Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное учреждение высшего образования  
«Пермский национальный исследовательский политехнический университет»

Электротехнический факультет  
Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т  
по лабораторной работе**

Дисциплина: «Основы алгоритмизации и программирования»

Выполнил  
студент группы ИВТ-21-1б  
Мифтахов И.Г

Проверил  
ст. п. кафедры ИТАС  
Яруллин Д.В.

Пермь 2022

Постановка задачи:

Для линейного и интерполяционного поиска: Дан массив целых чисел (или символов, если это дано по заданию), выполнить задание из своего варианта, используя указанный метод поиска. Массив можно заполнить вручную или случайными числами, размерность массива любая (не меньше 5). Массив для интерполяционного поиска предварительно отсортировать.

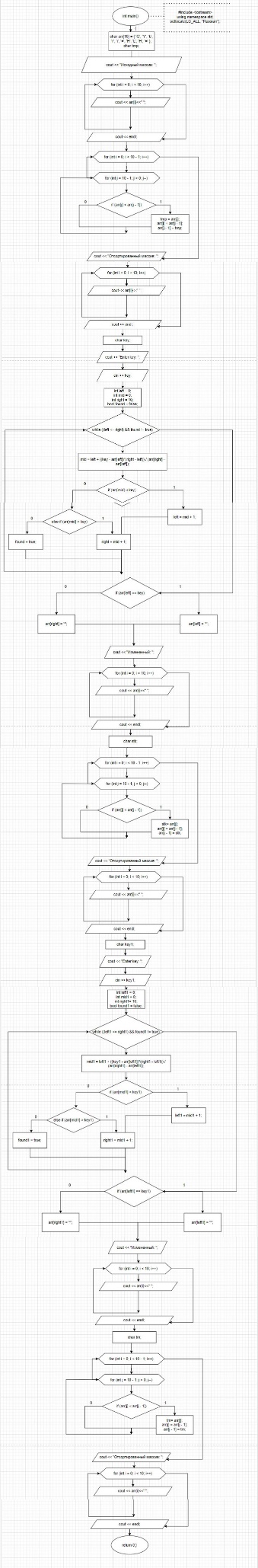
Для метода Бойера – Мура и Кнута – Морриса – Пратта: Дана строка и подстрока, которую нужно найти. Выполнить задание в соответствии со своим вариантом, используя указанный метод поиска. Значения строки и подстроки любые (если иного не указано по заданию).

Анализ задачи:

Инициализируем массив, выводим с помощью цикла for, сортируем методом Пузырька. Выводим отсортированный аналогично раннему методу.

Инициализируем переменные left, right, mid. Пока левая граница массива не равна правой или флажок false будем двигаться в цикле while. С помощью формулы считаем mid, прибавляем +1 к той части, где элемента нет и считаем заново. Далее сверяем найденный элемент с ключом, если он в левой границе или в правой, то меняем на звездочку. Вновь сортируем, выводим. Повторяем эту процедуру еще 1 раз с другим ключом. Выводится измененный и корректно отсортированный массив.

Блок-схема:



Код программы:

#include<iostream>

using namespace std;

int main()

{

setlocale(LC\_ALL, "Rus");

char arr[10] = { 'G', 'Y', 'b', 'r', 'i', '=', 'H', 'L', 'H', '=' };

char tmp;

cout << "Исходный массив: ";

for (int i = 0; i < 10; i++)

{

cout << arr[i]<<" ";

}

cout << endl;

for (int i = 0; i < 10 - 1; i++)

{

for (int j = 10 - 1; j > 0; j--)

{

if (arr[j] < arr[j - 1])

{

tmp = arr[j];

arr[j] = arr[j - 1];

arr[j - 1] = tmp;

}

}

}

cout << "Отсортированный массив: ";

for (int i = 0; i < 10; i++)

{

cout << arr[i]<<" ";

}

cout << endl;

char key;

cout << "Enter key: ";

cin >> key;

int left = 0;

int mid = 0;

int right = 10;

bool found = false;

while ((left <= right) && found != true)

{

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

if (arr[mid] < key) { left = mid + 1; }

else if (arr[mid] > key) { right = mid + 1; }

else { found = true; }

}

if (arr[left] == key)

{

arr[left] = '\*';

}

else if (arr[right] == key);

{

arr[right] = '\*';

}

cout << "Измененный: ";

for (int i = 0; i < 10; i++)

{

cout << arr[i] << " ";

}

cout << endl;

char stk;

for (int i = 0; i < 10 - 1; i++)

{

for (int j = 10 - 1; j > 0; j--)

{

if (arr[j] < arr[j - 1])

{

stk = arr[j];

arr[j] = arr[j - 1];

arr[j - 1] = stk;

}

}

}

cout << "Отсортированный массив: ";

for (int i = 0; i < 10; i++)

{

cout << arr[i] << " ";

}

cout << endl;

char key1;

cout << "Enter key: ";

cin >> key1;

int left1 = 1;

int mid1 = 1;

int right1 = 10;

bool found1 = false;

while ((left1 <= right1) && found1 != true)

{

mid1 = left1 + ((key1 - arr[left1])\*(right1 - left1)) / (arr[right1] - arr[left1]);

if (arr[mid1] < key1) { left1 = mid1 + 1; }

else if (arr[mid1] > key1) { right1 = mid1 + 1; }

else { found1 = true; }

}

if (arr[left1] == key1)

{

arr[left1] = '\*';

}

else if (arr[right1] == key1);

{

arr[right1] = '\*';

}

cout << "Измененный: ";

for (int i = 0; i < 10; i++)

{

cout << arr[i] << " ";

}

cout << endl;

char tm;

for (int i = 0; i < 10 - 1; i++)

{

for (int j = 10 - 1; j > 0; j--)

{

if (arr[j] < arr[j - 1])

{

tm = arr[j];

arr[j] = arr[j - 1];

arr[j - 1] = tm;

}

}

}

cout << "Отсортированный массив: ";

for (int i = 0; i < 10; i++)

{

cout << arr[i] << " ";

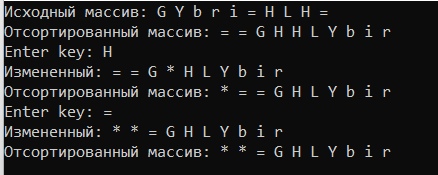
}

cout << endl;

return 0;

}

Скриншоты результатов:



Анализ результатов:

Программа работает корректно. Первые вхождения символов заменяются на звездочки.