

```
1  #include <bits/stdc++.h>
2  #define pb push back
3  #define mp make pair
4  #define F first
5  #define S second
6  #define eoq printf("eoq\n")
7  #define digitCountDec(a) (int)floor(1 + log10((double)a))
8  #define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
9  using namespace std;
10 typedef unsigned long long int llu;
11 typedef long long int ll;
12 typedef pair<ll, ll> ii;
13 typedef pair<double, double> dd;
14 typedef vector<ll> vi;
15 typedef vector<ii> vii;
16 int dr[] = { 0, 1, -1, 0, 1, -1, -1, 1};
17 int dc[] = {-1, 0, 0, 1, 1, 1, -1, -1};
18
19 ll gcd (ll x, ll y) {
20
21     return (y != 0 ? gcd (y, x%y) : x);
22 }
23
24 ll lcm (ll x, ll y) {
25
26     return (x / gcd (x, y) * y);
27 }
28
29 main () {
30
31
32 }
33
```

```
1  #include <bits/stdc++.h>
2  #define eps 1e-8
3  #define pi acos(-1)
4  using namespace std;
5  typedef long long ll;
6  double n, l, lo, hi, b, B, h, a;
7
8  double f (double x) {
9
10     return pi * n * (a*a*x*x*x / 3.0 + a*b*x*x + x*b*b);
11 }
12
13 main () {
14
15     int t; scanf ("%d", &t);
16     while (t--) {
17
18         scanf ("%lf %lf", &n, &l);
19         scanf ("%lf %lf %lf", &b, &B, &h);
20         a = (B-b) / h;
21         lo = 0, hi = h;
22
23         for (int i = 0; i < 100; ++i) {
24
25             double mid = (hi+lo) / 2.0;
26
27             if (f (mid) > l)
28                 hi = mid;
29             else
30                 lo = mid;
31         }
32
33         printf ("%0.2lf\n", lo);
34     }
35 }
36
```

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  typedef long long int ll;
4
5  ll n, x;
6  string s;
7
8  ll character [100005], BIT[100005];
9
10 ll get sum (ll index) {
11     ll sum = 0;
12     while (index > 0) {
13         sum += BIT[index];
14         index -= index & (-index);
15     }
16     cout << sum << '\n';
17 }
18
19 void update (int index, int val) {
20     index = index + 1;
21     while (index <= n) {
22         BIT[index] += val;
23         index += index & (-index);
24     }
25 }
26
27 void new BIT () {
28     for (int i = 1; i <= n; ++i)
29         BIT[i] = 0;
30     for (int i = 0; i < n; ++i)
31         update (i, character[i]);
32 }
33
34 main () {
35     ios base::sync with stdio (0);
36     cin.tie (0);
37     cin >> n;
38     for (int i = 0; i < n; ++i)
39         cin >> character[i];
40     new BIT ();
41     while (cin >> s >> x) {
42         --x;
43         if (s == "a")
44             update (x, -character[x]);
45         else
46             get sum (x);
47     }
48 }
```

```

1  #include <bits/stdc++.h>
2  #define pb push back
3  #define mp make pair
4  #define F first
5  #define S second
6  #define eoq printf("eoq\n")
7  #define digitCountDec(a) (int)floor(1 + log10((double)a))
8  #define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
9  using namespace std;
10 typedef unsigned long long int llu;
11 typedef long long int ll;
12 typedef pair<ll, ll> ii;
13 typedef pair<double, double> dd;
14 typedef vector<ll> vi;
15 typedef vector<ii> vii;
16 int dr[] = { 0, 1, -1, 0, 1, -1, -1, 1};
17 int dc[] = {-1, 0, 0, 1, 1, 1, -1, -1};
18
19 ll gcd (ll x, ll y) {
20
21     return (y != 0 ? gcd (y, x%y) : x);
22 }
23
24 ll lcm (ll x, ll y) {
25
26     return (x / gcd (x, y) * y);
27 }
28
29 const ll N = 1e4 + 10;
30 llu mask[N];
31
32 main () {
33
34     ll t; scanf ("%lld", &t);
35     while (t--) {
36
37         ll n; scanf ("%lld", &n);
38         for (ll i = 1; i <= n; ++i) {
39
40             mask[i] = 0;
41             ll m; scanf ("%lld", &m);
42             while (m--) {
43
44                 ll x; scanf ("%lld", &x);
45                 mask[i] |= (1LL << x);
46             }
47         }
48
49         ll m; scanf ("%lld", &m);
50         while (m--) {
51
52             ll op, x, y; scanf ("%lld %lld %lld", &op, &x, &y);
53
54             if (op == 1)
55                 printf ("%d\n", builtin_popcountll (mask[x] & mask[y]));
56
57             else
58                 printf ("%d\n", builtin_popcountll (mask[x] | mask[y]));
59         }
60     }
61 }
62

```

```
1  #include <bits/stdc++.h>
2  #define pb push back
3  #define mp make pair
4  #define F first
5  #define S second
6  #define eoq printf("eoq\n")
7  #define digitCountDec(a) (int)floor(1 + log10((double)a))
8  #define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
9  using namespace std;
10 typedef unsigned long long int llu;
11 typedef long long int ll;
12 typedef pair<ll, ll> ii;
13 typedef pair<double, double> dd;
14 typedef vector<ll> vi;
15 typedef vector<ii> vii;
16 int dr[] = { 0, 1, -1, 0, 1, -1, -1, 1};
17 int dc[] = {-1, 0, 0, 1, 1, 1, -1, -1};
18
19 ll gcd (ll x, ll y) {
20
21     return (y != 0 ? gcd (y, x%y) : x);
22 }
23
24 ll lcm (ll x, ll y) {
25
26     return (x / gcd (x, y) * y);
27 }
28
29 const ll N = 1e4 + 10;
30 bitset<70> mask[N];
31
32 main () {
33
34     ll t; scanf ("%lld", &t);
35     while (t--) {
36
37         ll n; scanf ("%lld", &n);
38         for (ll i = 1; i <= n; ++i) {
39
40             mask[i].reset ();
41             ll m; scanf ("%lld", &m);
42             while (m--) {
43
44                 ll x; scanf ("%lld", &x);
45                 mask[i][x] = true;
46             }
47         }
48
49         ll m; scanf ("%lld", &m);
50         while (m--) {
51
52             ll op, x, y; scanf ("%lld %lld %lld", &op, &x, &y);
53
54             if (op == 1)
55                 printf ("%lu\n", (mask[x] & mask[y]).count ());
56
57             else
58                 printf ("%lu\n", (mask[x] | mask[y]).count ());
59         }
60     }
61 }
62
```

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  bool flag = false;
5  set<int> numbers;
6  map<int, int> ans, l;
7  vector<int> pos;
8
9  main() {
10
11     ios base::sync with stdio (0);
12     cin.tie (0);
13
14     int n, v;
15
16     cin >> n >> v;
17     numbers.insert(v);
18
19     --n;
20     while (n--) {
21         cin >> v;
22         auto it = numbers.upper bound(v);
23
24         if (it != numbers.end() && l.count(*it) == 0)
25             l[*it] = v;
26
27         else
28             it--;
29
30         numbers.insert (v);
31         pos.push back (v);
32         ans[v] = *it;
33     }
34
35     cin >> n;
36     while (n--) {
37         cin >> v;
38
39         if (flag)
40             cout << ' ';
41
42         else
43             flag = true;
44
45         cout << ans[pos[v-2]];
46     }
47
48     cout << '\n';
49 }
50
51
52
```

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  typedef long long int ll;
4
5  ll n, m;
6
7  char v[205][205];
8  ll sum[205][205];
9
10 void make () {
11     for (ll i = 0; i <= n; ++i) {
12         for (ll j = 0; j <= m; ++j) {
13             if (!i || !j)
14                 sum[i][j] = 0;
15             else
16                 sum[i][j] = v[i][j] + sum[i-1][j] + sum[i][j-1] - sum[i-1][j-1];
17         }
18     }
19 }
20
21 char solve (ll x) {
22     for (ll i = 1; i <= n-x; ++i)
23         for (ll j = 1; j <= m-x; ++j)
24             if (sum[i+x][j+x] + sum[i-1][j-1] - sum[i-1][j+x] - sum[i+x][j-1] == 0)
25                 return true;
26     return false;
27 }
28
29 main () {
30     scanf ("%lld %lld", &n, &m);
31     for (ll i = 1; i <= n; ++i) {
32         for (ll j = 1; j <= m; ++j) {
33             ll x; scanf ("%lld", &x);
34             v[i][j] = !x;
35         }
36     }
37     make ();
38     ll t; scanf ("%lld", &t);
39     while (t--) {
40         ll x; scanf ("%lld", &x);
41         puts (solve (x-1) ? "yes" : "no");
42     }
43 }
```

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  typedef long long int ll;
4
5  const ll N = 1e6;
6  ll st[N], lazy[N];
7  ll n;
8
9  void update (ll p, ll l, ll r, ll a, ll b, ll v) {
10
11     if (lazy[p]!=0) {
12         st[p] += (r-l+1) * lazy[p];
13         if (l != r) {
14             lazy[p<<1] += lazy[p];
15             lazy[(p<<1)+1] += lazy[p];
16         }
17         lazy[p]=0;
18     }
19
20     if (r<a || l>b)
21         return;
22
23     if (a<=l && b>=r) {
24         st[p] += (r-l+1) * v;
25         if (l!=r) {
26             lazy[p<<1] += v;
27             lazy[(p<<1)+1] += v;
28         }
29         return;
30     }
31
32     ll mid = (l+r) >> 1;
33     update (p<<1, l, mid, a, b, v);
34     update ((p<<1)+1, mid+1, r, a, b, v);
35     st[p] = st[p<<1] + st[(p<<1)+1];
36 }
37
38 ll query (ll p, ll l, ll r, ll a, ll b) {
39
40     if (lazy[p]!=0) {
41         st[p] += (r-l+1) * lazy[p];
42         if (l!=r) {
43             lazy[p<<1] += lazy[p];
44             lazy[(p<<1)+1] += lazy[p];
45         }
46         lazy[p]=0;
47     }
48
49     if (r<a || l>b)
50         return 0;
51
52     if (a<=l && b>=r)
53         return st[p];
54
55     ll mid = (r+l) >> 1;
56     return query (p<<1, l, mid, a, b) + query ((p<<1)+1, mid+1, r, a, b);
57 }

```



```
72 }
73
74 main () {
75     ll tc; scanf ("%lld", &tc);
76     while (tc--){
77         ll q; scanf ("%lld %lld", &n, &q);
78
79         memset (st, 0, sizeof st);
80         memset (lazy, 0, sizeof lazy);
81
82         ll a, b, v;
83         while (q--) {
84             ll x; scanf ("%lld %lld %lld", &x, &a, &b);
85             if (x)
86                 printf ("%lld\n", query (1, 1, n, a, b));
87             else {
88                 scanf ("%lld", &v);
89                 update (1, 1, n, a, b, v);
90             }
91         }
92     }
93 }
94
95
96
97
98
99
100
```

```

1  #include <bits/stdc++.h>
2  #define pb push back
3  #define mp make pair
4  #define F first
5  #define S second
6  #define oioi printf("oioi\n")
7  #define digitCountDec(a) (int)floor(1 + log10((double)a))
8  #define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
9  using namespace std;
10 typedef unsigned long long int llu;
11 typedef long long int ll;
12 typedef vector<ll> vi;
13 typedef pair<ll, ll> ii;
14 typedef pair<double, double> pdd;
15 typedef vector<ii> vii;
16 int dr[] = {0, 1, -1, 0, 1, -1, -1, 1};
17 int dc[] = {-1, 0, 0, 1, 1, 1, -1, -1};
18
19 ll gcd (ll x, ll y) {
20
21     return (y != 0 ? gcd (y, x%y) : x);
22 }
23
24 ll lcm (ll x, ll y) {
25
26     return (x / gcd (x, y) * y);
27 }
28
29 double dist (pdd a, pdd b) {
30
31     return sqrt ((a.F-b.F)*(a.F-b.F) + (a.S-b.S)*(a.S-b.S));
32 }
33
34 const ll N = 2 * 1e3 + 10;
35 ll n;
36 pdd p0, points[N];
37 stack<pdd> s;
38
39 pdd nextToTop () {
40
41     pdd p = s.top (); s.pop ();
42     pdd res = s.top (); s.push (p);
43     return res;
44 }
45
46 int orientation (pdd p, pdd q, pdd r) {
47
48     double val = (q.S - p.S) * (r.F - q.F) - (q.F - p.F) * (r.S - q.S);
49
50     if (val == 0)
51         return 0;
52
53     return (val > 0) ? 1 : 2;
54 }
55
56 int compare (pdd a, pdd b) {
57
58     double o = orientation (p0, a, b);
59
60     if (o == 0)
61         return dist (p0, b) >= dist (p0, a);
62
63     return o == 2;
64 }
65
66 void solve () {
67
68     double ymin = points[0].S;
69     ll min = 0;
70     for (ll i = 1; i < n; ++i) {
71

