Knuth-Morris-Pratt (KMP) Algorithm Implementation

```cpp
#include <bits/stdc++.h>
using namespace std;

vector<int> computePrefix(string& pattern) {
    int m = pattern.size();
    vector<int> pi(m, 0);
    int k = 0;
    for (int q = 1; q < m; ++q) {
        while (k > 0 && pattern[k] != pattern[q]) {
            k = pi[k - 1];
        }
        if (pattern[k] == pattern[q]) {
            k++;
        }
        pi[q] = k;
    }
    return pi;
}

void kmp(string& text, string& pattern) {
    int n = text.size();
    int m = pattern.size();
    vector<int> pi = computePrefix(pattern);
    int q = 0;
    for (int i = 0; i < n; ++i) {
        while (q > 0 && pattern[q] != text[i]) {
            q = pi[q - 1];
        }
        if (pattern[q] == text[i]) {
            q++;
        }
        if (q == m) {
            cout << "Pattern found at index " << i - m + 1 <<
               " (using KMP)" << endl;
            q = pi[q - 1];
        }
    }
}
```