

```

1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int n,m;
5  map<string,int> mapa;
6  vector<int> parent,r;
7  vector<int> filhos;
8  int findSet(int i) {
9      return (parent[i] == -1) ? i : parent[i] = findSet(parent[i]);
10 }
11 bool isSameSet(int i, int j) {
12     return findSet(i) == findSet(j);
13 }
14 void unionF(int i,int j){
15     if (!isSameSet(i, j)) {
16         int x = findSet(i), y = findSet(j);
17         if (r[x] > r[y]) {
18             filhos[findSet(x)] += filhos[findSet(y)];
19             parent[y] = x;
20         } else {
21             filhos[findSet(y)] += filhos[findSet(x)];
22             parent[x] = y;
23             if (r[x] == r[y])
24                 r[y]++;
25         }
26     }
27 }
28 main(){
29     int i,j,k;
30     string a,b;
31
32     cin >> n;
33
34     for(i=0;i<n;i++){
35         cin >> m;
36         for(j=0;j<m;j++){
37             cin >> a >> b;
38             if(!mapa.count(a)){
39                 int aux = mapa.size();
40                 mapa[a] = aux;
41                 parent.push back(-1);
42                 filhos.push back(1);
43                 r.push back(0);
44             }
45             if(!mapa.count(b)){
46                 int aux = mapa.size();
47                 mapa[b] = aux;
48                 parent.push back(-1);
49                 filhos.push back(1);
50                 r.push back(0);
51             }
52             unionF(mapa[a],mapa[b]);
53             cout << filhos[findSet(mapa[a])] << endl;
54         }
55         mapa.clear();
56         parent.clear();
57         filhos.clear();
58         r.clear();
59     }
60 }
61

```

```

1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int n;
5  int A[21];
6  int M[21][21];
7
8  int dp[21][1<21];
9
10 int solve(int current, int w,int visi){
11     visi |= (1 << current);
12     if(w<=0)
13         return 0;
14     int ans = 0;
15     if(dp[current][visi]!=-1) return dp[current][visi];
16     for(int i=0;i<n;i++){
17         if(!(visi & (1 << i)) and w-(A[i]+M[current][i]) >= 0){
18             ans = max(ans,solve(i,w-(A[i]+M[current][i]),visi)+1);
19         }
20     }
21     return dp[current][visi] = ans;
22 }
23
24
25 main(){
26     ios base::sync with stdio(0);
27     cin.tie(0);
28     while(cin >> n and n){
29         memset(dp,-1,sizeof dp);
30         for(int i=0;i<n;i++){
31             cin >> A[i];
32         }
33         for(int i=0;i<n;i++){
34             for(int j=0;j<n;j++){
35                 cin >> M[i][j];
36             }
37         }
38         int ans = 0;
39         for(int i=0;i<n;i++){
40             int visi = (1 << i);
41             if(420 - A[i] >=0)
42                 ans = max(ans,solve(i,420 - A[i],visi)+1);
43         }
44
45         cout << ans << endl;
46     }
47 }
48
49 }
50

```

```

1  #include <bits/stdc++.h>
2  using namespace std;
3
4  double V[16][16];
5  vector<pair<int,int> > P;
6  int n;
7
8  double dp[20][1 << 16];
9
10 inline double calc(int i,int j){
11     return sqrt((P[i].first-P[j].first)*(P[i].first-P[j].first) + (P[i].second -
12     P[j].second)*(P[i].second - P[j].second));
13 }
14 double solve(int current, int mask){
15     if(mask == ((1<<(n+1))-1)){
16         return V[current][0];
17     }
18     if(dp[current][mask]!=-1) return dp[current][mask];
19     double ans = 1e9 + 10;
20     for(int i=1;i<=n;i++){
21         if(!(mask & (1<<i)))
22             ans = min(solve(i,mask | (1<<i))+V[current][i],ans);
23     }
24     return dp[current][mask] = ans;
25 }
26
27 main(){
28     ios base::sync with stdio(0);
29     cin.tie(0);
30     int x,y,a,b;
31     while(cin >> n and n){
32         P.clear();
33         for(int i=0;i<=n;i++){
34             for(int j=0;j<=(1<<(n+1));j++){
35                 dp[i][j] = -1;
36             }
37         }
38         cin >> x >> y;
39         P.push back(make pair(x,y));
40         for(int i=1;i<=n;i++){
41             cin >> a >> b;
42             P.push back(make pair(a,b));
43         }
44         for(int i=0;i<=n;i++){
45             for(int j=0;j<=n;j++){
46                 V[i][j] = calc(i,j);
47             }
48         }
49         cout << fixed << setprecision(2) << solve(0,1) << endl;
50     }
51 }
52
53
54
55
56

```

```

1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int n;
5  int M[19][19];
6  int A[19];
7  int dp[19][(1 << 19)];
8  int dead;
9  int solve(int current,int mask){
10
11     if(mask == dead)
12         return 0;
13
14     if(dp[current][mask]!=-1) return dp[current][mask];
15
16     int ans = 1e9+10;
17
18     for(int i=0;i<n;i++){
19         if(!(mask & A[i])){
20             ans = min(ans,solve(current+1,(mask | A[i]))+M[i][current]);
21         }
22     }
23     return dp[current][mask] = ans;
24 }
25
26 main(){
27     for(int i=0;i<=18;i++)
28         A[i] = 1 << i;
29     while(scanf("%d",&n) and n){
30         for(int i=0;i<n;i++)
31             for(int j=0;j<(1 << (n+1));j++)
32                 dp[i][j] = -1;
33         dead = (1 << n)-1;
34         for(int i=0;i<n;i++){
35             for(int j=0;j<n;j++){
36                 scanf("%d",&M[i][j]);
37             }
38         }
39         cout << solve(0,0) << endl;
40     }
41 }
42

```

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  typedef long long int ll;
5
6  main(){
7      ios base::sync with stdio(0);
8      cin.tie(0);
9      ll z,n;
10     cin >> z;
11     for(int k=0;k<z;k++){
12         cin >> n;
13         ll nr = n << 1;
14         ll M[nr][nr];
15         for(int i=0;i<n;i++){
16             for(int j=0;j<n;j++){
17                 cin >> M[i][j];
18                 M[i+n][j] = M[i][j];
19                 M[i][j+n] = M[i][j];
20                 M[i+n][j+n] = M[i][j];
21                 if(i>0) M[i][j] += M[i-1][j];
22                 if(j>0) M[i][j] += M[i][j-1];
23                 if(i>0 && j>0) M[i][j] -= M[i-1][j-1];
24             }
25         }
26         for(int i=0;i<nr;i++){
27             for(int j=0;j<nr;j++){
28                 if((i>=n or j>=n)){
29                     if(i>0){
30                         M[i][j] += M[i-1][j];
31                     }
32                     if(j>0){
33                         M[i][j] += M[i][j-1];
34                     }
35                     if(i>0 && j>0){
36                         M[i][j] -= M[i-1][j-1];
37                     }
38                 }
39             }
40         }
41         ll ans = -1000*100*100;
42         int xi,yi,xf,yf;
43         for(int i=0;i<n;i++){
44             for(int j=0;j<n;j++){
45                 for(int x=i;x<i+n;x++){
46                     for(int y=j;y<j+n;y++){
47                         ll at = M[x][y];
48                         if(i>0) at -= M[i-1][y];
49                         if(j>0) at -= M[x][j-1];
50                         if(i>0 and j>0) at += M[i-1][j-1];
51                         ans = max(ans,at);
52                         if(ans == at){
53                             xi = i;
54                             yi = j;
55                             xf = x;
56                             yf = y;
57                         }
58                     }
59                 }
60             }
61         }
62         if(xi==0 and yi==0 and ((xf==nr-1 and yf==n-1) or (xf==n-1 and yf==nr-1)))
63             ans -= M[n-1][n-1];
64         cout << ans << endl;
65         //cout << "COORD I:  " << xi << " " << yi << endl;
66         // << "COORD F:  " << xf << " " << yf << endl;
67     }
68 }
69
```

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  #define F first
4  #define S second
5  #define mp make pair
6  #define pb push back
7
8  typedef long long int ll;
9
10 ll sum[20005];
11 ll n,s;
12
13 main(){
14     ios base::sync with stdio(0);
15     cin.tie(0);
16     cin >> n;
17
18     for(int k=0;k<n;k++){
19         cin >> s;
20         ll ans = 0,ansx=1,ansy=1;
21         ll at = 0;
22         int from=1,to=1;
23         for(int i=0;i<s-1;i++){
24             cin >> sum[i];
25             at += sum[i];
26             if(at >= ans){
27                 if(at > ans or (ansy-ansx < i+2-from) or (ansy-ansx == i+2-from
28                     and from <= ansx)){
29                     ansx = from;
30                     ansy = i+2;
31                 }
32                 ans = at;
33             }
34             if(at<0){
35                 at = 0;
36                 from = i+2;
37             }
38         }
39         if(ans > 0)
40             cout << "The nicest part of route " << k+1 << " is between stops " <<
41                 ansx << " and " << ansy << endl;
42         else
43             cout << "Route " << k+1 << " has no nice parts" << endl;
44     }
```

```

1  #include <bits/stdc++.h>
2  using namespace std;
3
4  #define mp make pair
5  #define F first
6  #define S second
7  typedef long long int ll;
8  typedef pair<ll,ll> ii;
9  typedef vector<ll> vi;
10 typedef vector<ii> vii;
11
12 vi A;
13 vii st;
14 int n,m;
15
16 int getMax(vector<ll> e){
17     ll s = -1;
18     int p=-1;
19     for(int i=0;i<e.size();i++){
20         if(e[i]!=-1 and s < A[e[i]]){
21             s = A[e[i]];
22             p = i;
23         }
24     }
25     return p;
26 }
27
28 void build(int P,int L,int R){
29     if(L==R){
30         st[P] = mp(L,-1);
31         return;
32     }
33     if(L>R || R<L)
34         return;
35     int nxt = P << 1;
36     int mid = (L+R) >> 1;
37     build(nxt,L,mid);
38     build(nxt+1,mid+1,R);
39
40     vi e;
41     ll s1,s2;
42     int p;
43     e.push_back(st[nxt].F);
44     e.push_back(st[nxt].S);
45     e.push_back(st[nxt+1].F);
46     e.push_back(st[nxt+1].S);
47     p = getMax(e);
48     s1 = p==-1 ? -1:e[p];
49     e[p]=-1;
50     p = getMax(e);
51     s2 = p==-1 ? -1:e[p];
52
53     st[P] = mp(s1,s2);
54 }
55
56
57 void update (int p, int L, int R, int i, int value) {
58
59     // no overlap
60     if(L > i or R < i) return;
61
62     // total overlap
63     if(L == R and L == i) {
64         A[i] = value;
65         st[p] = mp(i,-1);
66         return;
67     }
68
69     int nxt = p << 1;
70     int mid = (L + R) >> 1;