```

```

72     double y = points[i].S;
73     if ((y < ymin) or (ymin == y and points[i].F < points[min].F))
74         ymin = points[i].S, min = i;
75 }
76
77 swap (points[0], points[min]);
78 p0 = points[0];
79 sort (points+1, points+n, compare);
80
81 ll m = 1;
82 for (ll i = 1; i < n; ++i) {
83     while (i < n-1 and orientation (p0, points[i], points[i+1]) == 0)
84         i++;
85     points[m++] = points[i];
86 }
87
88 s.push (points[0]); s.push (points[1]); s.push (points[2]);
89 for (ll i = 3; i < m; i++) {
90     while (orientation (nextToTop (), s.top (), points[i]) != 2)
91         s.pop ();
92     s.push (points[i]);
93 }
94
95 double ans = 0;
96 pdd fst = s.top ();
97 pdd before = s.top(); s.pop();
98 while (!s.empty()) {
99     ans += dist (before, s.top());
100    before = s.top (); s.pop ();
101 }
102
103 ans += dist (before, fst);
104 printf ("Tera que comprar uma fita de tamanho %.2lf.\n", ans);
105 }
106
107 void read () {
108     for (ll i = 0; i < n; ++i)
109         scanf ("%lf %lf", &points[i].F, &points[i].S);
110 }
111
112 main () {
113     while (scanf ("%lld", &n), n) {
114         read ();
115         solve ();
116     }
117 }
118
119
120
121
122
123
124
125
126

```

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  typedef long long int ll;
4  typedef unsigned long long int ull;
5  typedef pair<int,int> pii;
6  const int MAX = 51;
7
8  int n, m;
9
10 char board 1[MAX][MAX], board 2[MAX][MAX];
11 char visited [MAX][MAX][MAX][MAX];
12
13 bool validPosition (int x1, int y1, int x2, int y2) {
14     if (x1 < 0 || x1 == n || x2 < 0 || x2 == n)
15         return false;
16
17     if (y1 < 0 || y1 == m || y2 < 0 || y2 == m)
18         return false;
19
20     if (board 1[x1][y1] == 'B' || board 2[x2][y2] == 'B')
21         return false;
22
23     return true;
24 }
25
26 void bfs (pii source 1, pii source 2, pii destination 1, pii destination 2) {
27     memset (visited, false, sizeof visited);
28     queue< pair<pair<pii, pii>, int> > q;
29
30     q.push ({{source 1, source 2}, 0});
31     visited [source 1.first][source 1.second][source 2.first][source 2.second] =
32     true;
33
34     while (!q.empty ()) {
35
36         int x1 = q.front ().first.first.first;
37         int y1 = q.front ().first.first.second;
38         int x2 = q.front ().first.second.first;
39         int y2 = q.front ().first.second.second;
40         int moves = q.front ().second;
41         q.pop ();
42
43         if (moves == MAX) {
44             printf ("impossivel\n");
45             return;
46         }
47
48         if (x1 == destination 1.first && y1 == destination 1.second && x2 ==
49         destination 2.first && y2 == destination 2.second) {
50             printf ("%d\n", moves);
51             return;
52         }
53
54         // down
55         if (validPosition (x1+1, y1, x2+1, y2)) {
56             if (board 1[x1+1][y1] == '#' && board 2[x2+1][y2] != '#') {
57                 if (visited[x1][y1][x2+1][y2] == false) {
58                     q.push ({{{x1, y1}, {x2+1, y2}}, moves+1});
59                     visited[x1][y1][x2+1][y2] = true;
60                 }
61             }
62
63             if (board 1[x1+1][y1] != '#' && board 2[x2+1][y2] == '#') {
64

```

```
70         if (visited[x1+1][y1][x2][y2] == false) {
71             q.push ({{{x1+1, y1}, {x2, y2}}, moves+1});
72             visited[x1+1][y1][x2][y2] = true;
73         }
74     }
75 }
76
77 if (board 1[x1+1][y1] != '#' && board 2[x2+1][y2] != '#') {
78     if (visited[x1+1][y1][x2+1][y2] == false) {
79         q.push ({{{x1+1, y1}, {x2+1, y2}}, moves+1});
80         visited[x1+1][y1][x2+1][y2] = true;
81     }
82 }
83 }
84 }
85 }
86
87 // up
88 if (validPosition (x1-1, y1, x2-1, y2)) {
89     if (board 1[x1-1][y1] == '#' && board 2[x2-1][y2] != '#') {
90         if (visited[x1][y1][x2-1][y2] == false) {
91             q.push ({{{x1, y1}, {x2-1, y2}}, moves+1});
92             visited[x1][y1][x2-1][y2] = true;
93         }
94     }
95 }
96
97 if (board 1[x1-1][y1] != '#' && board 2[x2-1][y2] == '#') {
98     if (visited[x1-1][y1][x2][y2] == false) {
99         q.push ({{{x1-1, y1}, {x2, y2}}, moves+1});
100         visited[x1-1][y1][x2][y2] = true;
101     }
102 }
103
104 if (board 1[x1-1][y1] != '#' && board 2[x2-1][y2] != '#') {
105     if (visited[x1-1][y1][x2-1][y2] == false) {
106         q.push ({{{x1-1, y1}, {x2-1, y2}}, moves+1});
107         visited[x1-1][y1][x2-1][y2] = true;
108     }
109 }
110
111 // left
112 if (validPosition (x1, y1-1, x2, y2-1)) {
113     if (board 1[x1][y1-1] == '#' && board 2[x2][y2-1] != '#') {
114         if (visited[x1][y1][x2][y2-1] == false) {
115             q.push ({{{x1, y1}, {x2, y2-1}}, moves+1});
116             visited[x1][y1][x2][y2-1] = true;
117         }
118     }
119 }
120
121 if (board 1[x1][y1-1] != '#' && board 2[x2][y2-1] == '#') {
122     if (visited[x1][y1-1][x2][y2] == false) {
123         q.push ({{{x1, y1-1}, {x2, y2}}, moves+1});
124         visited[x1][y1-1][x2][y2] = true;
125     }
126 }
127
128 if (board 1[x1][y1-1] != '#' && board 2[x2][y2-1] != '#') {
129     if (visited[x1][y1-1][x2][y2-1] == false) {
130         q.push ({{{x1, y1-1}, {x2, y2-1}}, moves+1});
131         visited[x1][y1-1][x2][y2-1] = true;
132     }
133 }
134
135 if (board 1[x1][y1-1] != '#' && board 2[x2][y2-1] != '#') {
136     if (visited[x1][y1-1][x2][y2-1] == false) {
137         q.push ({{{x1, y1-1}, {x2, y2-1}}, moves+1});
138         visited[x1][y1-1][x2][y2-1] = true;
139     }
140 }
```

```
141         if (visited[x1][y1-1][x2][y2-1] == false) {
142             q.push ({{{x1, y1-1}, {x2, y2-1}}, moves+1});
143             visited[x1][y1-1][x2][y2-1] = true;
144         }
145     }
146 }
147
148 // right
149 if (validPosition (x1, y1+1, x2, y2+1)) {
150     if (board 1[x1][y1+1] == '#' && board 2[x2][y2+1] != '#') {
151         if (visited[x1][y1][x2][y2+1] == false) {
152             q.push ({{{x1, y1}, {x2, y2+1}}, moves+1});
153             visited[x1][y1][x2][y2+1] = true;
154         }
155     }
156     if (board 1[x1][y1+1] != '#' && board 2[x2][y2+1] == '#') {
157         if (visited[x1][y1+1][x2][y2] == false) {
158             q.push ({{{x1, y1+1}, {x2, y2}}, moves+1});
159             visited[x1][y1+1][x2][y2] = true;
160         }
161     }
162     if (board 1[x1][y1+1] != '#' && board 2[x2][y2+1] != '#') {
163         if (visited[x1][y1+1][x2][y2+1] == false) {
164             q.push ({{{x1, y1+1}, {x2, y2+1}}, moves+1});
165             visited[x1][y1+1][x2][y2+1] = true;
166         }
167     }
168 }
169 }
170 }
171 }
172 }
173 }
174 }
175 }
176 }
177 }
178 }
179 }
180 }
181
182 main () {
183     int t;
184     scanf ("%d", &t);
185     while (t--) {
186         scanf ("%d %d", &n, &m);
187         pii source 1, destination 1;
188         for (int i = 0; i < n; ++i) {
189             getchar ();
190             for (int j = 0; j < m; ++j) {
191                 scanf ("%c", &board 1[i][j]);
192                 if (board 1[i][j] == 'R')
193                     source 1 = {i, j};
194                 else if (board 1[i][j] == 'F')
195                     destination 1 = {i, j};
196             }
197         }
198         pii source 2, destination 2;
199         for (int i = 0; i < n; ++i) {
200             getchar ();
201             for (int j = 0; j < m; ++j) {
202                 scanf ("%c", &board 2[i][j]);
203                 if (board 2[i][j] == 'R')
204                     source 2 = {i, j};
205                 else if (board 2[i][j] == 'F')
206                     destination 2 = {i, j};
207             }
208         }
209         bfs (source 1, destination 1, source 2, destination 2);
210     }
211 }
```

```
212         scanf ("%c", &board 2[i][j]);
213
214         if (board 2[i][j] == 'R')
215             source 2 = {i, j};
216
217         else if (board 2[i][j] == 'F')
218             destination 2 = {i, j};
219     }
220 }
221
222 bfs (source 1, source 2, destination 1, destination 2);
223 }
224 }
225 }
```

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  #define MAX 1003
4
5  int n;
6
7  bool g [MAX][MAX];
8  char visited [MAX];
9
10 bool bfs (int source) {
11     queue< pair<int, int> > q;
12     memset (visited, 0, sizeof visited);
13
14     visited[source] = 1;
15     q.push (make pair (source, 2));
16
17     while (!q.empty ()) {
18         int i = q.front ().first;
19         int color = q.front ().second;
20         q.pop ();
21
22         for (int j = 0; j < n; ++j) {
23             if (g[i][j] == false) {
24                 if (visited[j] == 0)
25                     q.push (make pair (j, (color == 1) ? 2 : 1)), visited[j] = color;
26                 else if (visited[j] != color)
27                     return false;
28             }
29         }
30     }
31     return true;
32 }
33
34 main () {
35     scanf ("%d", &n);
36
37     for (int i = 0; i < n; ++i)
38         for (int j = 0; j < n; ++j)
39             scanf ("%d", &g[i][j]);
40
41     puts (bfs (0) ? "Bazinga!" : "Fail!");
42 }
43
44
45
46
47
48
49
50
```



```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  vector<int> g [10005];
5  int visited [10005];
6  bool loop;
7
8  void dfs (int i) {
9
10     visited[i] = 1;
11
12     for (int j = 0; loop == false && j < g[i].size (); ++j) {
13
14         if (visited[g[i][j]] == 0)
15             dfs (g[i][j]);
16
17         else if (visited[g[i][j]] == 1)
18             loop = true;
19     }
20
21     visited[i] = 2;
22 }
23
24 main () {
25
26     ios base::sync with stdio (0);
27     cin.tie (0);
28
29     int i, x, y, t, n, m;
30
31     cin >> t;
32     while (t--) {
33
34         cin >> n >> m;
35
36         for (i = 1; i <= n; ++i)
37             g[i].clear ();
38
39         while (m--) {
40
41             cin >> x >> y;
42             g[x].emplace back (y);
43         }
44
45         loop = false;
46         memset (visited, 0, sizeof (visited));
47
48         for (i = 1; loop == false && i <= n; ++i)
49             if (visited[i] != 1)
50                 dfs (i);
51
52         puts (loop ? "SIM" : "NAO");
53     }
54 }
55
```

```
1  #include <bits/stdc++.h>
2  #define eoq cout << "eoq" << endl
3  #define F first
4  #define S second
5  using namespace std;
6  typedef long long int ll;
7  typedef long long unsigned int llu;
8  typedef pair<int, int> pii;
9  const int N = 1e4 + 1;
10
11 ll n, vertex, weight;
12
13 vector<ll> g[N];
14 char visited[N];
15
16 void reset () {
17     for (ll i = 1; i <= n; ++i)
18         g[i].clear ();
19 }
20
21 void read () {
22     for (ll i = 2; i <= n; ++i) {
23         ll j;
24         scanf ("%lld", &j);
25         g[i].push back (j);
26         g[j].push back (i);
27     }
28 }
29
30 void bfs (int i, int wt) {
31     memset (visited, false, sizeof visited);
32     visited[i] = true;
33     queue<pii> q;
34     q.push (make pair (i, wt));
35     while (!q.empty ()) {
36         i = q.front ().F;
37         wt = q.front ().S;
38         q.pop ();
39         for (int j = 0; j < g[i].size (); ++j) {
40             if (visited[g[i][j]] == false) {
41                 visited[g[i][j]] = true;
42                 q.push (make pair (g[i][j], wt+1));
43             }
44         }
45         vertex = i;
46         weight = wt;
47     }
48 }
49
50 void solve () {
51     vertex = weight = 0;
52     bfs (1, 0);
53     bfs (vertex, 0);
54     printf ("%lld\n", weight/2 + (weight & 1));
55 }
56
57 main () {
58     while (scanf ("%lld", &n), n != -1) {
```

```
72
73     reset ();
74     read ();
75     solve ();
76   }
77 }
78
```

