходимо организовать по имени или по адресу (текстовым полям), то необходимо написать функцию, реализующую сравнение строк. Блок-схема данной функции представлена на рис. 7.2. Данная функция, получив в качестве параметров указатели на две строки символов, находит первый различный

символ строки и по нему сравнивает строки. Результат сравнения возвращается в виде числа (0 или 1).

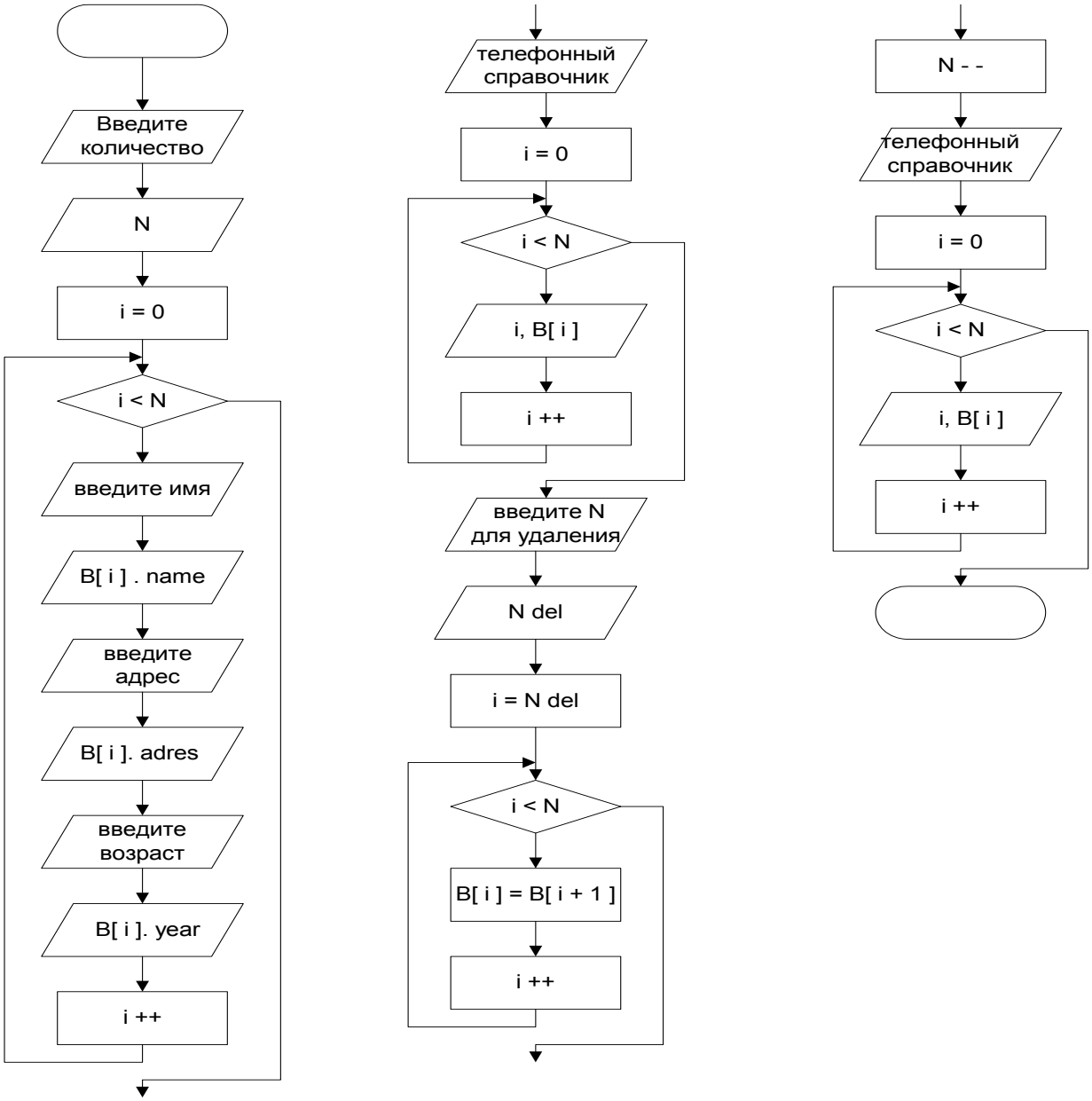


Рис. 7.1. Блок-схема программы ввода и удаления записей

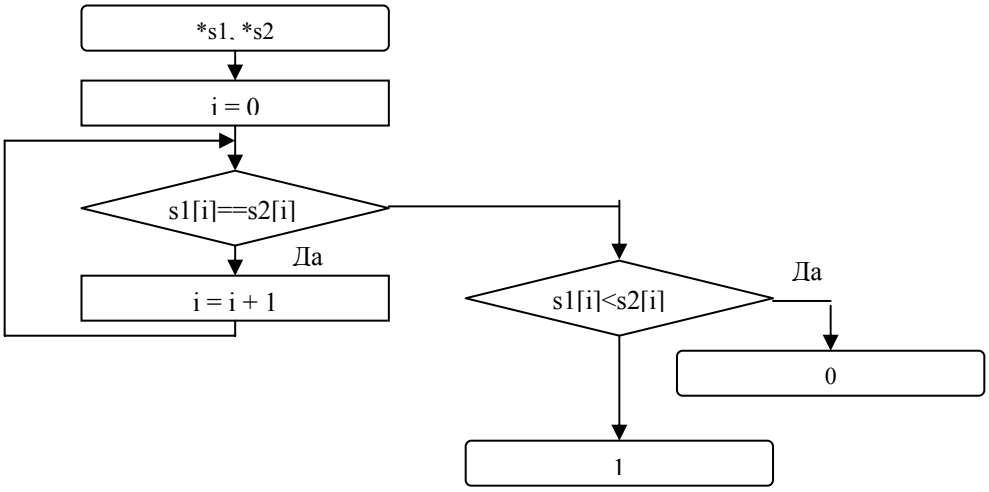


Рис. 7.2. Блок-схема функции реализующей сравнение строк

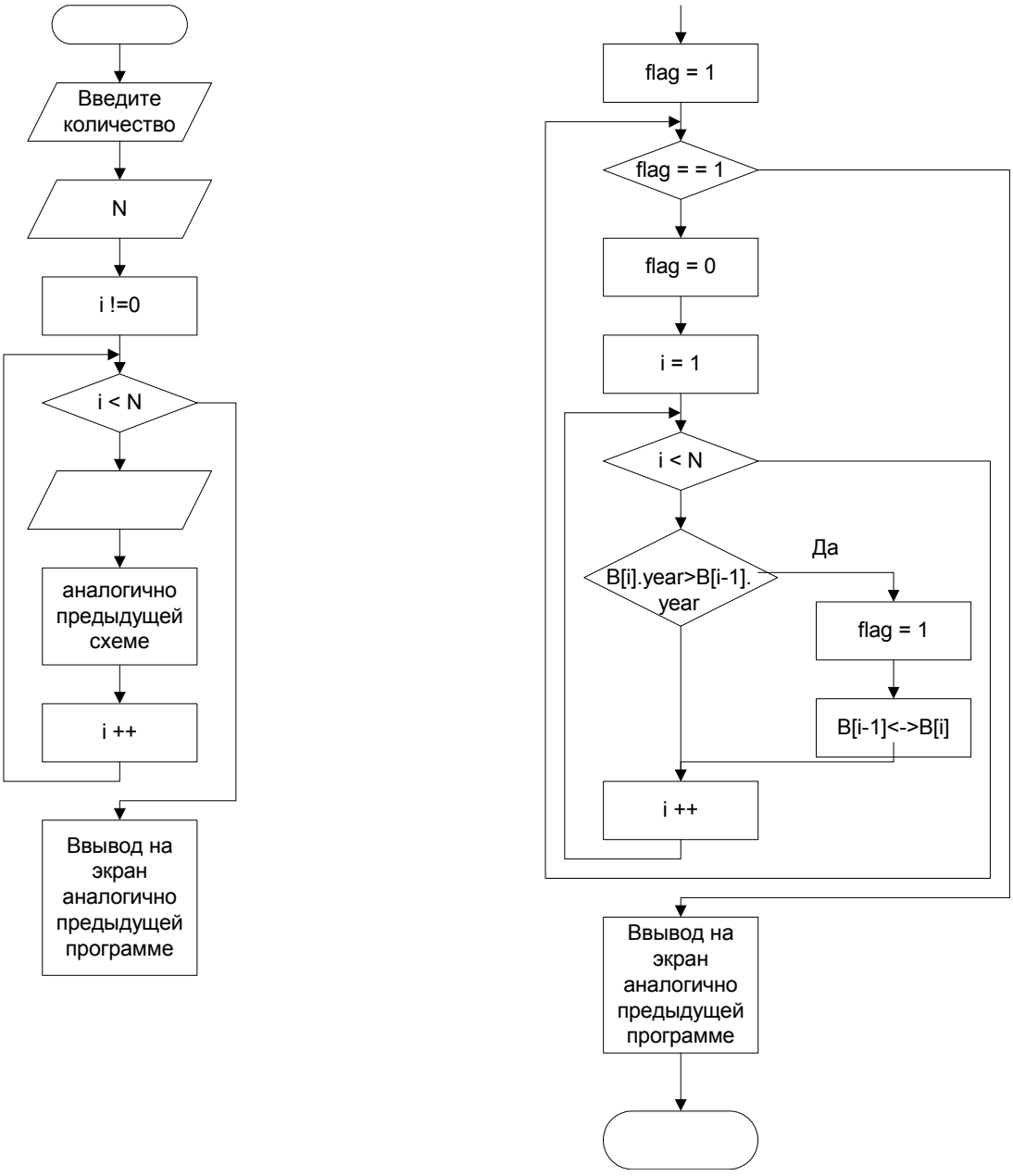


Рис. 7.3. Блок-схема программы сортировки записей по возрасту

По блок-схеме составить программу и ввести ее в компьютер, сохранить на диске (F2), проверить правильность работы (Ctrl+F9). При проверки необходимо проверить попадание точек во все области на рисунке.

Проверить результаты работы программы с помощью контролирующей программы. Для этого выбрать в контролирующей программе имя исполнимого файла. **Внимание!** Для правильной работы контролирующей программы вывод на экран необходимо организовать, используя функцию printf().

Проверяем работу созданной программы в пошаговом режиме в отладчике, контролируем изменение значения переменных (F7, F4, Ctrl + F7).

Оформляем отчет по лабораторной работе.

7.5. Требования к отчету по лабораторной работе

Отчет должен содержать:

1) титульный лист;
2) текст задания, включая заданный Вам формат структуры данных ;
3) блок-схему и листинг программы, желательно размещение операторов программы напротив соответствующих элементов блок-схемы;
4) пример работы программы.

Лабораторная работа 8. Программа телефонного справочника

8.1. Цель лабораторной работы

