Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Кафедра **«**Информационные технологии и автоматизированные системы**»**

**ОТЧЕТ**

Дисциплина: «Основы алгоритмизации и программирования»

Семестр 2

Тема: Поиски.

Выполнил работу

Студент группы РИС-22-1Б

Поважный В. Е.

Проверил

Доцент кафедры ИТАС

Полякова О.А.

г. Пермь-2023

**Введение**

Для выполнения лабораторной работы требуется сделать отчет программы. Создание программ – отличный способ практики программирования.

**Постановка задачи**

Реализовать методы различных поисков:

1. Линейный поиск
2. Бинарный поиск
3. Интерполяционный поиск
4. Поиск подстроки в строке
5. Поиск методом Бойера - Мура
6. Поиск Кнута - Морриса Пратта

**Код программы**

#include <iostream>

using namespace std;

int\* Shell(int A[], int n)

{

int c;

int d = n;

d = d / 2;

while (d > 0)

{

for (int i = 0; i < n - d; i++)

{

int j = i;

while (j >= 0 && A[j] > A[j + d])

{

c = A[j];

A[j] = A[j + d];

A[j + d] = c;

j--;

}

}

d = d / 2;

}

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

cout << A[i] << " ";

cout << endl;

return A;

}

// линейное

int lineSearch(int arr[], int size, int key)

{

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

if (arr[i] == key)

return i;

return -1;

}

//кнута морриса

void KMP(string text, string pattern)

{

int m = text.length();

int n = pattern.length();

int\* next = new int[n + 1];

for (int i = 0; i < n + 1; i++)

next[i] = 0;

for (int i = 1; i < n; i++)

{

int j = next[i + 1];

while (j > 0 && pattern[j] != pattern[i])

j = next[j];

if (j > 0 || pattern[j] == pattern[i])

next[i + 1] = j + 1;

}

for (int i = 0, j = 0; i < m; i++)

if (text[i] == pattern[j])

{

if (++j == n)

cout << "The pattern occurs with shift " << i - j + 1 << endl;

}

else

if (j > 0)

{

j = next[j];

i--;

}

}

// бинарный

int BINSearch(int arr[], int n, int key)

{

bool f = false;

int l = 0;

int r = 11;

int mid;

arr = Shell(arr, n);

while ((l <= r) && !f)

{

mid = (l + r) / 2;

if (arr[mid] == key)

f = true;

if (arr[mid] > key)

r = mid - 1;

else

l = mid + 1;

}

return mid;

}

// интерполяционный

int interpolSearch(int\* arr, int size, int key)

{

int l = 0;

int r = size;

int mid = 0;

bool found = false;

while ((l <= r) && !found)

{

mid = l + ((key - arr[l]) \* (r - l)) / (arr[r] - arr[l]);

if (arr[mid] < key)

l = mid + 1;

else

if (arr[mid] > key)

r = mid + 1;

else

found = true;

}

if (arr[l] == key)

return l;

else

if (arr[r] == key)

return r;

return -1;

}

// поиск подстроки в строке

int search(string str, string substr)

{

int strl, substrl, res = 1;

strl = str.size();

substrl = substr.size();

if (strl != 0 && substrl != 0)

{

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

for (int j = 0; j < substrl; j++)

{

if (substr[j] != str[i + j])

break;

else

if (j == substrl - 1)

{

res = i;

break;

}

}

return res;

}

return -1;

}

// бойера - мура

int BMSearch(string str, string substr)

{

int sl, ssl;

sl = str.size();

ssl = substr.size();

if (sl != 0 && ssl != 0)

{

int i, pos;

int bias[256];

for (i = 0; i < 256; i++)

bias[i] = ssl;

for (i = ssl - 2; i >= 0; i--)

if (bias[int((unsigned char)substr[i])] == ssl)

bias[int((unsigned char)substr[i])] == ssl - i - 1;

pos = ssl - 1;

while (pos < sl)

if (substr[ssl - 1] != str[pos])

pos += bias[int((unsigned char)str[pos])];

else

for (i = ssl - 1; i >= 0; i--)

if (substr[i] != str[pos - ssl + i + 1])

{

pos += bias[int((unsigned char)str[pos - ssl + i + 1])];

break;

}

else

if (i == 0)

return pos - ssl + 1;

}

return -1;

}

int main()

{

const int size = 10;

int key;

int arr[size];

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

cin >> arr[i];

// линейное

cin >> key;

cout << lineSearch(arr, size, key) << endl;

// с барьером

cin >> key;

cout << find(arr, size, key) << endl;

