

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. 9383

Рыбников Р.А.

Преподаватель

Фирсов М.А.

Санкт-Петербург

2021

Цель работы.

Изучить алгоритм Кнута-Морриса-Пратта поиска подстроки в строке.
Реализовать данный алгоритм на Языке Программирования C++.

Задание.

Реализуйте алгоритм КМП и с его помощью для заданных шаблона $P(|P| \leq 15000)$ и текста $T(|T| \leq 5000000)$ найдите все вхождения P в T .

Входные данные:

Первая строка — P .

Вторая строка — T .

Выходные данные:

Индексы начал вхождений P в T , разделенных запятой если P не входит в T , то вывести -1.

Заданы 2 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 сдвигом abcdef.

Входные данные:

Первая строка — A .

Вторая строка — B .

Выходные данные:

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

Основные теоретические положения.

На вход программе подаётся две строки. Первая строка(`m_Str`) это подстрока, которая будет искаться во второй строке(`m_Text`). Строка вида `m_Str + # + m_Text` (символ `#` – это разделитель, который не должен нигде более встречаться) подается аргументом функции `FindPrefix()`, которая высчитывает префиксную функцию и возвращает ее в виде вектора. Значения в векторе указывают на фактическую длину наибольшего префикса в строке. Если префикс-функция содержит значение, равное длине `m_Str`, значит `m_Str` входит в `m_Text`.

Для нахождения циклического сдвига необходимо найти префикс-функцию от строки вида `m_Text + # + m_Str + m_Str`. Если находится вхождение, то циклический сдвиг имеется, а индекс вхождения – это величина его сдвига.

Оценка сложности алгоритма.

Если считать, что длина строки, которую вычисляет префикс-функция, равна n , а длина всего текста равна m , то сложность алгоритма КМП по памяти равна $O(n+m)$.

Сложность алгоритма по вычисления префикс функции линейна – $O(n)$, где n – это длина строки, которая подаётся на вход префикс-функции.

Выполнение работы.

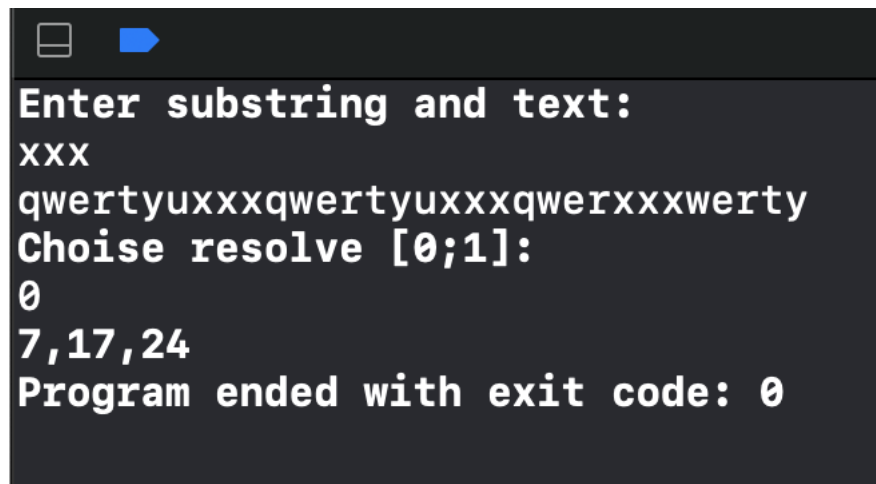
В программе реализован класс `KMP`, который содержит в себе два частных поля: `const std::string m_Text` и `const std::string m_Str`.

Так же в классе реализованы 4 метода:

- `std::vector<int> FindPrefix(const std::string &m_Text)` – поиск префикса строки.
- `std::vector<int> Resolve_1(const std::string &m_Text, const std::string &m_Str)` – метод, который реализовывает решение первого задания. Другими словами, метод позволяет искать заданную подстроку в данной строке(тексте).

- `std::vector<int> Resolve_2(const std::string &m_Text, const std::string &m_Str)` – метод, который реализовывает решение второго задания. Другими словами, метод позволяет узнать, является ли строка циклическим сдвигом другой подстроки, если да, то указать начало одной строки в другой.
- `std::vector<int> PrintResult(std::vector<int> res, const std::string *m_Text, const std::string &m_Str)` – метод, который позволяет печатать результат на экран.

Тестирование.