1. Освоить работу со структурами.
2. Освоить методы сортировки структурированных данных.

8.2. Используемое программное обеспечение

Для выполнения лабораторной работы используется Borland C++ версия 3.1 и контролирующая программа lab8.exe.

8.3. Порядок выполнения лабораторной работы

1. Запустить программу генерации задания lab8 и Borland C.
2. Зарегистрироваться в контролирующей программе.
3. Получить задание (генерируется контролирующей программой).
4. Составить программу.
5. Проверить правильность работы программы с помощью контролирующей программы и ручного расчета.
6. Проверить работу программы в месте с преподавателем.
7. Проверить работу программы в отладчике.

8.4. Пример выполнения лабораторной работы

Необходимо написать программу телефонного справочника, программа должна содержать средства выбора текущей записи (с помощью клавиш управления курсора), сортировки по различным полям, вывода на экран обработанного списка (для взаимодействия с контролирующей программой).

Блок-схема программы представлена на рис. 8.1. В целом программа повторяет программу представленную на рис. 7.1—7.3, дополнительно необходимо разработать интерфейс пользователя.

Предлагается сделать функцию void PutLine(int n, color c1, color c2) выводящую на экран в заданную позицию (определяется номером записи), запись с номером n. Цвета c1 и c2 определяют цвет выводимой строки (подсвеченной для активной строки или обычной). Функция void PutScr(void) выводит на экран содержимое всей базы данных.

Массив структур (B), их количество (N), номер активной записи (TekN), рекомендуется сделать глобальными переменными. Переменная redraw отвечает за перерисовку экрана. Если данная переменная имеет значение 1, то вызывается функция PutScr(), перерисовывающая экран.

