

```

#include <stdio.h>
#include <string.h>

#define d 256
#define q 101

int hashValue(char str[], int length, int base)
{
    int hash = 0,i;
    for (i = 0; i < length; i++) {
        hash = (hash * base + str[i]) % q;
    }
    return hash;
}

void rabinKarp(char text[], char pattern[]) {
    int n = strlen(text);
    int m = strlen(pattern);
    int patternHash = hashValue(pattern, m, d);
    int textHash = hashValue(text, m, d);
    int h = 1,i;

    for (i = 0; i < m - 1; i++) {
        h = (h * d) % q;
    }

    for (i = 0; i <= n - m; i++) {
        if (patternHash == textHash)
        {
            int j;

            for (j = 0; j < m; j++)
            {
                if (text[i + j] != pattern[j]) {
                    break;
                }
            }

            if (j == m)
            {
                printf("Pattern found at index %d \n", i);
            }
        }

        if (i < n - m) {
            textHash = (d * (textHash - text[i] * h) + text[i + m]) % q;

            if (textHash < 0) {
                textHash = textHash + q;
            }
        }
    }
}

```

```
}
```

```
int main() {  
    char text[] = "ABCCDDAEFG";  
    char pattern[] = "CDD";  
  
    rabinKarp(text, pattern);  
  
    return 0;  
}
```