Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе «Интерполяционный поиск»**

Дисциплина: Основы алгоритмизации и программирования

Выполнил:

Студентка группы ИВТ-22-2б

Мифтахов Марат Ринатович

Проверил:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2023**

**Постановка задачи**

Реализовать программу, которая будет удалять все вхождения символа N из массива с символами типа int.

**Анализ задачи**

1. Создается функция Print(mas, size), в которой производится вывод массива:

void Print(int\* mas, int size)

{

for (int i = 0; i < size; i++)

{

cout << mas[i] << " ";

}

cout << endl;

}

1. Затем описывается функция InterpolSearch(int\* mas, int size, int key), в которой производится Интерполяционный поиск по формуле:

int InterpolSearch(int\* mas, int size, int key)

{

int mid;

int left = 0;

int right = size - 1;

while ((mas[left] < key) && (key < mas[right]))

{

mid = left + ((key - mas[left]) \* (right - left) / (mas[right] - mas[left]));

if (mas[mid] < key)

{

left = mid + 1;

}

else

{

if (mas[mid] > key)

{

right = mid - 1;

}

else

{

return mid;

}

}

}

if (mas[left] == key)

{

return left;

}

else if (mas[right] == key)

{

return right;

}

return -1;

}

1. В главной функции пользователя запрашивается размерность массива:

do

{

cout << "Введите размерность массива: ";

cin >> size;

} while (size < 2);

1. Создается динамический массив:

int\* mas = new int[size];

1. Затем вводятся элементы массива:

cout << "Элементы массива: " << endl;

for (int i = 0; i < size; i++)

{

cin >> mas[i];

}

1. Далее вызывается функция Print(mas, size):

cout << endl << "Массив: " << endl;

Print(mas, size);

1. Затем происходит сортировка массива методом выбора:

for (int i = 0; i < size - 1; i++)

{

int min = i;

for (int j = i; j < size; j++)

{

if (mas[j] < mas[min])

{

min = j;

}

}

if (i != min)

{

int tmp = mas[i];

mas[i] = mas[min];

mas[min] = tmp;

}

}

1. Вызывается функция Print(mas, size), и выводится отсортированный массив:

cout << endl << "Отсортированный массив: " << endl;

Print(mas, size);

1. Далее у пользователя запрашивается элемент, все вхождения которого необходимо удалить.

cout << endl << "Введите элемент, который необходимо удалить из массива: ";

cin >> key;

1. Затем происходит удаление элементов:

do {

index = InterpolSearch(mas, size, key);

if (index != -1) {

for (int i = index; i < size - 1; i++) {

mas[i] = mas[i + 1];

}

size--;

}

} while (index != -1);

1. Вызывается функция Print(mas, size), и выводится массив после удаления:

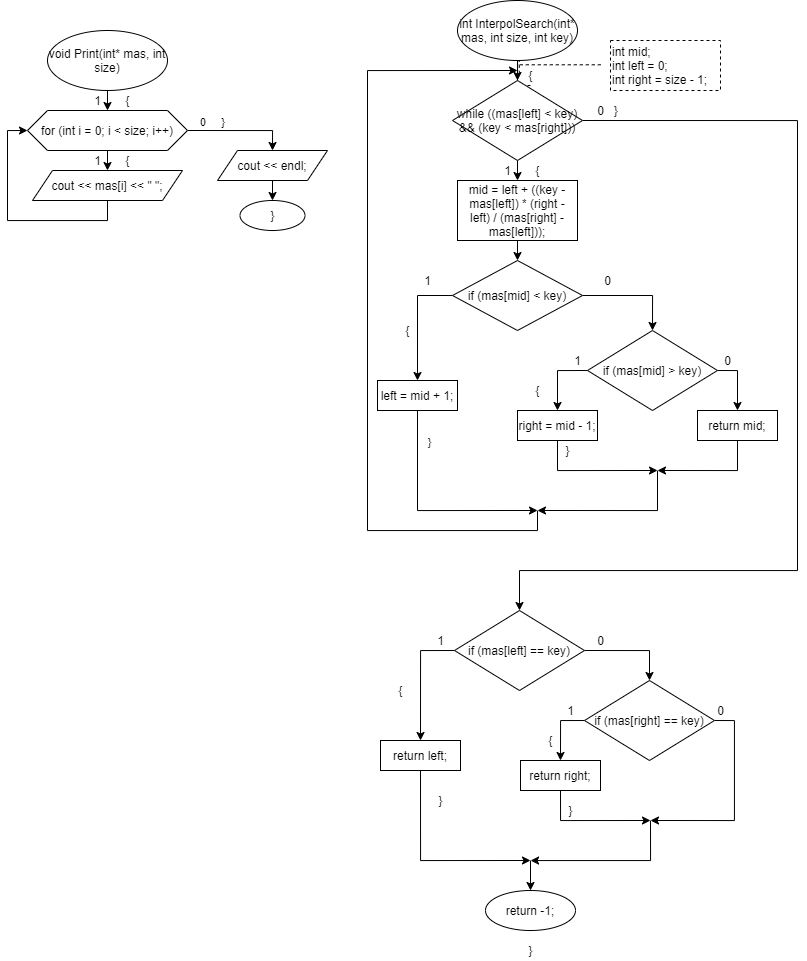
cout << endl << "Массив после удаления: " << endl;