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  #define pii pair<int,int>
4
5  int cost[26][26];
6  int minimum;
7  char visited[26];
8
9  int bfs (int source, int destination) {
10
11     priority queue <pii, vector<pii>, greater<pii> > pq;
12     memset (visited, false, sizeof visited);
13
14     pq.push (make pair (0, source));
15
16     while (!pq.empty ()) {
17
18         int qtd = pq.top ().first;
19         int i = pq.top ().second;
20
21         if (i == destination || qtd >= minimum)
22             return qtd;
23
24         pq.pop ();
25         visited[i] = true;
26         // mark visited after removing from queue
27
28         for (int j = 0; j < 26; ++j)
29             if (visited[j] == false)
30                 pq.push (make pair (qtd+cost[i][j], j));
31     }
32 }
33
34 main () {
35
36     ios base::sync with stdio (0);
37     cin.tie (0);
38
39     for (int i = 0; i < 26; ++i)
40         for (int j = 0; j < 26; ++j)
41             cin >> cost[i][j];
42
43     string s;
44     cin >> s;
45
46     int idx1 = 0;
47     int idx2 = s.length () - 1;
48     int answer = 0;
49
50     while (idx1 < idx2) {
51
52         if (s[idx1] != s[idx2]) {
53
54             minimum = 1e6;
55
56             for (int i = 0; i < 26; ++i)
57                 minimum = min (minimum, bfs (s[idx1] - 'a', i) + bfs (s[idx2] -
58                     'a', i));
59
60             answer += minimum;
61         }
62
63         ++idx1;
64         --idx2;
65     }
66
67     cout << answer << '\n';
68 }
```

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  typedef long long int ll;
4
5  char g[1025][1025];
6  int n, m, answer;
7
8  void dfs (int i, int j) {
9
10     ++g[i][j];
11
12     if (i > 0 && g[i-1][j] == '.')
13         dfs (i-1, j);
14
15     if (i < n-1 && g[i+1][j] == '.')
16         dfs (i+1, j);
17
18     if (j > 0 && g[i][j-1] == '.')
19         dfs (i, j-1);
20
21     if (j < m-1 && g[i][j+1] == '.')
22         dfs (i, j+1);
23 };
24
25 main () {
26
27     ios base::sync with stdio (0);
28     cin.tie (0);
29
30     cin >> n >> m;
31
32     for (int i = 0; i < n; ++i)
33         cin >> g[i];
34
35     answer = 0;
36     for (int i = 0; i < n; ++i) {
37         for (int j = 0; j < m; ++j) {
38             if (g[i][j] == '.') {
39                 ++answer;
40                 dfs (i, j);
41             }
42         }
43     }
44
45     cout << answer << endl;
46 }
47
48
49
50
```

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  typedef long long unsigned int llu;
4
5  llu cost[26][26];
6
7  void floydWarshall () {
8
9      for (llu k = 0; k < 26; k++)
10         for (llu i = 0; i < 26; i++)
11             for (llu j = 0; j < 26; j++)
12                 cost[i][j] = min (cost[i][j], cost[i][k] + cost[k][j]);
13 }
14
15 main () {
16
17     ios base::sync with stdio (0);
18     cin.tie (0);
19
20     for (llu i = 0; i < 26; ++i)
21         for (llu j = 0; j < 26; ++j)
22             cin >> cost[i][j];
23
24     for (llu i = 0; i < 26; ++i)
25         cost[i][i] = 0;
26
27     string s;
28     cin >> s;
29
30     llu idx1 = 0;
31     llu idx2 = s.length () - 1;
32     llu answer = 0;
33
34     floydWarshall ();
35
36     while (idx1 < idx2) {
37         if (s[idx1] != s[idx2]) {
38             llu minimum = 1e15;
39
40             for (llu i = 0; i < 26; ++i)
41                 minimum = min (minimum, cost[s[idx1] - 'a'][i] + cost[s[idx2] -
42                     'a'][i]);
43
44             answer += minimum;
45         }
46
47         ++idx1;
48         --idx2;
49     }
50
51     cout << answer << '\n';
52 }
53
54
```

```

1  #include <bits/stdc++.h>
2  #define pb push back
3  #define mp make pair
4  #define F first
5  #define S second
6  #define eoq printf("eoq\n")
7  #define digitCountDec(a) (int)floor(1 + log10((double)a))
8  #define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
9  using namespace std;
10 typedef unsigned long long int llu;
11 typedef long long int ll;
12 typedef vector<ll> vi;
13 typedef pair<ll, ll> ii;
14 typedef vector< ii > vii;
15 int dr[] = {0, 1, -1, 0, 1, -1, -1, 1};
16 int dc[] = {-1, 0, 0, 1, 1, 1, -1, -1};
17
18 ll gcd (ll x, ll y) {
19     return (y != 0 ? gcd (y, x%y) : x);
20 }
21
22 ll lcm (ll x, ll y) {
23     return (x / gcd (x, y) * y);
24 }
25
26 const ll N = 100;
27 map<string, ll> string to num;
28 string num to string[N];
29 vector<ll> g1[N], g2[N];
30 char visited[N];
31 stack<ll> s;
32 ll n, m, c = 1;
33 bool flag, endlime = false;
34
35 void reset () {
36     for (ll i = 0; i <= n; ++i)
37         g1[i].clear (), g2[i].clear ();
38     string to num.clear ();
39 }
40
41 void read () {
42     ll x, y, total = 0;
43     string a, b;
44     while (m--) {
45         cin >> a >> b;
46         if (!string to num.count (a))
47             string to num[a] = total, num to string[total] = a, total++;
48         if (!string to num.count (b))
49             string to num[b] = total, num to string[total] = b, total++;
50         x = string to num[a];
51         y = string to num[b];
52         g1[x].pb (y);
53         g2[y].pb (x);
54     }
55 }
56
57 void dfs1 (ll i) {
58     visited[i] = true;
59 }

```

```
72     for (ll j = 0; j < g1[i].size (); ++j)
73         if (!visited[g1[i][j]])
74             dfs1 (g1[i][j]);
75
76     s.push (i);
77 }
78
79 void dfs2 (ll i) {
80
81     if (flag == true)
82         printf (", ");
83
84     else
85         flag = true;
86
87     cout << num to string[i];
88
89     visited[i] = true;
90     for (ll j = 0; j < g2[i].size (); ++j)
91         if (!visited[g2[i][j]])
92             dfs2 (g2[i][j]);
93 }
94
95 void solve () {
96
97     memset (visited, false, sizeof visited);
98     for (ll i = 0; i < n; ++i)
99         if (!visited[i])
100             dfs1 (i);
101
102     ll current = 0LL; flag = false;
103     memset (visited, false, sizeof visited);
104
105     if (getline)
106         printf ("\n");
107
108     else
109         endl = true;
110
111     printf ("Calling circles for data set %lld:\n", c++);
112
113     while (!s.empty ()) {
114
115         ll i = s.top (); s.pop ();
116         if (!visited[i]) {
117
118             flag = false;
119             dfs2 (i);
120             printf ("\n");
121         }
122     }
123 }
124
125 main () {
126
127     while (scanf ("%lld %lld", &n, &m), n and m) {
128
129         reset ();
130         read ();
131         solve ();
132     }
133 }
134
```



```
1  #include <bits/stdc++.h>
2  using namespace std;
3  typedef long long int ll;
4
5  const ll N = 1e5 + 10;
6  vector< pair< ll, pair<ll, ll> > > g;
7  ll parent[N];
8  ll n, m, total;
9
10 void reset () {
11     for (int i = 0; i <= n; ++i)
12         parent[i] = i;
13
14     g.clear ();
15     total = 0;
16 }
17
18 ll findset (ll x) {
19     if (x != parent[x])
20         parent[x] = findset (parent[x]);
21
22     return parent[x];
23 }
24
25 void UNION (ll x, ll y) {
26     parent[x] = parent[y];
27 }
28
29 void solve () {
30     ll pu, pv, total = 0;
31     sort (g.begin (), g.end ());
32     for (ll i = 0; i < g.size (); i++) {
33         pu = findset (g[i].second.first);
34         pv = findset (g[i].second.second);
35
36         if (pu != pv) {
37             total += g[i].first;
38             UNION (pu, pv);
39         }
40     }
41     printf ("%lld\n", total);
42 }
43
44 void read () {
45     while (m--) {
46         ll x, y, w; scanf ("%lld %lld %lld", &x, &y, &w);
47         g.push back (make pair (w, make pair (x, y)));
48     }
49 }
50
51 main () {
52     scanf ("%lld %lld", &n, &m);
53     reset ();
54     read ();
55     solve ();
56 }
```

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  string str;
5  void convert (int n, int base) {
6
7      if (n >= base)
8          convert ((n / base), base);
9
10     char c = n % base;
11     if (c > 9)
12         str += (c + 'a' - 10);
13
14     else
15         str += (c + '0');
16 }
17
18 main () {
19
20     int t, c = 1;
21     string n, b;
22
23     cin >> t;
24     while (t--) {
25
26         cin >> n >> b;
27
28         cout << "Case " << c++ << ":\n";
29
30         if (b == "bin") {
31
32             str.clear ();
33             convert (stoi (n, nullptr, 2), 10);
34             cout << str << " dec\n";
35
36             str.clear ();
37             convert (stoi (n, nullptr, 2), 16);
38             cout << str << " hex\n";
39         }
40
41         else if (b == "dec") {
42
43             str.clear ();
44             convert (stoi (n, nullptr, 10), 16);
45             cout << str << " hex\n";
46
47             str.clear ();
48             convert (stoi (n, nullptr, 10), 2);
49             cout << str << " bin\n";
50         }
51
52         else if (b == "hex") {
53
54             str.clear ();
55             convert (stoi (n, nullptr, 16), 10);
56             cout << str << " dec\n";
57
58             str.clear ();
59             convert (stoi (n, nullptr, 16), 2);
60             cout << str << " bin\n";
61         }
62
63         cout << '\n';
64     }
65 }
66
```

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  typedef long long int ll;
4
5  ll convert (ll n) {
6
7      if (n == 0)
8          return 1;
9
10     if (n == 1)
11         return 1;
12
13     if (n == 2)
14         return 2;
15
16     if (n == 3)
17         return 6;
18
19     if (n == 4)
20         return 4;
21
22     if (n == 5)
23         return 2;
24
25     if (n == 6)
26         return 2;
27
28     if (n == 7)
29         return 4;
30
31     if (n == 8)
32         return 2;
33
34     if (n == 9)
35         return 8;
36
37     if (n == 10)
38         return 8;
39
40     if ((n/10) % 2 == 0)
41         return 6 * convert (n/5) * convert (n%10);
42
43     else
44         return 4 * convert (n/5) * convert (n%10);
45 }
46
47 main () {
48
49     ll n;
50     int c = 1;
51
52     while (scanf ("%lld", &n) != EOF)
53         printf ("Instancia %d\n%lld\n\n", c++, convert (n) % 10);
54 }
55
```

```
1  #include <bits/stdc++.h>
2  #define pb push back
3  #define mp make pair
4  #define F first
5  #define S second
6  #define oioi printf("oioi\n")
7  #define digitCountDec(a) (int)floor(1 + log10((double)a))
8  #define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
9  using namespace std;
10 typedef unsigned long long int llu;
11 typedef long long int ll;
12 typedef vector<ll> vi;
13 typedef pair<ll, ll> ii;
14 typedef vector< ii > vii;
15 int dr[] = {0, 1, -1, 0, 1, -1, -1, 1};
16 int dc[] = {-1, 0, 0, 1, 1, 1, -1, -1};
17
18 ll gcd (ll x, ll y) {
19     return (y != 0 ? gcd (y, x%y) : x);
20 }
21
22 ll lcm (ll x, ll y) {
23     return (x / gcd (x, y) * y);
24 }
25
26 main () {
27
28     ll n; scanf ("%lld", &n);
29
30     if (n == 1 or n == 2)
31         printf ("-1\n");
32
33     else if (n & 1)
34         printf ("%lld %lld\n", (n*n-1) >> 1, (n*n+1) >> 1);
35
36     else
37         printf ("%lld %lld\n", (n >> 1) * (n >> 1) - 1, (n >> 1) * (n >> 1) + 1);
38 }
39
40
```

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  typedef pair<int,int> ii;
4  typedef vector<ii> perm_t;
5
6  #define PERM_SZ 21
7  int cw[PERM_SZ] = { 13,8,3,18,14,9,4,0,19,15,10,5,1,20,16,11,6,2,17,12,7 }; //
8  int ccw[PERM_SZ] = { 7,12,17,2,6,11,16,20,1,5,10,15,19,0,4,9,14,18,3,8,13 }; //
9  // permutable faces with its affected values
10 int f[PERM_SZ] = { 16,17,18,3,19,20,21,37,6,22,23,24,40,9,25,26,27,43,28,29,30 };
11 int r[PERM_SZ] = { 18,15,12,21,37,38,39,46,24,40,41,42,49,27,43,44,45,52,30,33,36 };
12 int l[PERM_SZ] = { 10,13,16,48,1,2,3,19,51,4,5,6,22,54,7,8,9,25,34,31,28 };
13 int b[PERM_SZ] = { 12,11,10,39,46,47,48,1,42,49,50,51,4,45,52,53,54,7,36,35,34 };
14 int u[PERM_SZ] = { 48,47,46,1,10,11,12,39,2,13,14,15,38,3,16,17,18,37,19,20,21 };
15 int d[PERM_SZ] = { 25,26,27,9,28,29,30,43,8,31,32,33,44,7,34,35,36,45,54,53,52 };
16 // do-able moves (aka permutations) for the rubik cube
17 map<char,perm_t> moves;
18
19 perm_t makePerm (int *values, int *order) {
20     perm_t ret (PERM_SZ);
21
22     for (size_t i = 0; i < PERM_SZ; ++i)
23         ret[i].first = values[i], ret[i].second = values[ order[i] ];
24
25     return ret;
26 }
27
28 vector<int> getCubePerm (const char *seq) {
29     vector<int> ret(55), tmp(55);
30
31     for (size_t i = 0; i < ret.size (); ++i)
32         ret[i] = i;
33
34     for (size_t len = strlen (seq), i = 0; i < len; ++i) {
35         const perm_t& p = moves[seq[i]];
36
37         for (size_t j = 0; j < ret.size (); ++j)
38             tmp[j] = ret[j]; // back up
39
40         for (size_t j = 0; j < p.size (); ++j)
41             ret[ p[j].first ] = tmp[ p[j].second ]; // then permute
42     }
43
44     return ret;
45 }
46
47 int gcd (int a, int b) {
48     int r;
49
50     while (b > 0)
51         r = a%b, a = b, b = r;
52
53     return a;
54 }
55
56 inline int lcm (int a, int b) {
57     return ((a / gcd(a,b)) * b);
58 }
59
60 int gcdOfCyclesLen (const vector<int> &p) {
61     int ret = 1, cnt = 0;
62     vector<bool> used (0, p.size ());
63     size_t i, j;

