

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №4
по дисциплине «Построение и анализ алгоритмов»
Тема: Алгоритм Кнута-Морриса-Пратта

Студент гр. 7383

Корякин М.П.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2019

Цель работы.

Заданы две строки:

A ($|A| \leq 5000000$) и B ($|B| \leq 5000000$).

Определить, является ли A циклическим сдвигом B (это значит, что A и B имеют одинаковую длину и A состоит из суффикса B, склеенного с префиксом B). Например, defabc является циклическим сдвигом abcdef.

Вход:

Первая строка - A

Вторая строка - B

Выход:

Если AA является циклическим сдвигом BB, индекс начала строки B в A, иначе вывести -1. Если возможно несколько сдвигов вывести первый индекс.

Реализация алгоритма.

Для реализации данной задачи была написана функция, определяющая префикс-функцию строки A. Вычисленная префикс-функция строки была записана в вектор.

После определения префикс-функции начинается поиск циклического сдвига или его отсутствие. Поиск осуществляется следующим образом: создается строка temp, посредством дублирования строки A. И на новой строке temp мы сравниваем строку B, используя префикс-функцию, вычисленную ранее. Если итератор j становится равен строке B, то это означает, что циклический сдвиг есть, и алгоритм выводит значение индекса на экран. Иначе, если склеенная строка temp заканчивается, то алгоритм выводит значение -1.

Результат работы программы.

Входные данные	Выходные данные
defabc abcdef	3

Dsagdsfgdf Sadfsdgsdgg	-1
---------------------------	----

Выводы.

В ходе выполнения лабораторной работы был освоен алгоритм Кнута-Морриса-Пратта. Было решено две задачи на ЯП C++. Вторая задача представляла из себя вывод индексов элементов строки, в которых строка поиска входила в данную строку, исходный код представлен в приложении Б.

ПРИЛОЖЕНИЕ А.

ИСХОДНЫЙ КОД.

```
#include <iostream>
#include <vector>
using namespace std;

vector<int> prefix_function (string s) {
    size_t n = s.length();
    vector<int> pi(n);
    for (int i=1; i<n; ++i) {
        int j = pi[i-1];
        while ((j > 0) && (s[i] != s[j]))
            j = pi[j-1];
        if (s[i] == s[j])
            ++j;
        pi[i] = j;
    }
    return pi;
}

int search(string B, string A, vector<int> q){
    if (A.length() != B.length())
        return -1;
    string temp = A + A;
    int j = 0;
    int k = 0;
    int mean = -1;
    for (int i = 0; i < temp.length(); i++){
        while (j > 0 && B[j] != temp[i])
            j = q[j - 1];
        if (B[j] == temp[i])
            j++;
        if (j == B.length()){
            mean = (i + 1 - B.length());
            break;
        }
    }
    return mean;
}

int main(){
    string A;
    string B;
    vector<int> pref;
    cin>>A;
    cin>>B;
    cout<<search(B, A, prefix_function(A));
    return 0;
}
```

ПРИЛОЖЕНИЕ Б. ИСХОДНЫЙ КОД.

```
#include <iostream>
#include <vector>
using namespace std;
class A{
public:
vector<size_t> result;
vector<size_t> K_M_P(string A, string B){
    vector<size_t> prefix(A.length());
    size_t last_prefix = prefix[0] = 0;
    for (size_t i=1; i<A.length(); i++) {
        while (last_prefix > 0 && A[last_prefix] != A[i])
            last_prefix = prefix[last_prefix-1];

        if (A[last_prefix] == A[i])
            last_prefix++;
        prefix[i] = last_prefix;
    }
    last_prefix = 0;
    for (size_t i=0; i<B.length(); i++) {
        while (last_prefix > 0 && A[last_prefix] != B[i])
            last_prefix = prefix[last_prefix-1];

        if (A[last_prefix] == B[i])
            last_prefix++;

        if (last_prefix == A.length()) {
            result.push_back(i+1-A.length());
        }
    }
    return result;
}
};
int main() {
    A a;
    string A, B;
    cin >> A >> B;
    vector<size_t> t = a.K_M_P(A, B);
    if(t.size()>0){
        for(int i = 0;i<t.size()-1;i++){
            cout<<t[i]<<",";
        }
        cout<<t[t.size()-1];
    }
    else {
        cout<<-1;
    }
    return 0;
}
```