```

```

71     update (nxt, L, mid, i, value);
72     update (nxt + 1, mid + 1, R, i, value);
73
74     vi e;
75     ll s1,s2;
76     int V;
77     e.push_back(st[nxt].F);
78     e.push_back(st[nxt].S);
79     e.push_back(st[nxt+1].F);
80     e.push_back(st[nxt+1].S);
81     V = getMax(e);
82     s1 = V==-1? -1:e[V];
83     e[V]=-1;
84     V = getMax(e);
85     s2 = V==-1? -1:e[V];
86     st[p].F = s1;
87     st[p].S = s2;
88 }
89 ii query(int p, int L, int R, int i, int j){
90     // no overlap
91     if(i>R || j<L) return mp(-1,-1);
92
93     // total overlap
94     if(L>=i && R<=j) return st[p];
95
96     // partial overlap
97     int nxt = p << 1;
98     int mid = (L + R) >> 1;
99     ii p1 = query(nxt,L,mid,i,j);
100    ii p2 = query(nxt + 1,mid +1,R,i,j);
101
102    if(p1.F== -1 and p1.S== -1) return p2;
103    if(p2.F== -1 and p2.S== -1) return p1;
104
105    vi e;
106    ll s1,s2;
107    int V;
108    e.push_back(p1.F);
109    e.push_back(p1.S);
110    e.push_back(p2.F);
111    e.push_back(p2.S);
112    V = getMax(e);
113    s1 = V==-1? -1:e[V];
114    e[V] = -1;
115    V = getMax(e);
116    s2 = e[V];
117
118    return mp(s1,s2);
119 }
120
121
122 main(){
123     int i,j,k,a,b;
124     char o;
125     cin >> n;
126     st.resize(4*n);
127     A.resize(2*n);
128     st.assign(4*n,mp(-1,-1));
129     A.assign(2*n,-1);
130     for(i=0;i<n;i++){
131         cin >> A[i];
132     }
133     build(1,0,n-1);
134     cin >> m;
135     for(i=0;i<m;i++){
136         cin >> o >> a >> b;
137         if(o=='Q'){
138             ii aux = query(1,0,n-1,a-1,b-1);
139             cout << A[aux.F]+A[aux.S] << endl;
140         }

```



```
141         else{
142             update(1,0,n-1,a-1,b);
143         }
144     }
145
146 }
147
148
149
150
151
```

```

1  #include <bits/stdc++.h>
2  using namespace std;
3
4  typedef long long int ll;
5  typedef pair<ll,ll> ii;
6  typedef vector<ii> vii;
7  typedef vector<ll> vi;
8
9  vi st,lazy;
10 int n;
11
12 ll query(int p, int L, int R, int i,int j){
13
14     if(lazy[p]!=0){
15         st[p] += (R-L+1)*lazy[p];
16         if(R!=L){
17             lazy[p<<1] += lazy[p];
18             lazy[(p<<1)+1] += lazy[p];
19         }
20         lazy[p] = 0;
21     }
22     // no overlap
23     if(i>R || j<L) return 0;
24
25     // total overlap
26     if(L>=i && R<=j) return st[p];
27
28     // partial overlap
29     int nxt = p << 1;
30     int mid = (L + R) >> 1;
31
32     return query(nxt,L,mid,i,j) + query(nxt + 1,mid +1,R,i,j);
33 }
34 void update(int P,int L,int R, int i,int j, ll value){
35
36     if(lazy[P]!=0){
37         st[P] += (R-L+1)*lazy[P];
38         if(L!=R){
39             lazy[P << 1] += lazy[P];
40             lazy[(P << 1)+1] += lazy[P];
41         }
42         lazy[P] = 0;
43     }
44
45     // no overlap
46     if( L > j or R < i) return;
47
48     // total overlap
49     if(L >= i and R <= j){
50         st[P] += (R-L+1)*value;
51         if(L!=R){
52             lazy[P<<1] += value;
53             lazy[(P<<1)+1] += value;
54         }
55
56         return;
57     }
58
59     // partial overlap
60     int nxt = P << 1;
61     int mid = (L+R) >> 1;
62
63     update(nxt, L, mid, i, j, value);
64     update(nxt+1,mid+1,R,i,j,value);
65
66     st[P] = st[nxt]+st[nxt+1];
67
68 }
69
70 main(){

```

```
71     int i,j,q,z,a,b,o;
72     ll v;
73
74     cin >> z;
75
76     for(i=0;i<z;i++){
77         cin >> n >> q;
78         st.resize(n << 2);
79         st.assign(n << 2,0);
80         lazy.resize(n << 2);
81         lazy.assign(n << 2,0);
82         for(j=0;j<q;j++){
83             cin >> o;
84             if(o==1){
85                 cin >> a >> b;
86                 cout << query(1,0,n-1,a-1,b-1) << endl;
87             }
88             else{
89                 cin >> a >> b >> v;
90                 update(1,0,n-1,a-1,b-1,v);
91             }
92         }
93     }
94 }
95
```

```

1  #include <bits/stdc++.h>
2  using namespace std;
3
4  vector<vector<string > > Grafo(105);
5  map<string,int> V;
6  map<int,string> S;
7  map<string,int> grau;
8  int n,m;
9  vector<int> saida;
10 void kahn(){
11     int i;
12     priority queue<int> F;
13     for(i=0;i<n;i++){
14         if(grau[S[i]]==0)
15             F.push(-i);
16     }
17     while(!F.empty()){
18         int aux = -F.top();
19         saida.push back(aux);
20         F.pop();
21         for(i=0;i<Grafo[aux].size();i++){
22             if(--grau[Grafo[aux][i]]==0)
23                 F.push(-V[Grafo[aux][i]]);
24         }
25     }
26 }
27
28 main(){
29     int i,j,k,cont=1;
30     string e,from,to;
31     while(cin >> n){
32         cin.ignore();
33         for(i=0;i<n;i++){
34             cin >> e;
35             cin.ignore();
36             V[e]=i;
37             S[i]=e;
38             Grafo[i].clear();
39         }
40         cin >> m;
41         for(i=0;i<m;i++){
42             cin >> from >> to;
43             Grafo[V[from]].push back(to);
44             grau[to]++;
45         }
46         kahn();
47         cout << "Case #" << cont << ": Dilbert should drink beverages in this
order: ";
48         for(i=0;i<saida.size();i++){
49             cout << S[saida[i]];
50             if(i!=saida.size()-1)
51                 cout << " ";
52         }
53         cout << "." << endl;
54         V.clear();
55         S.clear();
56         saida.clear();
57         cont++;
58         cout << endl;
59     }
60 }
61

```

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  typedef vector<vector<int> > vvi;
4  typedef vector<int> vi;
5  typedef vector<pair<int,int> > vii;
6  typedef pair<int,int> ii;
7  vvi Grafo(100001);
8  bool visitados[100001];
9  vector<bool> visi;
10 stack<int> ordem;
11 int n,m;
12 void dfs(int n){
13     visitados[n] = true;
14     for(int i=0;i<Grafo[n].size();i++){
15         if(!visitados[Grafo[n][i]])
16             dfs(Grafo[n][i]);
17     }
18 }
19 void dfsOrd(int n){
20     visitados[n] = true;
21     for(int i=0;i<Grafo[n].size();i++){
22         if(!visitados[Grafo[n][i]])
23             dfs(Grafo[n][i]);
24     }
25     ordem.push(n);
26 }
27 void reset(){
28     for(int i=0;i<n;i++)
29         visitados[i]=false;
30 }
31
32 main(){
33     ios base::sync with stdio(0);
34     cin.tie(0);
35     int i,j,z,from,to;
36     cin >> z;
37
38     for(i=0;i<z;i++){
39         cin >> n >> m;
40         for(j=0;j<n;j++)
41             Grafo[j].clear();
42         for(j=0;j<m;j++){
43             cin >> from >> to;
44             Grafo[from-1].push back(to-1);
45         }
46         reset();
47
48         for(j=0;j<n;j++){
49             if(!visitados[j])
50                 dfsOrd(j);
51         }
52         reset();
53         int cont = 0;
54         while(!ordem.empty()){
55             int x = ordem.top();
56             ordem.pop();
57             if(!visitados[x]){
58                 dfs(x);
59                 cont++;
60             }
61         }
62         cout << cont << endl;
63     }
64 }
65
66

```

```

1  #include <bits/stdc++.h>
2  using namespace std;
3
4  typedef long long int ll;
5
6  int n,m;
7  int produtos[1010][2],P[110];
8  ll dp[1010][30];
9  inline ll solve(ll current, ll w){
10
11     if(current<0 or w <= 0) return 0LL;
12
13     if(dp[current][w]!=-1) return dp[current][w];
14
15     ll ans;
16     if(produtos[current][1]<=w)
17         ans = solve(current-1,w-produtos[current][1])+produtos[current][0];
18
19     ans = max(solve(current-1,w),ans);
20
21     return dp[current][w] = ans;
22 }
23
24 main(){
25     int t;
26
27     scanf("%d",&t);
28
29     for(int k=0;k<t;k++){
30
31         scanf("%d",&n);
32
33         for(int i=0;i<n;i++){
34             scanf("%d %d",&produtos[i][0], &produtos[i][1]);
35         }
36
37         scanf("%d",&m);
38         memset(dp,-1,sizeof dp);
39         ll ans = 0;
40         for(int i=0;i<m;i++){
41             scanf("%d",&P[i]);
42             ans += solve(n-1,P[i]);
43         }
44         printf("%lld\n",ans);
45     }
46 }
47

```