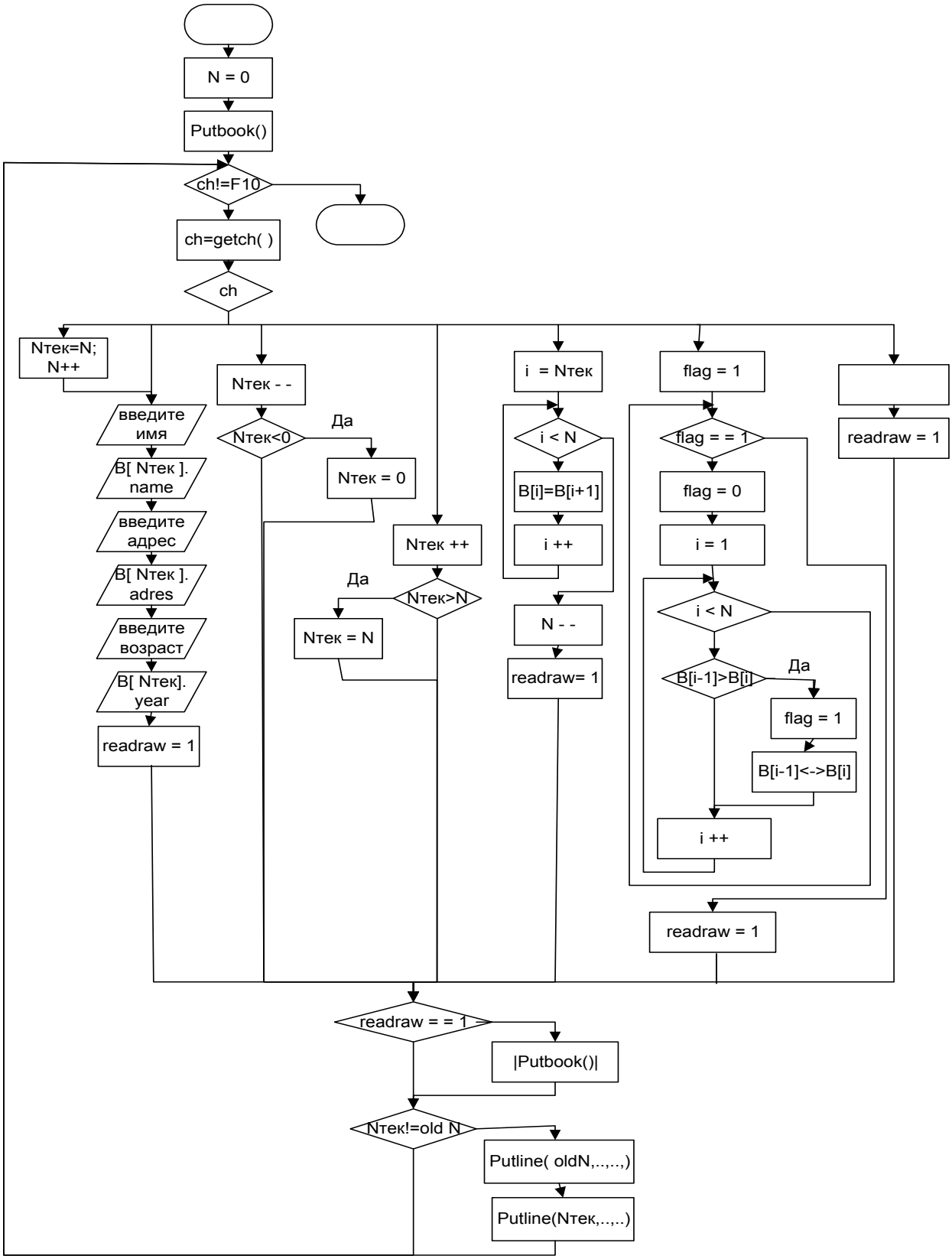


Рис. 8.1. Блок-схема программы телефонного справочника

По блок-схеме составить программу и ввести ее в компьютер, сохранить на диске (F2), проверить правильность работы (Ctrl+F9).

Проверить результаты работы программы с помощью контролирующей программы. Для этого выбирать в контролирующей программе имя исполнимого файла. **Внимание!** Для правильной работы контролирующей программы вывод на экран необходимо организовать, используя функцию printf().

Проверить работу созданной программы в пошаговом режиме в отладчике, контролируем изменение значения переменных (F7, F4, Ctrl + F7).

Оформить отчет по лабораторной работе.

8.5. Требования к отчету по лабораторной работе

Отчет должен содержать:

1) титульный лист;
2) текст задания, включая заданный Вам формат структуры данных ;
3) блок-схему и листинг программы;
4) пример работы программы.

Лабораторная работа 9, работа с текстовыми файлами

9.1. Цель лабораторной работы

1. Освоить работу с функциями стандартной библиотеки языка Си, предназначенными для работы с текстовыми файлами.
2. Освоить методы обработки текстовых файлов.

9.2. Используемое программное обеспечение

Для выполнения лабораторной работы используется Borland C++ версия 3.1 и контролирующая программа lab9.exe.

9.3. Порядок выполнения лабораторной работы

1. Запустить программу генерации задания lab9.exe и Borland C.
2. Зарегистрироваться в контролирующей программе.
3. Получить задание (генерируется контролирующей программой).
4. Составить программу.
5. Проверить правильность работы программы с помощью контролирующей программы и ручного расчета.
6. Проверить работу программы в отладчике.

9.4. Пример выполнения лабораторной работы

Контролирующая программа генерирует задание, например, необходимо написать программу, копирующую содержимое одного файла в другой файл. Блок-схема данной программы представлена на рис. 9.1.

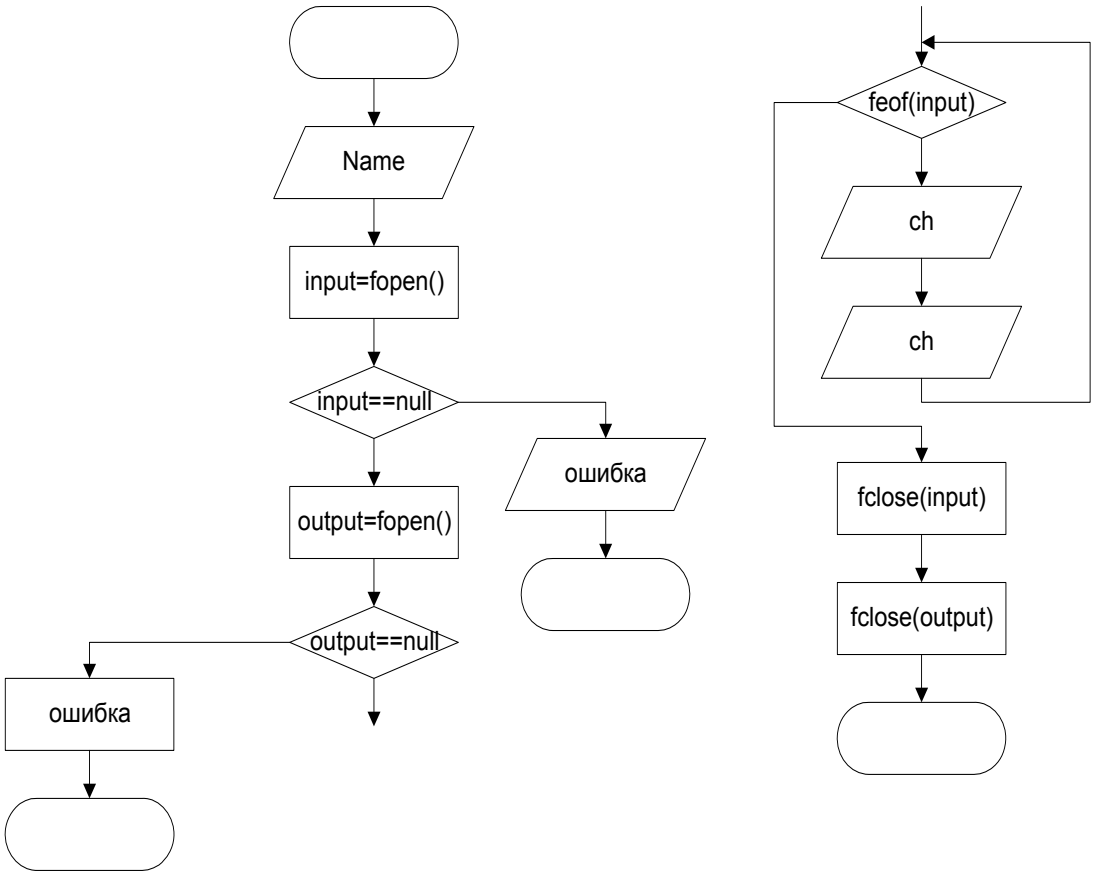


Рис. 9.1. Программа копирования файла

В программе используются следующие функции библиотеки языка Си, предназначенные для работы с текстовыми файлами через указатель на переменную типа FILE.

FILE * fopen(<Имя файла>,<Режим доступа>) ;

fprintf(<Указатель на файл>,"<Строка>",[<Переменные>]) ;

fscanf(<Указатель на файл>,"<Формат>",<Адрес переменных>) ;

fclose(<Указатель на файл>) ;

feof(<Указатель на файл>) ;

Функция fopen возвращает указатель на открытый файл. Если операция закончилась неудачно (файл не удалось открыть), возвращается значение NULL, это обязательно надо проверить, иначе могут быть непредсказуемые сбои. В качестве аргументов функции передаются имя открываемого файла и режим доступа. Режим доступа может быть одним из:

"r" - открыть для чтения (существующий);

"w" - открыть для записи, если файл был, то он уничтожается;

"a" - добавлять в конец файла.

