

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования «Рязанский государственный радиотехнический
университет имени В.Ф. Уткина»

Кафедра «ЭВМ»

Отчёт о лабораторной работе № 10
«Структуры»
по курсу
«ОАиООП»

Выполнили:
студенты группы 247
Сидоров А.С.,
Воробьёва М.П.

Проверил:
асс. каф. ЭВМ
Тарасов А.С.,
асс. каф. ЭВМ
Панина И.С.

Рязань, 2023

Цель работы: приобретение навыков разработки алгоритмов и программ с использованием структур.

Вариант №2

Практическая часть

1. Постановка задачи

Описать структуру с именем MARSH, содержащую следующие поля:

- название начального пункта маршрута;
- название конечного пункта маршрута;
- номер маршрута.

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из восьми элементов типа MARSH;
- вывод таблицы на экран;
- записи упорядочить по номерам маршрутов;
- вывод отсортированной таблицы на экран;
- вывод на экран информации о маршруте, номер которого введен с клавиатуры;
- если таких маршрутов нет, выдать на дисплей соответствующее сообщение.

Входные данные: массив из 8 структур MARSH, номер маршрута для поиска.

Выходные данные: 2 таблицы со структурами, найденная по вводимому номеру маршрута структура.

Рекомендуемый вид экрана во время выполнения программы:

Введите информацию о маршруте: ...

Изначальная таблица: ...

Отсортированная таблица: ...

Введите номер маршрута для поиска: ...

Если он найден, вывод информации о нём, иначе вывод: такого маршрута нет.

2. Математическая модель

Массив маршрутов будет отсортирован по возрастанию пузырьковой сортировкой по номеру маршрута. Так как поиск маршрута по номеру будет проводится только после сортировки, будет применён бинарный поиск.

3. Разработка алгоритма

Блок-схемы программы представлены на рисунках 1 и 2.