```

1  #include <bits/stdc++.h>
2  using namespace std;
3
4  typedef long long ll;
5
6  ll w,T;
7  ll wt[40];
8  ll A[40][40];
9  int n;
10
11 inline ll knapsack(){
12
13     ll K[n+1][T+1];
14
15     for(int i=0;i<=n;i++){
16         for(int j=0;j<=T;j++){
17             if(i==0 or j==0)
18                 K[i][j] = 0;
19             else if(A[i-1][1] <= j){
20                 K[i][j] = max(A[i-1][0]+K[i-1][j-A[i-1][1]],K[i-1][j]);
21             }
22             else
23                 K[i][j] = K[i-1][j];
24         }
25     }
26     ll total B = 0;
27     ll total w = 0;
28     vector<pair<ll,ll> > V;
29     for(int i=n,j=T;i>0;i--){
30         if(K[i][j]!=K[i-1][j]){
31             V.push back(make pair(wt[i-1],A[i-1][0]));
32             ++total B;
33             j-=A[i-1][1];
34         }
35     }
36     cout << K[n][T] << endl
37         << total B << endl;
38     for(int i=V.size()-1;i>=0;i--){
39         cout << V[i].first << " " << V[i].second << endl;
40     }
41     main(){
42         bool f = true;
43         while(cin >> T >> w){
44             if(!f)
45                 cout << endl;
46             cin >> n;
47             for(int i=0;i<n;i++){
48                 cin >> A[i][1] >> A[i][0];
49                 wt[i] = A[i][1];
50                 A[i][1] = (2*w*A[i][1]) + (w*A[i][1]);
51             }
52             knapsack();
53             f = false;
54         }
55
56
57     }
58

```

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  int w;
4  vector<int> car;
5  int dp[10001][10001];
6  int solve(int current,int s1,int s2){
7      if(current>=car.size())return dp[s1][s2] = 0LL;
8
9      if(dp[s1][s2]!=-1) return dp[s1][s2];
10
11     int ans = 0;
12     if(car[current]+s1 <= w)
13         ans = max(solve(current+1,s1+car[current],s2)+1,ans);
14     if(car[current]+s2 <= w)
15         ans = max(solve(current+1,s1,s2+car[current])+1,ans);
16
17     return dp[s1][s2] = ans;
18 }
19 void print(int current, int s1, int s2){
20
21     if(current>=car.size()) return;
22     if(s1+car[current] <=w and dp[s1][s2] - 1 == dp[s1+car[current]][s2]){
23         printf("port\n");
24         print(current+1,s1+car[current],s2);
25     }
26     else if(s2+car[current] <=w and dp[s1][s2]-1 == dp[s1][s2+car[current]]){
27         printf("starboard\n");
28         print(current+1,s1,s2+car[current]);
29     }
30 }
31
32 }
33 main(){
34     int n,aux;
35     scanf("%d",&n);
36     for(int k=0;k<n;k++){
37         scanf("%d",&w);
38         w*=100;
39         while(scanf("%d",&aux) and aux){
40             car.push back(aux);
41         }
42         memset(dp,-1,sizeof dp);
43         printf("%d\n",solve(0,0,0));
44
45         print(0,0,0);
46         car.clear();
47         if(k<n-1)
48             puts("");
49     }
50 }
51

```



```

1  #include <bits/stdc++.h>
2  using namespace std;
3
4  typedef vector<vector<int> > vvi;
5  typedef vector<int> vi;
6  typedef vector<pair<int,int> > vii;
7  vvi Grafo(1005);
8  vii pontes;
9  int dfs low[1005];
10 int dfs num[1005];
11 int dfs parent[1005];
12 bool articulation vertex[1005];
13 int dfsNumberCounter,Children,dfsRoot,n,arti;
14 void print dfs(){
15     cout << pontes.size() << " critical links\n";
16     for(int i=0;i<pontes.size();i++){
17         cout << pontes[i].first << " - " << pontes[i].second << "\n";
18     }
19     cout << "\n";
20 }
21 void dfs(int u){
22     dfs low[u] = dfs num[u] = dfsNumberCounter++;
23     for(int i=0;i<Grafo[u].size();i++){
24         int v = Grafo[u][i];
25         if(dfs num[v]==-1){
26             dfs parent[v] = u;
27             if(u==dfsRoot)
28                 Children++;
29             dfs(v);
30
31             if(dfs low[v]>=dfs num[u]){
32                 articulation vertex[u] = true;}
33 //             if(dfs low[v]>dfs num[u])
34 //                 pontes.push back(make pair(u,v));
35             dfs low[u] = min(dfs low[u],dfs low[v]);
36         }
37         else if(v!=dfs parent[u])
38             dfs low[u] = min(dfs low[u],dfs num[v]);
39     }
40 }
41 void reset(){
42     for(int i=0;i<n;i++){
43         Grafo[i].clear();
44         dfs num[i] = -1;
45         dfs low[i] = 0;
46         dfs parent[i] = 0;
47         articulation vertex[i] = false;
48     }
49     //pontes.clear();
50     dfsNumberCounter = 0;
51     arti = 0;
52 }
53 void solve(){
54     for(int i=0;i<n;i++){
55         if(dfs num[i]==-1){
56             dfsRoot = i;
57             Children = 0;
58             dfs(i);
59             articulation vertex[i] = (Children>1);
60         }
61     }
62     for(int i=0;i<n;i++){
63         if(articulation vertex[i])
64             arti++;
65     }
66     cout << arti << "\n";
67 }
68
69 main(){
70     int i,j,k,from,to,m;

```

```
71
72     while(cin >> n and n){
73         reset();
74         for(i=0;i<n+1;i++){
75             cin >> from;
76             if(from==0)
77                 break;
78             while(cin >> to and to){
79                 Grafo[from-1].push back(to-1);
80                 Grafo[to-1].push back(from-1);
81                 if(getchar()=='\n')
82                     break;
83             }
84         }
85         solve();
86     }
87 }
88
```

```

1  #include <bits/stdc++.h>
2  #define S second
3  #define F first
4  using namespace std;
5  typedef pair<int,int> ii;
6  typedef pair<int,ii> iii;
7  typedef vector<int> vi;
8  typedef vector<iii> viii;
9  int dy[] = {1,-1,0,0};
10 int dx[] = {0,0,1,-1};
11 bool Grafo[1001][1001];
12 int n,m;
13
14 bool valid(int i,int j){
15     if(Grafo[i][j]) return false;
16     if(i<0 or i>=n) return false;
17     if(j<0 or j>=m) return false;
18     return true;
19 }
20
21 int bfs(ii ini,ii dest){
22     queue<iii> q;
23     q.push(make pair(0,make pair(ini.F,ini.S)));
24     Grafo[ini.F][ini.S]=true;
25     while(!q.empty()){
26         iii x = q.front();
27         ii p = x.second;
28         q.pop();
29         if(x.second == dest)
30             return x.first;
31         for(int i=0;i<4;i++){
32             if(valid(p.first+dy[i],p.second+dx[i])){
33                 Grafo[p.first+dy[i]][p.second+dx[i]] = true;
34                 q.push(make pair(x.first+1,make pair(p.first+dy[i],p.second+dx[i])));
35             }
36         }
37     }
38 }
39
40 }
41
42 main(){
43     ios base::sync with stdio(0);
44     cin.tie(0);
45     int l,c,i,j,k,z,xi,yi,xd,yd,b;
46     while(cin >> n >> m and n and m){
47         cin >> z;
48         for(i=0;i<n;i++){
49             for(j=0;j<m;j++){
50                 Grafo[i][j] = false;
51             }
52         }
53         for(i=0;i<z;i++){
54             cin >> l >> b;
55             for(j=0;j<b;j++){
56                 cin >> c;
57                 Grafo[l][c] = true;
58             }
59         }
60         cin >> xi >> yi;
61         cin >> xd >> yd;
62         cout << bfs(make pair(xi,yi),make pair(xd,yd)) <<"\n";
63     }
64 }
65
66

```

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  typedef pair<int,int> ii;
4  typedef vector<ii> vii;
5  typedef vector<vii> vvii;
6  typedef vector<pair<int,ii> > viii;
7  typedef long long int ll;
8  viii Grafo,MST;
9  int parent[200001];
10 ll total;
11 int findset(int x){
12     if(x!=parent[x])
13         parent[x] = findset(parent[x]);
14     return parent[x];
15 }
16 void UNION(int x,int y){
17     parent[x] = parent[y];
18 }
19 void kruskal(){
20     int pu,pv;
21     sort(Grafo.begin(),Grafo.end());
22     for(int i=0;i<Grafo.size();i++){
23         pu = findset(Grafo[i].second.first);
24         pv = findset(Grafo[i].second.second);
25         if(pu!=pv){
26             total+=Grafo[i].first;
27             UNION(pu,pv);
28         }
29     }
30 }
31
32 main(){
33     int n,m,from,to,w;
34
35     while(cin >> n >> m and n and m){
36         Grafo.clear();
37         ll t = 0;
38         for(int i=0;i<m;i++){
39             cin >> from >> to >> w;
40             Grafo.push back(make pair(w,make pair(from,to)));
41             t+=w;
42         }
43         for(int i=0;i<n;i++){
44             parent[i] = i;
45         }
46         total = 0;
47         kruskal();
48         cout << t-total << endl;
49     }
50 }
51

```

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  typedef vector<vector<int> > vvi;
4  typedef vector<int> vi;
5  vvi Grafo(2005);
6  bool visitados[2005];
7  int n,m;
8  void dfs(int n){
9      visitados[n] = true;
10     for(int i=0;i<Grafo[n].size();i++){
11         if(!visitados[Grafo[n][i]])
12             dfs(Grafo[n][i]);
13     }
14 }
15
16
17 main(){
18     ios base::sync with stdio(0);
19     cin.tie(0);
20     int i,j,z,from,to,way;
21
22     while(cin >> n >> m and n and m){
23         for(j=0;j<n;j++){Grafo[j].clear();}
24         for(j=0;j<m;j++){
25             cin >> from >> to >> way;
26             Grafo[from-1].push back(to-1);
27             if(way==2)
28                 Grafo[to-1].push back(from-1);
29         }
30         bool falha = false;
31         for(j=0;j<n;j++){
32             memset(visitados,false,sizeof(visitados));
33             dfs(j);
34             for(int k=0;k<n;k++){if(visitados[k]==false){falha = true;break;}}
35             if(falha)
36                 break;
37         }
38         if(falha)
39             cout << 0 << endl;
40         else
41             cout << 1 << endl;
42     }
43 }
44
45
```

