

ЛАБОРАТОРНАЯ РАБОТА №7

ИССЛЕДОВАНИЕ АЛГОРИТМОВ ОБРАБОТКИ ОДНОМЕРНЫХ ДИНАМИЧЕСКИХ МАССИВОВ СТРУКТУР

1. Цель работы:

Исследование способов описания структур данных на языке С. Приобретение практических навыков разработки алгоритмов обработки табличной информации, представленной в виде динамического массива структур на языках С/С++.

2. Постановка задачи и вариант задания:

Описать структуру с именем route, содержащую следующие поля: порядковый номер, номер маршрута, начальный пункт, конечный пункт, стоимость проезда, время в пути. Данные сортировать по полю «стоимость проезда». Выполнить поиск количества маршрутов, курсирующих по кругу (у которых названия начального и конечного пунктов совпадают).

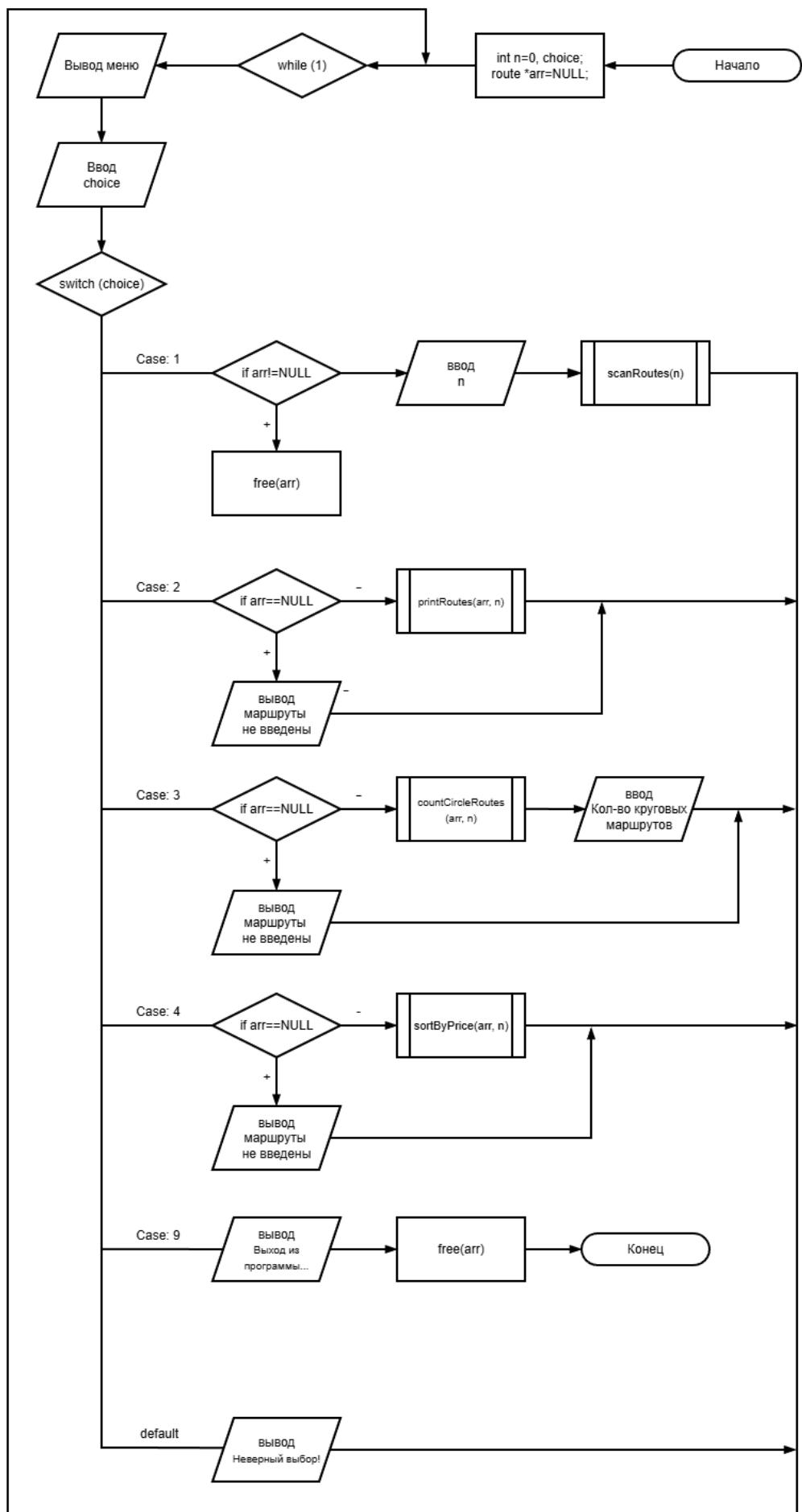
3. Краткие теоретические сведения

Для выполнения лабораторной работы необходимо изучить особенности объявления, описания и вызова структур в языках С/С++.