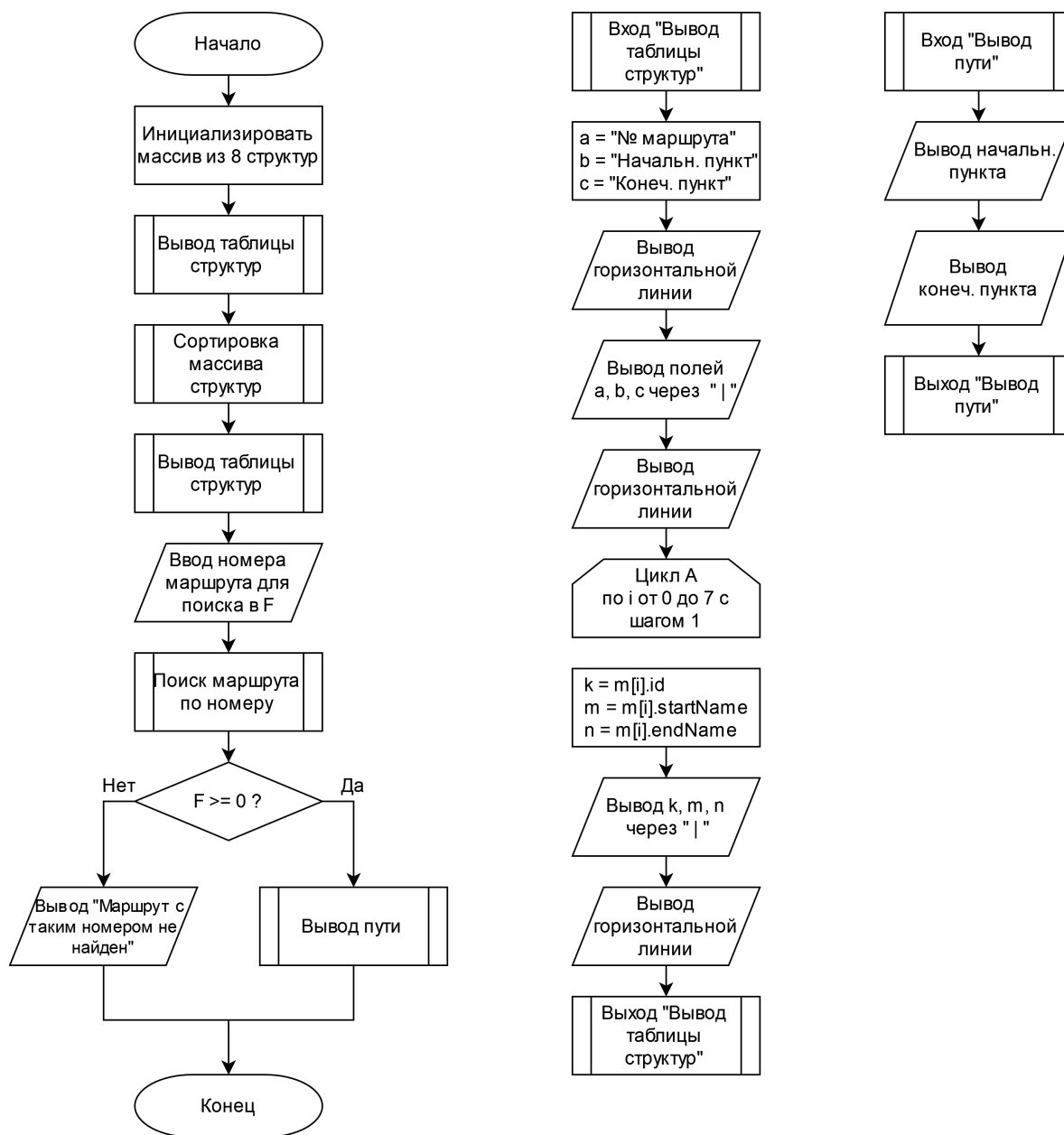


Рисунок 1 – Блок-схемы программы, часть 1

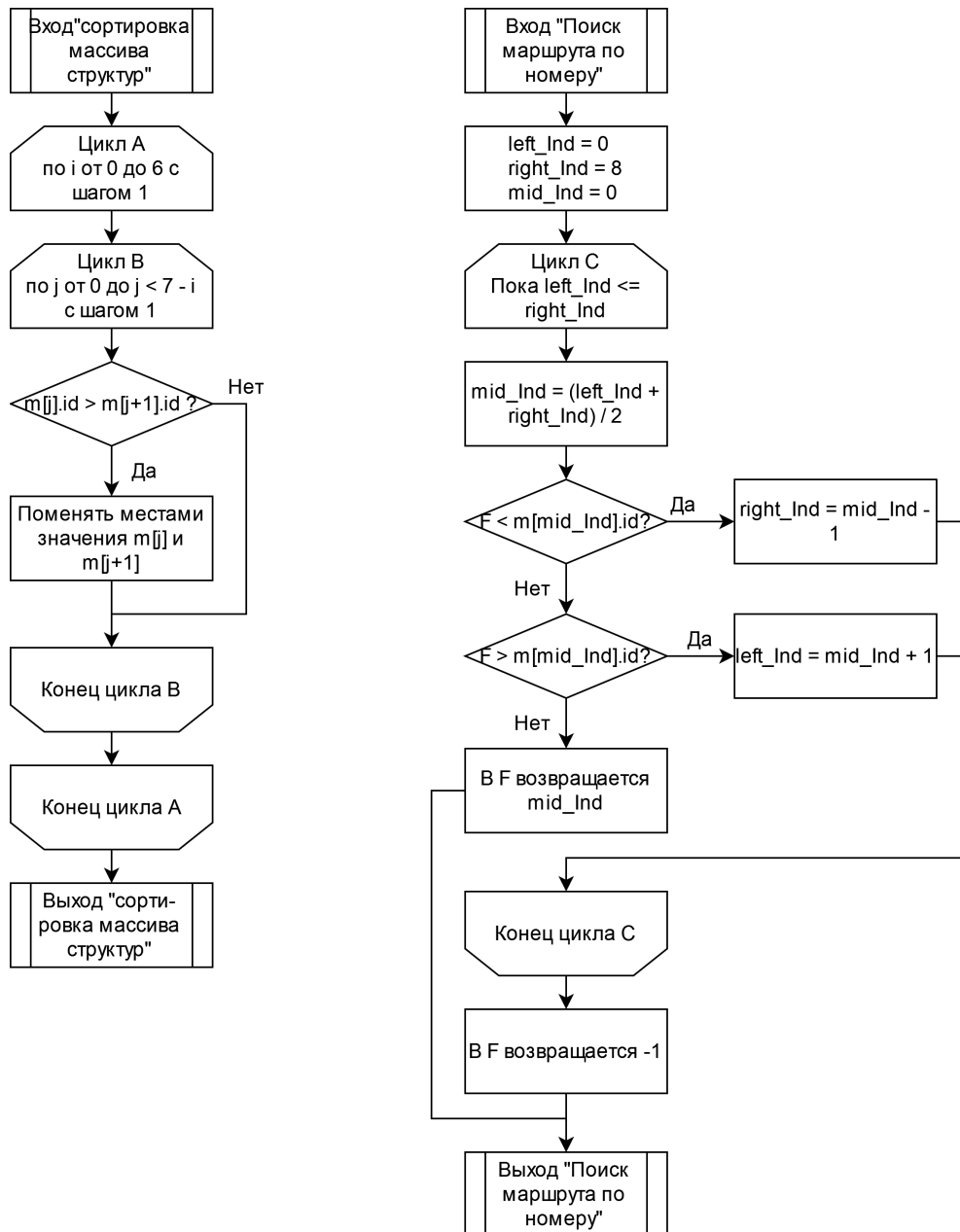


Рисунок 2 – Блок-схемы программы, часть 2

4. Программирование

Код файла Marsh.h с описанием структуры Marsh:

```

#pragma once
struct Marsh
{
    // Название начального пункта маршрута
    char startName[15];

    // Название конечного пункта маршрута
    char endName[15];

    // Номер маршрута
    int id;
};
  
```

Код основной программы представлен ниже.

```
#include <iostream>
#include <iomanip>
#include "Marsh.h"
using std::cout;
using std::cin;
using std::endl;
using std::setw;

/*
Описать структуру с именем MARSH, содержащую следующие поля:
    - название начального пункта маршрута;
    - название конечного пункта маршрута;
    - номер маршрута.
Написать программу, выполняющую следующие действия:
    - ввод с клавиатуры данных в массив, состоящий из восьми
      элементов типа MARSH;
    - вывод таблицы на экран;
    - записи упорядочить по номерам маршрутов;
    - вывод отсортированной таблицы на экран;
    - вывод на экран информации о маршруте, номер которого
      введен с клавиатуры;
    - если таких маршрутов нет, выдать на дисплей
      соответствующее сообщение.
*/

void PrintMarshesTable(Marsh* marshes, int dbSize);

void PrintMarshInfo(Marsh marsh);

void SortMarshesById(Marsh* marshes, int dbSize);

int BinaryIdSearch(int toFind, Marsh* marshes, int dbSize);

int main()
{
    setlocale(LC_ALL, "Russian");

    const int dbSize = 8;
    struct Marsh marshes[dbSize];

    for (int i = 0; i < dbSize; i++)
    {
        cout << "Введите название начального пункта маршрута:\n";
        cin >> marshes[i].startName;

        cout << "Введите название конечного пункта маршрута:\n";