Функции fscanf(...) и fprintf(...) аналогичны функциям scanf(...) и printf(...), но ввод/вывод происходят из/в файла, а не с клавиатуры/экрана. Первым аргументом в них передается указатель на файл с которым необходимо работать.

Функция fclose(...) закрывает открытый файл. Это обязательно необходимо сделать, иначе содержимое файла будет утраченным, а на диске могут образоваться потерянные сектора.

Функция feof(...) возвращает Да (1), если достигнут конец файла, чаще, пользуются обратным условием, - пока не достигнут конец файла.

Пример программы, составленной по блок-схеме (рис. 9.1), приведен на листинге 9.1.

Листинг 9.1. Программа посимвольного копирования файла

```
#include <dos.h>
#include <stdlib.h>
int main(void)
{
   char Name[20],ch ;
   FILE *Input , *Output ;
   printf("\n Копирование файлов.\n Введите имя исх. файла:") ;
   scanf("%s",Name) ;
   printf("\n Копирование %s в RESERV.DAT") ;
   Input=fopen(Name,"r+") ;
   if(Input==NULL)
   {
         printf("\n Ошибка открытия файла \"%s\" ",Name) ; return(0)
;
   } /* Кон. if */
   Output=fopen("RESERV.DAT","w+") ;
   if(Output==NULL)
   {
         printf("\n Ошибка открытия файла \"RESERV.DAT\" ") ;
         return(0) ;
   }/* Кон. if */
   while(!feof(Input))
   {
         fcanf(Input,"%c",&ch) ;
         fprintf(Output,"%c",ch) ;
   } /* Кон. while */
```

```
    fclose(Input) ;
    fclose(Output) ;
    return(1) ;
} /* Кон. main() */
```

Согласно полученному заданию составить алгоритм и программу, вести ее в компьютер, сохранить на диске (F2), проверит правильность работы (Ctrl+F9).

Проверить результаты работы программы с помощью контролирующей программы. Для этого выбирать в контролирующей программе имя исполнимого файла написанной Вами программы.

Проверить работу созданной программы в пошаговом режиме в отладчике, контролируем изменение значения переменных (F7, F4, Ctrl + F7).

Оформить отчет по лабораторной работе.

9.5. Требования к отчету по лабораторной работе

Отчет должен содержать:

1) титульный лист;
2) текст задания, включая заданный Вам формат структуры данных ;
3) блок-схему и листинг программы;
4) пример работы программы.

Лабораторная работа 10, работа с бинарными файлами

10.1. Цель лабораторной работы

1. Освоить работу с бинарными файлами.
2. Освоить методы блочных операций ввода/вывода.
3. Освоить методы обработки бинарных данных, представленных в виде файлов.

10.2. Используемое программное обеспечение

Для выполнения лабораторной работы используется Borland C++ версия 3.1 и контролирующая программа lab10.exe.

10.3. Порядок выполнения лабораторной работы

1. Запустить программу генерации задания lab10 и Borland C.
2. Зарегистрироваться в контролирующей программе.
3. Получить задание (генерируется контролирующей программой).
4. Составить программу.

5. Проверить правильность работы программы с помощью контролирующей программы и ручного расчета.
6. Проверить работу программы в отладчике.

10.4. Пример выполнения лабораторной работы

Контролирующая программа генерирует задание, например, необходимо написать программу чтения и перезаписи в файл матриц. Блок-схема и листинг данной программы и представлены на рис. 10.1 и листинге 10.1.

В программе используются следующие функции работы с бинарными файлами. Идентификатором файла, с которым выполняются операции ввода/вывода, является целая переменная int Handle.

Handle=open(<Имя>,<Режим>,<Атр.>) – открыть файл в указанном режиме, если не открылся Handle== –1

read(Handle,<Куда>,<Сколько>) – прочитать из файла <Сколько> байт то адресу <Куда>.

write(Handle,<Откуда>,<Сколько>) – записать в файл с адреса <Откуда> указанное число байт.

lseek(Handle,<Смещение>,SEEK_SET) – переместить указатель в файле.

eof(Handle) – функция возвращает «истину», если достигнут конец файла.

close(Handle) – закрыть файл.

В функции open() используются следующие атрибуты открытия файла, соединенные арифметической операцией ИЛИ («|»):

O_BINARY – двоичный файл;
O_CREAT – создать;
O_TRUNC – открыть с усечением (обнулить размер);
O_APPEND – открыть для дополнения;
O_RDWR – открыть для чтения и записи.

Атрибуты создаваемого файла рекомендуется установить как:

S_IWRITE – разрешение записи, если он не указан, файл создается с атрибутом только для чтения.

В функции перемещения по файлу lseek() используется атрибут SEEK_SET, указывающий, что отсчет смещения производится от начала файла.