ХОД РАБОТЫ

4. Структурная схема алгоритма

Структурная схема алгоритма, а также всех функций представлены на рисунках 7.1 – 7.5:

Рисунок 7.1 – Структурная схема функции `main()`

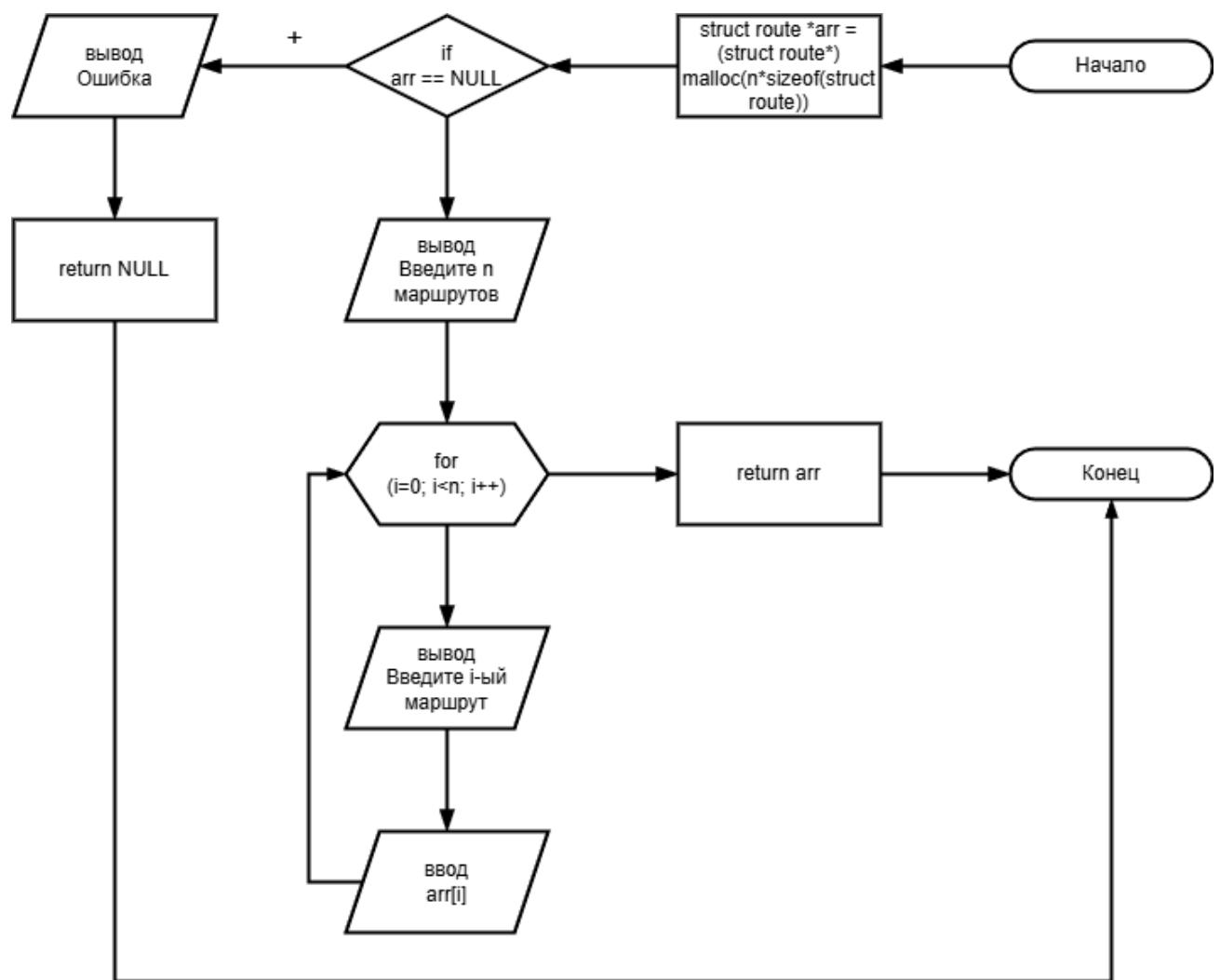


Рисунок 7.2 – Структурная схема функции `scanRoutes(n)`

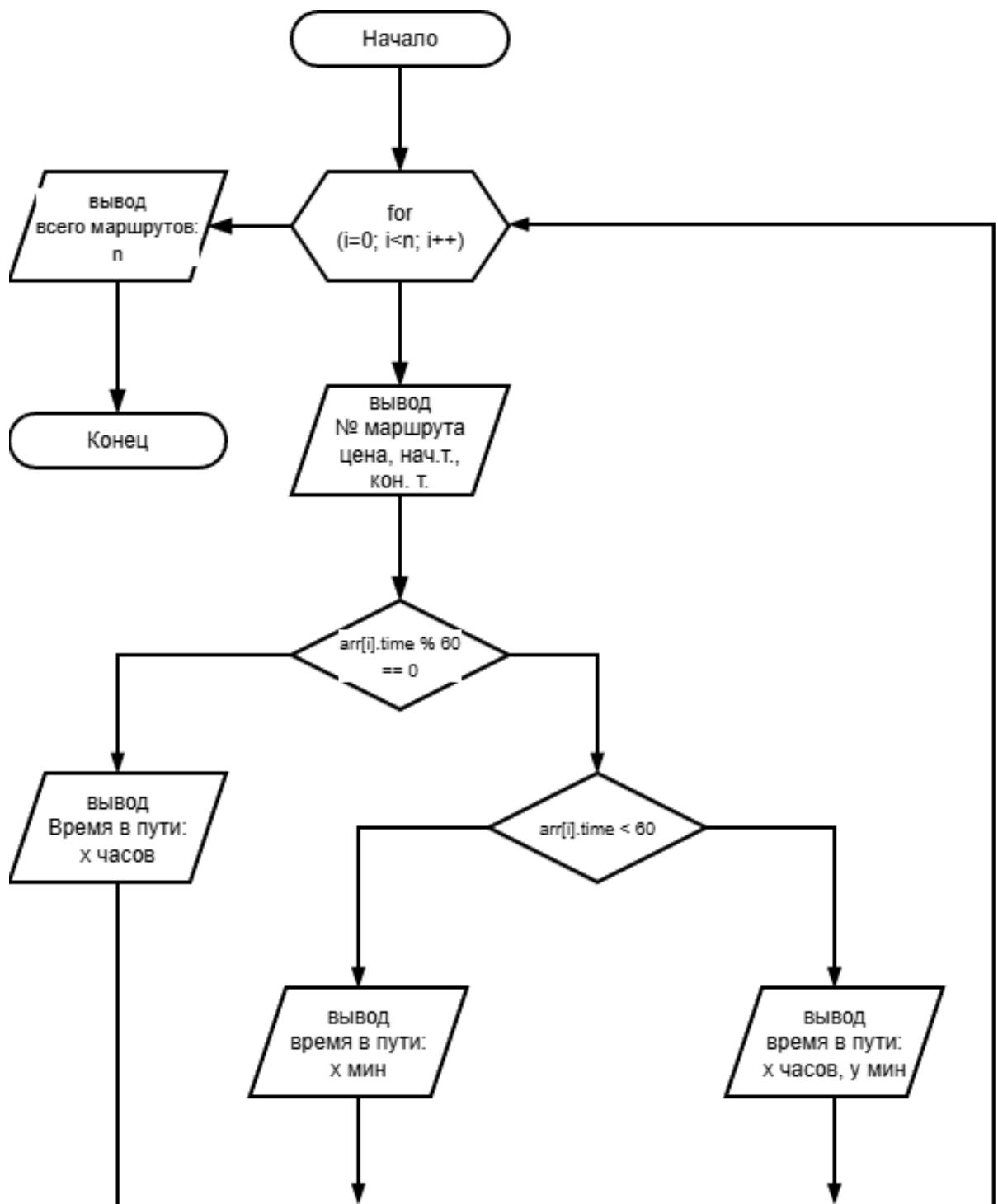


Рисунок 7.3 – Структурная схема функции printRoutes(*arr, n)