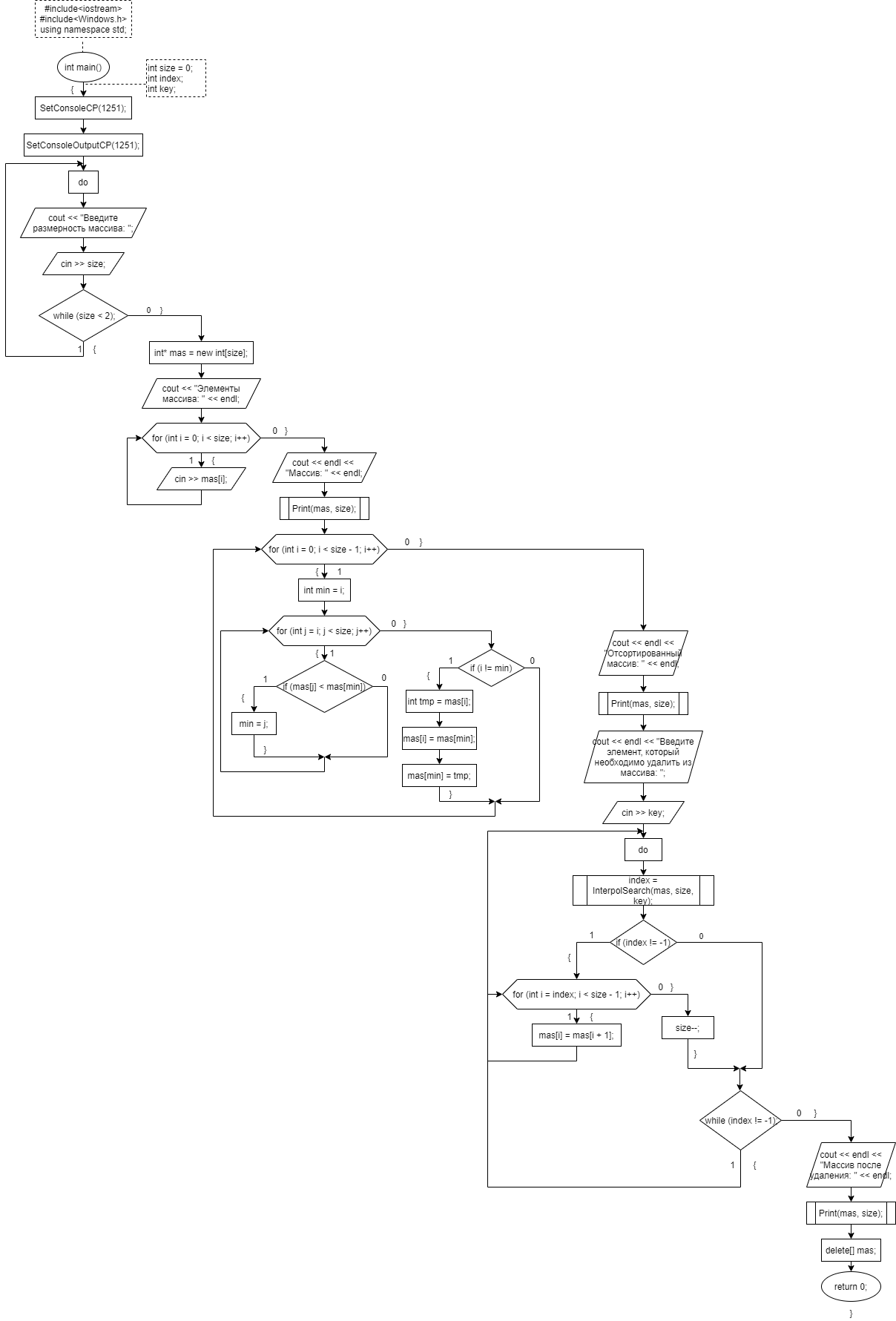
Print(mas, size);

