

```

#include <iostream>
#include <string>
#include <vector>

using namespace std;

//패턴pat의 실패함수
vector<int> failure(string& s) {
    vector<int> f(s.size());

    int j = 0;
    for (int i=1; i<s.size(); i++) {
        while (j>0 && s[i]!=s[j]) j = f[j-1];
        if (s[i]==s[j]) f[i] = ++j;
    } //i loop

    return f;
}

int main() {
    string str, pat;
    cin >> str >> pat;

    vector<int> f = failure(pat);
    int j = 0;
    for (int i=0; i<str.size(); i++) {
        while (j>0 && str[i] != pat[j]) j = f[j-1];
        if (str[i] == pat[j]) j++;
        if (j == pat.size()) { //str에서 pat을 발견한 경우-
            cout << 1;
            return 0;
            /*
                str에서 pat을 단 한 번 발견하고 싶으면 return 0 처리로 main 함수를 종료해도 되지만,
                str에서 pat을 여러 번 발견하고 싶으면 return 0 대신
                    j = f[j-1];
                을 뒤서 반복문을 중단하지 않도록 만든다.
                str에서 pat의 위치는 (0부터 시작하는 index 기준) i-j+1 이다.
            */
        }
    } //i loop
    cout << 0;
    return 0;
}

```