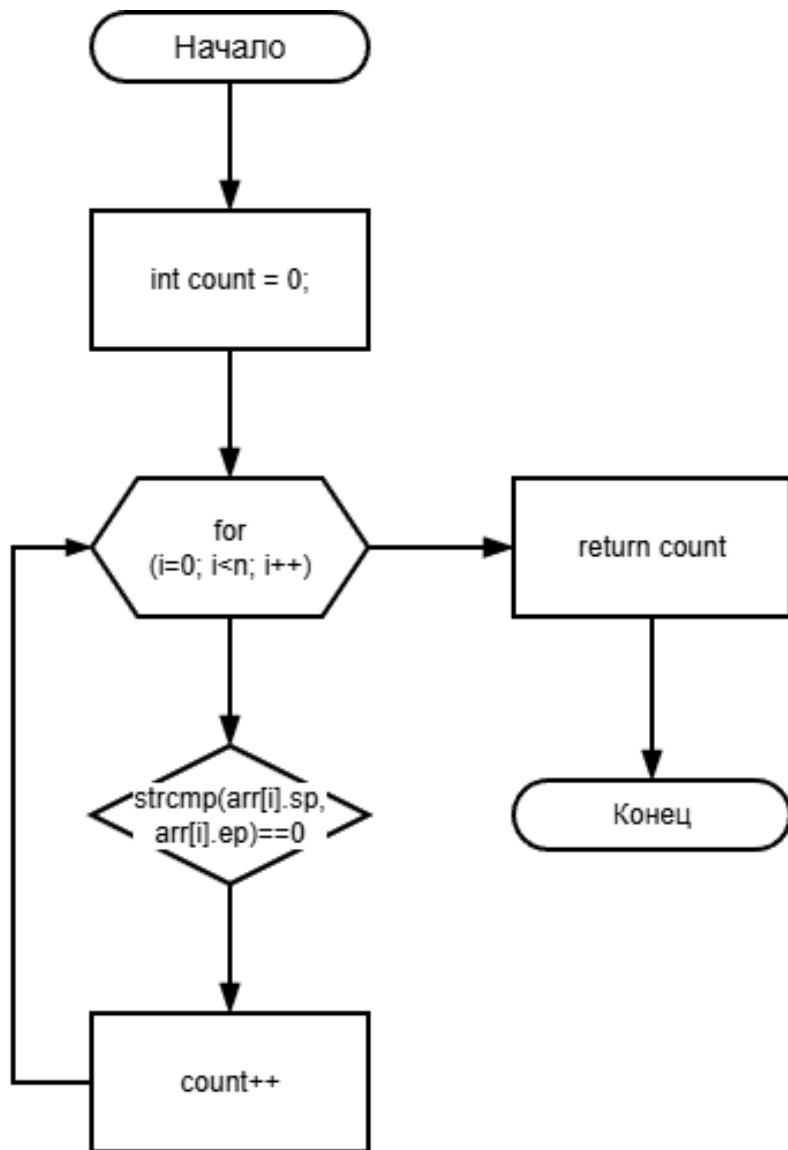


Рисунок 7.4 – Структурная схема функции `countCircleRoutes(*arr, n)`

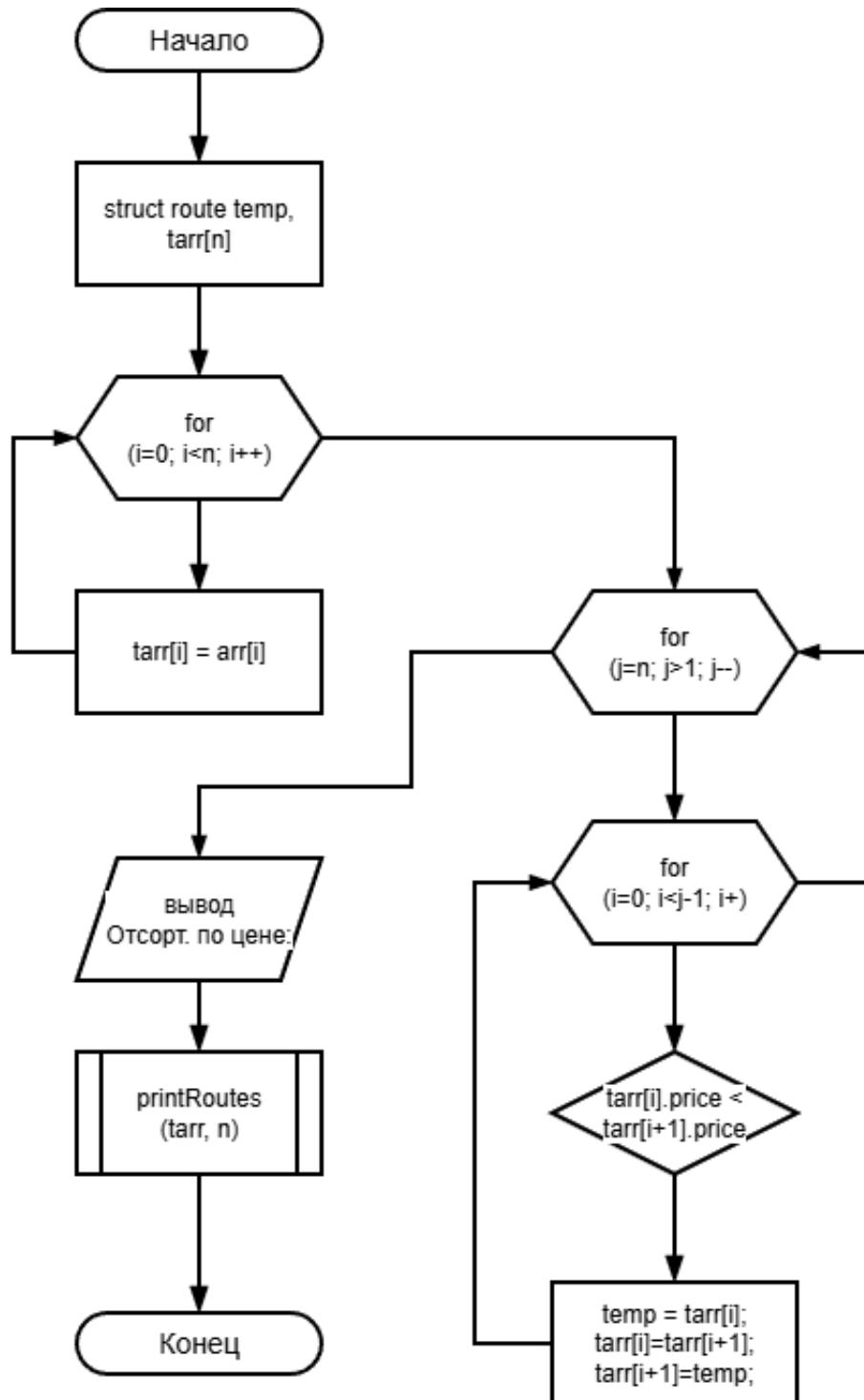


Рисунок 7.5 – Структурная схема функции `sortByPrice(*arr, n)`

5. Текст кода на языке С. Тестирование и отладка программы

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

// Создание структуры "route"
struct route {
    int number;
    int routeNumber;
    char startPoint[100];
    char endPoint[100];
    float price;
    int time;
};

// Объявление функций
struct route* scanRoutes(int n);
void printRoutes(struct route *arr, int n);
int countCircleRoutes(struct route *arr, int n);
void sortByPrice(struct route *arr, int n);

int main() {
    int choice, n = 0;
    struct route *arr=NULL;
    while (1) {
        printf("\n===== М Е НЮ =====\n");
        printf("1. Ввести все маршруты\n");
        printf("2. Вывести все маршруты\n");
        printf("3. Найти количество круговых маршрутов\n");
        printf("4. Отсортировать маршруты по стоимости проезда\n");
        printf("9. Выход\n");
        printf("-> ");
        scanf("%d", &choice);
        printf("\n");

        switch(choice) {
            case 1:
                if (arr != NULL) {
                    free(arr);
                }
                n = 0;
                printf("Введите количество маршрутов: ");
                scanf("%d", &n);
                if (n <= 0) {
                    printf("Количество маршрутов должно быть > 0");
                    break;
                }
                arr = scanRoutes(n);
                break;
        }
    }
}

```

```

        case 2:
            if (arr == NULL) {
