```cpp
#include <bits/stdc++.h>
using namespace std;

int n,m;
map<string,int> mapa;
vector<int> parent,r;
vector<int> filhos;
int findSet(int i) {
        return (parent[i] == -1) ? i : parent[i] = findSet(parent[i]);
}
bool isSameSet(int i, int j) {
        return findSet(i) == findSet(j);
}
void unionF(int i,int j){
    if (!isSameSet(i, j)) {
        int x = findSet(i), y = findSet(j);
        if (r[x] > r[y]) {
            filhos[findSet(x)] += filhos[findSet(y)];
            parent[y] = x;
        } else {
            filhos[findSet(y)] += filhos[findSet(x)];
            parent[x] = y;
            if (r[x] == r[y])
                r[y]++;
        }
    }
}
main(){
    int i,j,k;
    string a,b;

    cin >> n;

    for(i=0;i<n;i++){
        cin >> m;
        for(j=0;j<m;j++){
            cin >> a >> b;
            if(!mapa.count(a)){
                int aux = mapa.size();
                mapa[a] = aux;
                parent.push back(-1);
                filhos.push back(1);
                r.push back(0);
            }
            if(!mapa.count(b)){
                int aux = mapa.size();
                mapa[b] = aux;
                parent.push back(-1);
                filhos.push back(1);
                r.push back(0);
            }
            unionF(mapa[a],mapa[b]);
            cout << filhos[findSet(mapa[a])] << endl;
        }
        mapa.clear();
        parent.clear();
        filhos.clear();
        r.clear();
    }
}
```

```cpp
#include <bits/stdc++.h>
using namespace std;

int n;
int A[21];
int M[21][21];

int dp[21][1<<21];

int solve(int current, int w,int visi){
    visi |= (1 << current);
    if(w<=0)
        return 0;
    int ans = 0;
    if(dp[current][visi]!=-1) return dp[current][visi];
    for(int i=0;i<n;i++){
        if(!(visi & (1 << i)) and w-(A[i]+M[current][i]) >= 0){
            ans = max(ans,solve(i,w-(A[i]+M[current][i]),visi)+1);
        }
    }
    return dp[current][visi] = ans;
}


main(){
    ios base::sync with stdio(0);
    cin.tie(0);
    while(cin >> n and n){
        memset(dp,-1,sizeof dp);
        for(int i=0;i<n;i++){
            cin >> A[i];
        }
        for(int i=0;i<n;i++){
            for(int j=0;j<n;j++){
                cin >> M[i][j];
            }
        }
        int ans = 0;
        for(int i=0;i<n;i++){
            int visi = (1 << i);
            if(420 - A[i] >=0)
                ans = max(ans,solve(i,420 - A[i],visi)+1);
        }

        cout << ans << endl;

    }

}
```

```cpp
#include <bits/stdc++.h>
using namespace std;

double V[16][16];
vector<pair<int,int> > P;
int n;

double dp[20][1 << 16];

inline double calc(int i,int j){
    return sqrt((P[i].first-P[j].first)*(P[i].first-P[j].first) + (P[i].second - P[j].second)*(P[i].second - P[j].second));
}

double solve(int current, int mask){
    if(mask == ((1<<(n+1))-1)){
        return V[current][0];
    }

    if(dp[current][mask]!=-1) return dp[current][mask];

    double ans = 1e9 + 10;

    for(int i=1;i<=n;i++){
        if(!(mask & (1<<i)))
            ans = min(solve(i,mask | (1<<i))+V[current][i],ans);
    }
    return dp[current][mask] = ans;
}


main(){
    ios base::sync with stdio(0);
    cin.tie(0);
    int x,y,a,b;
    while(cin >> n and n){
        P.clear();
        for(int i=0;i<=n;i++){
            for(int j=0;j<=(1<<(n+1));j++){
                dp[i][j] = -1;
            }
        }
        cin >> x >> y;
        P.push back(make pair(x,y));
        for(int i=1;i<=n;i++){
            cin >> a >> b;
            P.push back(make pair(a,b));
        }
        for(int i=0;i<=n;i++){
            for(int j=0;j<=n;j++){
                V[i][j] = calc(i,j);
            }
        }
        cout << fixed << setprecision(2) << solve(0,1) << endl;
    }
}
```

```cpp
#include <bits/stdc++.h>
using namespace std;

int n;
int M[19][19];
int A[19];
int dp[19][(1 << 19)];
int dead;
int solve(int current,int mask){

    if(mask == dead)
        return 0;

    if(dp[current][mask]!=-1) return dp[current][mask];

    int ans = 1e9+10;

    for(int i=0;i<n;i++){
        if(!(mask & A[i])){
            ans = min(ans,solve(current+1,(mask | A[i]))+M[i][current]);