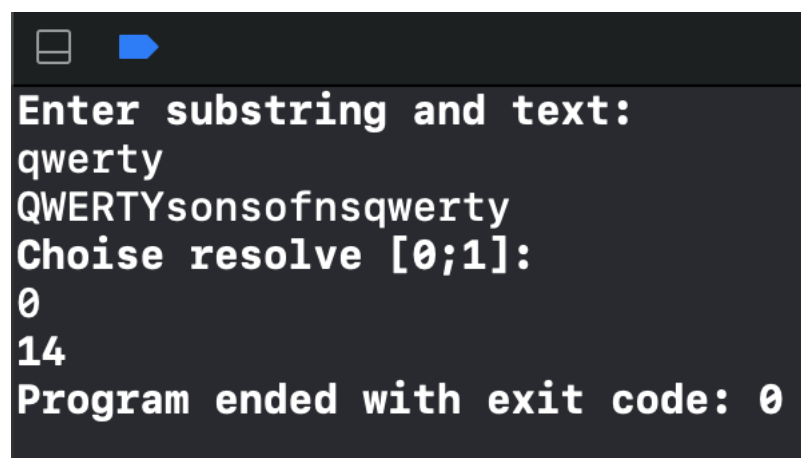


```

Enter substring and text:
xxx
qwertyuxxxqwertyuxxxqwerxxxwerty
Choise resolve [0;1]:
0
7,17,24
Program ended with exit code: 0

```

Рисунок 1 – Тестирование программы на выполнение первого задания.

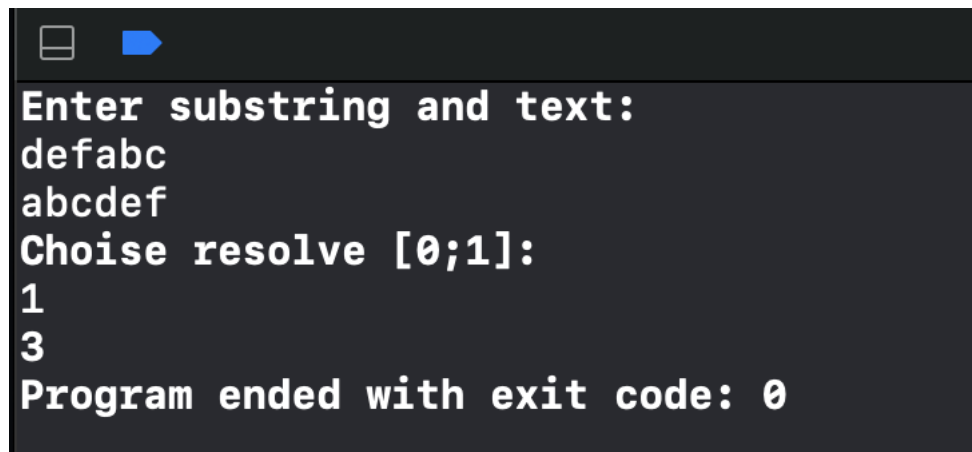


```

Enter substring and text:
qwerty
QWERTYsonsofnsqwerty
Choise resolve [0;1]:
0
14
Program ended with exit code: 0

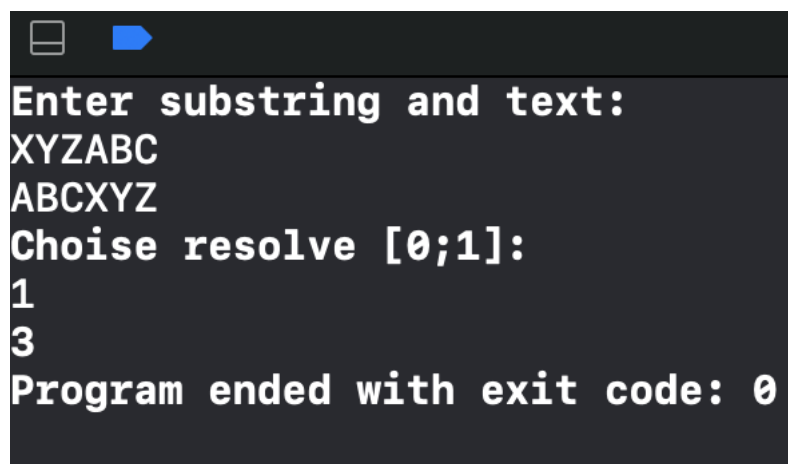
```

Рисунок 2 – Тестирование программы на выполнение первого задания.

A terminal window with a dark background and white text. The text shows the program's execution flow: it prompts for a substring and text, receives 'defabc' and 'abcdef', asks for a choice to resolve, receives '1' and '3', and finally reports that the program ended with exit code 0.

```
Enter substring and text:
defabc
abcdef
Choise resolve [0;1]:
1
3
Program ended with exit code: 0
```

Рисунок 3 – Тестирование программы на выполнение второго задания.

A terminal window with a dark background and white text, similar to Figure 3. It shows the program being tested with different inputs: 'XYZABC' and 'ABCXYZ'. The user enters '1' and '3' for the choice to resolve, and the program ends with exit code 0.

```
Enter substring and text:
XYZABC
ABCXYZ
Choise resolve [0;1]:
1
3
Program ended with exit code: 0
```

Рисунок 4 – Тестирование программы на выполнение второго задания.

Выводы.

Изучен алгоритм Кнута-Морриса-Пратта. Реализована программа на C++, которая для заданного текста находит все вхождения заданной строки и определяет циклический сдвиг.