

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ  
УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

**ІКНІ**  
Кафедра ПЗ



**ЗВІТ**

До лабораторної роботи №10  
**на тему:** “Метод пошуку Кнута-Мориса-Прата”  
**з дисципліни:** "Алгоритми і структури даних”

**Лектор:**  
доцент кафедри ПЗ  
Коротєєва Т. О.

**Виконав:**  
студент групи ПЗ-24  
Губик А. С.

**Прийняв:**  
асистент кафедри ПЗ  
Вишневський К. О.

## Тема роботи

Метод пошуку Кнута-Мориса-Прата.

## Мета роботи

Вивчити алгоритм пошуку Кнута-Мориса-Прата. Здійснити програмну реалізацію алгоритму пошуку Кнута-Мориса-Прата.

## Індивідуальне завдання

У файлі задано дві стрічки  $S$  та  $P$ . Довжина стрічок більша 0 і менша 10000, стрічки містять лише букви латинського алфавіту. Вивести номери символів в порядку спадання починаючи з яких стрічка  $P$  входить в стрічку  $S$ .

## Індивідуальне завдання

1. Для заданого слова визначити префікс-функцію  $D$
2. Встановити  $i=0, j=0, d=1$ .
3. Поки  $j < m, i < n$
4. Перевірка: якщо  $S[i]=P[j]$ , то  $d++$ ,  $i++$ ,  $j++$  поки  $d \neq m$ .
5. Якщо  $j == n-1$  повернути  $i-j$ , інакше повернути  $-1$
6. Кінець.

## Теоретичні відомості

На вхід поступають два масиви символів -  $S$  розміром  $n$  (текст) та  $P$  розміром  $m$  (слово). Необхідно знайти перше входження слова в текст. Схема алгоритму полягає у поступовому порівнянні слова з текстом та у разі знайденого неспівпадіння зробити зсув по тексту на крок, що дорівнює різниці кількості співпалих символів та значення відповідної префікс-функції. Алгоритм використовує просте спостереження, що коли відбувається неспівпадіння тексту і слова, то слово містить у собі достатньо інформації для того, щоб визначити, де наступне входження може початися, таким чином пропускаючи кілька разів перевірку попередньо порівняних символів. Попередньо проводиться дослідження слова та визначається префікс-функція.

Префікс-функція стрічки – це довжина найбільшого префікса стрічки  $S[1..i]$ , який не співпадає з цією стрічкою і одночасно є її суфіксом. Простіше кажучи, це довжина найбільшого початку стрічки, що одночасно є її кінцем.

## Вихідний код

```
#include <iostream>
#include <fstream>
#include <vector>
#include <string>

std::vector<int> computeLPS(const std::string& pattern) {
    int patternLength = pattern.length();
    std::vector<int> lps(patternLength, 0);
```

```

    int j = 0; // Length of the previous longest prefix suffix

    for (int i = 1; i < patternLength; ++i) {
        while (j > 0 && pattern[i] != pattern[j]) {
            j = lps[j - 1];
        }

        if (pattern[i] == pattern[j]) {
            lps[i] = ++j;
        }
    }
    return lps;
}

int findFirstOccurrence(const std::string& text, const std::string& pattern) {
    int textLength = text.length();
    int patternLength = pattern.length();
    std::vector<int> lps = computeLPS(pattern);
    for(int i : lps)
        std::cout << i << ' ';
    std::cout << std::endl;
    int j = 0; // Index for pattern[]

    for (int i = 0; i < textLength; ++i) {
        while (j > 0 && text[i] != pattern[j]) {
            j = lps[j - 1];
        }

        if (text[i] == pattern[j]) {
            if (j == patternLength - 1) {
                return i - j; // Return the start index of the first occurrence
            } else {
                j++;
            }
        }
    }
    return -1; // Pattern not found in text
}

int main() {
    std::ifstream file("input.txt"); // Replace "input.txt" with your file name
    if (!file.is_open()) {
        std::cout << "Unable to open file!" << std::endl;
        return 1;
    }

    std::string S, P;
    if (std::getline(file, S) && std::getline(file, P)) {
        int index = findFirstOccurrence(S, P);
        if (index != -1) {
            std::cout << "Pattern P found at index: " << index << std::endl;
        } else {

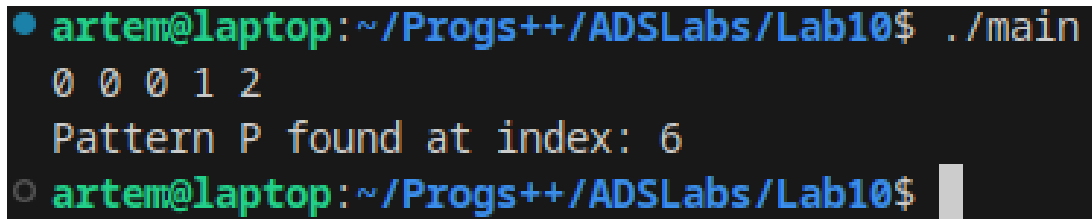
```

```

        std::cout << "Pattern P not found in string S" << std::endl;
    }
} else {
    std::cout << "File does not contain both strings!" << std::endl;
}

file.close();
return 0;
}

```



A terminal window with a dark background. The prompt is 'artem@laptop:~/Progs++/ADSLabs/Lab10\$'. The user enters './main'. The output is '0 0 0 1 2' followed by 'Pattern P found at index: 6'. The prompt returns to 'artem@laptop:~/Progs++/ADSLabs/Lab10\$'.

```

● artem@laptop:~/Progs++/ADSLabs/Lab10$ ./main
0 0 0 1 2
Pattern P found at index: 6
○ artem@laptop:~/Progs++/ADSLabs/Lab10$

```

Рис. 1:

## Висновок

Я ознайомився з алгоритмом Кнута-Моріса-Прата і що таке Longest Suffix Prefix.