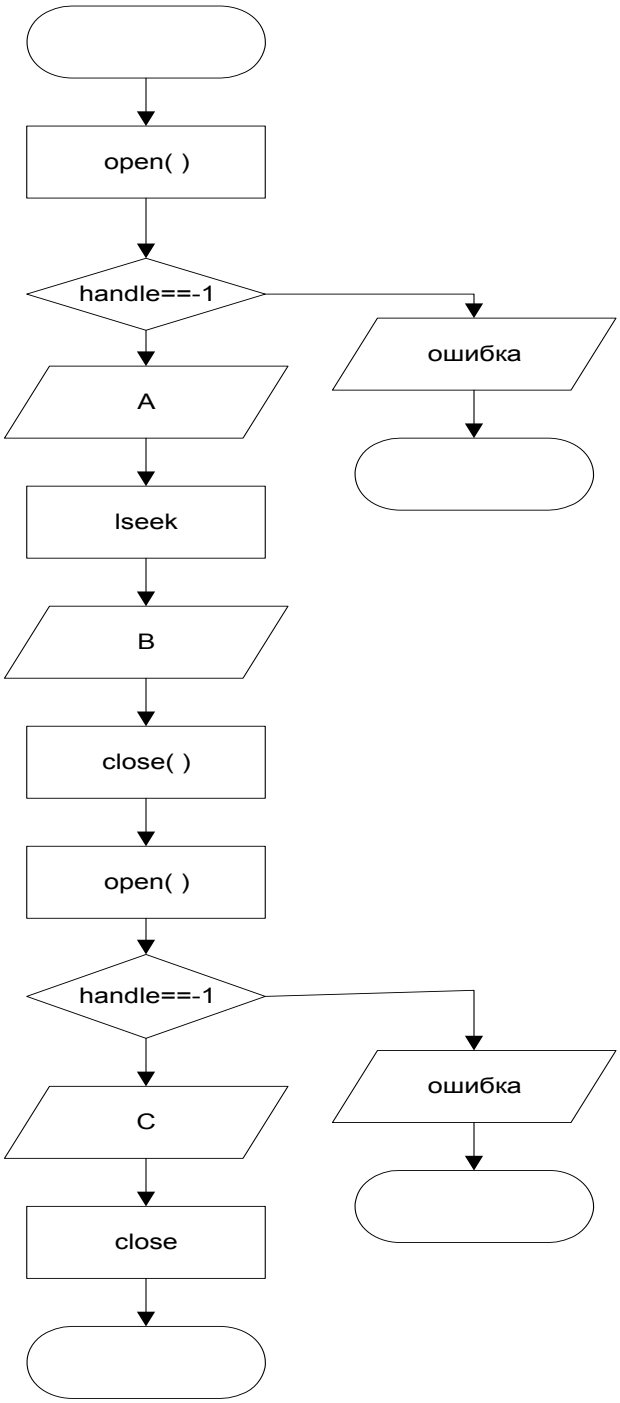


Рис. 10.1. Блок-схема программы работы с бинарными файлами

Листинг 10.1. Программа работы с бинарными файлами

```
#include <stdio.h>
#include <io.h>
#include <fcntl.h>
#include <sys\stat.h>
int main(void)
```

```
{
        float A[10],B[10],C[10] ;
        int Handle ;
        Handle=open("test.dat",O_BINARY) ;
        if(handle==-1)
                {
                        printf("\n Ошибка открытия файла test.dat") ; re-
                        turn(0) ;
                } /* Кон. if() */
        read(Handle,A,sizeof(A)) ;
        lseek(Handle,0L,SEEK_SET) ;
        read(Handle,B,sizeof(B)) ;
        close(Handle);
        Handle=open("res.dat",O_BINARY|O_CREAT|O_TRUNC|
                O_RDWR,  S_IWRITE);
        if(handle==-1)
                {
                        printf("\n Ошибка создания файла res.dat") ;
                        return(0) ;
                } /* Кон. if() */
        write(Handle,C,sizeof(C)) ;
        close(Handle) ;
        return(1) ;
} /* Кон. main() */
```

Согласно полученному заданию составить алгоритм и программу, ввести ее в компьютер, сохранить на диске (F2), проверить правильность работы (Ctrl+F9).

Проверить результаты работы программы с помощью контролирующей программы. Для этого выбираем в контролирующей программе имя исполнимого файла написанной Вами программы.

Проверить работу созданной программы в пошаговом режиме в отладчике, контролируем изменение значения переменных (F7, F4, Ctrl + F7).

Оформит отчет по лабораторной работе.

10.5. Требования к отчету по лабораторной работе

Отчет должен содержать:

1) титульный лист;
2) текст задания, включая заданный Вам формат структуры данных;
3) блок-схему и листинг программы;
4) пример работы программы.

Лабораторная работа 11, Интерпретатор математического (логического) выражения

11.1. Цель лабораторной работы

1. Освоить механизм вызова рекурсивных функций.
2. Освоить возвращение из функции числовых значений.

11.2. Используемое программное обеспечение

Для выполнения лабораторной работы используется Borland C++ версия 3.1 и контролирующая программа lab11.exe.

11.3. Порядок выполнения лабораторной работы

1. Запустить программу генерации задания lab11.exe и Borland C.
2. Зарегистрироваться в контролирующей программе.
3. Получить задание (генерируется контролирующей программой).
4. Составить программу.
5. Проверить правильность работы программы с помощью контролирующей программы и ручного расчета.
6. Проверить работу программы в отладчике.

11.4. Пример выполнения лабораторной работы

Контролирующая программа генерирует задание, например, необходимо написать интерпретатор математического выражения, содержащего указанный набор функций, переменную с именем t, однозначные цифры и любой уровень вложенности скобочек.

На основании сгенерированных контролирующей программой синтаксических правил записи выражений строятся синтаксические диаграммы и блок-схема программы (рис.11.1–11.11).

Выражение S1

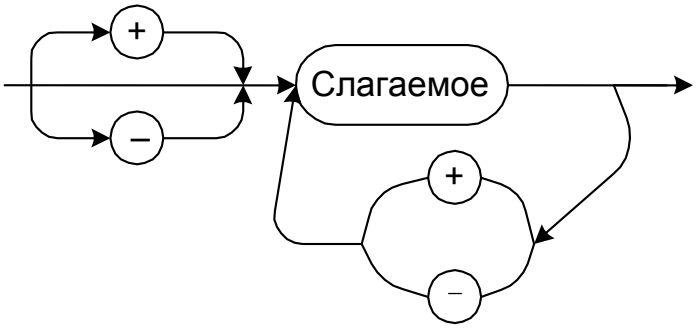


Рис. 11.1. Синтаксическая диаграмма вычисления математического выражения

Слагаемое S2

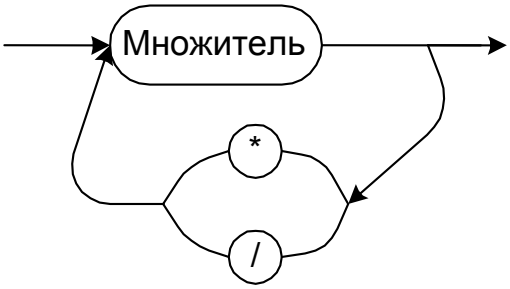


Рис. 11.2. Синтаксическая диаграмма вычисления слагаемого

Множитель S3

Слагаемое S2

Слагаемое S2

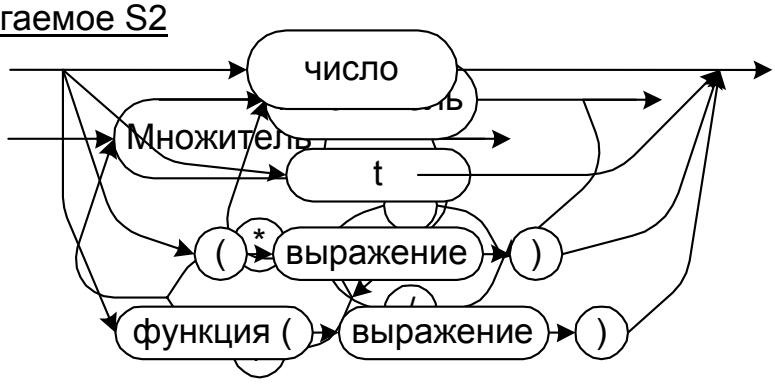


Рис. 11.3. Синтаксическая диаграмма вычисления множителя

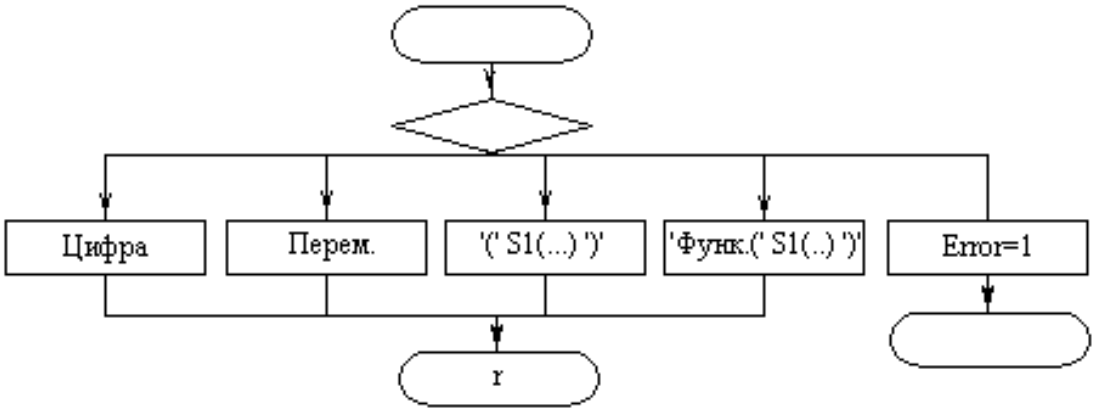


Рис. 11.4. Блок схема функции вычисления множителя S3()

