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»

Тема: 3 метода поиска

Семестр 2

Выполнил работу

Студент группы РИС-22-1Б

Аксёнов Д.О.

Проверил

Доцент кафедры ИТАС

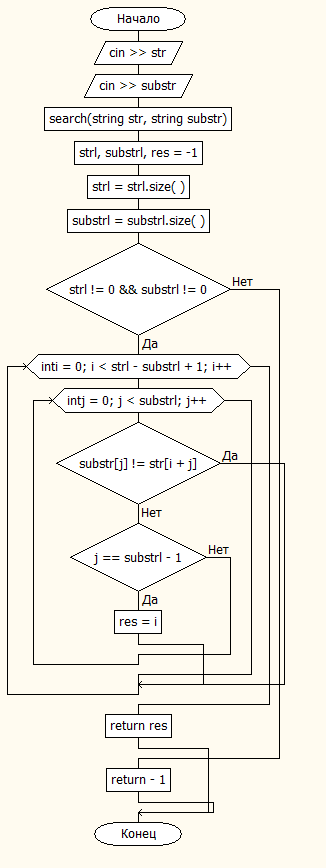
Полякова Ольга Андреевна

Г. Пермь-2023

**Постановка задачи**

Задача: реализовать 4 метода поиски (прямая, интерполяционная, Бойера-Мура, Кнута–Морриса-Пратта)

**Алгоритм программы(прямая)**



**Код программы(быстрая)**

#include <iostream>

#include <string>

using namespace std;

int main()

{

string str,substr;

cin>>str;

cin>>substr;

int search (string str,string substr);

{

int strl,substrl,res=-1;

strl=strl.size();substrl=substrl.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;

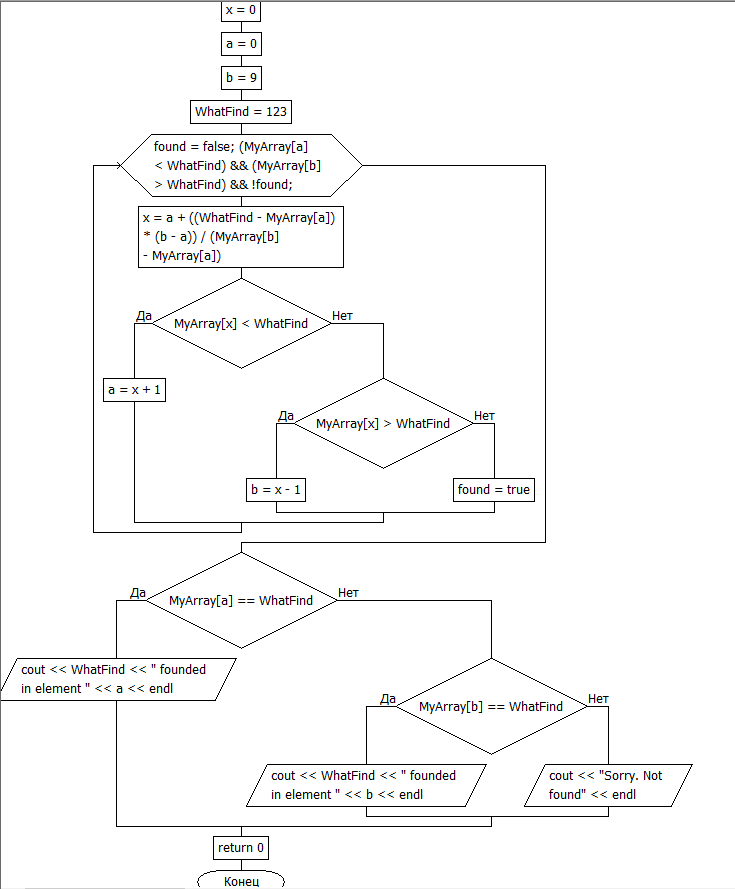
}

return -1;

}

}

**Алгоритм программы(интерполяционная)**



**Код программы (интерполяционная)**

#include <iostream>

using namespace std;

int main()

{

int MyArray[] { 1, 2, 4, 6, 7, 89, 123, 231, 1000, 1235 };

int x = 0;

int a = 0;

int b = 9;

int WhatFind = 123;

bool found;

for (found = false; (MyArray[a] < WhatFind) && (MyArray[b] > WhatFind) && !found; )

{

x = a + ((WhatFind - MyArray[a]) \* (b - a)) / (MyArray[b] - MyArray[a]);

if (MyArray[x] < WhatFind)

a = x + 1;

else if (MyArray[x] > WhatFind)

b = x - 1;

else

found = true;

}

if (MyArray[a] == WhatFind)

cout << WhatFind << " founded in element " << a << endl;

else if (MyArray[b] == WhatFind)

cout << WhatFind << " founded in element " << b << endl;

