

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерных технологий

Продвинутые алгоритмы и структуры данных

Отчет о задаче № F

Выполнила
студентка

Ершова А. И.

Группа № P4115

Преподаватель: Косяков Михаил Сергеевич

г. Санкт-Петербург

2024

Условие задачи

Г. Праздничная лента

Ограничение времени	2 секунды
Ограничение памяти	256Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

У вас есть лента, на которой написано n латинских букв. Буквы пронумерованы от 1 до n .

Также у вас есть текст, состоящий ровно из m слов.

Вы хотите выбрать и закрасить некоторое количество букв на ленте, чтобы получить текст, состоящий в точности из заданных слов. Для того чтобы текст был читаемым и понятным, необходимо, чтобы после закрашивания всех выбранных букв было выполнено следующее:

- между каждой парой слов была хотя бы одна закрашенная буква (при этом, наличие закрашенных букв перед первым или после последнего слова текста не обязательно);
- в полученном тексте не было никаких других слов, кроме указанных;
- все слова текста должны следовать друг за другом в указанном порядке.

Для заданной строки s , описывающей ленту, и m слов, составляющих текст, вам необходимо определить, можете ли вы закрашиванием некоторого количества букв получить текст, удовлетворяющий **всем** условиям выше.

Например, если на ленте написаны следующие 17 букв: «HHappyabNewQYear», и вы хотите получить текст, состоящий из трех слов «Happy», «New» и «Year», то для выполнения всех вышеуказанных условий достаточно закрасить буквы с индексами 1, 2, 8, 9 и 13.

Формат ввода

Первая строка входных данных содержит ровно два целых положительных числа n ($1 \leq n \leq 10^5$) и m ($1 \leq m \leq 10^3$) — количества букв на ленте и слов в тексте соответственно. Вторая строка входных данных содержит строку s , состоящую из n латинских букв, которые могут быть как строчными, так и заглавными.

Далее следуют m строк, каждая из которых содержит слово текста t_i ($1 \leq |t_i| \leq 10^5$). Каждое слово t_i также состоит из латинских букв, которые могут быть как строчными, так и заглавными.

Гарантируется, что сумма длин всех строк t_i не превышает числа n .

Формат вывода

Выведите:

- «YES», если можно выбрать и закрасить некоторое количество букв на ленте таким образом, чтобы получился заданный текст из m слов и все условия, указанные в задаче, были выполнены;
- «NO» иначе.

Вы можете выводить ответ в любом регистре (например, строки «yEs», «yes», «Yes» и «YES» будут распознаны как положительный ответ).

Пример 1

Ввод 

17 3
HHHappyabNewQYear
Happy
New
Year

Вывод 

YES

Пример 2

Ввод 

9 2
HelloITMO
Hello
ITMO

Вывод 

NO

Решение:

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

using namespace std;

int main() {
    ifstream fin;
    fin.open("../input.txt");
    int n, m;
    fin >> n >> m;
    string str;
    fin >> str;
    int str_pointer = 0;
    int prev_str_pointer = 0;
    int word_pointer = 0;
    bool fool_word = false;
    string ans = "NO";

    vector<string> words(m);
    for (int i = 0; i < m; ++i) {
```

```

fin >> words[i];
for (int j = str_pointer; j < n; ++j) {
    if (words[i][word_pointer] == str[j]) {
        word_pointer++;
        if (word_pointer == (int)words[i].size()) {
            word_pointer = 0;
            str_pointer = j + 2;
            fool_word = true;
            break;
        }
    } else if (words[i][0] == str[j]) {
        word_pointer = 1;
        if (words[i].size() == 1) {
            word_pointer = 0;
            str_pointer = j + 2;
            fool_word = true;
            break;
        }
    } else if (word_pointer != 0) {
        word_pointer = 0;
    }
}
if (word_pointer == 0 && i == m - 1 && fool_word) {
    ans = "YES";
}
if (str_pointer == prev_str_pointer) {
    break;
} else {
    prev_str_pointer = str_pointer;
}
fool_word = false;
}
cout << ans << '\n';
}

```

Объяснение решения:

Есть несколько указателей: `str_pointer`, `prev_str_pointer` и `word_pointer`. По ходу считывания `m` слов программа сразу проверяет можно ли найти слово в имеющейся строке. Если первого же слова программой не будет найдено, то считаются все слова, и программа завершится, не произведя никаких дополнительных действий.

`str_pointer` является указателем в первоначальной строке, он указывает на следующий символ, который будут проверять на наличие символов, совпадающими со словами. `word_pointer` указывает на символ в текущем проверяемом слове. Если слово найдено в строке, то этот указатель обнуляется. Также он обнуляется, если не все символы в строке совпадают со словом. `prev_str_pointer` необходим для того, чтобы

проверять нашлось ли запрашиваемое слово. Если `prev_str_pointer` равен `str_pointer` значит слово не было найдено. При нахождении слова `str_pointer` сразу увеличивается на 2, т.к. по условию необходимо, чтоб между словами был хотя бы 1 символ.