

```

1  #include <bits/stdc++.h>
2
3  using namespace std;
4
5  using ll = long long;
6  using vi = vector<int>;
7  using vll = vector<ll>;
8  using vvi = vector<vector<int>>;
9  using vvll = vector<vector<ll>>;
10
11 const int MAX_STRING_SIZE = 1000000;
12
13 ll base = 1007;
14
15 vector<ll> MODS({2147483647, 6700417, 524287, 1000000007})
16
17 );
18
19 vector<vector<ll>> invs(MODS.size()), pows(MODS.size());
20
21 ll powm(ll e, ll x, ll m) {
22     if (x == 0) return 1;
23     if (x == 1) return e;
24     if (x % 2 == 0) return powm((e * e) % m, x / 2, m);
25     return (e * powm(e, x - 1, m)) % m;
26 }
27
28 void RHinit() {
29     int i = 0;
30     for (auto &p : MODS) {
31         invs[i].resize(MAX_STRING_SIZE + 1);
32         invs[i][0] = 1;
33         ll inv = powm(base, p - 2, p);
34         for (int j = 1; j <= MAX_STRING_SIZE; j++) {
35             invs[i][j] = (invs[i][j - 1] * inv) % p;
36         }
37         pows[i].resize(MAX_STRING_SIZE + 1);
38         pows[i][0] = 1;
39         for (int j = 1; j <= MAX_STRING_SIZE; j++) {
40             pows[i][j] = (pows[i][j - 1] * base) % p;
41         }
42         i++;
43     }
44 }
45
46 struct RH {
47     int n;
48     string s;
49     vector<vector<ll>> hs;
50

```

```

51  RH(string &s) : s(s) {
52      if (pows[0].empty()) {
53          RHinit();
54      }
55      n = s.size();
56      hs.resize(MODS.size());
57      for (int i = 0; i < MODS.size(); i++) {
58          hs[i].resize(n + 1);
59          hs[i][0] = 0;
60          for (int j = 0; j < n; j++) {
61              hs[i][j + 1] = (hs[i][j] + pows[i][j] * s[j]) % MODS[i];
62          }
63      }
64  }
65
66  vector<ll> substr(int l, int r) {
67      vector<ll> ret(MODS.size());
68      for (int i = 0; i < MODS.size(); i++) {
69          ret[i] = (hs[i][r] - hs[i][l] + MODS[i]) % MODS[i];
70          ret[i] *= invs[i][l];
71          ret[i] %= MODS[i];
72      }
73      return ret;
74  }
75 };
76
77 struct N {
78     vector<ll> v;
79 };
80
81 bool operator<(const N &l, const N &r) {
82     if (l.v[0] == r.v[0]) {
83         if (l.v[1] == r.v[1]) {
84             if (l.v[2] == r.v[2]) {
85                 return l.v[3] < r.v[3];
86             }
87             return l.v[2] < r.v[2];
88         }
89         return l.v[1] < r.v[1];
90     }
91     return l.v[0] < r.v[0];
92 }
93
94 int main() {
95     int n, m;
96     cin >> n >> m;
97     string s;
98     cin >> s;
99     RH h(s);
100     int l = 0, r = 1;

```

```
101  set<N> ss;
102  for (int i = 0; i < m; i++) {
103      string t;
104      cin >> t;
105      if (t == "L++") {
106          l++;
107          r;
108      } else if (t == "L--") {
109          l--;
110          r;
111      } else if (t == "R++") {
112          l;
113          r++;
114      } else if (t == "R--") {
115          l;
116          r--;
117      }
118      ss.insert(N{h.substr(l, r)});
119  }
120  cout << ss.size() << endl;
121  return 0;
122  }
123
```