        cin >> marshes[i].endName;

        cout << "Введите номер маршрута: ";
        cin >> marshes[i].id;
    }

    cout << "До сортировки" << endl;
    PrintMarshesTable(marshes, dbSize);

    // Сортировка по номеру маршрута
    SortMarshesById(marshes, dbSize);
```

```

cout << "\nПосле сортировки" << endl;
PrintMarshesTable(marshes, dbSize);

cout << "\nВведите номер маршрута для поиска" << endl;
int toFind;
cin >> toFind;

// Так как список отсортирован, можно применить
// бинарный поиск
int found = BinaryIdSearch(toFind, marshes, dbSize);

if (found >= 0)
{
    PrintMarshInfo(marshes[found]);
}
else
{
    cout << "Маршрут с номером " << toFind << " не найден" << endl;
}

system("pause");
}

void PrintMarshesTable(Marsh* marshes, int dbSize)
{
    const int width = 15;
    // 3 основных столбца + 4 символа '|' + '\0'
    char horizontalLine[width * 3 + 5] =
    {
        "-----"
    };
    // Выравнивание по левой границе
    cout.setf(std::ios::left);

    // Шапка
    cout << horizontalLine << endl;
    cout << "|"
        << setw(width) << "Номер маршрута" << "|"
        << setw(width) << "Начальный пункт" << "|"
        << setw(width) << "Конечный пункт" << "|"
        << endl;
    cout << horizontalLine << endl;

    for (int i = 0; i < dbSize; i++)
    {
        cout << "|"
            << setw(width) << marshes[i].id << "|"
            << setw(width) << marshes[i].startName << "|"
            << setw(width) << marshes[i].endName << "|"
            << endl;

        cout << horizontalLine << endl;
    }
}

void PrintMarshInfo(Marsh marsh)
{
    cout << "Начальный пункт: " << marsh.startName << endl;
    cout << "Конечный пункт: " << marsh.endName << endl;
}