```

```
70
71     for (i = 1; i < p.size (); ++i) {
72
73         if (!used[i]) {
74
75             used[i] = true, cnt++, j = p[i];
76
77             while (j != i)
78                 used[j] = true, cnt++, j = p[j];
79
80             ret = lcm (cnt, ret), cnt = 0;
81         }
82     }
83
84     return ret;
85 }
86
87 main () {
88
89     moves['U'] = makePerm(u,cw), moves['u'] = makePerm(u,ccw);
90     moves['F'] = makePerm(f,cw), moves['f'] = makePerm(f,ccw);
91     moves['D'] = makePerm(d,cw), moves['d'] = makePerm(d,ccw);
92     moves['R'] = makePerm(r,cw), moves['r'] = makePerm(r,ccw);
93     moves['L'] = makePerm(l,cw), moves['l'] = makePerm(l,ccw);
94     moves['B'] = makePerm(b,cw), moves['b'] = makePerm(b,ccw);
95
96     char input[85];
97     while (gets(input) != NULL)
98         printf ("%d\n", gcdOfCyclesLen (getCubePerm (input)));
99 }
100
```

```
1  #include <bits/stdc++.h>
2
3  typedef unsigned long long int lli;
4
5  lli i, j, k, lim = 20000000;
6
7  std::vector<bool> isprime (20000000, true);
8  std::vector<bool> v;
9
10 void primes () {
11     for (i = 2; i < lim; ++i)
12         if (isprime[i])
13             for (j = i*i; j < lim; j += i)
14                 isprime[j] = false;
15 }
16
17
18 main () {
19     primes ();
20
21     lli t, n, elem, i;
22
23     scanf ("%llu", &t);
24
25     for (; t; --t) {
26         scanf ("%llu", &n);
27
28         v = isprime;
29
30         for (i = 0; n; --n) {
31             scanf ("%llu", &elem);
32
33             if (elem < lim)
34                 v[elem] = false;
35         }
36
37         for (i = 1; i < lim; ++i)
38             if (v[i])
39                 break;
40
41         printf ("%llu\n", i-1);
42     }
43 }
44
45
46
47
```

```

1  #include <bits/stdc++.h>
2  #define pb push back
3  #define mp make pair
4  #define F first
5  #define S second
6  #define oioi printf("oioi\n")
7  #define digitCountDec(a) (int)floor(1 + log10((double)a))
8  #define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
9  using namespace std;
10 typedef unsigned long long int llu;
11 typedef long long int ll;
12 typedef vector<ll> vi;
13 typedef pair<ll, ll> ii;
14 typedef vector<ii> vii;
15 int dr[] = {0, 1, -1, 0, 1, -1, -1, 1};
16 int dc[] = {-1, 0, 0, 1, 1, 1, -1, -1};
17 ll gcd (ll x, ll y) {
18
19     return (y != 0 ? gcd (y, x%y) : x);
20 }
21
22 ll lcm (ll x, ll y) {
23
24     return (x / gcd (x, y) * y);
25 }
26
27 ll i, j;
28
29 ll end (ll n, ll i) {
30
31     if (i == (n >> 1) + (n & 1))
32         return n*n;
33
34     ll a1 = (n << 2) - 4;
35     ll r = -8;
36     ll ai = a1 + (i-1) * r;
37
38     return ((a1 + ai) * i) >> 1;
39 }
40
41 ll begin (ll n, ll i) {
42
43     if (i == 1)
44         return 1;
45
46     return end (n, i-1) + 1;
47 }
48
49 ll go right (ll n, ll layer, ll pos, ll current) {
50
51     ll aux = current + (n - ((layer-1) << 1)) - 1;
52     j += min (aux, pos) - current;
53     return aux;
54 }
55
56 ll go down (ll n, ll layer, ll pos, ll current) {
57
58     ll aux = current + (n - ((layer-1) << 1)) - 1;
59     i += min (aux, pos) - current;
60     return aux;
61 }
62
63 ll go left (ll n, ll layer, ll pos, ll current) {
64
65     ll aux = current + (n - ((layer-1) << 1)) - 1;
66     j -= min (aux, pos) - current;
67     return aux;
68 }
69
70 ll go up (ll n, ll layer, ll pos, ll current) {
71

```



```
72     ll aux = current + (n - ((layer-1) << 1)) - 2;
73     i -= min (aux, pos) - current;
74     return aux;
75 }
76
77 void solve (ll n, ll layer, ll pos, ll idx begin) {
78     i = j = layer;
79     ll current = idx begin;
80     if (current < pos)
81         current = go right (n, layer, pos, current);
82
83     if (current < pos)
84         current = go down (n, layer, pos, current);
85
86     if (current < pos)
87         current = go left (n, layer, pos, current);
88
89     if (current < pos)
90         current = go up (n, layer, pos, current);
91
92     printf ("%lld %lld\n", i, j);
93 }
94
95 main () {
96     ll n, pos; scanf ("%lld %lld", &n, &pos);
97     ll lo = 1; // first layer
98     ll hi = (n >> 1) + (n & 1); // last layer
99
100     while (true) {
101         ll mid = (lo + hi) >> 1;
102         ll idx begin = begin (n, mid);
103         ll idx end = end (n, mid);
104
105         if (pos >= idx begin and pos <= idx end) {
106             solve (n, mid, pos, idx begin);
107             break;
108         }
109
110         if (pos > idx end)
111             lo = mid+1;
112
113         if (pos < idx begin)
114             hi = mid;
115     }
116 }
117
118 }
119
120 }
121
```

```

1  #include <bits/stdc++.h>
2  #define pb push back
3  #define mp make pair
4  #define F first
5  #define S second
6  #define oioi printf("oioi\n")
7  #define digitCountDec(a) (int)floor(1 + log10((double)a))
8  #define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
9  using namespace std;
10 typedef unsigned long long int llu;
11 typedef long long int ll;
12 typedef vector<ll> vi;
13 typedef pair<ll, ll> ii;
14 typedef pair<double, double> pdd;
15 typedef vector<ii> vii;
16 int dr[] = {0, 1, -1, 0, 1, -1, -1, 1};
17 int dc[] = {-1, 0, 0, 1, 1, 1, -1, -1};
18
19 ll gcd (ll x, ll y) {
20
21     return (y != 0 ? gcd (y, x%y) : x);
22 }
23
24 ll lcm (ll x, ll y) {
25
26     return (x / gcd (x, y) * y);
27 }
28
29 const ll N = 1e2 + 10;
30 ll dp[N][N][10], value[N], n;
31 char s[N];
32
33 ll solve (ll current, ll before, ll exponent) {
34
35     if (current == n)
36         return 0LL;
37
38     if (dp[current][before][exponent] != -1)
39         return dp[current][before][exponent];
40
41     ll ans = 0;
42     ll pt = value[s[current] - 'a'];
43
44     if (s[current] == s[before]) {
45
46         pt = min (pt*(ll)pow(2, exponent), pt * 512);
47         ans = max (solve (current+1, before, exponent), pt + solve (current+1,
48             current, min (exponent+1, 9LL)));
49
50     }
51     else
52         ans = max (solve (current+1, before, exponent), pt + solve (current+1,
53             current, 1));
54
55     return dp[current][before][exponent] = ans;
56 }
57
58 main () {
59
60     ll t; scanf ("%lld", &t);
61     while (t--) {
62
63         for (ll i = 0; i < 3; ++i)
64             scanf ("%lld", value + i);
65
66         scanf ("%s", s); n = strlen (s);
67         memset (dp, -1, sizeof dp);
68         printf ("%lld\n", solve (0, 0, 0));
69     }
70 }

```

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  typedef long long int ll;
4
5  int n, total;
6
7  int coin [30], T[10000001], R[10000001];
8
9  void coinChange () {
10
11     T[0] = 0;
12
13     for (int i = 1; i <= total; ++i) {
14
15         T[i] = INT MAX;
16         R[i] = -1;
17     }
18
19     for (int j = 0; j < n; ++j) {
20
21         for (int i = 1; i <= total; ++i) {
22
23             if (i >= coin[j]) {
24
25                 if (T[i - coin[j]] + 1 < T[i]) {
26
27                     T[i] = 1 + T[i - coin[j]];
28                     R[i] = j;
29                 }
30             }
31         }
32     }
33
34     cout << T[total] << '\n';
35 }
36
37 // discover used coins
38 void print coinChange () {
39
40     int i = total;
41
42     while (i != 0) {
43
44         int j = R[i];
45         cout << coin[j] << '\n';
46         i -= coin[j];
47     }
48 }
49
50 main () {
51
52     ios base::sync with stdio (0);
53     cin.tie (0);
54
55     int t;
56     cin >> t;
57     while (t--) {
58
59         cin >> n >> total;
60
61         for (int i = 0; i < n; ++i)
62             cin >> coin[i];
63
64         coinChange ();
65     }
66 }
67
```

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  typedef long long int ll;
4  #define MAX COIN 30
5  #define MAX TOTAL 1000006
6
7  int n;
8
9  int coin [MAX COIN], dp [MAX TOTAL];
10
11 int coinChange (int total, int idx) {
12     if (total == 0)
13         return 0;
14
15     if (idx == n || total < 0)
16         return MAX TOTAL;
17
18     if (dp[total] != -1)
19         return dp[total];
20
21     // coinChange with repetition
22     dp[total] = min (coinChange (total, idx+1), 1 + coinChange (total-coin[idx],
23     idx));
24
25     // coinChange 1-0
26     //dp[total] = min (coinChange (total, idx+1), min (1 + coinChange
27     (total-moeda[idx], idx+1), 1 + coinChange(total-moeda[idx], idx)));
28     return dp[total];
29 }
30
31 main () {
32     ios base::sync with stdio (0);
33     cin.tie (0);
34
35     int t;
36     cin >> t;
37     while (t--) {
38         int total;
39         cin >> n >> total;
40
41         for (int i = 0; i < n; ++i)
42             cin >> coin[i];
43
44         for (int i = 0; i <= total; ++i)
45             dp[i] = -1;
46
47         cout << coinChange (total, 0) << '\n';
48     }
49 }
50
51
52
```

```

1  #include <bits/stdc++.h>
2  #define pb push back
3  #define mp make pair
4  #define F first
5  #define S second
6  #define oioi printf("oioi\n")
7  #define digitCountDec(a) (int)floor(1 + log10((double)a))
8  #define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
9  using namespace std;
10 typedef unsigned long long int llu;
11 typedef long long int ll;
12 typedef vector<ll> vi;
13 typedef pair<ll, ll> ii;
14 typedef vector<ii> vii;
15 int dr[] = {0, 1, -1, 0, 1, -1, -1, 1};
16 int dc[] = {-1, 0, 0, 1, 1, 1, -1, -1};
17
18 ll gcd (ll x, ll y) {
19     return (y != 0 ? gcd (y, x%y) : x);
20 }
21
22 ll lcm (ll x, ll y) {
23     return (x / gcd (x, y) * y);
24 }
25
26 const ll N = 1e3 + 10;
27 ii v[N];
28 ll n, max elem, min elem;
29
30 ll calculate (ll line, ll i) {
31     ll ans = 1; ii before = v[i++];
32     for (; i < n; ++i) // search points on curve
33         if (v[i].F != before.F and v[i].S != before.S and abs (v[i].S - line) == 1) // new point on line
34             ++ans, before = v[i];
35     return ans;
36 }
37
38 ll get max (ll line) {
39     ll i;
40     for (i = 0; i < n and (abs (v[i].S - line) != 1); ++i); // search first point
41     if (i == n)
42         return 0;
43
44     ll ans = calculate (line, i++);
45     for (; i < n and v[i].F == v[i-1].F; ++i) // search all first points with same x
46         if (abs (v[i].S - line) == 1)
47             ans = max (ans, calculate (line, i));
48     return ans;
49 }
50
51 void solve () {
52     ll ans = 0;
53     for (ll line = min elem + 1; line < max elem; ++line)
54         ans = max (ans, get max (line));
55     cout << ans << '\n';
56 }
57
58 main () {
59     ios base::sync with stdio (0);
60     cin.tie (0);
61
62     while (cin >> n) {
63         min_elem = LLONG_MAX;