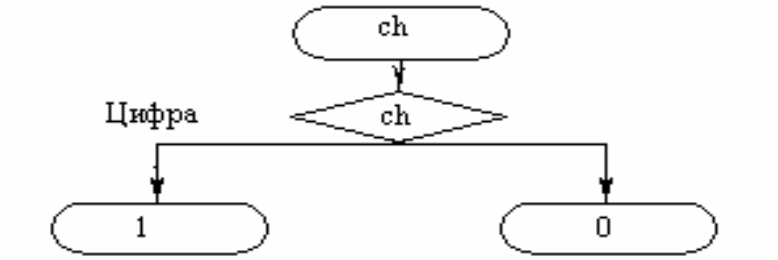


Рис. 11.5. Блок-схема функции определения цифры

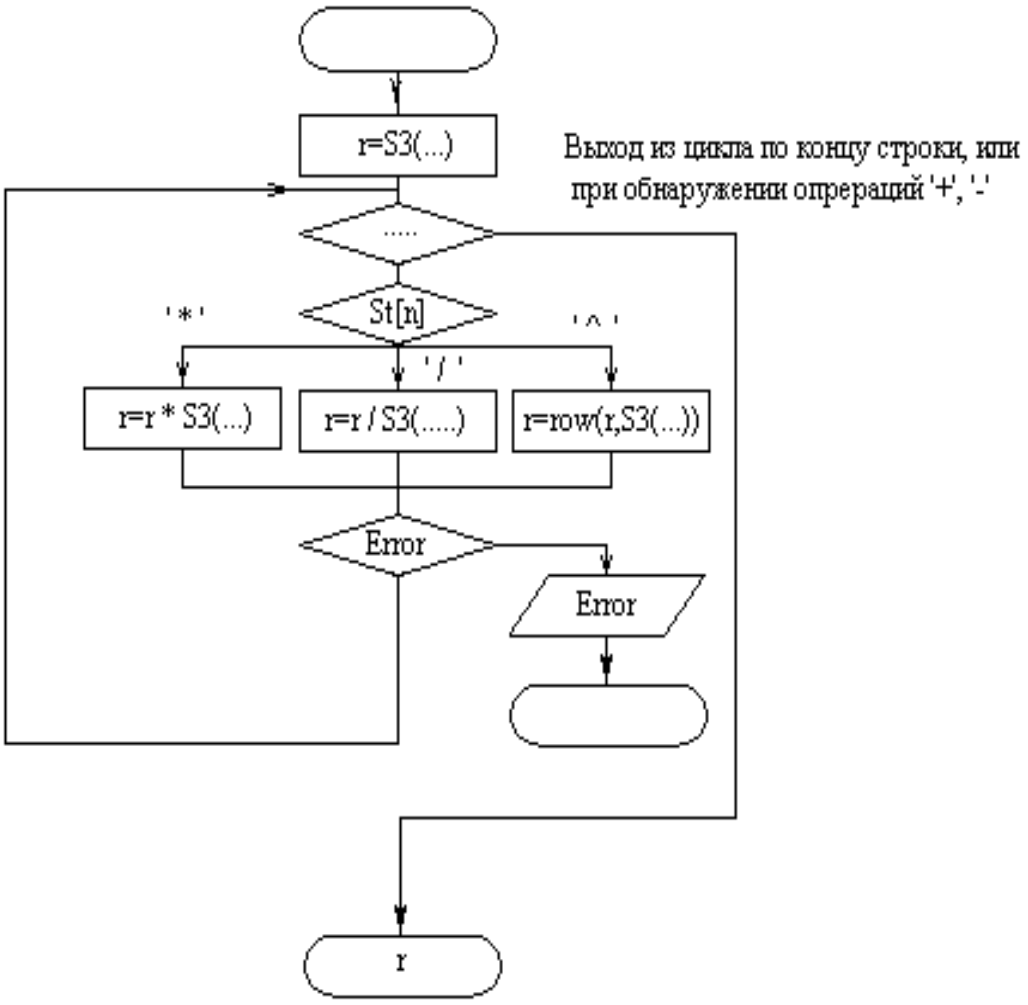


Рис. 11.6. Блок схема функции вычисления множителя S3()

При организации программы Вам предлагается использовать глобальные переменные int n, error, char st[80]. Переменная n указывает на номер обрабатываемого в настоящий момент символа, переменная error – на признак ошибки (0 или 1), строка st содержит запись математического выражения.

Для организации рекурсивной работы функций согласно синтаксическим диаграммам рис. 11.1-11.3, необходимо описать их в следующем порядке и ввести прототип для функции s1(). В программе это должно выглядеть следующим образом:

```
int IfNum(char ch)
        {
                . . .
        }
float s1(void) ;  // прототип функции s1()
float s3(void)
        {
```

```
        . . .
    }
float s2(void)
    {
        . . .
    }
float s1(void)
    {
        . . .
    }
```

Вещественная переменная r в каждой из функций является локальной.