                printf("Маршруты не введены! \n");
            }
            else {
                printf("Все маршруты:\n");
                printRoutes(arr, n);
            }
            break;

        case 3:
            if (arr == NULL) {
                printf("Маршруты не введены! \n");
            }
            else {
                printf("Количество круговых маршрутов:
%d\n", countCircleRoutes(arr, n));
            }
            break;

        case 4:
            if (arr == NULL) {
                printf("Маршруты не введены! \n");
            }
            else {
                sortByPrice(arr, n);
            }
            break;

        case 9:
            free(arr);
            printf("\nВыход из программы . . .");
            return 0;

        default:
            printf("Неверный выбор! Попробуйте снова.");
            break;
    }

}

struct route* scanRoutes(int n) {
    struct route *arr = (struct route*) malloc(n * sizeof(struct route));
    if (arr == NULL) {
        printf("Ошибка выделения памяти! \n");
        return NULL;
    }

    printf("Введите %d маршрутов: \n", n);
    for (int i = 0; i < n; i++) {

```

```

        arr[i].number = i+1;
        printf("Введите номер маршрута: ");
        scanf("%d", &arr[i].routeNumber);
        while (getchar() != '\n');

        printf("Введите начальную точку маршрута (до 100
символов): ");
        gets(arr[i].startPoint);

        printf("Введите конечную точку маршрута (до 100 символов):
");
        gets(arr[i].endPoint);

        printf("Введите цену маршрута: ");
        scanf("%f", &arr[i].price);

        printf("Введите время в пути (в минутах): ");
        scanf("%d", &arr[i].time);