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  typedef long long ll;
4  typedef pair<int,int> ii;
5  typedef vector<ii> vii;
6  typedef vector<vii> vvii;
7  typedef vector<int> vi;
8  typedef vector<ll> vll;
9  vvii Grafo(1001);
10 int n,m;
11
12 bool ford(int ini){
13     vll dist(n+1,LLONG_MAX);
14     dist[ini] = 0;
15     for(int i=0;i<n;i++){
16         for(int k=0;k<n;k++){
17             for(int j=0;j<Grafo[k].size();j++){
18                 ii v = Grafo[k][j];
19                 if(dist[k]!=LLONG_MAX)
20                     dist[v.first] = min(dist[v.first],dist[k]+v.second);
21             }
22         }
23     }
24     // bool negative = false;
25     for(int i=0;i<n;i++){
26         for(int j = 0;j<Grafo[i].size();j++){
27             ii v = Grafo[i][j];
28             if(dist[v.first]!=LLONG_MAX and dist[v.first] > dist[i]+v.second){
29                 return true;
30             }
31         }
32     }
33     return false;
34 }
35 void reset(){
36     for(int i=0;i<n;i++){
37         Grafo[i].clear();
38     }
39 }
40
41 main(){
42     int k,from,to,w,i,j;
43
44     cin >> k;
45
46     for(i=0;i<k;i++){
47         cin >> n >> m;
48         reset();
49         for(j=0;j<m;j++){
50             cin >> from >> to >> w;
51             Grafo[from].push back(make pair(to,w));
52         }
53         if(ford(0))
54             cout << "possible\n";
55         else
56             cout << "not possible\n";
57     }
58 }
59

```

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  typedef long long int ll;
4  ll Grafo[101][101];
5  bool visitados[101];
6  int m;
7
8  main(){
9      int n,z,from,to,x,y;
10     scanf("%d",&z);
11     for(int v =0;v<z;v++){
12         scanf("%d", &m);
13         scanf("%d", &n);
14         for(int i=0;i<m;i++){
15             for(int j=0;j<m;j++){
16                 if(i==j)
17                     Grafo[i][j] = 0;
18                 else
19                     Grafo[i][j] = 100000000;
20             }
21         }
22         for(int q=0;q<n;q++){
23             scanf("%d %d",&from,&to);
24             Grafo[from][to] = 1;
25             Grafo[to][from] = 1;
26         }
27         scanf("%d %d",&x,&y);
28         for (int k = 0; k < m; k++){
29             for (int i = 0; i < m; i++){
30                 for (int j = 0; j < m; j++){
31                     Grafo[i][j] = min(Grafo[i][j], Grafo[i][k] + Grafo[k][j]);
32                 }
33             }
34         }
35         ll saida = 0;
36         for(int i=0;i<m;i++)
37             saida = max(Grafo[x][i]+Grafo[i][y],saida);
38
39         cout << "Case " << v+1<< ": " << saida << endl;
40     }
41 }
42
```

```

1  #include <bits/stdc++.h>
2  using namespace std;
3
4  typedef long long int ll;
5  map<string,int> cidades;
6  map<int,string> vertices;
7  vector<vector<pair<int,int> > > Grafo(30);
8  int pai[30];
9  int n,m;
10
11 void dijkstra(string o,string des){
12
13     int dist[n];
14     for(int i=0;i<n;i++) dist[i] = 10101010;
15     priority queue<pair<int,int>, vector<pair<int,int> >, greater<pair<int,int> > >
16     pq;
17
18     pq.push(make pair(0,cidades[o]));
19     dist[cidades[o]] = 0;
20
21     while(!pq.empty()){
22
23         int d = pq.top().first;
24         int v = pq.top().second;
25         pq.pop();
26         for(int i=0;i<Grafo[v].size();i++){
27             if(dist[Grafo[v][i].first] > Grafo[v][i].second+d){
28                 pq.push(make pair(Grafo[v][i].second+d,Grafo[v][i].first));
29                 pai[Grafo[v][i].first] = v;
30                 dist[Grafo[v][i].first] = Grafo[v][i].second+d;
31             }
32         }
33     }
34
35 }
36
37 main(){
38
39     memset(pai,-1,sizeof pai);
40     string o,des,e,aux;
41     int d;
42     cin >> n;
43     cin.ignore();
44     for(int i=0;i<n;i++){
45         cin >> e;
46         cidades[e] = i;
47         vertices[i] = e;
48     }
49     cin >> m;
50     for(int i=0;i<m;i++){
51         cin.ignore();
52         cin >> e >> aux >> d;
53         Grafo[cidades[e]].push back(make pair(cidades[aux],d));
54         Grafo[cidades[aux]].push back(make pair(cidades[e],d));
55     }
56     cin >> o >> des;
57
58     dijkstra(o,des);
59     vector<string> ans;
60     while(pai[cidades[des]]!=-1){
61         ans.push back(des);
62         des = vertices[pai[cidades[des]]];
63     }
64     ans.push back(o);
65     for(int i=ans.size()-1;i>=0;i--){
66         if(i!=ans.size()-1)
67             cout << "- ";
68         cout << ans[i];
69     }

```



```
70     cout << endl;  
71 }  
72
```

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  typedef long long int ll;
5  ll n;
6  int moedas[] = {1,5,10,25,50};
7  ll dp[10][1000010];
8
9  inline ll solve(ll current,ll sum){
10
11     if(sum==0) return 1LL;
12     if(current < 0 or sum<0) return 0LL;
13
14     if(dp[current][sum]!=-1) return dp[current][sum];
15
16     ll ans = solve(current,sum-moedas[current])+solve(current-1,sum);
17
18     return dp[current][sum] = ans;
19 }
20
21
22 main(){
23     ios base::sync with stdio(0);
24     cin.tie(0);
25     memset(dp,-1,sizeof dp);
26     ll ans;
27     while(scanf("%lld",&n)!=EOF){
28         ans = solve(4,n);
29         cout << "There " << (ans==1? "is only ":"are ") << ans << (ans==1? "
30             way":" ways") << " to produce " << n << " cents change." << endl;
31     }
32 }
33
```

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  #define EPS 1e-2
4
5  typedef long long int ll;
6  int M[] = {10000,5000,2000,1000,500,200,100,50,20,10,5};
7
8  ll dp[11][40100];
9  ll n;
10
11 ll solve(ll current, ll sum){
12     if(sum==0) return dp[current][sum] = 1LL;
13     if(current < 0 or sum < 0) return 0LL;
14
15     if(dp[current][sum]!=-1) return dp[current][sum];
16
17     ll ans = solve(current,sum-M[current]) + solve(current-1,sum);
18
19     return dp[current][sum] = ans;
20 }
21
22 }
23
24 main(){
25     float aux;
26     memset(dp,-1,sizeof dp);
27     while(cin >> aux and aux!= 0.00){
28         n = (ll)(aux*100);
29         if(fabs(aux*100 - n) > EPS)
30             n++;
31         printf("%6.2f %16lld\n", aux, solve(10,n));
32     }
33 }
34
35 }
```

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  typedef long long int ll;
5
6  ll BIT[100100];
7  int n;
8
9  inline void update(ll index){
10
11     for(;index<=n;index += index &(-index))
12         BIT[index] += 1;
13 }
14 inline ll query(ll index){
15     ll ans =0;
16     for(;index>0;index -= index & (-index))
17         ans += BIT[index];
18     return ans;
19 }
20
21
22 main(){
23     vector<ll> num;
24     scanf("%d",&n);
25     ll ans = 0,aux;
26     for(int i=1;i<=n;i++){
27         scanf("%lld",&aux);
28         num.push back(aux);
29     }
30     for(int i=num.size()-1;i>=0;i--){
31         update(num[i]);
32         ans += query(num[i]-1);
33     }
34     printf("%lld\n",ans);
35 }
36
37
```

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  vector<vector<int> > Grafo(205);
5  int cores[205];
6  bool visitados[205];
7  int n,m;
8
9  bool bfs bi(){
10     memset(visitados,false,sizeof(visitados));
11     memset(cores,-1,sizeof(cores));
12     queue<int> F;
13     F.push(0);
14     cores[0] = false;
15     while(!F.empty()){
16         int aux = F.front();
17         F.pop();
18         if(!visitados[aux]){
19             visitados[aux] = true;
20             for(int i=0;i<Grafo[aux].size();i++){
21                 if(cores[Grafo[aux][i]]==-1)
22                     cores[Grafo[aux][i]] = 1-cores[aux];
23                 else if(cores[Grafo[aux][i]]==cores[aux])
24                     return false;
25                 cores[Grafo[aux][i]] = 1-cores[aux];
26                 F.push(Grafo[aux][i]);
27             }
28         }
29     }
30     return true;
31 }
32
33 main(){
34     int i,j,aux,from,to;
35
36     while(scanf("%d",&n) and n){
37         cin >> m;
38         for(i=0;i<n;i++){Grafo[i].clear();}
39         for(i=0;i<m;i++){
40             scanf("%d %d",&from,&to);
41             Grafo[from].push back(to);
42         }
43         if(bfs bi())
44             printf("BICOLORABLE.\n");
45         else
46             printf("NOT BICOLORABLE.\n");
47     }
48 }
49
50 }
```

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  #define EPS 1e-6
4
5  typedef long long int ll;
6
7
8  main(){
9      int p,q,r,s,t,u;
10
11
12      while(cin >> p >> q >> r >> s >> t >> u){
13          double ini= 0.000000,fim = 1.000000, mid = 0.500000;
14          bool achou = false;
15
16          while(ini<=fim){
17              double ans = p*exp(-mid)+q*sin(mid)+r*cos(mid)+s*tan(mid)+t*(mid*mid)+u;
18              if(fabs(ans) <= EPS){
19                  cout << fixed << setprecision(4) << mid << endl;
20                  achou = true;
21                  break;
22              }
23              else if(ans < EPS)
24                  fim = mid - 0.000000001;
25              else
26                  ini = mid + 0.000000001;
27              mid = (ini+fim)/2;
28          }
29          if(!achou)
30              cout << "No solution\n";
31      }
32
33
34 }
35
```

```

1  #include <bits/stdc++.h>
2  using namespace std;
3
4  typedef long long int ll;
5
6
7  ll D[10100];
8  ll n,num,X,mid;
9
10 ll binary_search(ll lo, ll hi){
11     mid = (lo+hi)/2;
12     if(lo > hi or mid == X) return -1;
13     if(num==D[mid]) return mid;
14     else if(num < D[mid]) return binary_search(lo,mid-1);
15     else if(num > D[mid]) return binary_search(mid+1,hi);
16 }
17
18 main(){
19     ios base::sync with stdio(0);
20     cin.tie(0);
21     ll y,ans,ans1,aux;
22
23     while(cin >> n){
24         for(int i=0;i<n;i++){
25             cin >> D[i];
26         }
27         cin >> y;
28         sort(D,D+n);
29         ans = -1;
30         ans1 = -1;
31         for(X=0;X<n;X++){
32             num = y - D[X];
33             aux = -1;
34             if(num < D[X])
35                 aux = binary_search(0,X-1);
36             if(aux != -1){
37                 if(ans == -1){
38                     ans = aux;
39                     ans1 = X;
40                 }
41                 else{
42                     if(abs(D[X]-D[aux]) < abs(D[ans]-D[ans1])){
43                         ans = aux;
44                         ans1 = X;
45                     }
46                 }
47             }
48         }
49         if(ans <= ans1)
50             cout << "Peter should buy books whose prices are " << D[ans] << " and " << D[ans1] << "." << endl;
51         else
52             cout << "Peter should buy books whose prices are " << D[ans1] << " and " << D[ans] << "." << endl;
53         cout << endl;
54     }
55 }
56

```

```
1  import java.math.BigInteger;
2  import java.util.Scanner;
3
4  public class Main{
5
6      public Main(){}
7
8      public static String reverse(String a){
9          String aux = "";
10
11          for(int i=a.length()-1;i>=0;i--){
12              aux = aux.concat(a.substring(i,i+1));
13          }
14          return aux;
15      }
16
17      public static void main(String[] args){
18          long n;
19          Scanner ler = new Scanner(System.in);
20          BigInteger a,b;
21          String s1,s2,aux,aux2;
22
23          n = ler.nextLong();
24
25          for(long i = 0;i< n;i++){
26              a = ler.nextBigInteger();
27              b = ler.nextBigInteger();
28
29              s1 = reverse(a.toString());
30              s2 = reverse(b.toString());
31              a = new BigInteger(s1);
32              b = new BigInteger(s2);
33
34              s1 = reverse(a.add(b).toString());
35              a = new BigInteger(s1);
36              System.out.println(a);
37          }
38      }
39  }
40
41  }
42
```



