문자열들의 탐색과 정렬

문제

- 문자열들을 키워드 문자열이 처음 등장하는 위치에 따라 오름차순으로 정렬합니다.

  ex) 키워드 문자열 pi, 문자열 piaa 일 때 키워드 문자열이 처음 등장하는 위치는 0.

  키워드 문자열 pi, 문자열 aapiaa 일 때 키워드 문자열이 처음 등장하는 위치는 2.

- 키워드가 등장하지 않는 문자열은 위치를 –1로 간주합니다.

- 정렬은 stable 해야 합니다.

- KMP 알고리즘을 사용하여 스트링 탐색을 하고, 코드의 해당 부분에 주석으로 표시하세요. (불이행 시 오답 처리)

조건

- 첫 번째 줄에 키워드 문자열이 입력됩니다.

- 이어서, 임의의 문자열이 입력됩니다. 0이 입력되면 입력이 종료됩니다.

- 키워드의 길이는 8자 이하입니다.

- 각 문자열의 길이는 250자 이하입니다.

입력과 출력 예시

pio asdpiof apiosdf aspiodf asdfpio pioasdf 0 pioasdf apiosdf aspiodf asdpiof asdfpio	pio abpiocd abpiocd apiobcd abcpiod 0 apiobcd abpiocd abpiocd abcpiod

모범 답안

```cpp
#include <iostream>
#include <queue>
#include <algorithm>
#include <string>
using namespace std;

int* InitNext(string p) {
    const int m = (int)p.size();
    int j = 0;
    int *in = new int[m];
    in[0] = -1;
    for (int i = 1; i < m; i++) {
        while (j > 0 && p[i] != p[j])
            j = in[j - 1];
        if (p[i] == p[j])
            in[i] = ++j;
    }
    return in;
}

int kmp(string p, string t, int* next) {
    int m = (int)p.size(), n = (int)t.size(), i, j;
    for (i = 0, j = 0; j < m && i < n; i++, j++)
        while (j >= 0 && t[i] != p[j])
            j = next[j];
    if (j == m) return i - m;
    else return -1;
}

bool cmp(pair<int, string> a, pair<int, string> b) {
    return a.first < b.first;
}

int main() {
    int i = 0, j = 0;
    string k, s;

    pair<int, string> p[100];
    cin >> k;
    int* next = InitNext(k);
    while (true) {
        cin >> s;
        if (string(s) == "0") break;
        p[i++] = make_pair(kmp(k, s, next), s);
    }
    stable_sort(p, p + i, cmp);
    while (j != i) {
        cout << p[j++].second << endl;
    }

    return 0;
}
```