```

```
71         max elem = LLONG MIN;
72
73         for (ll i = 0; i < n; ++i) {
74
75             cin >> v[i].F >> v[i].S;
76             min elem = min (min elem, v[i].S);
77             max elem = max (max elem, v[i].S);
78         }
79
80         sort (v, v + n);
81         solve ();
82     }
83 }
84
```

```
1  #include <bits/stdc++.h>
2  #define eps 1e-8
3  #define maxn 1000100
4  #define mod 1000000LL
5  using namespace std;
6
7  typedef long long ll;
8
9  ll n, m, o;
10
11 struct mat {
12     ll v[2][2];
13
14     mat () {
15         memset(v, 0, sizeof v);
16     }
17
18     void init () {
19         v[0][0] = v[1][1] = 1;
20     }
21
22     mat operator * (mat other) {
23         mat res;
24         for (int i = 0; i < 2; ++i)
25             for (int j = 0; j < 2; ++j)
26                 for (int k = 0; k < 2; ++k)
27                     res.v[i][j] = (res.v[i][j] + v[i][k]*other.v[k][j]) % mod;
28
29         return res;
30     }
31 };
32
33 main () {
34     while (~scanf ("%lld %lld %lld", &n, &m, &o)) {
35         m %= mod;
36         o %= mod;
37         n /= 5;
38         mat ans, aux;
39         ans.init ();
40         aux.v[0][1] = 1;
41         aux.v[1][0] = o;
42         aux.v[1][1] = m;
43         while (n > 0) {
44             if (n & 1) ans = ans * aux;
45             aux = aux*aux;
46             n /= 2;
47         }
48
49         printf ("%06lld\n", (ans.v[0][1] * m + ans.v[0][0]) % mod);
50     }
51 }
```

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  typedef long long int ll;
4  #define W 50
5
6  int n;
7
8  int wt[1005], val[1005], dp[1005][W+1];
9
10 void knapSack () {
11     for (int i = 0; i <= n; i++) {
12         for (int j = 0; j <= W; j++) {
13             if (i==0 || j==0)
14                 dp[i][j] = 0;
15
16             // knapsack 0-1
17             else if (wt[i-1] <= j)
18                 dp[i][j] = max (dp[i-1][j], val[i-1] + dp[i-1][j-wt[i-1]]);
19
20             // knapsack with repetition
21             // dp[i][j] = max (dp[i-1][j], val[i-1] + dp[i][j-wt[i-1]]);
22
23             else
24                 dp[i][j] = dp[i-1][j];
25         }
26     }
27
28     // discover used val and wt
29     int total b = 0;
30     int total w = 0;
31     int j = W;
32
33     for (int i = n; i > 0; --i) {
34         if (dp[i][j] != dp[i-1][j]) {
35             total w += wt[i-1];
36             ++total b;
37             j -= wt[i-1];
38         }
39     }
40
41     printf ("%d brinquedos\n", dp[n][W]);
42     printf ("Peso: %d kg\n", total w);
43     printf ("sobra(m) %d pacote(s)\n\n", n - total b);
44 }
45
46 main () {
47     ios base::sync with stdio (0);
48     cin.tie (0);
49
50     int t;
51     cin >> t;
52     while (t--) {
53         cin >> n;
54         for (int i = 0; i < n; ++i)
55             cin >> val[i] >> wt[i];
56
57         knapSack ();
58     }
59 }
```



```

1  #include <bits/stdc++.h>
2  #define pb push back
3  #define mp make pair
4  #define F first
5  #define S second
6  #define oioi printf("oioi\n")
7  #define digitCountDec(a) (int)floor(1 + log10((double)a))
8  #define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
9  using namespace std;
10 typedef unsigned long long int llu;
11 typedef long long int ll;
12 typedef vector<ll> vi;
13 typedef pair<ll, ll> ii;
14 typedef vector<ii> vii;
15 int dr[] = {0, 1, -1, 0, 1, -1, -1, 1};
16 int dc[] = {-1, 0, 0, 1, 1, 1, -1, -1};
17
18 ll gcd (ll x, ll y) {
19     return (y != 0 ? gcd (y, x%y) : x);
20 }
21
22 ll lcm (ll x, ll y) {
23     return (x / gcd (x, y) * y);
24 }
25
26 const ll N = 1e5 + 10;
27 ll hist[N], n;
28
29 void read () {
30     scanf ("%lld", &n);
31     for (ll i = 0; i < n; ++i)
32         scanf ("%lld", &hist[i]);
33 }
34
35 void solve () {
36     stack<ll> s;
37     ll max area = 0;
38     ll tp;
39     ll area with top;
40
41     ll i = 0;
42     while (i < n) {
43         if (s.empty () or hist[s.top ()] <= hist[i])
44             s.push (i++);
45         else {
46             tp = s.top ();
47             s.pop ();
48             area with top = hist[tp] * (s.empty () ? i : i - s.top () - 1);
49             if (max area < area with top)
50                 max area = area with top;
51         }
52     }
53
54     while (!s.empty ()) {
55         tp = s.top ();
56         s.pop ();
57         area with top = hist[tp] * (s.empty () ? i : i - s.top () - 1);
58         if (max area < area with top)
59             max area = area with top;
60     }
61 }
62
63 while (!s.empty ()) {
64     tp = s.top ();
65     s.pop ();
66     area with top = hist[tp] * (s.empty () ? i : i - s.top () - 1);
67     if (max area < area with top)
68         max area = area with top;
69 }
70
71

```

```
72     printf ("%lld\n", max area);
73 }
74
75 main () {
76     while (true) {
77         read ();
78         if (!n)
79             break;
80         solve ();
81     }
82 }
83
84
85
86
87
```

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  typedef long long int ll;
4
5  int n, c, maxDiff;
6
7  main () {
8
9      ios base::sync with stdio (0);
10     cin.tie (0);
11
12     cin >> n >> c;
13
14     int prices [n], dp [n];
15
16     for (int i = 0; i < n; ++i)
17         cin >> prices[i];
18
19     dp[0] = 0;
20     maxDiff = -prices[0];
21
22     for (int i = 1; i <= n; ++i) {
23
24         dp[i] = max (dp[i - 1], prices[i-1] + maxDiff - c);
25         maxDiff = max (maxDiff, dp[i-1] - prices[i-1]);
26     }
27
28     cout << dp[n] << endl;
29 }
30
```

```

1  #include <bits/stdc++.h>
2  #define pb push back
3  #define mp make pair
4  #define F first
5  #define S second
6  #define eoq printf("eoq\n")
7  #define digitCountDec(a) (int)floor(1 + log10((double)a))
8  #define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
9  using namespace std;
10 typedef unsigned long long int llu;
11 typedef long long int ll;
12 typedef pair<ll, ll> ii;
13 typedef pair<double, double> dd;
14 typedef vector<ll> vi;
15 typedef vector<ii> vii;
16 int dr[] = { 0, 1, -1, 0, 1, -1, -1, 1};
17 int dc[] = {-1, 0, 0, 1, 1, 1, -1, -1};
18
19 ll gcd (ll x, ll y) {
20
21     return (y != 0 ? gcd (y, x%y) : x);
22 }
23
24 ll lcm (ll x, ll y) {
25
26     return (x / gcd (x, y) * y);
27 }
28
29 const ll N = 2 * 1e5 + 10;
30 ll dp[N][2], v[N];
31 ll n, c;
32
33 void read () {
34
35     scanf ("%lld %lld", &n, &c);
36
37     for (ll i = 0; i < n; ++i)
38         scanf ("%lld", v+i);
39
40     memset (dp, -1, sizeof dp);
41 }
42
43 ll solve (ll current, ll has) {
44
45     if (current == n)
46         return 0LL;
47
48     if (dp[current][has] != -1)
49         return dp[current][has];
50
51     if (has == 1)
52         return dp[current][has] = max (solve (current+1, 0) + v[current], solve
            (current+1, 1));
53
54     else
55         return dp[current][has] = max (solve (current+1, 1) - v[current] - c,
            solve (current+1, 0));
56 }
57
58 main () {
59
60     read ();
61     printf ("%lld\n", solve (0, 0));
62 }
63

```

```

1  #include <bits/stdc++.h>
2  #define pb push back
3  #define mp make pair
4  #define F first
5  #define S second
6  #define eoq printf("eoq\n")
7  #define digitCountDec(a) (int)floor(1 + log10((double)a))
8  #define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
9  using namespace std;
10 typedef unsigned long long int llu;
11 typedef long long int ll;
12 typedef pair<ll, ll> ii;
13 typedef pair<double, double> dd;
14 typedef vector<ll> vi;
15 typedef vector<ii> vii;
16 int dr[] = { 0, 1, -1, 0, 1, -1, -1, 1};
17 int dc[] = {-1, 0, 0, 1, 1, 1, -1, -1};
18
19 ll gcd (ll x, ll y) {
20
21     return (y != 0 ? gcd (y, x%y) : x);
22 }
23
24 ll lcm (ll x, ll y) {
25
26     return (x / gcd (x, y) * y);
27 }
28
29 const ll INF = 1e14 + 10;
30 const ll N = 1e2 + 10;
31 ll color[N], cost[N][N], dp[N][N][N];
32 ll n, m, k;
33
34 void read () {
35
36     scanf ("%lld %lld %lld", &n, &m, &k);
37
38     for (ll i = 0; i < n; i++)
39         scanf ("%lld", &color[i]);
40
41     for (ll i = 0; i < n; i++)
42         for (ll j = 1; j <= m; j++)
43             scanf ("%lld", &cost[i][j]);
44
45     memset (dp, -1, sizeof dp);
46 }
47
48 ll solve (ll current, ll before, ll beauty) {
49
50     if (beauty > k)
51         return INF;
52
53     if (current == n)
54         return (beauty == k) ? 0LL : INF;
55
56     if (dp[current][before][beauty] != -1)
57         return dp[current][before][beauty];
58
59     ll ans = INF;
60
61     if (color[current] == 0)
62         for (ll i = 1; i <= m; i++)
63             ans = min (ans, cost[current][i] + solve (current+1, i, (i == before)
64                 ? beauty : beauty+1));
65
66     else if (current == 0)
67         ans = solve (current+1, color[current], 1);
68
69     else
70         ans = solve (current+1, color[current], (color[current] == before) ?
71             beauty : beauty+1);

```

```
70
71     return dp[current][before][beauty] = ans;
72 }
73
74 int main () {
75     read ();
76     ll ans = solve (0, color[0], 0);
77     printf ("%lld\n", (ans == INF) ? -1 : ans);
78 }
79
80
```

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  typedef long long int ll;
4  const int N = 3 * 1e3 + 10;
5
6  int r, c, k;
7
8  char v[N][N];
9  int m[N][N];
10
11 void read () {
12     for (int i = 0; i < r; ++i) {
13         for (int j = 0; j < c; ++j) {
14             getchar ();
15             for (int j = 0; j < c; ++j)
16                 scanf ("%c", &v[i][j]);
17         }
18     }
19 }
20
21 void make () {
22     memset (m, 0, sizeof m);
23
24     for (int i = 0; i < r; ++i)
25         for (int j = 0; j < c; ++j)
26             m[i+1][j+1] = (v[i][j] == '.' ? 1 : 0) + m[i][j+1] + m[i+1][j] - m[i][j];
27 }
28
29 void solve () {
30     int ans = r*c;
31
32     for (int c1 = 0; c1 < c; ++c1) {
33         for (int c2 = c1; c2 < c; ++c2) {
34             for (int r1 = 0, r2 = 0; r2 < r; ++r2) {
35                 while (r1 < r2 && m[r2 + 1][c2 + 1] - m[r1 + 1][c2 + 1] - m[r2 + 1][c1] + m[r1 + 1][c1] >= k)
36                     ++r1;
37
38                 if (m[r2 + 1][c2 + 1] - m[r1][c2 + 1] - m[r2 + 1][c1] + m[r1][c1] >= k)
39                     ans = min (ans, (c2 - c1 + 1) * (r2 - r1 + 1));
40             }
41         }
42     }
43     printf ("%d\n", ans);
44 }
45
46 main () {
47     while (scanf ("%d %d %d", &r, &c, &k)) {
48         if (!r && !c && !k)
49             break;
50
51         read ();
52         make ();
53         solve ();
54     }
55 }
```