        printf("-----\n");
    }

    return arr;
}

void printRoutes(struct route *arr, int n) {
    printf("-----\n");

    for (int i = 0; i < n; i++) {
        printf("%d Номер маршрута: %d\n", arr[i].number,
arr[i].routeNumber);
        printf("Начальная точка: %s \nКонечная точка: %s\n",
arr[i].startPoint, arr[i].endPoint);
        printf("Цена проезда: %.2f\n", arr[i].price);
        if (arr[i].time % 60 == 0) {
            printf("Время в пути: %dч\n", arr[i].time/60);
        }
        else if (arr[i].time < 60) {
            printf("Время в пути: %dmин\n", arr[i].time);
        }
        else {
            printf("Время в пути: %dч %dmин\n", arr[i].time/60,
arr[i].time%60);
        }
        printf("-----\n");
    }

    printf("Всего маршрутов: %d\n", n);
}

int countCircleRoutes(struct route *arr, int n) {

```

```
int count = 0;

for (int i = 0; i < n; i++) {
    if (strcmp(arr[i].startPoint, arr[i].endPoint) == 0) {
        count++;
    }
}

return count;
}

void sortByPrice(struct route *arr, int n) {
    struct route temp, tarr[n];

    for (int i = 0; i < n; i++) {
        tarr[i] = arr[i];
    }

    for (int j = n; j > 1; j--) {
        for (int i = 0; i < j-1; i++) {
            if (tarr[i].price < tarr[i+1].price) {
                temp=tarr[i];
                tarr[i]=tarr[i+1];
                tarr[i+1]=temp;
            }
        }
    }

    printf("Отсортированный список маршрутов по полю \"Цена проезда\":\n");
    printRoutes(tarr, n);
}
```

На Рисунках 7.6, 7.7 отображены результаты работы кода согласно всем условиям по заданию для языка С.

===== М Е НЮ =====

1. Ввести все маршруты
 2. Вывести все маршруты
 3. Найти количество круговых маршрутов
 4. Отсортировать маршруты по стоимости проезда
 9. Выход
- > 1

Введите количество маршрутов: 3

Введите 3 маршрутов:

Введите номер маршрута: 10

Введите начальную точку маршрута (до 100 символов): Центр

Введите конечную точку маршрута (до 100 символов): Депо-1

Введите цену маршрута: 30

Введите время в пути (в минутах): 125

Введите номер маршрута: 77

Введите начальную точку маршрута (до 100 символов): Вокзал

Введите конечную точку маршрута (до 100 символов): Юмашева

Введите цену маршрута: 45

Введите время в пути (в минутах): 180

Введите номер маршрута: 5

Введите начальную точку маршрута (до 100 символов): Кино

Введите конечную точку маршрута (до 100 символов): Кино

Введите цену маршрута: 13

Введите время в пути (в минутах): 45

===== М Е НЮ =====

1. Ввести все маршруты
 2. Вывести все маршруты
 3. Найти количество круговых маршрутов
 4. Отсортировать маршруты по стоимости проезда
 9. Выход
- > 2

Все маршруты:

1) Номер маршрута: 10

Начальная точка: Центр

Конечная точка: Депо-1

Цена проезда: 30.00₽

Время в пути: 2ч 5мин

2) Номер маршрута: 77

Начальная точка: Вокзал

Конечная точка: Юмашева

Цена проезда: 45.00₽

Время в пути: 3ч

3) Номер маршрута: 5

Начальная точка: Кино

Конечная точка: Кино

Цена проезда: 13.00₽

Время в пути: 45мин

Всего маршрутов: 3

Рисунок 7.6 – Результат работы программы на языке С

===== М Е НЮ =====

1. Ввести все маршруты
 2. Вывести все маршруты
 3. Найти количество круговых маршрутов
 4. Отсортировать маршруты по стоимости проезда
 9. Выход
- > 3

Количество круговых маршрутов: 1

===== М Е НЮ =====

1. Ввести все маршруты
 2. Вывести все маршруты
 3. Найти количество круговых маршрутов
 4. Отсортировать маршруты по стоимости проезда
 9. Выход
- > 4

Отсортированный список маршрутов по полю "Цена проезда":

2) Номер маршрута: 77
Начальная точка: Вокзал
Конечная точка: Юмашева
Цена проезда: 45.00₽
Время в пути: 3ч

1) Номер маршрута: 10
Начальная точка: Центр
Конечная точка: Депо-1
Цена проезда: 30.00₽
Время в пути: 24 5мин

3) Номер маршрута: 5
Начальная точка: Кино
Конечная точка: Кино
Цена проезда: 13.00₽
Время в пути: 45мин

Всего маршрутов: 3

===== М Е НЮ =====

1. Ввести все маршруты
 2. Вывести все маршруты
 3. Найти количество круговых маршрутов
 4. Отсортировать маршруты по стоимости проезда
 9. Выход
- > 9

Выход из программы . . .

Рисунок 7.7 – Результат работы программы на языке С

6. Текст кода на языке C++. Тестирование и отладка программы