```

```

void SortMarshesById(Marsh* marshes, int dbSize)
{
    for (int i = 0; i < dbSize - 1; i++)
    {
        for (int j = 0; j < dbSize - i - 1; j++)
        {
            if (marshes[j].id > marshes[j + 1].id)
            {
                struct Marsh temp = marshes[j];
                marshes[j] = marshes[j + 1];
                marshes[j + 1] = temp;
            }
        }
    }
}

int BinaryIdSearch(int toFind, Marsh* marshes, int dbSize)
{
    int leftIndex = 0;
    int rightIndex = dbSize;
    int midIndex = 0;

    while (leftIndex <= rightIndex)
    {
        midIndex = (leftIndex + rightIndex) / 2;

        if (toFind < marshes[midIndex].id)
        {
            rightIndex = midIndex - 1;
        }
        else if (toFind > marshes[midIndex].id)
        {
            leftIndex = midIndex + 1;
        }
        else return midIndex;
    }

    // Маршрут не найден. Предполагается, что
    // маршрутов с отрицательным номером не будет
    return -1;
}

```

5. Тестирование

Результат программы представлен на рисунках 3 и 4.

```
D:\Code\Lab9-10\x64\Debug\Lab10.exe
Маршрут 0
Введите название начального пункта маршрута:
a
Введите название конечного пункта маршрута:
b
Введите номер маршрута: 5

Маршрут 1
Введите название начального пункта маршрута:
c
Введите название конечного пункта маршрута:
d
Введите номер маршрута: 1

Маршрут 2
Введите название начального пункта маршрута:
e
Введите название конечного пункта маршрута:
f
Введите номер маршрута: 3246

Маршрут 3
Введите название начального пункта маршрута:
h
Введите название конечного пункта маршрута:
d
Введите номер маршрута: 45

Маршрут 4
Введите название начального пункта маршрута:
m
Введите название конечного пункта маршрута:
p
Введите номер маршрута: 23

Маршрут 5
Введите название начального пункта маршрута:
v
Введите название конечного пункта маршрута:
h
Введите номер маршрута: 666

Маршрут 6
Введите название начального пункта маршрута:
j
Введите название конечного пункта маршрута:
k
Введите номер маршрута: 322

Маршрут 7
Введите название начального пункта маршрута:
b
Введите название конечного пункта маршрута:
a
Введите номер маршрута: 2

До сортировки
-----
|Номер маршрута |Начальный пункт|Конечный пункт |
|5              |a              |b              |
-----
```

Рисунок 3 – Результат программы, часть 1


```
D:\Code\Lab9-10\x64\Debug\Lab10.exe
Введите номер маршрута: 2
До сортировки
-----
|Номер маршрута |Начальный пункт|Конечный пункт |
-----
|5              |a              |b              |
-----
|1              |c              |d              |
-----
|3246           |e              |f              |
-----
|45             |h              |d              |
-----
|23             |m              |p              |
-----
|666            |v              |h              |
-----
|322            |j              |k              |
-----
|2              |b              |a              |
-----

После сортировки
-----
|Номер маршрута |Начальный пункт|Конечный пункт |
-----
|1              |c              |d              |
-----
|2              |b              |a              |
-----
|5              |a              |b              |
-----
|23             |m              |p              |
-----
|45             |h              |d              |
-----
|322            |j              |k              |
-----
|666            |v              |h              |
-----
|3246           |e              |f              |
-----

Введите номер маршрута для поиска
666
Начальный пункт: v
Конечный пункт: h
Для продолжения нажмите любую клавишу . . .
```

Рисунок 4 – Результат программы, часть 2

Заключение

В этой лабораторной работе были получены навыки разработки алгоритмов и программ с использованием структур.