```

1  #include <bits/stdc++.h>
2  #define inf 0x3F3F3F3F
3  #define mp make pair
4  using namespace std;
5
6  int n, m, k;
7  char s[111][111];
8  int id[111][111];
9  int x[111], y[111];
10 int dist[20][20];
11 int vis[111][111];
12
13 int di[] = {-2, -2, -1, 1, 2, 2, 1, -1};
14 int dj[] = {1, -1, 2, 2, 1, -1, -2, -2};
15
16 void bfs(int ini) {
17     memset (dist[ini], inf, sizeof dist[ini]);
18     dist[ini][ini] = 0;
19
20     memset (vis, inf, sizeof vis);
21     vis[x[ini]][y[ini]] = 0;
22
23     queue<pair<int, int> > q;
24     q.push (mp(x[ini], y[ini]));
25
26     while (!q.empty ()) {
27         int i = q.front().first, j = q.front().second;
28         q.pop();
29
30         for(int k = 0; k < 8; ++k) {
31             int ii = i+di[k], jj = j+dj[k];
32
33             if (ii < 0 || jj < 0 || ii >= n || jj >= m)
34                 continue;
35
36             if(s[ii][jj] == '#')
37                 continue;
38
39             if(vis[ii][jj] != inf)
40                 continue;
41
42             vis[ii][jj] = vis[i][j]+1;
43             q.push(mp(ii, jj));
44
45             if (id[ii][jj] != -1)
46                 dist[ini][ id[ii][jj] ] = vis[ii][jj];
47         }
48     }
49 }
50
51 int dp[1<<15][16];
52 int solve (int mask, int pos) {
53     if (mask == (1<<k)-1)
54         return dist[pos][k];
55
56     int &p = dp[mask][pos];
57
58     if (p != -1) return p;
59     p = inf;
60
61     for(int i = 0; i < k; ++i) {
62         if (mask & (1<<i))
63             continue;
64
65         p = min (p, solve (mask|(1<<i), i) + dist[pos][i]);
66     }
67 }

```



```
72
73     return p;
74 }
75
76 main () {
77     while (1) {
78         int cnt = 0;
79         scanf ("%d %d %d", &n, &m, &k);
80         if (!n && !m && !k)
81             break;
82
83         memset (id, -1, sizeof id);
84         for (int i = 0; i < n; ++i) {
85             scanf ("%s", s[i]);
86             for (int j = 0; j < m; ++j) {
87                 if (s[i][j] == 'P') {
88                     x[cnt] = i;
89                     y[cnt] = j;
90                     id[i][j] = cnt++;
91                 }
92                 if (s[i][j] == 'C') {
93                     x[k] = i;
94                     y[k] = j;
95                     id[i][j] = k;
96                 }
97             }
98         }
99
100         for (int i = 0; i <= k; ++i)
101             bfs(i);
102
103         memset (dp, -1, sizeof dp);
104         printf ("%d\n", solve (0, k));
105     }
106 }
107
108
109
110
111
112
113
114
```

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  const int alph=256;
4  const int INF=1e9;
5  const int MPOW=16;
6  const int N=1<<MPOW-1;
7  const int N2=N<<1;
8  struct sg_tree {
9
10     int arr[N2];
11
12     void build(vector<int> x,int n) {
13
14         fill(arr,arr+N2,INF);
15
16         for(int i=0;i<n;i++)
17             arr[i+N] = x[i];
18
19         for(int i=N-1;i>0;i--)
20             arr[i] = min(arr[i<<1],arr[(i<<1)+1]);
21     }
22
23     int get_min(int c,int cl,int cr,int l,int r) {
24
25         if(l==cl && r==cr)
26             return arr[c];
27
28         if(l > r)
29             return INF;
30
31         int cm = cl + cr>>1;
32
33         return min
34             (get_min(c<<1,cl,cm,l,min(r,cm)),get_min((c<<1)+1,cm+1,cr,max(l,cm+1),r));
35     }
36
37     int get_min (int l,int r) {
38         return get_min (1, 0, N-1, l, r);
39     }
40 };
41
42 pair<vector<int>,vector<int>> compute(string &s)
43 {
44     int n=s.size();
45     int maxn=n+alph;
46
47     vector<int> p(n),c(n),cnt(maxn,0);
48     for(int i=0;i<n;i++)
49         cnt[s[i]]++;
50     for(int i=1;i<maxn;i++)
51         cnt[i]+=cnt[i-1];
52     for(int i=0;i<n;i++)
53         p[--cnt[s[i]]]=i;
54     int cl=0;
55     c[p[0]]=cl;
56     for(int i=1;i<n;i++)
57     {
58         if(s[p[i]]!=s[p[i-1]])cl++;
59         c[p[i]]=cl;
60     }
61     vector<int> lcp(n,0);
62     for(int i=1;i<n;i++)
63         lcp[i]=c[p[i]]==c[p[i-1]];
64     vector<int> pn(n),cn(n),lcpn(n);
65     vector<int> rpos(n),lpos(n);
66     sg_tree rmq;
67     int k=1;
68     while(k<n)
69     {
70         fill(begin(cnt),end(cnt),0);

```

```

71     for(int i=0;i<n;i++)
72         rpos[c[p[i]]]=i;
73     for(int i=n-1;i>=0;i--)
74         lpos[c[p[i]]]=i;
75     for(int i=0;i<n;i++)
76     {
77         pn[i]=p[i]-k;
78         if(pn[i]<0)pn[i]+=n;
79     }
80     for(int i=0;i<n;i++)
81         cnt[c[i]]++;
82     for(int i=1;i<maxn;i++)
83         cnt[i]+=cnt[i-1];
84     for(int i=n-1;i>=0;i--)
85         p[--cnt[c[pn[i]]]]=pn[i];
86     cl=0;
87     cn[p[0]]=0;
88     for(int i=1;i<n;i++)
89     {
90         int m1=(p[i]+k)%n,m2=(p[i-1]+k)%n;
91         if(c[p[i]]!=c[p[i-1]] || c[m1]!=c[m2])cl++;
92         cn[p[i]]=cl;
93     }
94     rmq.build(lcp,n);
95     for(int i=1;i<n;i++)
96     {
97         int a=p[i],b=p[i-1];
98         if(c[a]!=c[b])
99             lcpn[i]=lcp[lpos[c[a]]];
100         else
101         {
102             int aa=(a+k)%n,bb=(b+k)%n;
103             if(c[aa]==c[bb])
104                 lcpn[i]=k<<1;
105             else
106                 lcpn[i]=k+rmq.get_min(lpos[c[bb]]+1,rpos[c[aa]]);
107         }
108         lcpn[i]=min(n,lcpn[i]);
109     }
110     copy(begin(cn),end(cn),begin(c));
111     copy(begin(lcpn),end(lcpn),begin(lcp));
112     k<<=1;
113 }
114 return {p,lcp};
115 }
116
117 struct suffix tree
118 {
119     struct edge
120     {
121         int from;
122         int to;
123         int next vert;
124         int suffix here;
125     };
126
127     struct vertex
128     {
129         vector<edge> go;
130     };
131
132     string str;
133     vector<vertex> data;
134
135     static bool comp(const edge &b,const char &a)
136     {
137         return 1;
138     }
139
140     void build(string &s)

```

```

142     {
143         pair<vector<int>,vector<int>> info=compute(s);
144         vector<int> p=info.first,lcp=info.second;
145
146         int n=s.size();
147         str=s;
148
149         vector<int> p vert;
150         vector<int> p edge;
151         vector<int> p dist;
152
153
154         vertex v;
155         edge e;
156         e.from=p[0];
157         e.to=n;
158         e.next vert=-1;
159         e.suffix here=p[0];
160         v.go.push back(e);
161         data.push back(v);
162
163         p vert.push back(0);
164         p edge.push back(0);
165         p dist.push back(0);
166
167         for(int i=1;i<n;i++)
168         {
169             int c lcp=lcp[i];
170
171             while(p dist.back()>c lcp)
172             {
173                 edge &E=data[p vert.back()].go[p edge.back()];
174                 if(E.next vert+1)
175                 {
176                     int m=data[E.next vert].go.size();
177                     for(int j=0;j<m;j++)
178                         E.suffix here=min(E.suffix here,data[E.next vert].go[j].suffix
179
180                 }
181                 p vert.pop back();
182                 p edge.pop back();
183                 p dist.pop back();
184             }
185             vertex v;
186             edge e;
187             int c v=p vert.back();
188             int c e=p edge.back();
189             int At=data[c v].go[c e].from+c lcp-p dist.back();
190
191             p dist.push back(c lcp);
192
193             e.next vert=-1;
194             e.suffix here=p[i];
195             e.from=p[i]+c lcp;
196             e.to=n;
197
198             if(At==data[c v].go[c e].from)
199             {
200                 data[c v].go.push back(e);
201                 p vert.push back(c v);
202                 p edge.push back(data[c v].go.size()-1);
203             }
204             else
205             {
206                 v.go.push back(data[c v].go[c e]);
207                 v.go.back().from=At;
208                 v.go.push back(e);
209                 data.push back(v);
210                 data[c v].go[c e].next vert=data.size()-1;
211                 data[c_v].go[c_e].to=At;

```

```

211         p vert.push back(data.size()-1);
212         p edge.push back(1);
213     }
214 }
215
216 while(!p dist.empty())
217 {
218     edge &E=data[p vert.back()].go[p edge.back()];
219     if(E.next vert+1)
220     {
221         int m=data[E.next vert].go.size();
222         for(int j=0;j<m;j++)
223             E.suffix here=min(E.suffix here,data[E.next vert].go[j].suffix here);
224     }
225
226     p vert.pop back();
227     p edge.pop back();
228     p dist.pop back();
229 }
230 }
231
232 int search str(string &s)
233 {
234     int n=s.size();
235     int cur v=0;
236     int cur e;
237     char t;
238
239     for(int i=0;i<n;)
240     {
241         if(cur v==-1)break;
242         t=s[i];
243         int cur e;
244         for(cur e=0;cur e<data[cur v].go.size();cur e++)
245             if(str[data[cur v].go[cur e].from]>=t)break;
246         for(int j=data[cur v].go[cur e].from;i<n &&
247             j<data[cur v].go[cur e].to;j++,i++)
248             if(str[j]!=s[i])
249                 i=n+1;
250         if(i==n)
251             return data[cur v].go[cur e].suffix here;
252         cur v=data[cur v].go[cur e].next vert;
253     }
254     return -1;
255 }
256
257 int print(int x)
258 {
259     if(x==-1)return 0;
260     int ans=0;
261     for(int i=0;i<data[x].go.size();i++)
262         ans+=data[x].go[i].to-data[x].go[i].from+print(data[x].go[i].next vert)-1;
263     return ans;
264 }
265
266 int main() {
267     ios::sync with stdio(0);
268     cin.tie(0);
269     string a;
270     cin>>a;
271     a+='#';
272     suffix tree sf;
273     sf.build(a);
274     cout<<sf.print(0)<<endl;
275 }

```

```

1  #include <bits/stdc++.h>
2  #define pb push back
3  #define mp make pair
4  #define F first
5  #define S second
6  #define oioi printf("oioi\n")
7  #define digitCountDec(a) (int)floor(1 + log10((double)a))
8  #define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
9  using namespace std;
10 typedef unsigned long long int llu;
11 typedef long long int ll;
12 typedef vector<ll> vi;
13 typedef pair<ll, ll> ii;
14 typedef vector<ii> vii;
15 int dr[] = {0, 1, -1, 0, 1, -1, -1, 1};
16 int dc[] = {-1, 0, 0, 1, 1, 1, -1, -1};
17
18 ll gcd (ll x, ll y) {
19     return (y != 0 ? gcd (y, x%y) : x);
20 }
21
22 ll lcm (ll x, ll y) {
23     return (x / gcd (x, y) * y);
24 }
25
26 const ll N = 85;
27 char x[N], y[N];
28 ll len x, len y;
29 ll dp[N][N];
30 ll c = 0;
31
32 void print solution () {
33     // 1: insert, 2: delete, 3: replace
34     stack<ii> s; // first: command second: position
35
36     ll i = len x, j = len y;
37
38     while (true) {
39         if (!i and !j)
40             break;
41
42         else if (!i)
43             s.push (make pair (1LL, i)), --j; // insert
44
45         else if (!j)
46             s.push (make pair (2LL, i)), --i; // delete
47
48         else {
49             if (dp[i][j] == dp[i-1][j-1] + (x[i-1] != y[j-1])) {
50                 if (x[i-1] != y[j-1])
51                     s.push (make pair (3LL, i-1)); // replace
52                 --i, --j;
53             }
54
55             else if (dp[i][j] - 1 == dp[i][j-1])
56                 s.push (make pair (1LL, i)), --j; // insert
57
58             else
59                 s.push (make pair (2LL, i)), --i; // delete
60         }
61     }
62
63     ll idx = 1;
64     ll current = 0;