```
1
2 import java.math.BigInteger;
3 import java.util.Scanner;
4
5 public class Main{
6
7     public Main(){
8     }
9
10    public static void main(String[] args){
11
12        BigInteger b,ans;
13        int n=-1,a=-1,cont=1;
14        Scanner ler = new Scanner(System.in);
15
16        while(n!=0 && a!=0){
17            n = ler.nextInt();
18            a = ler.nextInt();
19            if(n==0 && a==0)
20                break;
21            ans = BigInteger.ZERO;
22            for(int i = 0;i<n;i++){
23                b = ler.nextBigInteger();
24                ans = ans.add(b);
25            }
26            b = ans.divide(BigInteger.valueOf(a));
27            System.out.println("Bill #"+cont+" costs "+ans.toString()+" : each
28            friend should pay "+b.toString()+"\n");
29            cont++;
30        }
31    }
32 }
33
34 }
35
```

```
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
3  using namespace std;
4  typedef long long int ll;
5
6  ll BIT[1005][1005], n, m;
7
8  ll next(ll index){
9      return index & (-index);
10 }
11
12 void update(ll x, ll y, ll value)
13 {
14     ll x1, y1;
15     x1 = x;
16     while(x1 <= n)
17     {
18         y1 = y;
19         while(y1 <= m)
20         {
21             BIT[x1][y1] += value;
22             y1 += next(y1);
23         }
24         x1 += next(x1);
25     }
26 }
27
28
29 ll query(ll x, ll y)
30 {
31     ll x1, y1, ans = 0;
32     x1 = x;
33
34     while(x1)
35     {
36         y1 = y;
37         while(y1)
38         {
39             ans += BIT[x1][y1];
40             y1 -= next(y1);
41         }
42         x1 -= next(x1);
43     }
44
45     return ans;
46 }
47
48
49 main()
50 {
51     ios base::sync with stdio(0);
52     cin.tie(0);
53     ll x, y, p, q, num;
54     ll x1, y1, x2, y2;
55     ll ax1, ay1, ax2, ay2;
56     char c;
57
58     while(cin >> n >> m >> p && (n+m+p))
59     {
60         memset(BIT, 0, sizeof BIT);
61
62         cin >> q;
63         while(q-->0)
64         {
65             cin.ignore();
66             cin >> c;

```

```
67
68     if (c == 'A')
69     {
70         cin >> num >> x >> y;
71         update(x+1, y+1, num);
72     }
73     else if (c == 'P')
74     {
75         cin >> ax1 >> ay1 >> ax2 >> ay2;
76         ax1++; ay1++; ax2++; ay2++;
77
78         ll ans;
79         if (ax1 > ax2 and ay1 < ay2){
80             x1 = ax2;
81             y1 = ay1;
82             x2 = ax1;
83             y2 = ay2;
84         }
85         else if (ax1 < ax2 and ay1 > ay2){
86             x1 = ax1;
87             y1 = ay2;
88             x2 = ax2;
89             y2 = ay1;
90         }
91         else if (ax1 >= ax2 and ay1 >= ay2){
92             x1 = ax2;
93             y1 = ay2;
94             x2 = ax1;
95             y2 = ay1;
96         }
97         else{
98             x1 = ax1; y1 = ay1;
99             x2 = ax2; y2 = ay2;
100         }
101
102         //cout << x1 << " " << y1 << " " << x2 << " " << y2 << endl;
103         ans = query(x2, y2) - query(x2, y1-1) - query(x1-1, y2) + query(x1-1, y1-1);
104         cout << ans*p << endl;
105     }
106 }
107
108 cout << endl;
109 }
110 }
111
```

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  typedef long long int ll;
4  typedef struct{
5      ll maxsum, sum, leftsum, rightsum;
6  }data;
7
8  vector<data>st;
9  vector<ll>vet;
10 data aux;
11 ll n;
12
13 ll left(ll p){ return (p << 1); }
14 ll right(ll p){ return (p << 1) + 1; }
15
16 void build(ll p, ll l, ll r)
17 {
18     if (l == r)
19         st[p].maxsum = st[p].sum = st[p].leftsum = st[p].rightsum = vet[l];
20     else
21     {
22         ll p1 = left(p);
23         ll p2 = right(p);
24         ll mid = (l+r)/2;
25
26         build(p1, l, mid);
27         build(p2, mid+1, r);
28
29         st[p].sum = st[p1].sum + st[p2].sum;
30         st[p].leftsum = max(st[p1].leftsum, st[p1].sum + st[p2].leftsum);
31         st[p].rightsum = max(st[p2].rightsum, st[p2].sum + st[p1].rightsum);
32
33         ll m1 = max(st[p].leftsum, st[p].rightsum);
34         ll m2 = max(st[p1].maxsum, st[p2].maxsum);
35         st[p].maxsum = max(st[p1].rightsum + st[p2].leftsum, max(m1, m2));
36     }
37 }
38
39 void update(ll p, ll l, ll r, ll a, ll value)
40 {
41     if (a > r || a < l) return;
42
43     if (l == r and l == a)
44     {
45         st[p].sum = st[p].maxsum = st[p].leftsum = st[p].rightsum = value;
46         return;
47     }
48
49     ll p1 = left(p), p2 = right(p);
50     ll mid = (l+r)/2;
51
52     update(p1, l, mid, a, value);
53     update(p2, mid+1, r, a, value);
54
55     st[p].sum = st[p1].sum + st[p2].sum;
56     st[p].leftsum = max(st[p1].leftsum, st[p1].sum + st[p2].leftsum);
57     st[p].rightsum = max(st[p2].rightsum, st[p2].sum + st[p1].rightsum);
58
59     ll m1 = max(st[p].leftsum, st[p].rightsum);
60     ll m2 = max(st[p1].maxsum, st[p2].maxsum);
61     st[p].maxsum = max(st[p1].rightsum + st[p2].leftsum, max(m1, m2));
62 }
63
64 data query(ll p, ll l, ll r, ll a, ll b)
65 {
66     if (a > r || b < l)

```

```

67     return aux;
68
69     if (l >= a and r <= b)
70         return st[p];
71
72     ll p1 = left(p), p2 = right(p);
73     ll mid = (l+r)/2;
74
75     data res1 = query(p1, l, mid, a, b);
76     data res2 = query(p2, mid+1, r, a, b);
77
78     data result;
79     result.sum = res1.sum + res2.sum;
80     result.leftsum = max(res1.leftsum, res1.sum + res2.leftsum);
81     result.rightsum = max(res2.rightsum, res2.sum + res1.rightsum);
82
83     ll m1 = max(result.leftsum, result.rightsum);
84     ll m2 = max(res1.maxsum, res2.maxsum);
85
86     result.maxsum = max(res1.rightsum + res2.leftsum, max(m1, m2));
87     return result;
88 }
89
90 main()
91 {
92     ll num, m, c, x, y;
93     cin >> n;
94     aux.sum = aux.maxsum = aux.leftsum = aux.rightsum = INT_MIN;
95     st.assign(4*n, aux);
96     vet.clear();
97
98     for (ll i = 0; i < n; i++)
99     {
100         cin >> num;
101         vet.push back(num);
102     }
103
104     build(1, 0, n-1);
105
106     cin >> m;
107     while(m--)
108     {
109         cin >> c >> x >> y;
110
111         if (c == 0)
112             update(1, 0, n-1, x-1, y);
113         else{
114             data res = query(1, 0, n-1, x-1, y-1);
115             cout << res.maxsum << endl;
116         }
117     }
118 }
119

