

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

ЭЛЕКТРОТЕХНИЧЕСКИЙ ФАКУЛЬТЕТ

КАФЕДРА «ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И
АВТОМАТИЗИРОВАННЫХ СИСТЕМ»

ОТЧЁТ

**«ЛАБОРАТОРНАЯ №14:
АЛГОРИТМ КНУТА — МОРРИСА — ПРАТА»**

Дисциплина: «Программирование»

Выполнил:

Студент группы ИВТ-21-26

Безух Владимир Сергеевич

Проверил:

Доцент кафедры ИТАС

Полякова Ольга Андреевна

Пермь, 2022

Содержание

1.	Постановка задачи	3
2.	Анализ задачи.....	4
3.	Исходный код.....	5
4.	Анализ результатов.....	7

1. Постановка задачи

Найти образец строки в тексте с помощью алгоритма Кнута — Морриса — Прата.

2. Анализ задачи

1. Считать значения префикс-функции $\pi[i]$ по очереди: от $i = 1$ к $i = n - 1$ (значение $\pi[0]$ присвоим равным нулю).
2. Для подсчёта текущего значения $\pi[i]$ мы заводим переменную j , обозначающую длину текущего рассматриваемого образца. Изначально $j = \pi[i - 1]$.
3. Тестируем образец длины j , для чего сравниваем символы $s[j]$ и $s[i]$. Если они совпадают — то полагаем $\pi[i] = j + 1$ и переходим к следующему индексу $i + 1$. Если же символы отличаются, то уменьшаем длину j , полагая её равной $\pi[j - 1]$, и повторяем этот шаг алгоритма с начала.
4. Если мы дошли до длины $j = 0$ и так и не нашли совпадения, то останавливаем процесс перебора образцов и полагаем $\pi[i] = 0$ и переходим к следующему индексу $i + 1$.

3. Исходный код

```
#include <iostream>

void computeLPSArray(char* pattern, int pattern_length, int* longest_proper_suffix)
{
    int length = 0; longest_proper_suffix[0] = 0;

    int i = 1;
    while (i < pattern_length)
    {
        if (pattern[i] == pattern[length])
        {
            length++;
            longest_proper_suffix[i] = length;
            i++;
        }
        else // (pattern[i] != pattern[length])
        {
            if (length != 0)
                length = longest_proper_suffix[length - 1];
            else // if (length == 0)
            {
                longest_proper_suffix[i] = 0;
                i++;
            }
        }
    }
}

void KMPSearch(char* pattern, char* text)
{
    int pattern_length = strlen(pattern);
    int text_length = strlen(text);

    int* longest_proper_suffix = new int[pattern_length];
    computeLPSArray(pattern, pattern_length, longest_proper_suffix);

    int i = 0;
    int j = 0;
    while (i < text_length)
    {
        if (pattern[j] == text[i]) { j++; i++; }

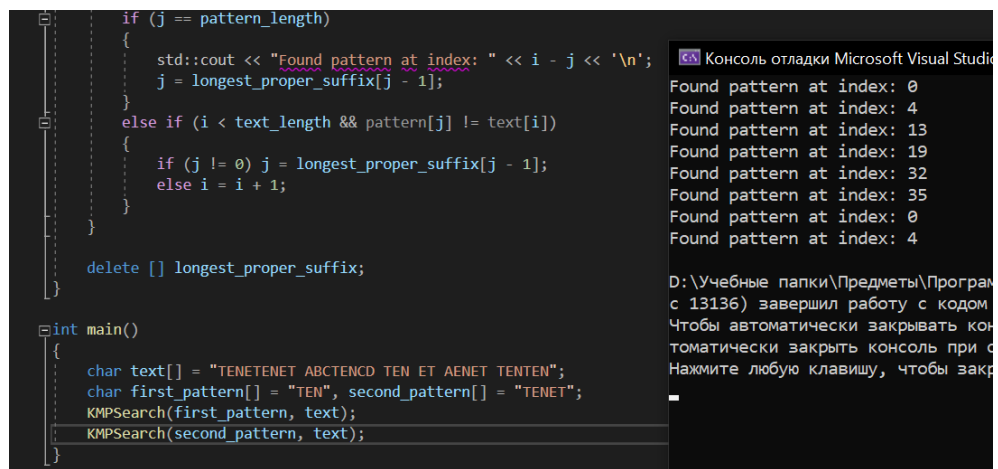
        if (j == pattern_length)
        {
            std::cout << "Found pattern at index: " << i - j << '\n';
            j = longest_proper_suffix[j - 1];
        }
        else if (i < text_length && pattern[j] != text[i])
        {
            if (j != 0) j = longest_proper_suffix[j - 1];
            else i = i + 1;
        }
    }

    delete [] longest_proper_suffix;
}
```

```
int main()
{
    char text[] = "TENETENET ABCTENCD TEN ET AENET TENTEN";
    char first_pattern[] = "TEN", second_pattern[] = "TENET";
    KMPSearch(first_pattern, text);
    KMPSearch(second_pattern, text);
}
```

4. Анализ результатов

Результаты работы программы (рис. 1).



The image shows a screenshot of a C++ program in Visual Studio. The program implements the KMP search algorithm. The code is as follows:

```
if (j == pattern_length)
{
    std::cout << "Found pattern at index: " << i - j << '\n';
    j = longest_proper_suffix[j - 1];
}
else if (i < text_length && pattern[j] != text[i])
{
    if (j != 0) j = longest_proper_suffix[j - 1];
    else i = i + 1;
}

delete [] longest_proper_suffix;

int main()
{
    char text[] = "TENETENET ABCTENCD TEN ET AENET TENTEN";
    char first_pattern[] = "TEN", second_pattern[] = "TENET";
    KMPSearch(first_pattern, text);
    KMPSearch(second_pattern, text);
}
```

The output of the program, displayed in the Visual Studio console, is:

```
Found pattern at index: 0
Found pattern at index: 4
Found pattern at index: 13
Found pattern at index: 19
Found pattern at index: 32
Found pattern at index: 35
Found pattern at index: 0
Found pattern at index: 4
```

The console also shows a message from the Windows operating system: "D:\Учебные папки\Предметы\Программирование\13136) завершил работу с кодом. Чтобы автоматически закрывать консоль при окончании работы программы, нажмите любую клавишу, чтобы закрыть консоль."

Рисунок 1 — Результаты