1. Удаляется память, выделенная под массив:

delete[] mas;

**Блок-схема**





**Исходный код**

#include<iostream>

#include<Windows.h>

using namespace std;

void Print(int\* mas, int size)

{

for (int i = 0; i < size; i++)

{

cout << mas[i] << " ";

}

cout << endl;

}

int InterpolSearch(int\* mas, int size, int key)

{

int mid;

int left = 0;

int right = size - 1;

while ((mas[left] < key) && (key < mas[right]))

{

mid = left + ((key - mas[left]) \* (right - left) / (mas[right] - mas[left]));

if (mas[mid] < key)

{

left = mid + 1;

}

else

{

if (mas[mid] > key)

{

right = mid - 1;

}

else

{

return mid;

}

}

}

if (mas[left] == key)

{

return left;

}

else if (mas[right] == key)

{

return right;

}

return -1;

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int size = 0;

do

{

cout << "Введите размерность массива: ";

cin >> size;

} while (size < 2);

int\* mas = new int[size];

cout << "Элементы массива: " << endl;

for (int i = 0; i < size; i++)

{

cin >> mas[i];

}

cout << endl << "Массив: " << endl;

Print(mas, size);

for (int i = 0; i < size - 1; i++)

{

int min = i;

for (int j = i; j < size; j++)

{

if (mas[j] < mas[min])

{

min = j;

}

}

if (i != min)

{

int tmp = mas[i];

mas[i] = mas[min];

mas[min] = tmp;

}

}

cout << endl << "Отсортированный массив: " << endl;

Print(mas, size);

int index;

int key;

cout << endl << "Введите элемент, который необходимо удалить из массива: ";

cin >> key;

do {

index = InterpolSearch(mas, size, key);

if (index != -1) {

for (int i = index; i < size - 1; i++) {

mas[i] = mas[i + 1];

}

size--;

}

} while (index != -1);

cout << endl << "Массив после удаления: " << endl;

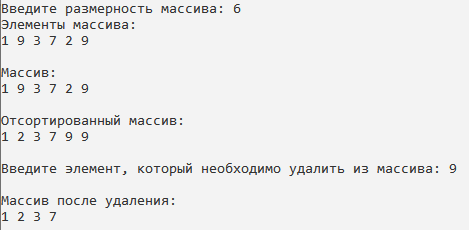
Print(mas, size);

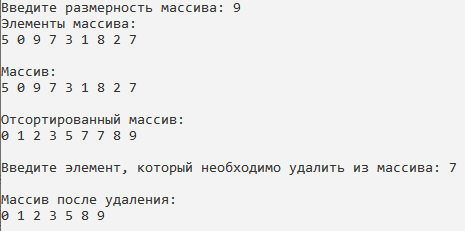
delete[] mas;

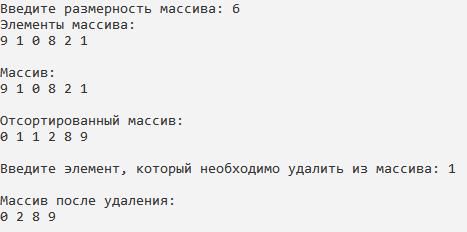
return 0;

}

**Скриншоты**







**Анализ (объяснение скриншотов)**

Программа запрашивает у пользователя размерность и все элементы массива. Выводит полностью массив, а затем выводит отсортированный массив. Далее запрашивает у пользователя элемент, который необходимо удалить. Затем программа выводит массив без данного элемента.