```

```

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  #define endl '\n'
3  #define F first
4  #define S second
5  using namespace std;
6  typedef int ll;
7  typedef pair<ll, ll> ii;
8  map< ll, ii >mapa;
9  map< ll, ii >::iterator it, ant;
10 vector<ii>positions;
11 ll n, dx[] = {1, -1, 0, 0}, dy[] = {0, 0, 1, -1};
12 ll distances[110], dp[15][15][110][110];
13 char graph[15][15];
14 bool visited[15][15], flag;
15
16 inline ll solve(ll x, ll y, ll food, ll quan)
17 {
18     //cout << x << " " << y << " " << food << " " << quan << " " <<
19     positions.size() << endl;
20     if(food == positions.size())
21         return 1LL;
22     if(quan > distances[food])
23         return 0LL;
24
25     char letra = (char)(food + 'A');
26     if(graph[x][y] == letra)
27         return dp[x][y][food][quan] = solve(positions[food].F, positions[food].S,
28         food+1, 0);
29
30     if(dp[x][y][food][quan] != -1)
31         return dp[x][y][food][quan];
32
33     ll ans = 0LL;
34     for(ll i = 0; i < 4; i++)
35     {
36         ll nx = x + dx[i];
37         ll ny = y + dy[i];
38
39         if(nx < 0 || nx >= n || ny < 0 || ny >= n || graph[nx][ny] == '#') continue;
40
41         if(isalpha(graph[nx][ny]))
42         {
43             if(positions[food].F == nx && positions[food].S == ny)
44                 ans += solve(nx, ny, food+1, 0);
45
46             if(graph[nx][ny] < graph[positions[food].F][positions[food].S])
47                 ans += solve(nx, ny, food, quan+1);
48         }
49         else
50             ans += solve(nx, ny, food, quan+1);
51     }
52
53     return dp[x][y][food][quan] = ans%20437;
54 }
55
56 inline ll bfs(ll x1, ll y1, ll x2, ll y2)
57 {
58     memset(visited, false, sizeof visited);
59     queue< pair<ll, ii> >pq;
60     pq.push( {0, {x1, y1}} );
61
62     while(!pq.empty())
63     {
64         ll x = pq.front().S.F;
```

```
65     ll y = pq.front().S.S;
66     ll w = pq.front().F;
67     pq.pop();
68
69     if(x == x2 && y == y2)
70         return w;
71
72     visited[x][y] = true;
73
74     for(ll i = 0; i < 4; i++)
75     {
76         ll nx = x + dx[i];
77         ll ny = y + dy[i];
78
79         if(nx < 0 || nx >= n || ny < 0 || ny >= n || graph[nx][ny] == '#')
80             continue;
81
82         if(!visited[nx][ny]){
83             if(isalpha(graph[nx][ny]))
84             {
85                 if(graph[nx][ny] <= graph[x2][y2]){
86                     pq.push( {w+1, {nx, ny}} );
87                     visited[nx][ny] = true;
88                     continue;
89                 }
90                 else
91                     continue;
92             }
93
94             pq.push( { w+1, {nx, ny}} );
95             visited[nx][ny] = true;
96         }
97     }
98
99     return -1;
100 }
101
102 inline ll calculate()
103 {
104     ll sum = 0;
105     positions.clear();
106     ant = mapa.begin();
107     ll indice = 1;
108     for(it = mapa.begin(); it != mapa.end(); it++)
109     {
110         positions.push back( {it->S.F, it->S.S} );
111         if(it != mapa.begin())
112         {
113             ll res = bfs(ant->S.F, ant->S.S, it->S.F, it->S.S);
114             distances[indice++] = res;
115             sum += res;
116             if(res == -1){
117                 flag = false;
118                 break;
119             }
120         }
121         ant = it;
122     }
123
124     return sum;
125 }
126
127 inline void read()
128 {
129     for(ll i = 0; i < n; i++){
```

```
130         cin.ignore();
131
132         for(ll j = 0; j < n; j++){
133             cin >> graph[i][j];
134
135             if(isalpha(graph[i][j]))
136             {
137                 ll num = (ll)graph[i][j] - 'A';
138                 mapa[num] = {i, j};
139             }
140         }
141     }
142 }
143
144 main()
145 {
146     ll s, t, cases=1;
147     while(cin >> n && n)
148     {
149         flag = true;
150         mapa.clear();
151
152         read();
153         ll tot = calculate();
154
155         cout << "Case " << cases++ << ": ";
156         if(!flag){
157             cout << "Impossible" << endl;
158             continue;
159         }
160
161         memset(dp, -1, sizeof dp);
162         ll res = solve(positions[0].F, positions[0].S , 1, 0);
163         cout << tot << " " << res << endl;
164     }
165 }
166
```

```
1  #include <bits/stdc++.h>
2
3  using namespace std;
4  vector<vector<int>> >grafo(10005);
5  vector<int>inimigos, amigos, degree(10005), res;
6  int n, m, s, cnt;
7
8  void khan()
9  {
10     queue<int>p;
11
12     for (int i = 1; i <= n; i++)
13         if (degree[i] == 0)
14             p.push(i);
15
16     while(!p.empty())
17     {
18         int u = p.front();
19         p.pop();
20         res.push back(u);
21
22         for (int i = 0; i < grafo[u].size(); i++)
23             if (--degree[grafo[u][i]] == 0)
24                 p.push(grafo[u][i]);
25
26         cnt++;
27     }
28 }
29
30 void reset()
31 {
32     inimigos.clear();
33     amigos.clear();
34     res.clear();
35     inimigos.push back(0);
36     amigos.push back(s);
37     fill(degree.begin(), degree.end(), 0);
38     cnt = 0;
39 }
40
41 main()
42 {
43     int a, b;
44     while(cin >> n >> m >> s && (n+m+s))
45     {
46         reset();
47         for (int i = 1; i <= n; i++){
48             grafo[i].clear();
49             cin >> a;
50             inimigos.push back(a);
51         }
52
53         for (int i = 1; i <= n; i++)
54         {
55             cin >> b;
56             amigos.push back(b);
57         }
58
59         while(m--)
60         {
61             cin >> a >> b;
62             grafo[a].push back(b);
63             degree[b]++;
64         }
65
66         khan();
```

```
67
68     bool flag = true;
69
70     if (cnt != n) flag = false;
71
72     for(int i = 0; i < res.size() && flag; i++)
73     {
74         //cout << amigos[0] << " " << inimigos[res[i]] << " " << res[i] << endl;
75         if(amigos[0] > inimigos[res[i]])
76             amigos[0] += amigos[res[i]];
77         else
78             flag = false;
79     }
80
81     if (flag) cout << "possivel" << endl;
82     else cout << "impossivel" << endl;
83 }
84 }
85
```

```
1  #include <bits/stdc++.h>
2  #define tam arvore 20
3  using namespace std;
4  typedef long long int ll;
5  vector<vector< ll > > grafo(100005);
6  int n, parent[100005], height[100005];
7  ll length[100005], lca[100005][25], lulu[100005];
8
9  void build lca(int u, int v)
10 {
11     parent[u] = v;
12     lulu[u] = lulu[parent[u]] + length[u];
13     height[u] = height[v] + 1;
14     lca[u][0] = v;
15
16     for(int j = 1; j < tam arvore; j++)
17         lca[u][j] = lca[lca[u][j-1]][j-1];
18
19     for(ll i = 0; i < grafo[u].size(); i++)
20     {
21         if(grafo[u][i] != v)
22             build lca(grafo[u][i], u);
23     }
24 }
25
26 ll get lca(int x, int y)
27 {
28     if(height[x] > height[y])
29         swap(x, y);
30
31     int dist = height[y] - height[x];
32
33     for(int j = 0; j < tam arvore; j++)
34         if(dist & (1 <= j))
35             y = lca[y][j];
36
37     if(x == y)
38         return x;
39
40     for(int j = tam arvore-1; j >= 0; j--)
41         if(lca[x][j] != lca[y][j])
42             x = lca[x][j], y = lca[y][j];
43
44     return lca[x][0];
45 }
46
47 main()
48 {
49     ll a, l, s, t;
50     while(scanf("%d", &n) && n)
51     {
52         for(ll i = 0; i < n; i++)
53             grafo[i].clear();
54
55         parent[0] = 0;
56         height[0] = 0;
57         lulu[0] = 0;
58
59         for(int i = 1; i <= n-1; i++)
60         {
61             height[i] = 1;
62             parent[i] = i;
63             lulu[i] = 0;
64
65             scanf("%lld %lld", &a, &l);
66             length[i] = l;
```

```
67         grafo[a].push back( i );
68         grafo[i].push back( a );
69     }
70
71     int q;
72     scanf("%d", &q);
73     build lca(0, 0);
74
75     for(int i = 0; i < q; i++)
76     {
77         scanf("%lld %lld", &s, &t);
78
79         if(i) printf(" ");
80         printf("%lld", lulu[s] + lulu[t] - 2*lulu[get lca(s, t)]);
81     }
82     printf("\n");
83 }
84 }
85
```

```

1  #include <bits/stdc++.h>
2  #define endl '\n'
3  #define tam arvore 25
4  using namespace std;
5  typedef long long int ll;
6  typedef pair<ll, ll> ii;
7  vector< vector< ii > > grafo(20005);
8  vector< pair<ll, ll> > edges;
9  ll n, parent[20005], height[20005], lulu[20005][28], lca[20005][28], tot, use;
10
11 ll findset(ll x){
12     if (x != parent[x])
13         parent[x] = findset(parent[x]);
14
15     return parent[x];
16 }
17
18 void UNION(ll x, ll y){
19     parent[x] = parent[y];
20 }
21
22 void kruskal()
23 {
24     ll pu, pv;
25     for (ll i = 0; i < edges.size(); i++)
26     {
27         ll u = edges[i].second.first;
28         ll v = edges[i].second.second;
29         ll w = edges[i].first;
30         pu = findset(u);
31         pv = findset(v);
32
33         if (pu != pv)
34         {
35             grafo[u].push back( {v, w} );
36             grafo[v].push back( {u, w} );
37
38             UNION(pu, pv);
39             use = findset(pu);
40         }
41     }
42 }
43
44 void build lca(ll u, ll v, ll w)
45 {
46     parent[u] = v;
47     lulu[u][0] = w;
48     height[u] = height[v] + 1;
49     lca[u][0] = v;
50
51     for(ll j = 1; j < tam arvore; j++){
52         lca[u][j] = lca[lca[u][j-1]][j-1];
53         lulu[u][j] = min(lulu[u][j-1], lulu[lca[u][j-1]][j-1]);
54     }
55
56     for(ll i = 0; i < grafo[u].size(); i++)
57         if(grafo[u][i].first != v)
58             build lca(grafo[u][i].first, u, grafo[u][i].second);
59 }
60
61 ll get lca(ll x, ll y)
62 {
63     if(height[x] > height[y])
64         swap(x, y);
65
66     ll dist = height[y] - height[x];

```



```

67     tot = LLONG MAX;
68
69     for(ll j = 0; j < tam arvore; j++)
70         if(dist & (1LL << j)){
71             tot = min(tot, lulu[y][j]);
72             y = lca[y][j];
73         }
74
75     if(x == y)
76         return x;
77
78     for(ll j = tam arvore; j >= 0; j--)
79         if(lca[x][j] != lca[y][j]){
80             tot = min(tot, min(lulu[x][j], lulu[y][j]));
81             x = lca[x][j], y = lca[y][j];
82         }
83
84     tot = min(tot, min(lulu[x][0], lulu[y][0]));
85     return lca[x][0];
86 }
87
88 void reset();
89 main()
90 {
91     ll m, s, a, b, p;
92     while(cin >> n >> m >> s)
93     {
94         reset();
95
96         while(m--)
97         {
98             cin >> a >> b >> p;
99             edges.push back( {p, {a, b} } );
100         }
101
102         sort(edges.rbegin(), edges.rend());
103         kruskal();
104         build lca(use, use, LLONG MAX);
105
106         while(s--)
107         {
108             cin >> a >> b;
109             ll ohcomoelevai = get lca(a,b);
110             cout << tot << endl;
111         }
112     }
113 }
114
115 void reset()
116 {
117     for(ll i = 0; i <= n; i++){
118         grafo[i].clear();
119         parent[i] = i;
120         height[i] = 0;
121
122         for(ll j = 0; j < tam arvore; j++)
123             lulu[i][j] = INT MAX;
124     }
125     edges.clear();
126 }
127

