

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  #define ll long long
4  #define inf (int)1e9
5
6  Rabin Karp Matching:
7  //to find all occurrences of s in t
8
9  vector<int> rabin_karp_matching(string const& s, string const& t){
10     const int p1=1333, p2=137; // 31 , 53 , 137 , 1331 , 1333
11     const int m1=1e9+9, m2=1e9+7; // 1e9+7 , 1e9+9
12     int S = s.size(), T = t.size();
13
14     vector<ll> pow1(max(S , T)), pow2 (max(S , T));
15     vector<ll> h1(T+1 , 0), h2(T+1 , 0);
16     pair<ll , ll> h_s = {0 , 0};
17
18     pow1[0] = 1 , pow2[0] = 1;
19     for (int i = 1; i < (int)pow1.size(); i++) {
20         pow1[i] = (pow1[i-1] * p1) % m1;
21         pow2[i] = (pow2[i-1] * p2) % m2;
22     }
23
24     h1[0]=t[0]-'a'+1;
25     h2[0]=t[0]-'a'+1;
26     for(int i=1 ; i<T ; i++){
27         h1[i] = (h1[i-1]*p1 + t[i]-'a'+1)%m1;
28         h2[i] = (h2[i-1]*p2 + t[i]-'a'+1)%m2;
29     }
30
31     for (int i = 0; i < S; i++) {
32         h_s.first = (h_s.first*p1 + s[i]-'a'+1)%m1;
33         h_s.second = (h_s.second*p2 + s[i]-'a'+1)%m2;
34     }
35
36     vector<int> occurrences;
37     for (int i = 0; i + S - 1 < T; i++) {
38         int j = i+S-1;
39         pair<ll , ll> h_t = {0 , 0};
40
41         //calc first hash
42         h_t.first = h1[j];
43         if(i) h_t.first -= h1[i-1] * pow1[j-i+1];
44         h_t.first = (h_t.first%m1 + m1)%m1;
45
46         //calc second hash
47         h_t.second = h2[j];
48         if(i) h_t.second -= h2[i-1] * pow2[j-i+1];
49         h_t.second = (h_t.second%m2 + m2)%m2;
50
51         if(h_t==h_s) occurrences.push_back(i);
52     }
53
54     return occurrences;
55 }

```