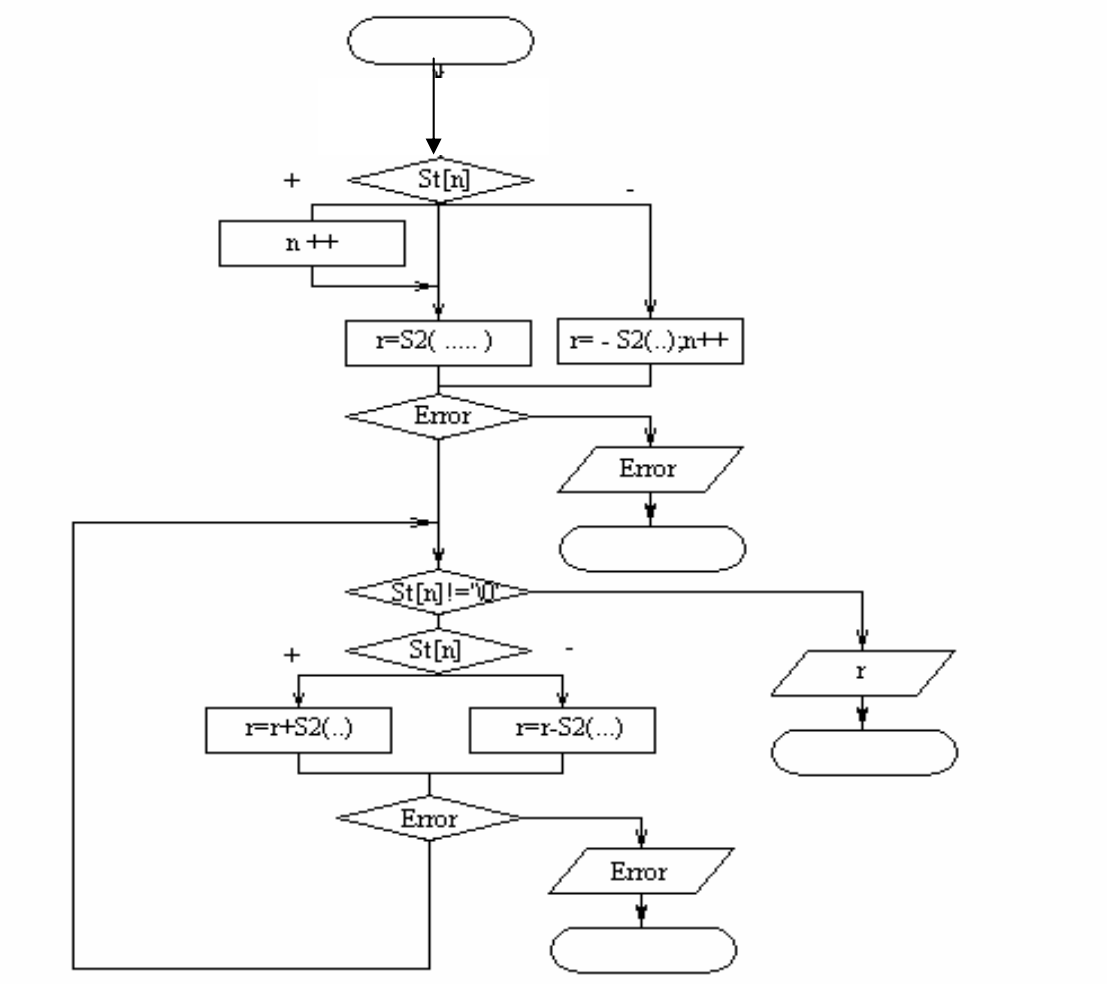


Рис. 11.7. Блок-схема функции вычисления математического выражения

Согласно полученному заданию составить алгоритм и программу, ввести ее в компьютер, сохранить на диске (F2), проверить правильность работы (Ctrl+F9).

Проверить результаты работы программы с помощью контролирующей программы. Для этого выбрать в контролирующей программе имя исполнимого файла.

Проверить работу созданной программы в пошаговом режиме в отладчике, контролируем изменение значения переменных (F7, F4, Ctrl + F7).

Оформить отчет по лабораторной работе.

11.5. Требование к отчету по лабораторной работе

Отчет должен содержать:

1) титульный лист;
2) текст задания, включая заданные Вам правила формирования математического выражения;
3) синтаксические диаграммы;
4) блок-схему и листинг программы;
5) пример работы программы.

Лабораторная работа №12. Работа с указателями

12.1. Цель лабораторной работы

1. Освоить работу с указателями.
2. Освоить возвращение из функции нескольких значений числовых значений через указатели.
3. Освоить работу с проектом и подключением внешних функций.
4. Разобраться с моделями памяти в Си.

12.2. Используемое программное обеспечение

Для выполнения лабораторной работы используется Borland C версия 3.1 и контролирующая программа lab12.exe.

12.3. Порядок выполнения лабораторной работы

1. Запустить программу генерации задания lab12.exe и Borland C.
2. Зарегистрироваться в контролирующей программе.
3. Получить задание (генерируется контролирующей программой).
4. Составить программу.
5. Проверить правильность работы программы с помощью контролирующей программы и ручного расчета.
6. Проверить работу программы в отладчике.

12.4. Пример выполнения лабораторной работы

Контролирующая программа генерирует задание, например, необходимо ввести две строки и скопировать их в одну. Данное дейст-

вие должно выполняться в собственной функции, аргументы и результаты передаются через указатели. Кроме этого для проверки Вам предлагается еще одна функция, выдаваемая контролирующей программой в виде объектного модуля, параметры в которую передаются через указатели.

Иллюстрация задания приведена на рис. 12.1.

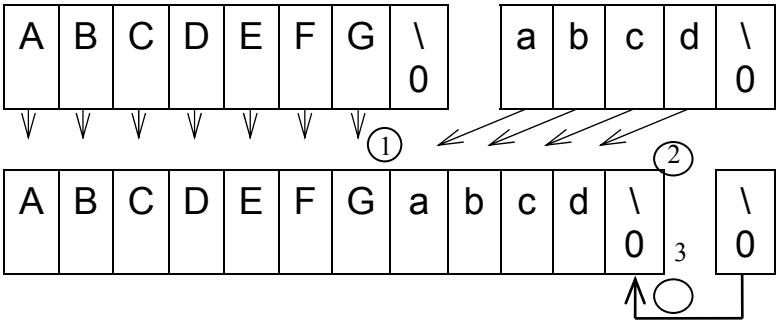


Рис. 12.1. Организация копирования строк

В данном примере в функцию CopyStr() указателями передаются два массива. Язык си++ не осуществляет контроля за размерами массивов, это необходимо учитывать в программе. Признак конца строки является символ '\0'.

Листинг 12.1. Фрагмент программы слияния двух строк

```
#include <stdio.h>
void CopyStr(char *st1,char *st2,char *Rez)
{
        Rez[i]=st1[i];
        . . .  /* Символ '\0' обозначает конец строки.*/
} /* Кон. CopyStr() */
void main(void)
{
        chat s1[20],s2[20],rezultat[40] ;
        int n ;
        printf("\n Введите две строки") ;
        scatf("%20s",s1) ;
        scanf("%20s",s2) ;
        CopyStr(s1,s2,resultat,&n) ;
        printf("\n Результат : %s",resultat) ;
        Test(s1,s2,result,&n) ;
} /* кон. main() */
```

Функция Test() предлагается Вам в виде объектного модуля, для ее подключения Вам необходимо включить в свою программу прототип, содержащий описание этой функции void Test(char *st1, char *st2,

char *st3, int n), имена аргументов функции не обязательно должны совпадать с именами переменных программы. После этого Вам необходимо создать проект и указать в нем, какой файл Вы собираетесь транслировать и какой файл подключается на этапе компоновки. **Внимание!** Модель памяти, с при которой Вы собираетесь транслировать Вашу программу, должна совпадать с моделью памяти, подключаемого Вами модуля.

Проверить результаты работы программы с помощью контролирующей программы. Для этого выбирать в контролирующей программе имя исполнимого файла.

Проверить работу созданной программы в пошаговом режиме в отладчике, контролируя изменение значения переменных (F7, F4, Ctrl + F7).

Оформить отчет по лабораторной работе.

12.1. Требование к отчету по лабораторной работе

Отчет должен содержать:

1) титульный лист;
2) текст задания;
3) блок-схему и листинг программы;
4) пример работы программы.

Приложение 1.
Работа с контролирующими программами

При запуске на выполнение контролирующая программа пытается установить соединение с сервером. Вид окна, соответствующего данному режиму, представлен на рис. П1.1. В том случае, если соединение не установлено, то программа предлагает выбрать адрес сервера из списка или ввести его вручную.

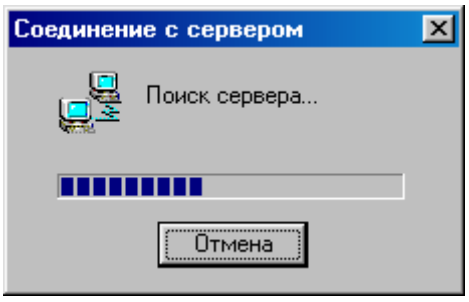


Рис. П1.1. Вид окна поиска сервера

Если соединение с сервером не установлено, это означает, что сервер выключен или Вы не имеете связи с компьютером, на котором он запущен.

После соединения с сервером предлагается выбрать группу и сформировать бригаду студентов. Вид окна выбора приведен на рис. П1.2. Выбор учетных записей каждого из студентов подтверждается паролем.