```

```

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 }
98
99 if (board 1[x1-1][y1] != '#' && board 2[x2-1][y2] == '#') {
100     if (visited[x1-1][y1][x2][y2] == false) {
101         q.push ({{{x1-1, y1}, {x2, y2}}, moves+1});
102         visited[x1-1][y1][x2][y2] = true;
103     }
104 }
105 }
106 }
107
108 if (board 1[x1-1][y1] != '#' && board 2[x2-1][y2] != '#') {
109     if (visited[x1-1][y1][x2-1][y2] == false) {
110         q.push ({{{x1-1, y1}, {x2-1, y2}}, moves+1});
111         visited[x1-1][y1][x2-1][y2] = true;
112     }
113 }
114 }
115 }
116 }
117
118 // left
119 if (validPosition (x1, y1-1, x2, y2-1)) {
120     if (board 1[x1][y1-1] == '#' && board 2[x2][y2-1] != '#') {
121         if (visited[x1][y1][x2][y2-1] == false) {
122             q.push ({{{x1, y1}, {x2, y2-1}}, moves+1});
123             visited[x1][y1][x2][y2-1] = true;
124         }
125     }
126 }
127 }
128 }
129
130 if (board 1[x1][y1-1] != '#' && board 2[x2][y2-1] == '#') {
131     if (visited[x1][y1-1][x2][y2] == false) {
132         q.push ({{{x1, y1-1}, {x2, y2}}, moves+1});
133         visited[x1][y1-1][x2][y2] = true;
134     }
135 }
136 }
137 }
138
139 if (board 1[x1][y1-1] != '#' && board 2[x2][y2-1] != '#') {
140     if (visited[x1][y1-1][x2][y2-1] == false) {
141         q.push ({{{x1, y1-1}, {x2, y2-1}}, moves+1});
142         visited[x1][y1-1][x2][y2-1] = true;
143     }
144 }
145 }
```

```
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) {
```

```
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) {
59         reset ();
60         read ();
61         solve ();
62     }
63 }
```



```
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  #define endl "\n"
3  using namespace std;
4  typedef long long int ll;
5
6  vector<vector<ll> > grafo(55);
7  vector<ll> dfs num, dfs low, dfs parent;
8  ll counter, root, rootChildren, cont;
9
10 void reset(int v)
11 {
12     counter = 0;
13     dfs num.assign(v+1, -1);
14     dfs low.assign(v+1, 0);
15     dfs parent.assign(v+1, 0);
16 }
17
18 void dfs(int u)
19 {
20     dfs low[u] = dfs num[u] = counter++;
21     for (int i = 0; i < grafo[u].size(); i++)
22     {
23         int v = grafo[u][i];
24         if (dfs num[v] == -1)
25         {
26             dfs parent[v] = u;
27             if (u == root) rootChildren++;
28             dfs(v);
29
30             //Bridges
31             if (dfs low[v] > dfs num[u])
32                 cont++;
33             //Articulation Points
34             //if (dfs low[v] >= dfs num[u])
35             //    points++
36
37             dfs low[u] = min(dfs low[u], dfs low[v]);
38         }
39         else if (v != dfs parent[u])
40             dfs low[u] = min(dfs low[u], dfs num[v]);
41     }
42 }
43
44 main()
45 {
46     ll c, p, x, y;
47     while(cin >> c >> p)
48     {
49         for (ll i = 0; i <= c; i++)
50             grafo[i].clear();
51         reset(c);
52         while(p-->0)
53         {
54             cin >> x >> y;
55             grafo[x-1].push back(y-1);
56             grafo[y-1].push back(x-1);
57         }
58         counter = 0;
59         cont = 0;
60         for (ll i = 0; i < c; i++)

```



```
67     {
68         if (dfs num[i] == -1)
69         {
70             root = i;
71             rootChildren = 0;
72             dfs(i);
73         }
74     }
75
76     cout << cont << endl;
77 }
78 }
79
80
```

```
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
20     return (y != 0 ? gcd (y, x%y) : x);
21 }
22
23 ll lcm (ll x, ll y) {
24
25     return (x / gcd (x, y) * y);
26 }
27
28 main () {
29
30     ll n; scanf ("%lld", &n);
31
32     if (n == 1 or n == 2)
33         printf ("-1\n");
34
35     else if (n & 1)
36         printf ("%lld %lld\n", (n*n-1) >> 1, (n*n+1) >> 1);
37
38     else
39         printf ("%lld %lld\n", (n >> 1) * (n >> 1) - 1, (n >> 1) * (n >> 1) + 1);
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 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 = 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         printf ("%06lld\n", (ans.v[0][1] * m + ans.v[0][0]) % mod);
49     }
50 }
```

```
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 }
60
61
62
63
64
65
66
67
```



```

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  #define EPS 1e-6
3  #define endl '\n'
4  using namespace std;
5  double p, q, r, s, t, u;
6
7  double solve(double x){
8      return p*exp(-x) + q*sin(x) + r*cos(x) + s*tan(x) + t*x*x + u;
9  }
10
11 main()
12 {
13     //p = r = u = 1;
14     //q = s = t = -1;
15     //cout << solve(0.500) << endl;
16     //cout << solve(0.600) << endl;
17     //cout << EPS << endl;
18     while(cin >> p >> q >> r >> s >> t >> u)
19     {
20         double lo = 0.0, hi = 1.0, mid;
21         double ans = -1;
22
23         for (int i = 0; i < 100; i++)
24         {
25             mid = (lo+hi)/2;
26
27             if(solve(mid) > 0.0){
28                 lo = mid;
29             }
30             else{
31                 hi = mid;
32             }
33         }
34
35         if(fabs(solve(mid)) <= EPS)
36             cout << fixed << setprecision(4) << mid << endl;
37         else
38             cout << "No solution" << endl;
39     }
40 }
41
42
```

```

1  #include <bits/stdc++.h>
2  #define endl '\n'
3  using namespace std;
4  typedef long long int ll;
5  ll res[11], dp[10][3][3][25], numero;
6  vector<ll> va, vb, vc;
7
8  ll solve(ll pos, bool isequal, bool started, ll sum)
9  {
10     //cout << pos << " " << isequal << " " << started << " " << sum << endl;
11     if(pos == vc.size())
12         return sum;
13
14     ll &lulu = dp[pos][isequal][started][sum];
15     if(lulu != -1)
16         return lulu;
17
18     ll ans = 0LL;
19
20     if(isequal){
21         for(ll i = 0; i <= vc[pos]; i++){
22             {
23                 ll aux = sum;
24
25                 if(numero == 0 && started && i == 0)
26                     aux++;
27                 else if(i == numero && numero)
28                     aux++;
29
30                 ans += solve(pos+1, i == vc[pos], started|(i != 0), aux);
31             }
32         }
33     }
34     else{
35         for(ll i = 0; i <= 9; i++){
36             {
37                 ll aux = sum;
38
39                 if(numero == 0 && started && i == 0)
40                     aux++;
41                 else if(i == numero && numero)
42                     aux++;
43
44                 ans += solve(pos+1, false, started|(i != 0), aux);
45             }
46         }
47
48         return lulu = ans;
49     }
50
51     main()
52     {
53         string sa, sb;
54         ll a, b;
55
56         while(cin >> a >> b && (a+b))
57         {
58             a--;
59             va.clear();
60             vb.clear();
61             memset(res, 0, sizeof res);
62
63             sa = to_string(a);
64             sb = to_string(b);
65
66             for(ll i = 0; i < sa.size(); i++){

```

```
67         ll num = (ll)sa[i] - '0';
68         va.push_back(num);
69     }
70
71     for(ll i = 0; i < sb.size(); i++){
72         ll num = (ll)sb[i] - '0';
73         vb.push_back(num);
74     }
75
76     vc = va;
77     for(ll i = 0; i <= 9; i++)
78     {
79         memset(dp, -1, sizeof dp);
80         numero = i;
81         res[i] -= solve(0, true, false, 0);
82     }
83
84     vc = vb;
85     for(ll i = 0; i <= 9; i++)
86     {
87         memset(dp, -1, sizeof dp);
88         numero = i;
89         res[i] += solve(0, true, false, 0);
90     }
91
92     for(ll i = 0; i < 10; i++)
93     {
94         if(i)
95             cout << " ";
96         cout << res[i];
97     }
98     cout << endl;
99 }
100 }
101
```

```
1  #include <bits/stdc++.h>
2  #define endl '\n'
3  using namespace std;
4  typedef long long int ll;
5  ll n, d, m, p, dp[2005][2005];
6  bool flag;
7  vector<ll>manutencao, venda;
8
9  ll solve(ll pos, ll age)
10 {
11     if(pos == n)
12         return 0LL;
13
14     if(dp[pos][age] != -1)
15         return dp[pos][age];
16
17     ll ans = INT MAX;
18
19     if(age == m)
20         ans = min(ans, solve(pos+1, 1) - venda[age-1] + p + manutencao[0]);
21     else
22     {
23         ans = min(ans, solve(pos+1, age+1) + manutencao[age]);
24         ans = min(ans, solve(pos+1, 1) - venda[age-1] + p + manutencao[0]);
25     }
26
27     return dp[pos][age] = ans;
28 }
29
30 void print(ll pos, ll age)
31 {
32     if(pos == n) return;
33
34     ll ans = INT MAX;
35
36     if(age == m){
37         if(solve(pos+1, 1) - venda[age-1] + p + manutencao[0] == dp[pos][age]){
38             if(flag)
39                 cout << " ";
40             cout << pos+1;
41             flag = true;
42             print(pos+1, 1);
43         }
44     }
45     else
46     {
47         if(solve(pos+1, 1) - venda[age-1] + p + manutencao[0] == dp[pos][age]){
48             if(flag)
49                 cout << " ";
50             cout << pos+1;
51             flag = true;
52             print(pos+1, 1);
53         }
54         else if(solve(pos+1, age+1) + manutencao[age] == dp[pos][age]){
55             print(pos+1, age+1);
56         }
57     }
58 }
59
60 main()
61 {
62     ios base::sync with stdio(0);
63     cin.tie(0);
64     ll num;
65     while(cin >> n >> d >> m >> p)
66     {
```

```
67     flag = false;
68     manutencao.clear();
69     venda.clear();
70
71     for (int i = 0; i < m; i++)
72     {
73         cin >> num;
74         manutencao.push back(num);
75     }
76
77     for (int i = 0; i < m; i++)
78     {
79         cin >> num;
80         venda.push back(num);
81     }
82
83     memset(dp, -1, sizeof dp);
84     cout << solve(0, d) << endl;
85     print(0, d);
86     if(!flag)
87         cout << 0;
88     cout << endl;
89 }
90 }
91
```

```
1  #include <bits/stdc++.h>
2
3  using namespace std;
4  typedef long long int ll;
5  ll mat[105][105], n, maxi, maxStart, maxEnd, maxSum, leftBound, rightBound,
   upBound, lowBound;
6  vector<ll>vet;
7
8  void kadane()
9  {
10     ll currentStart, maxSoFar;
11
12     maxi = currentStart = maxSoFar = 0;
13     maxStart = maxEnd = -1;
14
15     for(int i=0; i < vet.size(); i++){
16         maxSoFar += vet[i];
17         if(maxSoFar < 0){
18             maxSoFar = 0;
19             currentStart = i+1;
20         }
21         if(maxi < maxSoFar){
22             maxStart = currentStart;
23             maxEnd = i;
24             maxi = maxSoFar;
25         }
26     }
27 }
28
29 ll solve()
30 {
31     for (ll left = 0; left < n; left++)
32     {
33         vet.assign(n, 0);
34         for (ll right = left; right < n; right++)
35         {
36             for (ll i = 0 ; i < n; i++)
37                 vet[i] += mat[i][right];
38
39             kadane();
40
41             if (maxi > maxSum){
42                 maxSum = maxi;
43                 leftBound = left;
44                 rightBound = right;
45                 upBound = maxStart;
46                 lowBound = maxEnd;
47             }
48         }
49     }
50
51     return maxSum;
52 }
53
54 main()
55 {
56     cin >> n;
57
58     for(ll i = 0 ; i < n; i++)
59         for (ll j = 0; j < n; j++)
60             cin >> mat[i][j];
61
62     cout << solve() << endl;
63 }
64
```

```

1  #include <bits/stdc++.h>
2  #define endl '\n'
3  using namespace std;
4  typedef int ll;
5  vector< pair<ll, ll> >coord;
6  ll n;
7  double dp[18][(1 << 18)];
8
9  inline void reset()
10 {
11     coord.clear();
12
13     for(ll i = 0; i <= n; i++)
14         for(ll j = 0; j <= (1 << (n+1)); j++)
15             dp[i][j] = -1;
16 }
17
18 inline double dist(ll x1, ll y1, ll x2, ll y2){
19     return sqrt( (x1-x2)*(x1-x2) + (y1-y2)*(y1-y2) );
20 }
21
22 inline double solve(ll current, ll mask)
23 {
24     if(mask == (1LL << n)-1)
25         return 0;
26
27     if(dp[current][mask] != -1)
28         return dp[current][mask];
29
30     if(mask & (1LL << current))
31         return dp[current][mask] = solve(current+1, mask);
32
33     double ans = 10555555;
34
35     mask |= (1LL << current);
36     for(ll i = current+1; i < n; i++)
37     {
38         if(i == current) continue;
39         if(!(mask & (1 << i)))
40         {
41             ll opt = mask | (1LL << i);
42             ans = min(ans, solve(current+1, opt) +
43                 dist(coord[current].first, coord[current].second, coord[i].first,
44                     coord[i].second));
45         }
46     }
47
48     return dp[current][mask] = ans;
49 }
50
51 main()
52 {
53     ll t, x, y;
54     scanf("%d", &t);
55
56     while(t--)
57     {
58         scanf("%d", &n);
59         reset();
60
61         for(ll i = 0 ; i < n; i++)
62         {
63             scanf("%d %d", &x, &y);
64             coord.push back( {x, y} );
65         }