```

```
72     while (!s.empty ()) {
73
74         i = s.top ().F; j = s.top ().S;
75         s.pop ();
76
77         if (i == 1)
78             printf ("%lld Insert %lld,%c\n", idx++, j + current + 1, y[j +
79                                     current]), ++current;
80
81         if (i == 2)
82             printf ("%lld Delete %lld\n", idx++, j + current), --current; // %c,
83             x[j - 1];
84
85         if (i == 3)
86             printf ("%lld Replace %lld,%c\n", idx++, j + current + 1, y[j +
87                                     current]);
88     }
89 }
90
91 void solve () {
92     len x = strlen (x), len y = strlen (y);
93
94     for (ll i = 0; i <= len x; ++i) {
95         for (ll j = 0; j <= len y; ++j) {
96             if (!i)
97                 dp[i][j] = j; // empty (x) -> insert
98
99             else if (!j)
100                 dp[i][j] = i; // empty (y) -> delete
101
102             else
103                 dp[i][j] = min (dp[i-1][j-1] + (x[i-1] != y[j-1]), min
104                                 (dp[i][j-1], dp[i-1][j]) + 1);
105                 // horizontal (j-1) -----> insert
106                 // vertical (i-1) -----> delete
107                 // diagonal (i-1, j-1) --> replace or nothing
108         }
109     }
110
111     if (c++)
112         printf ("\n");
113
114     printf ("%lld\n", dp[len x][len y]);
115     print solution ();
116 }
117
118 main () {
119     while (gets (x) != NULL) {
120
121         gets (y);
122         solve ();
123     }
124 }
125
```

```

1  #include <bits/stdc++.h>
2  #define pb push back
3  #define mp make pair
4  #define F first
5  #define S second
6  #define oioi printf("oioi\n")
7  #define digitCountDec(a) (int)floor(1 + log10((double)a))
8  #define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
9  using namespace std;
10 typedef unsigned long long int llu;
11 typedef long long int ll;
12 typedef vector<ll> vi;
13 typedef pair<ll, ll> ii;
14 typedef vector<ii> vii;
15 int dr[] = {0, 1, -1, 0, 1, -1, -1, 1};
16 int dc[] = {-1, 0, 0, 1, 1, 1, -1, -1};
17
18 ll gcd (ll x, ll y) {
19     return (y != 0 ? gcd (y, x%y) : x);
20 }
21
22 ll lcm (ll x, ll y) {
23     return (x / gcd (x, y) * y);
24 }
25
26 const ll N = 1e3 + 10;
27 ll dp[N][N];
28 string x, y;
29 ll len x, len y;
30
31 void solve () {
32     for (ll i = 0; i <= x.length (); ++i) {
33         for (ll j = 0; j <= y.length (); ++j) {
34             if (!i)
35                 dp[i][j] = j;
36             else if (!j)
37                 dp[i][j] = i;
38             else
39                 dp[i][j] = min (dp[i-1][j-1] + (x[i-1] != y[j-1]), min
40                     (dp[i][j-1], dp[i-1][j]) + 1);
41             if (i > 1 and j > 1 and x[i-1] == y[j-2] and x[i-2] == y[j-1])
42                 dp[i][j] = min (dp[i][j], dp[i-2][j-2] + 1); // transposition
43         }
44     }
45     printf ("%lld\n", dp[x.length ()][y.length ()]);
46 }
47
48 main () {
49     while (cin >> x) {
50         cin >> y;
51         solve ();
52     }
53 }
54
55
56
57
58
59
60
61
62
63
64

```



```
1  #include <bits/stdc++.h>
2  using namespace std;
3  typedef long long int ll;
4
5  ll hamming distance (const char *x, const char *y) {
6
7      ll distance = 0;
8
9      for (ll i = 0; i < strlen (x); ++i)
10         if (x[i] != y[i])
11             distance++;
12
13     return distance;
14 }
15
16 const ll N = 1e5 + 10;
17
18 main() {
19
20     char x[N], y[N]; scanf ("%s", x);
21     ll n; scanf ("%lld", &n);
22     ll position = 1;
23     ll distance = N;
24
25     for (ll i = 1; i <= n; ++i) {
26
27         scanf ("%s", y);
28         ll d = hamming distance (x, y);
29
30         if (d < distance) {
31
32             position = i;
33             distance = d;
34         }
35     }
36
37     printf ("%lld\n", position);
38     printf ("%lld\n", distance);
39 }
40
```

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  typedef long long int ll;
4  typedef long long unsigned int llu;
5  const int N = 1e5 + 10;
6
7  llu hashString (char *str) { // base 33
8
9      llu hash = 5381;
10     int c;
11
12     while (c = *str++)
13         hash = ((hash << 5) + hash) + c;
14
15     return hash;
16 }
17
18 main () {
19
20     map<llu, llu> okay;
21     okay.clear ();
22
23     int n; scanf ("%d", &n);
24     while (n--) {
25
26         char s[N]; scanf ("%s", s);
27
28         llu hash = hashString (s);
29         if (okay.find (hash) == okay.end ()) {
30
31             printf ("OK\n");
32             okay[hash] = 1;
33         }
34
35         else {
36
37             printf ("%s%llu\n", s, okay[hash]);
38             okay[hash]++;
39         }
40     }
41 }
42
```

```
1  #include <bits/stdc++.h>
2  #define pb push back
3  #define mp make pair
4  #define F first
5  #define S second
6  #define oioi printf("oioi\n")
7  #define digitCountDec(a) (int)floor(1 + log10((double)a))
8  #define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
9  using namespace std;
10 typedef unsigned long long int llu;
11 typedef long long int ll;
12 typedef vector<ll> vi;
13 typedef pair<ll, ll> ii;
14 typedef vector<ii> vii;
15 int dr[] = {0, 1, -1, 0, 1, -1, -1, 1};
16 int dc[] = {-1, 0, 0, 1, 1, 1, -1, -1};
17
18 ll gcd (ll x, ll y) {
19     return (y != 0 ? gcd (y, x%y) : x);
20 }
21
22 ll lcm (ll x, ll y) {
23     return (x / gcd (x, y) * y);
24 }
25
26 const ll N = 100;
27 ll back table[N];
28 ll len x, len y, ans;
29 string x, y;
30
31 void make () {
32     ll i = 0, j = -1;
33     back table[0] = -1;
34     while (i < len y) {
35         while (j >= 0 and y[i] != y[j])
36             j = back table[j];
37         i++; j++;
38         back table[i] = j;
39     }
40 }
41
42 void kmp () {
43     ans = 1;
44     ll i = len y, j = 0; // start after end of len x
45     while (i < len x) {
46         while (j >= 0 and x[i] != y[j])
47             j = back table[j];
48         i++; j++;
49         if (j == len y) {
50             ans++;
51             j = back table[j];
52         }
53     }
54 }
55
56 main () {
57     ll c = 0;
58     ll t; scanf ("%lld", &t);
59     while (t--) {
```

```
72
73     cin >> x;
74     len x = x.length ();
75     len y = 0;
76
77     while (true) {
78
79         ++len y;
80         if (len x % len y)
81             continue;
82
83         y = x.substr (0, len y);
84         make ();
85         kmp ();
86
87         if (ans == len x / len y)
88             break;
89     }
90
91     if (c++)
92         printf ("\n");
93     printf ("%lld\n", len y);
94 }
95 }
96
```

```

1  #include <bits/stdc++.h>
2  #define pb push back
3  #define mp make pair
4  #define F first
5  #define S second
6  #define oioi printf("oioi\n")
7  #define digitCountDec(a) (int)floor(1 + log10((double)a))
8  #define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
9  using namespace std;
10 typedef unsigned long long int llu;
11 typedef long long int ll;
12 typedef vector<ll> vi;
13 typedef pair<ll, ll> ii;
14 typedef vector<ii> vii;
15 int dr[] = {0, 1, -1, 0, 1, -1, -1, 1};
16 int dc[] = {-1, 0, 0, 1, 1, 1, -1, -1};
17
18 ll gcd (ll x, ll y) {
19     return (y != 0 ? gcd (y, x%y) : x);
20 }
21
22 ll lcm (ll x, ll y) {
23     return (x / gcd (x, y) * y);
24 }
25
26 const ll N = 1e3 + 10;
27 ll n;
28 char x[N], y[N];
29 ll len x, len y;
30 ll dp[N][N];
31 void solve ();
32
33 main () {
34     while (scanf ("%lld", &n), n)
35         solve ();
36 }
37
38 void solve () {
39     scanf ("%s %s", x, y);
40     len x = strlen (x), len y = strlen (y);
41     for (ll i = 1; i <= len x; ++i) {
42         for (ll j = 1; j <= len y; ++j) {
43             ll ans = 0;
44             dp[i][j] = max (dp[i-1][j], dp[i][j-1]);
45             while (i-1-ans >= 0 and j-1-ans >= 0 and x[i-1-ans] == y[j-1-ans]) {
46                 ++ans;
47                 if (ans >= n)
48                     dp[i][j] = max (dp[i][j], dp[i-ans][j-ans] + ans);
49             }
50             if (ans >= n)
51                 dp[i][j] = max (dp[i][j], dp[i-ans][j-ans] + ans);
52         }
53     }
54     printf ("%lld\n", dp[len x][len y]);
55 }
56
57
58
59
60
61
62
63
64
65
66
67
68

```

```

1  #include <bits/stdc++.h>
2  #define pb push back
3  #define mp make pair
4  #define F first
5  #define S second
6  #define oioi printf("oioi\n")
7  #define eq cout << "eq" << endl
8  #define digitCountDec(a) (int)floor(1 + log10((double)a))
9  #define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
10 using namespace std;
11 typedef unsigned long long int llu;
12 typedef long long int ll;
13 typedef vector<ll> vi;
14 typedef pair<ll, ll> ii;
15 typedef vector<ii> vii;
16 int dr[] = {0, 1, -1, 0, 1, -1, -1, 1};
17 int dc[] = {-1, 0, 0, 1, 1, 1, -1, -1};
18
19 ll gcd (ll x, ll y) {
20
21     return (y != 0 ? gcd (y, x%y) : x);
22 }
23
24 ll lcm (ll x, ll y) {
25
26     return (x / gcd (x, y) * y);
27 }
28
29 const ll N = 3 * 1e3 + 10;
30 ll c = 1;
31 ll dp[N][N];
32 map<llu, string> word;
33
34 llu encrypt string (const char *str) { // base 33
35
36     llu hash = 5381;
37     llu c;
38
39     while (c = *str++)
40         hash = ((hash << 5) + hash) + c;
41
42     return hash;
43 }
44
45 vector<llu> read () {
46
47     string s;
48     vector<llu> v;
49     while (cin >> s and s != "#") {
50
51         llu hash = encrypt string (s.c str ());
52         word[hash] = s;
53         v.push back (hash);
54     }
55
56     return v;
57 }
58
59 void print dp (vector<llu> x, vector<llu> y, ll i, ll j) {
60
61     stack<string> s;
62     while (i and j) {
63
64         if (dp[i][j] == dp[i-1][j])
65             i--;
66
67         else if (dp[i][j] == dp[i][j-1])
68             j--;
69
70         else {
71

```

```
72         s.push (word[x[i-1]]);
73         i--;
74         j--;
75     }
76 }
77
78 if (!s.empty ()) {
79     cout << s.top ();
80     s.pop ();
81 }
82
83 while (!s.empty ()) {
84     cout << ' ' << s.top ();
85     s.pop ();
86 }
87 cout << '\n';
88 }
89
90 void solve (vector<llu> x, vector<llu> y) {
91     for (ll i = 0; i <= x.size (); ++i) {
92         for (ll j = 0; j <= y.size (); ++j) {
93             if (!i || !j)
94                 dp[i][j] = 0;
95             else if (x[i-1] == y[j-1])
96                 dp[i][j] = dp[i-1][j-1] + 1;
97             else
98                 dp[i][j] = max (dp[i-1][j], dp[i][j-1]);
99         }
100     }
101     print dp (x, y, x.size (), y.size ());
102 }
103
104 main () {
105     ios base::sync with stdio (0);
106     cin.tie (0);
107     cin.tie (0);
108
109     while (true) {
110         word.clear ();
111         vector<llu> x = read ();
112         if (x.size () == 0)
113             break;
114
115         vector<llu> y = read ();
116         solve (x, y);
117     }
118 }
119
120
121
122
123
124
125
126
127
128
129
130
```

```

1  #include <bits/stdc++.h>
2  #define pb push back
3  #define mp make pair
4  #define F first
5  #define S second
6  #define oioi printf("oioi\n")
7  #define digitCountDec(a) (int)floor(1 + log10((double)a))
8  #define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
9  using namespace std;
10 typedef unsigned long long int llu;
11 typedef long long int ll;
12 typedef vector<ll> vi;
13 typedef pair<ll, ll> ii;
14 typedef vector<ii> vii;
15 int dr[] = {0, 1, -1, 0, 1, -1, -1, 1};
16 int dc[] = {-1, 0, 0, 1, 1, 1, -1, -1};
17
18 ll gcd (ll x, ll y) {
19     return (y != 0 ? gcd (y, x%y) : x);
20 }
21
22 ll lcm (ll x, ll y) {
23     return (x / gcd (x, y) * y);
24 }
25
26 const ll N = 250*250 + 10;
27 ll idx[N], dp[N], tail[N];
28 ll n, p, q;
29 ll c = 1;
30
31 void read () {
32     memset (idx, -1, sizeof idx);
33     scanf ("%lld %lld %lld", &n, &p, &q);
34     for (ll i = 0; i <= p; ++i) {
35         ll x; scanf ("%lld", &x);
36         idx[x] = i;
37     }
38     for (ll i = 0; i <= q; ++i) {
39         ll x; scanf ("%lld", &x);
40         dp[i] = idx[x];
41     }
42 }
43
44 ll get index (int lo, int hi, int key) {
45     while (hi-lo > 1) {
46         ll mid = lo + ((hi-lo) >> 1);
47         if (tail[mid] >= key)
48             hi = mid;
49         else
50             lo = mid;
51     }
52     return hi;
53 }
54
55 void solve () {
56     // n log n LIS on modified vector
57     memset (tail, 0, sizeof tail);
58     ll length = 1;
59     tail[0] = dp[0];

```