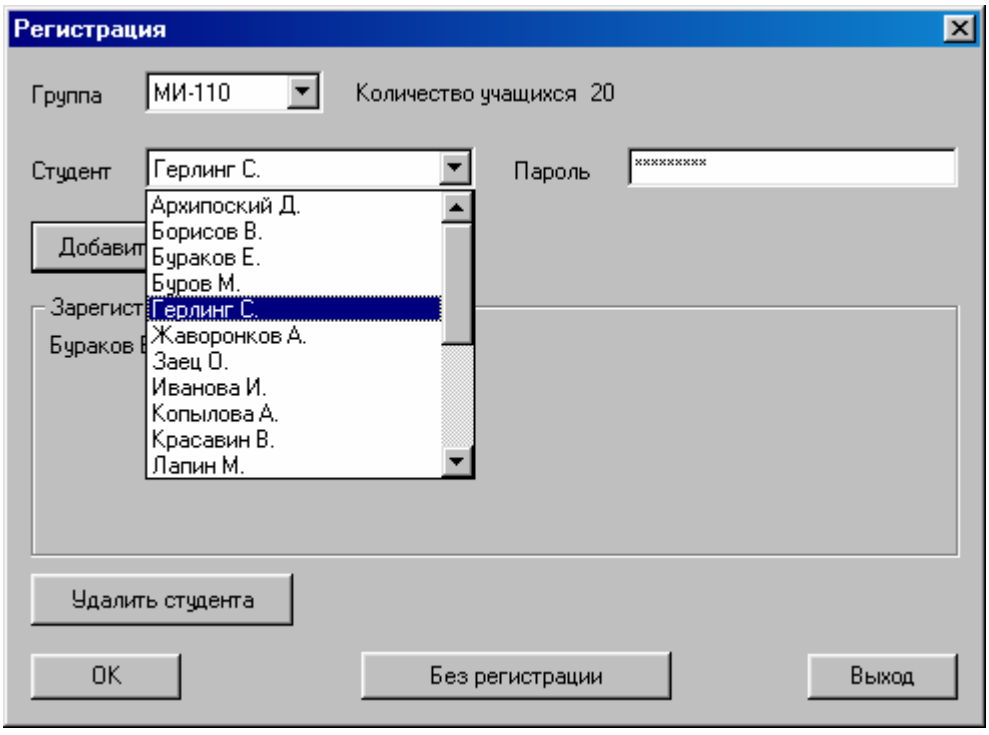


Рис. П1.2. Вид окна регистрации студентов

В случае, если Вы выполняете лабораторную работу без регистрации (в автономном режиме), Ваши результаты не фиксируются автоматически на сервере и Вам необходимо предъявлять их преподавателю в очном режиме.

Приложение 2.

Работа с интегрированной средой Borland C версия 3.1

При работе с интегрированной средой Borland C++ все действия можно выполнить, либо выбрав пункт меню, либо нажав установленные функциональные клавиши:

'+' – одновременное нажатие нескольких клавиш.

',' – последовательное нажатие клавиш.

F1	– помощь.
Cntrl+F1	– помощь по конкретному слову.
F10	– выход в меню.
Alt+X	– выход в DOS.
F10,'F','N'	– создание нового файла.
F3	– загрузка существующего файла.
F2	– запись редактируемого файла на диск.
Ctrl+'K','B'	– отметить начало блока.
Ctrl+'K','K'	– отметить конец блока.
Ctrl+'K','C'	– копировать отмеченный блок в указанное место.
Ctrl+'Y'	– удалить строку.
F6	– переход между окнами.
Alt+<Номер>	– переход к окну с номером.
Alt+F3	– закрытие окна.
F9	– трансляция программы (проверить ошибки).
Cntrl+F9	– транслировать и запустить на выполнение.
F7	– пошаговое выполнение программы.
F4	– выполнить до указанного места.
Cntrl+F7	– контроль значений переменных.
Alt+F5	– показать экран.

Возможные проблемы

Borland C не запускается, выдается сообщение: Disk full! Not enough swap space. При работе в компьютерных классах УИЦ ИТТ, скорее всего, Вы запустили программу Borland C++ с защищенного от записи диска, и он не может записать на него свой временный файл. В этом случае Вам необходимо сделать ярлык для программы и указать в нем в поле «рабочий каталог» диск, доступный для записи, например, ”d:\temp”. Аналогичная проблема может возникнуть, если на Вашем рабочем диске будет недостаточно свободного места. Для нормальной работы Borland C++ необходимо порядка 1 Мбайт свободного места.

Если при попытке транслировать программу выдается большое количество сообщений об **ошибках компоновки**, причем Вы уверены, что имена вызываемых функций указанны правильно. Наиболее вероятно, в опциях Borland C++, в разделе директории неправильно указан путь до каталогов include и lib.

Если при трансляции программы выдается сообщение **о невозможности создать файла OBJ**, это вызвано, невозможностью записать файл по указанному пути, например, каталог для исполнимых файлов в настройках Borland C не существует. Для устранения ошибки требуется создать необходимый каталог или исправить пути в опциях Borland C, раздел директории.

При запуске программы на трансляцию **транслируется не Ваш файл, а другая программа**. Это вызвано тем, что у Вас автоматически открылся чужой проект, находящийся в Вашем рабочем каталоге. Его необходимо закрыть, после чего можно продолжить работу.

Приложение 3
Требования к оформлению блок-схем

При оформлении блок-схем необходимо придерживаться определенных требований на оформление блок-схем, приведенных в ГОСТ. Для оформления блок-схем рекомендуется воспользоваться пакетом VisioPro или средствами векторной графики Ms Word. Допустимо также выполнение блок-схем от руки.

Типовые элементы для оформления блок-схем приведены на рис. П3.1.

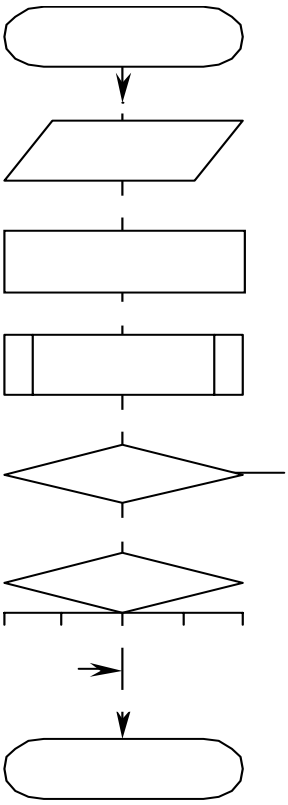


Рис. П3.1. Типовые элементы для записи блок-схем

Литература

1. Аксёнкин М.А., Целобёнок О.Н. Язык С. Минск: "Уіверсітэцкае", 1995.
2. Бочков С.О., Субботин Д.М. Язык программирования Си для персонального компьютера. М.: Радио и связь, 1990.
3. Прокофьев Б.П. и др. Графические средства Turbo C и Turbo C++. М.: "Финансы и статистика", 1992.
4. Справочник по функциям Borland C++ 3.1/4.0. Киев: Диалектика, 1994.
5. Страуструп Б. Язык программирования Си++. М.: Радио и связь, 1991.
6. Красов А.В. Программирование на языке Си. Ч.1 / СПбГУТ, СПб, 2001.
7. Красов А.В. Программирование на языке Си. Ч2. Использование графики и программная обработка данных / СПбГУТ, СПб, 2001.
8. Красов А.В. Программирование на Borland C под Ms Windows / СПбГУТ, СПб, 2001.

Оглавление