```



```
66
67     double res = solve(0, 0);
68     printf("%.2lf\n", res);
69 }
70 }
71
```

```

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 here);
179                 }
180                 p vert.pop back();
181                 p edge.pop back();
182                 p dist.pop back();
183             }
184             vertex v;
185             edge e;
186             int c v=p vert.back();
187             int c e=p edge.back();
188             int At=data[c v].go[c e].from+c lcp-p dist.back();
189
190             p dist.push back(c lcp);
191
192             e.next vert=-1;
193             e.suffix here=p[i];
194             e.from=p[i]+c lcp;
195             e.to=n;
196
197             if(At==data[c v].go[c e].from)
198             {
199                 data[c v].go.push back(e);
200                 p vert.push back(c v);
201                 p edge.push back(data[c v].go.size()-1);
202             }
203             else
204             {
205                 v.go.push back(data[c v].go[c e]);
206                 v.go.back().from=At;
207                 v.go.push back(e);
208                 data.push back(v);
209                 data[c v].go[c e].next vert=data.size()-1;
210                 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
42     for (ll i = 1; i <= len x; ++i) {
43         for (ll j = 1; j <= len y; ++j) {
44             ll ans = 0;
45             dp[i][j] = max (dp[i-1][j], dp[i][j-1]);
46
47             while (i-1-ans >= 0 and j-1-ans >= 0 and x[i-1-ans] == y[j-1-ans]) {
48                 ++ans;
49                 if (ans >= n)
50                     dp[i][j] = max (dp[i][j], dp[i-ans][j-ans] + ans);
51             }
52
53             if (ans >= n)
54                 dp[i][j] = max (dp[i][j], dp[i-ans][j-ans] + ans);
55         }
56     }
57
58     printf ("%lld\n", dp[len x][len y]);
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 }

```

```
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 }
```

---

**Combinação:**  $\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n(n-1)(n-2)\cdots(n-k+1)}{k!}$

$$\binom{n}{k} = \binom{n}{n-k} \qquad \binom{n-1}{k-1} + \binom{n-1}{k} = \binom{n}{k}$$

---

**Permutação com Repetição:**  $P_n^{k,j,\dots,m} = \frac{n!}{k! \cdot j! \cdot \dots \cdot m!}$

---

$$\mathbf{J = P . i . n}$$

**Juros Simples:**

Onde:

**J** = juros

**P** = principal (capital)

**i** = taxa de juros

**n** = número de períodos

---

$$\mathbf{M = P . (1 + i)^n}$$

**Juros Composto:**

$$\mathbf{J = M - P}$$

---

---

**Progressão Aritmética:**  $a_n = a_1 + (n - 1) \cdot r$

$$S_n = \frac{(a_1 + a_n)n}{2}$$

---

**Progressão Geométrica:**  $a_n = a_1 \cdot q^{(n - 1)}$

$$S_n = \frac{a_1(q^n - 1)}{q - 1} \quad S_{(p,q)} = \frac{a_p(1 - q^{q-p+1})}{1 - q}$$

---

**Logarítmos:**

$$\log_a x/y = \log_a x - \log_a y$$

$$\log_a (x * y) = \log_a x + \log_a y$$

$$\log_b a = \frac{\log_c a}{\log_c b}$$

$$\log_a x^m = m * \log_a x$$

---

---

**Área/Perímetro/Volume:**

Área – Círculo:  $\text{PI} * \text{raio} * \text{raio}$

Área – Paralelogramo:  $\text{base} * \text{altura}$

Área – Triângulo:  $(\text{base} * \text{altura}) / 2$

Perímetro – Circunferência:  $2 * \text{PI} * \text{raio}$

Perímetro – Paralelogramo:  $2 * (\text{lado1} + \text{lado2})$

Perímetro – Triângulo:  $\text{lado1} + \text{lado2} + \text{lado3}$

Volume – Esfera:  $(4 * \text{PI} * \text{raio} * \text{raio} * \text{raio}) / 3$

Volume – Paralelepípedo:  $\text{comprimento} * \text{largura} * \text{altura}$

Volume – Pirâmide:  $(\text{Área\_da\_base} * \text{altura}) / 3$

---

### **Divisibilidade por 2**

Um número natural é divisível por 2 quando ele termina em 0, ou 2, ou 4, ou 6, ou 8, ou seja, quando ele é par.

Exemplos:

1) 5040 é divisível por 2, pois termina em 0.

### **Divisibilidade por 3**

Um número é divisível por 3 quando a soma dos valores absolutos dos seus algarismos for divisível por 3.

Exemplo:

234 é divisível por 3, pois a soma de seus algarismos é igual a  $2+3+4=9$ , e como 9 é divisível por 3, então 234 é divisível por 3.

### **Divisibilidade por 4**

Um número é divisível por 4 quando termina em 00 ou quando o número formado pelos dois últimos algarismos da direita for divisível por 4.

Exemplo:

1800 é divisível por 4, pois termina em 00.

4116 é divisível por 4, pois 16 é divisível por 4.

1324 é divisível por 4, pois 24 é divisível por 4.

3850 não é divisível por 4, pois não termina em 00 e 50 não é divisível por 4.

### **Divisibilidade por 5**

Um número natural é divisível por 5 quando ele termina em 0 ou 5.

Exemplos:

1) 55 é divisível por 5, pois termina em 5.

2) 90 é divisível por 5, pois termina em 0.

3) 87 não é divisível por 5, pois não termina em 0 nem em 5.

### **Divisibilidade por 6**

Um número é divisível por 6 quando é divisível por 2 e por 3.

Exemplos:

1) 312 é divisível por 6, porque é divisível por 2 (par) e por 3 (soma: 6).

2) 5214 é divisível por 6, porque é divisível por 2 (par) e por 3 (soma: 12).

3) 716 não é divisível por 6, (é divisível por 2, mas não é

divisível por 3).

### **Divisibilidade por 8**

Um número é divisível por 8 quando termina em 000, ou quando o número formado pelos três últimos algarismos da direita for divisível por 8.

Exemplos:

- 1) 7000 é divisível por 8, pois termina em 000.
- 2) 56104 é divisível por 8, pois 104 é divisível por 8.
- 3) 61112 é divisível por 8, pois 112 é divisível por 8.
- 4) 78164 não é divisível por 8, pois 164 não é divisível por 8.

### **Divisibilidade por 9**

Um número é divisível por 9 quando a soma dos valores absolutos dos seus algarismos for divisível por 9.

Exemplo:

2871 é divisível por 9, pois a soma de seus algarismos é igual a  $2+8+7+1=18$ , e como 18 é divisível por 9, então 2871 é divisível por 9.

### **Divisibilidade por 10**

Um número natural é divisível por 10 quando ele termina em 0.

Exemplos:

- 1) 4150 é divisível por 10, pois termina em 0.
- 2) 2106 não é divisível por 10, pois não termina em 0.

### **Divisibilidade por 11**

Um número é divisível por 11 quando a diferença entre as somas dos valores absolutos dos algarismos de ordem ímpar e a dos de ordem par é divisível por 11.

O algarismo das unidades é de 1ª ordem, o das dezenas de 2ª ordem, o das centenas de 3ª ordem, e assim sucessivamente.

Exemplos:

- 1) 87549

$$Si \text{ (soma das ordens ímpares)} = 9+5+8 = 22$$

$$Sp \text{ (soma das ordens pares)} = 4+7 = 11$$

$$Si-Sp = 22-11 = 11$$

Como 11 é divisível por 11, então o número 87549 é divisível por 11.

- 2) 439087

$$Si \text{ (soma das ordens ímpares)} = 7+0+3 = 10$$

$$Sp \text{ (soma das ordens pares)} = 8+9+4 = 21$$

$$Si-Sp = 10-21$$

Como a subtração não pode ser realizada, acrescenta-se o

menor múltiplo de 11 (diferente de zero) ao minuendo, para que a subtração possa ser realizada:  $10+11 = 21$ . Então temos a subtração  $21-21 = 0$ .

### **Divisibilidade por 12**

Um número é divisível por 12 quando é divisível por 3 e por 4.

Exemplos:

- 1) 720 é divisível por 12, porque é divisível por 3 (soma=9) e por 4 (dois últimos algarismos, 20).
- 2) 870 não é divisível por 12 (é divisível por 3, mas não é divisível por 4).
- 3) 340 não é divisível por 12 (é divisível por 4, mas não é divisível por 3).

### **Divisibilidade por 15**

Um número é divisível por 15 quando é divisível por 3 e por 5.

Exemplos:

- 1) 105 é divisível por 15, porque é divisível por 3 (soma=6) e por 5 (termina em 5).
- 2) 324 não é divisível por 15 (é divisível por 3, mas não é divisível por 5).
- 3) 530 não é divisível por 15 (é divisível por 5, mas não é divisível por 3).

### **Divisibilidade por 25**

Um número é divisível por 25 quando os dois algarismos finais forem 00, 25, 50 ou 75.

Exemplos:

200, 525, 850 e 975 são divisíveis por 25.