```

#include <iostream>
#include <stdlib.h>
#include <string>
#include <iomanip>
#include <math.h>

using namespace std;

// Создание структуры "route"
struct route
{
    int number;
    int routeNumber;
    string startPoint;
    string endPoint;
    float price;
    int time;
};

// Объявление функций
route *scanRoutes(int n);
void printRoutes(route *arr, int n);
int countCircleRoutes(route *arr, int n);
void sortByPrice(route *arr, int n);

int main()
{
    int choice, n = 0;
    route *arr = NULL;
    while (1)
    {
        cout << "\n===== М Е Н Й =====\n";
        cout << "1. Ввести все маршруты" << endl;
        cout << "2. Вывести все маршруты" << endl;
        cout << "3. Найти количество круговых маршрутов" << endl;
        cout << "4. Отсортировать маршруты по стоимости проезда"
<< endl;
        cout << "9. Выход" << endl;
        cout << "-> ";
        cin >> choice;
        cout
            << endl;

        switch (choice)
        {
        case 1:
            if (arr != NULL)
            {
                delete[] arr;
            }
            n = 0;
            cout << "Введите количество маршрутов: ";

```

```

    cin >> n;
    if (n <= 0)
    {
        cout << "Количество маршрутов должно быть > 0" <<
endl;
        break;
    }
    arr = scanRoutes(n);
    break;

case 2:
    if (arr == NULL)
    {
        cout << "Маршруты не введены!" << endl;
    }
    else
    {
        cout << "Все маршруты:" << endl;
        printRoutes(arr, n);
    }
    break;

case 3:
    if (arr == NULL)
    {
        cout << "Маршруты не введены!" << endl;
    }
    else
    {
        cout << "Количество круговых маршрутов: " <<
countCircleRoutes(arr, n) << endl;
    }
    break;

case 4:
    if (arr == NULL)
    {
        cout << "Маршруты не введены!" << endl;
    }
    else
    {
        sortByPrice(arr, n);
    }
    break;

case 9:
    delete[] arr;
    cout << "Выход из программы . . .";
    return 0;

default:
    cout << "Неверный выбор! Попробуйте снова." << endl;
    break;

```

```

        }
    }

route *scanRoutes(int n)
{
    route *arr = new route[n];

    if (arr == NULL)
    {
        cout << "Ошибка выделения памяти!" << endl;
        return NULL;
    }

    cout << "Введите " << n << " маршрутов:" << endl;
    for (int i = 0; i < n; i++)
    {
        arr[i].number = i + 1;
        cout << "Введите номер маршрута: ";
        cin >> arr[i].routeNumber;
        while (getchar() != '\n')
            ;

        cout << "Введите начальную точку маршрута (до 100
СИМВОЛОВ): ";
        getline(cin, arr[i].startPoint);

        cout << "Введите конечную точку маршрута (до 100
СИМВОЛОВ): ";
        getline(cin, arr[i].endPoint);

        cout << "Введите цену маршрута: ";
        cin >> arr[i].price;

        cout << "Введите время в пути (в минутах): ";
        cin >> arr[i].time;

        cout << "-----"
<< endl;
    }

    return arr;
}

void printRoutes(route *arr, int n)
{
    cout << "-----" <<
endl;

    for (int i = 0; i < n; i++)
    {
        cout << fixed << setprecision(2);
    }
}

```

```

        cout << arr[i].number << ") Номер маршрута: " <<
arr[i].routeNumber << endl;
        cout << "Начальная точка:" << arr[i].startPoint << endl
            << "Конечная точка: " << arr[i].endPoint << endl;
        cout
            << "Цена проезда: " << arr[i].price << "₽" << endl;
        if (arr[i].time % 60 == 0)
        {
            cout << "Время в пути" << arr[i].time / 60 << "ч" <<
endl;
        }
        else if (arr[i].time < 60)
        {
            cout << "Время в пути" << arr[i].time << "мин" <<
endl;
        }
        else
        {
            cout << "Время в пути: " << arr[i].time / 60 << "ч "
<< arr[i].time % 60 << "мин" << endl;
        }
        cout << "-----"
<< endl;
    }