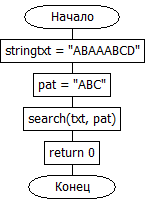
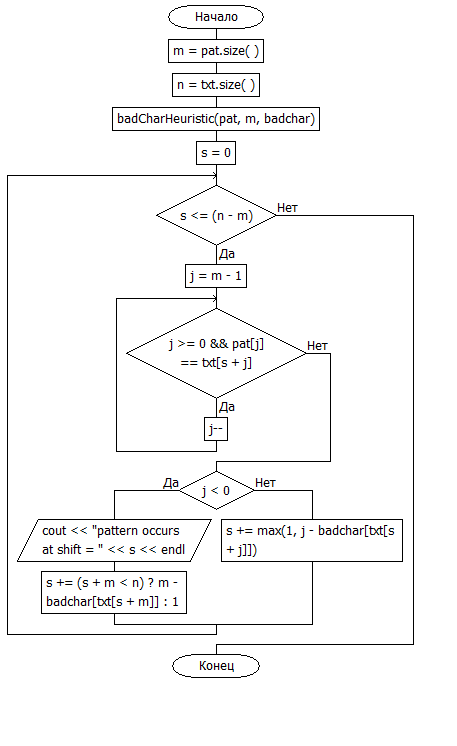
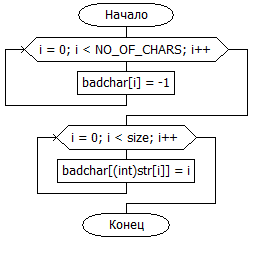
else

cout << "Sorry. Not found" << endl;

return 0;

}

**Алгоритм программы (Бойера-Мур)**



**Код программы (Бойера-Мур)**

#include <bits/stdc++.h>

using namespace std;

# define NO\_OF\_CHARS 256

void badCharHeuristic( string str, int size,

int badchar[NO\_OF\_CHARS])

{

int i;

for (i = 0; i < NO\_OF\_CHARS; i++)

badchar[i] = -1;

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

badchar[(int) str[i]] = i;

}

void search( string txt, string pat)

{

int m = pat.size();

int n = txt.size();

int badchar[NO\_OF\_CHARS];

badCharHeuristic(pat, m, badchar);

int s = 0;

while(s <= (n - m))

{

int j = m - 1;

while(j >= 0 && pat[j] == txt[s + j])

j--;

if (j < 0)

{

cout << "pattern occurs at shift = " << s << endl;

s += (s + m < n)? m-badchar[txt[s + m]] : 1;

}

else

s += max(1, j - badchar[txt[s + j]]);

}

}

int main()

{

string txt= "ABAAABCD";

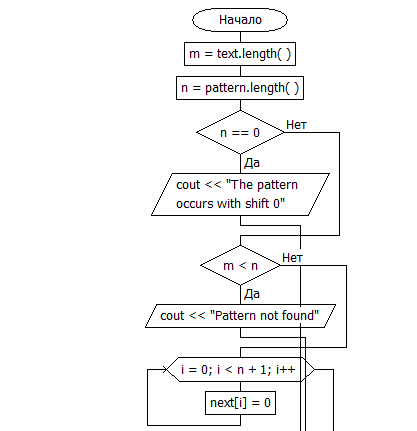
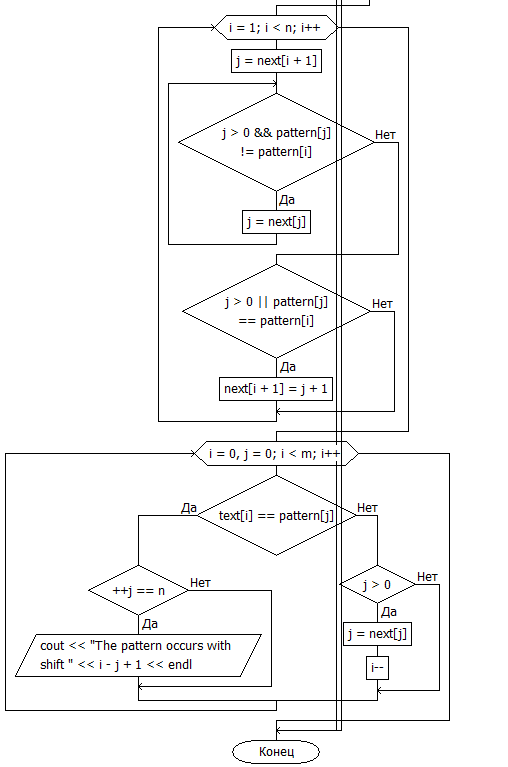
string pat = "ABC";

search(txt, pat);

return 0;

**}**

**Алгоритм программы(Кнута – Морриса – Пратта)**

**Код программы (Кнута – Морриса – Пратта)**

#include <iostream>

using namespace std;

void KMP(string text, string pattern)

{

int m = text.length();

int n = pattern.length();

if (n == 0)

{

cout << "The pattern occurs with shift 0";

return;

}

if (m < n)

{

cout << "Pattern not found";

return;

}

int next[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--;

}

}

}

int main()

{

string text = "ABCABAABCABAC";

string pattern = "CAB";

KMP(text, pattern);

return 0;

}

**Работа программ.**