// бинарный

cin >> key;

cout << BINSearch(arr, size, key) << endl;

// интерполяционный

cin >> key;

cout << interpolSearch(arr, size, key) << endl;

// поиск подстроки в строке

string str, substr;

cin >> str >> substr;

cout << search(str, substr) << endl;

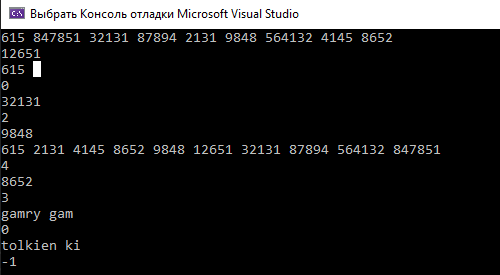
// бойера - мура

cin >> str >> substr;

cout << BMSearch(str, substr) << endl;

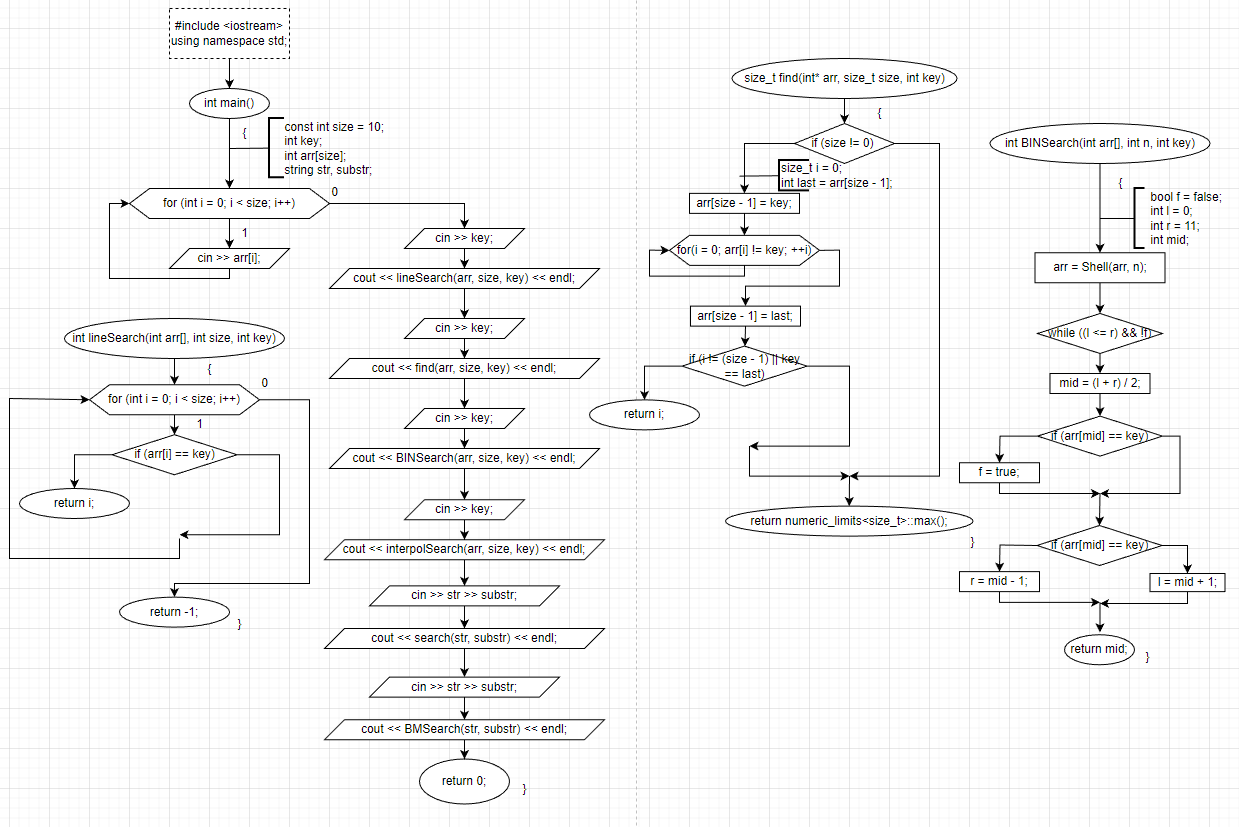
}

**Вывод программы**

****

***Рисунок 1 – вывод программы***

**Блок-схема**

****

***Рисунок 3 – Схема алгоритма программы***

**Вывод**

Программа выполняет свою задачу.

**Заключение**

Для решения задачи потребовались знания языка программирования, а конкретнее C++. Программа выполняет те условия, что были указаны в постановке задачи и работает без проблем.