    cout << "Всего маршрутов: " << n << endl;
}

int countCircleRoutes(route *arr, int n)
{
    int count = 0;

    for (int i = 0; i < n; i++)
    {
        if (arr[i].startPoint == arr[i].endPoint)
        {
            count++;
        }
    }

    return count;
}

void sortByPrice(route *arr, int n)
{
    route temp, tarr[n];

    for (int i = 0; i < n; i++)
    {
        tarr[i] = arr[i];
    }

    for (int j = n; j > 1; j--)

```

```
{  
    for (int i = 0; i < j - 1; i++)  
    {  
        if (tarr[i].price < tarr[i + 1].price)  
        {  
            temp = tarr[i];  
            tarr[i] = tarr[i + 1];  
            tarr[i + 1] = temp;  
        }  
    }  
}  
  
cout << "Отсортированный список маршрутов по полю \"Цена  
проезда\" :" << endl;  
printRoutes(tarr, n);  
}
```

На Рисунках 7.8, 7.9 отображены результаты работы кода согласно всем условиям по заданию для языка C++.

```
===== М Е НЮ =====
1. Ввести все маршруты
2. Вывести все маршруты
3. Найти количество круговых маршрутов
4. Отсортировать маршруты по стоимости проезда
9. Выход
-> 1

Введите количество маршрутов: 3
Введите 3 маршрутов:
Введите номер маршрута: 77
Введите начальную точку маршрута (до 100 символов): Вокзал
Введите конечную точку маршрута (до 100 символов): Юмашева
Введите цену маршрута: 30
Введите время в пути (в минутах): 125
-----
Введите номер маршрута: 10
Введите начальную точку маршрута (до 100 символов): Центр
Введите конечную точку маршрута (до 100 символов): Депо-1
Введите цену маршрута: 45
Введите время в пути (в минутах): 75
-----
Введите номер маршрута: 5
Введите начальную точку маршрута (до 100 символов): Кино
Введите конечную точку маршрута (до 100 символов): Кино
Введите цену маршрута: 15
Введите время в пути (в минутах): 25
-----

===== М Е НЮ =====
1. Ввести все маршруты
2. Вывести все маршруты
3. Найти количество круговых маршрутов
4. Отсортировать маршруты по стоимости проезда
9. Выход
-> 2

Все маршруты:
-----
1) Номер маршрута: 77
Начальная точка: Вокзал
Конечная точка: Юмашева
Цена проезда: 30.00₽
Время в пути: 24 5мин
-----
2) Номер маршрута: 10
Начальная точка: Центр
Конечная точка: Депо-1
Цена проезда: 45.00₽
Время в пути: 1ч 15мин
-----
3) Номер маршрута: 5
Начальная точка: Кино
Конечная точка: Кино
Цена проезда: 15.00₽
Время в пути: 25мин
-----
Всего маршрутов: 3
```

Рисунок 7.8 – Результат работы программы на языке C++

```
===== М Е НЮ =====
1. Ввести все маршруты
2. Вывести все маршруты
3. Найти количество круговых маршрутов
4. Отсортировать маршруты по стоимости проезда
9. Выход
-> 3
```

Количество круговых маршрутов: 1

```
===== М Е НЮ =====
1. Ввести все маршруты
2. Вывести все маршруты
3. Найти количество круговых маршрутов
4. Отсортировать маршруты по стоимости проезда
9. Выход
-> 4
```

Отсортированный список маршрутов по полю "Цена проезда":

2) Номер маршрута: 10
Начальная точка: Центр
Конечная точка: Депо-1
Цена проезда: 45.00₽
Время в пути: 1ч 15мин

1) Номер маршрута: 77
Начальная точка: Вокзал
Конечная точка: Юмашева
Цена проезда: 30.00₽
Время в пути: 2ч 5мин

3) Номер маршрута: 5
Начальная точка: Кино
Конечная точка: Кино
Цена проезда: 15.00₽
Время в пути: 25мин

Всего маршрутов: 3

```
===== М Е НЮ =====
1. Ввести все маршруты
2. Вывести все маршруты
3. Найти количество круговых маршрутов
4. Отсортировать маршруты по стоимости проезда
9. Выход
-> 9
```

Выход из программы . . .

Рисунок 7.9 – Результат работы программы на языке C++

ВЫВОД

В ходе лабораторной работы были исследованы особенности представления и обработки структур данных в языках C/C++. На практике были успешно реализованы алгоритмы работы со структурами данных, включая их создание, обработку и освобождение памяти под массив структур. Экспериментально подтверждена и изучена неразрывная связь между указателями и массивами структур данных. Приобретены навыки эффективного использования динамического выделения памяти под структуры данных на языках C/C++.