```
72     for (ll i = 1; i <= q; i++) {
73
74         if (dp[i] < tail[0])
75             tail[0] = dp[i];
76
77         else if (dp[i] > tail[length-1])
78             tail[length++] = dp[i];
79
80         else
81             tail[get index (-1, length-1, dp[i])] = dp[i];
82     }
83
84     printf ("Case %lld: %lld\n", c++, length);
85 }
86
87 main () {
88
89     ll t; scanf ("%lld", &t);
90     while (t--) {
91
92         read ();
93         solve ();
94     }
95 }
96
```

```

1  #include <bits/stdc++.h>
2  #define pb push back
3  #define mp make pair
4  #define F first
5  #define S second
6  #define oioi printf("oioi\n")
7  #define digitCountDec(a) (int)floor(1 + log10((double)a))
8  #define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
9  using namespace std;
10 typedef unsigned long long int llu;
11 typedef long long int ll;
12 typedef vector<ll> vi;
13 typedef pair<ll, ll> ii;
14 typedef vector<ii> vii;
15 int dr[] = {0, 1, -1, 0, 1, -1, -1, 1};
16 int dc[] = {-1, 0, 0, 1, 1, 1, -1, -1};
17
18 ll gcd (ll x, ll y) {
19     return (y != 0 ? gcd (y, x%y) : x);
20 }
21
22 ll lcm (ll x, ll y) {
23     return (x / gcd (x, y) * y);
24 }
25
26 const ll N = 1e3 + 10;
27 ll c = 1;
28 ll dp[N][N];
29
30 void solve (string x, string y) {
31     ll len x = x.length ();
32     ll len y = y.length ();
33
34     for (ll i = 0; i <= len x; ++i) {
35         for (ll j = 0; j <= len y; ++j) {
36             if (!i || !j)
37                 dp[i][j] = 0;
38             else if (x[i-1] == y[j-1])
39                 dp[i][j] = dp[i-1][j-1] + 1;
40             else
41                 dp[i][j] = max (dp[i-1][j], dp[i][j-1]);
42         }
43     }
44
45     printf ("Case #%lld: you can visit at most %lld cities.\n", c++,
46            dp[len x][len y]);
47 }
48
49 main () {
50     string x, y;
51     while (getline (cin, x), x != "#") {
52         getline (cin, y);
53         solve (x, y);
54     }
55 }
56
57
58
59
60
61
62
63
64

```

```

1  #include <bits/stdc++.h>
2  #define pb push back
3  #define mp make pair
4  #define F first
5  #define S second
6  #define oioi printf("oioi\n")
7  #define digitCountDec(a) (int)floor(1 + log10((double)a))
8  #define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
9  using namespace std;
10 typedef unsigned long long int ll;
11 typedef long long int ll;
12 typedef vector<ll> vi;
13 typedef pair<ll, ll> ii;
14 typedef vector<ii> vii;
15 int dr[] = {0, 1, -1, 0, 1, -1, -1, 1};
16 int dc[] = {-1, 0, 0, 1, 1, 1, -1, -1};
17
18 ll gcd (ll x, ll y) {
19     return (y != 0 ? gcd (y, x%y) : x);
20 }
21
22 ll lcm (ll x, ll y) {
23     return (x / gcd (x, y) * y);
24 }
25
26 const ll N = 2 * 1e3 + 10;
27 char special[N], x[N]; ll len;
28 ii dp[N][N]; // F-> special positions, S -> total palindrome
29
30 void read () {
31     scanf ("%s", x);
32     len = strlen (x);
33
34     memset (special, false, sizeof special);
35     ll n; scanf ("%lld", &n);
36     for (ll i = 0; i < n; ++i) {
37         ll x; scanf ("%lld", &x);
38         special[x-1] = true;
39     }
40 }
41
42 void solve () {
43     for (ll k = 1; k <= len; ++k) {
44         for (ll i = 0; i < len - k + 1; ++i) {
45             if (k == 1) {
46                 dp[i][i].F = special[i];
47                 dp[i][i].S = 1;
48                 continue;
49             }
50             ii a; a.F = a.S = 0;
51             ll j = i + k - 1;
52             if (x[i] == x[j]) {
53                 if (k > 2)
54                     a = dp[i+1][j-1];
55                 a.F += special[i] + special[j];
56                 a.S += 2;
57             }
58             dp[i][j] = max (a, max (dp[i+1][j], dp[i][j-1]));
59         }
60     }
61 }

```

```
72     }
73 }
74
75     printf ("%lld\n", dp[0][len - 1].S);
76 }
77
78 main () {
79
80     read ();
81     solve ();
82 }
83
```

```

1  #include <bits/stdc++.h>
2  #define pb push back
3  #define mp make pair
4  #define F first
5  #define S second
6  #define eoq printf("eoq\n")
7  #define digitCountDec(a) (int)floor(1 + log10((double)a))
8  #define digitCount(a, b) (int)floor(1 + log10((double)a) / log10((double)b))
9  using namespace std;
10 typedef unsigned long long int llu;
11 typedef long long int ll;
12 typedef pair<ll, ll> ii;
13 typedef pair<double, double> dd;
14 typedef vector<ll> vi;
15 typedef vector<ii> vii;
16 int dr[] = { 0, 1, -1, 0, 1, -1, -1, 1};
17 int dc[] = {-1, 0, 0, 1, 1, 1, -1, -1};
18
19 ll gcd (ll x, ll y) {
20
21     return (y != 0 ? gcd (y, x%y) : x);
22 }
23
24 ll lcm (ll x, ll y) {
25
26     return (x / gcd (x, y) * y);
27 }
28
29 const ii zero = mp (0, 0);
30 const ll N = 2 * 1e3 + 10;
31 char s[N], special[N];
32 ll len;
33 ii dp[N][N];
34
35 void read () {
36
37     memset (special, false, sizeof special);
38
39     ll n; scanf ("%s %lld", s, &n);
40     while (n--) {
41
42         ll x; scanf ("%lld", &x);
43         special[--x] = true;
44     }
45
46     len = strlen (s);
47     memset (dp, -1, sizeof dp);
48 }
49
50 ii solve (ll i, ll j) {
51
52     if (i > j)
53         return dp[i][j] = zero;
54
55     if (i == j)
56         return dp[i][j] = mp (special[i], 1);
57
58     if (dp[i][j].F != -1)
59         return dp[i][j];
60
61     dp[i][j] = max (solve (i+1, j), solve (i, j-1));
62
63     if (s[i] == s[j]) {
64
65         ii ans = solve (i+1, j-1);
66         return dp[i][j] = max (dp[i][j], mp (ans.F + special[i] + special[j],
67             ans.S + 2));
68     }
69
70     return dp[i][j];

```

```
71  
72 main () {  
73  
74     read ();  
75     printf ("%lld\n", solve (0, len-1).S);  
76 }  
77
```

```
1  #include <bits/stdc++.h>
2  #define F first
3  #define S second
4  using namespace std;
5  typedef long long int ll;
6
7  const ll N = 1e3 + 10;
8  int caso = 1;
9  char v[N][N];
10 string s1, s2;
11 pair<int, int> dp[N][N];
12
13 void solve () {
14     int n = s1.length ();
15     int m = s2.length ();
16
17     for (int i = n; i >= 0; --i) {
18         for (int j = m; j >= 0; --j) {
19             if (i == n and j == m) {
20                 dp[i][j].F = 0;
21                 dp[i][j].S = 0;
22             }
23             else if (i == n) {
24                 dp[i][j].F = 0;
25                 dp[i][j].S = m-j;
26             }
27             else if (j == m) {
28                 dp[i][j].F = n-i;
29                 dp[i][j].S = 0;
30             }
31             else if (s1[i] == s2[j]) {
32                 dp[i][j].F = dp[i+1][j+1].F;
33                 dp[i][j].S = dp[i+1][j+1].S;
34             }
35             else if (dp[i+1][j].F + dp[i+1][j].S <= dp[i][j+1].F + dp[i][j+1].S) {
36                 dp[i][j].F = dp[i+1][j].F + 1;
37                 dp[i][j].S = dp[i+1][j].S;
38             }
39             else {
40                 dp[i][j].F = dp[i][j+1].F;
41                 dp[i][j].S = dp[i][j+1].S + 1;
42             }
43         }
44     }
45
46     printf ("Case %d: %d %d\n", caso++, dp[0][0].F , dp[0][0].S);
47 }
48
49 main () {
50     int t; scanf ("%d", &t);
51     while (t--) {
52         int h, w; scanf ("%d %d", &h, &w);
53         for (int i = 0; i < h; ++i) {
```

```
72         getchar ();
73         for (int j = 0; j < w; ++j)
74             scanf ("%c", &v[i][j]);
75     }
76
77     int n, x, y;
78     scanf ("%d %d %d", &n, &x, &y); getchar ();
79     --x; --y;
80     s1.clear ();
81     s1.push back (v[y][x]);
82     while (n--) {
83
84         char c; scanf ("%c", &c);
85         switch (c) {
86
87             case 'N':
88                 --y;
89                 break;
90
91             case 'E':
92                 ++x;
93                 break;
94
95             case 'W':
96                 --x;
97                 break;
98
99             case 'S':
100                 ++y;
101                 break;
102         }
103
104         s1.push back (v[y][x]);
105     }
106
107     scanf ("%d %d %d", &n, &x, &y); getchar ();
108     --x; --y;
109     s2.clear ();
110     s2.push back (v[y][x]);
111     while (n--) {
112
113         char c; scanf ("%c", &c);
114         switch (c) {
115
116             case 'N':
117                 --y;
118                 break;
119
120             case 'E':
121                 ++x;
122                 break;
123
124             case 'W':
125                 --x;
126                 break;
127
128             case 'S':
129                 ++y;
130                 break;
131         }
132
133         s2.push back (v[y][x]);
134     }
135
136     solve ();
137 }
138 }
139 }
```



```

1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int n;
5  string s;
6  int suffixArr[100010];
7
8  struct suffix {
9
10     int index;
11     int rank[2];
12 };
13
14 int compare (struct suffix a, struct suffix b) {
15
16     return (a.rank[0] == b.rank[0]) ? (a.rank[1] < b.rank[1] ? 1 : 0) : (a.rank[0]
17     < b.rank[0] ? 1 : 0);
18 }
19
20 void build () {
21     n = s.length ();
22     struct suffix suffixes[n];
23
24     for (int i = 0; i < n; ++i) {
25
26         suffixes[i].index = i;
27         suffixes[i].rank[0] = s[i] - 'a';
28         suffixes[i].rank[1] = ((i+1) < n)? (s[i + 1] - 'a') : -1;
29     }
30
31     sort (suffixes, suffixes+n, compare);
32
33     int ind[n];
34
35     for (int k = 4; k < 2*n; k = k*2) {
36
37         int rank = 0;
38         int prev rank = suffixes[0].rank[0];
39         suffixes[0].rank[0] = rank;
40         ind[suffixes[0].index] = 0;
41
42         for (int i = 1; i < n; ++i) {
43
44             if (suffixes[i].rank[0] == prev rank && suffixes[i].rank[1] ==
45             suffixes[i-1].rank[1]) {
46
47                 prev rank = suffixes[i].rank[0];
48                 suffixes[i].rank[0] = rank;
49             }
50             else {
51
52                 prev rank = suffixes[i].rank[0];
53                 suffixes[i].rank[0] = ++rank;
54             }
55
56             ind[suffixes[i].index] = i;
57         }
58
59         for (int i = 0; i < n; ++i) {
60
61             int nextindex = suffixes[i].index + k/2;
62             suffixes[i].rank[1] = (nextindex < n) ?
63             suffixes[ind[nextindex]].rank[0] : -1;
64         }
65
66         sort (suffixes, suffixes+n, compare);
67     }
68
69     for (int i = 0; i < n; ++i)

```

```
69     suffixArr[i] = suffixes[i].index;
70 }
71
72 void solve () {
73     vector<int> lcp (n, 0);
74     vector<int> invSuff (n, 0);
75
76     for (int i = 0; i < n; ++i)
77         invSuff[suffixArr[i]] = i;
78
79     int k = 0;
80     int ans = 0, pos = 0;
81
82     for (int i = 0; i < n; ++i) {
83         if (invSuff[i] == n-1) {
84             k = 0;
85             continue;
86         }
87         int j = suffixArr[invSuff[i]+1];
88         while (i+k < n and j+k < n and s[i+k] == s[j+k])
89             k++;
90         lcp[invSuff[i]] = k;
91         if (k > ans)
92             pos = i, ans = k;
93     }
94     if (ans > 2)
95         cout << s.substr (pos, ans) << '\n';
96     else
97         cout << '*' << '\n';
98 }
99
100 main () {
101     cin >> s;
102     build ();
103     solve ();
104 }
```