```
#include <iostream>

using namespace std;


#define d 256  // number of characters in the input alphabet


void rabinKarp(string text, string pattern, int q) {

    int n = text.length();

    int m = pattern.length();

    int h = 1;

    int p = 0, t = 0; // hash values for pattern and text

    int i, j;


    // h = pow(d, m-1) % q

    for (i = 0; i < m - 1; i++)

        h = (h * d) % q;


    // calculate hash for pattern and first window

    for (i = 0; i < m; i++) {

        p = (d * p + pattern[i]) % q;

        t = (d * t + text[i]) % q;

    }


    // slide the pattern over text

    for (i = 0; i <= n - m; i++) {

        if (p == t) {

            // check characters one by one
            for (j = 0; j < m; j++)
                if (text[i + j] != pattern[j])
                    break;
            if (j == m)
                cout << "Pattern found at index " << i << endl;
        }
        // calculate hash for next window
        if (i < n - m) {
            t = (d * (t - text[i] * h) + text[i + m]) % q;
            if (t < 0)
                t += q;
        }
    }
}

int main() {
    string text = "ABCCDDAEFG";
    string pattern = "CDD";
    int q = 101; // A prime number
    rabinKarp(text, pattern, q);
    return 0;
}
```