        }
    }
    return dp[current][mask] = ans;
}

main(){
    for(int i=0;i<=18;i++)
        A[i] = 1 << i;
    while(scanf("%d",&n) and n){
        for(int i=0;i<n;i++)
            for(int j=0;j<(1 << (n+1));j++)
                dp[i][j] = -1;
        dead = (1 << n)-1;
        for(int i=0;i<n;i++){
            for(int j=0;j<n;j++){
                scanf("%d",&M[i][j]);
            }
        }
        cout << solve(0,0) << endl;
    }
}
```

```cpp
#include <bits/stdc++.h>
using namespace std;

typedef long long int ll;

main(){
    ios base::sync with stdio(0);
    cin.tie(0);
    ll z,n;
    cin >> z;
    for(int k=0;k<z;k++){
        cin >> n;
        ll nr = n << 1;
        ll M[nr][nr];
        for(int i=0;i<n;i++){
            for(int j=0;j<n;j++){
                cin >> M[i][j];
                M[i+n][j] = M[i][j];
                M[i][j+n] = M[i][j];
                M[i+n][j+n] = M[i][j];
                if(i>0) M[i][j] += M[i-1][j];
                if(j>0) M[i][j] += M[i][j-1];
                if(i>0 && j>0) M[i][j] -= M[i-1][j-1];
            }
        }
        for(int i=0;i<nr;i++){
            for(int j=0;j<nr;j++){
                if((i>=n or j>=n)){
                    if(i>0){
                        M[i][j] += M[i-1][j];
                    }
                    if(j>0){
                        M[i][j] += M[i][j-1];
                    }
                    if(i>0 && j>0){
                        M[i][j] -= M[i-1][j-1];
                    }
                }
            }
        }
        ll ans = -1000*100*100;
        int xi,yi,xf,yf;
        for(int i=0;i<n;i++){
            for(int j=0;j<n;j++){
                for(int x=i;x<i+n;x++){
                    for(int y=j;y<j+n;y++){
                        ll at = M[x][y];
                        if(i>0) at -= M[i-1][y];
                        if(j>0) at -= M[x][j-1];
                        if(i>0 and j>0) at += M[i-1][j-1];
                        ans = max(ans,at);
                        if(ans == at){
                            xi = i;
                            yi = j;
                            xf = x;
                            yf = y;
                        }
                    }
                }
            }
        }
        if(xi==0 and yi==0 and ((xf==nr-1 and yf==n-1) or (xf==n-1 and yf==nr-1)))
                ans -= M[n-1][n-1];
        cout << ans << endl;
        //cout << "COORD I:  " << xi << " " << yi << endl
            // << "COORD F:  " << xf << " " << yf << endl;
    }
}

```

```cpp
#include <bits/stdc++.h>
using namespace std;
#define F first
#define S second
#define mp make pair
#define pb push back

typedef long long int ll;

ll sum[20005];
ll n,s;

main(){
    ios base::sync with stdio(0);
    cin.tie(0);
    cin >> n;

    for(int k=0;k<n;k++){
        cin >> s;
        ll ans = 0,ansx=1,ansy=1;
        ll at = 0;
        int from=1,to=1;
        for(int i=0;i<s-1;i++){
            cin >> sum[i];
            at += sum[i];
            if(at >= ans){
                if(at > ans or (ansy-ansx < i+2-from) or (ansy-ansx == i+2-from
                and from <= ansx)){
                    ansx = from;
                    ansy = i+2;
                }
                ans = at;
            }
            if(at<0){
                at = 0;
                from = i+2;
            }
        }
        if(ans > 0)
            cout << "The nicest part of route " << k+1 << " is between stops " <<
            ansx << " and " << ansy << endl;
        else
            cout << "Route " << k+1 << " has no nice parts" << endl;
    }
}

```

```cpp
#include <bits/stdc++.h>
using namespace std;

#define mp make pair
#define F first
#define S second
typedef long long int ll;
typedef pair<ll,ll> ii;
typedef vector<ll> vi;
typedef vector<ii > vii;

vi A;
vii st;
int n,m;

int getMax(vector<ll> e){
    ll s = -1;
    int p=-1;
    for(int i=0;i<e.size();i++){
        if(e[i]!=-1 and s < A[e[i]]){
            s = A[e[i]];
            p = i;
        }
    }
    return p;
}

void build(int P,int L,int R){
    if(L==R){
        st[P] = mp(L,-1);
        return;
    }
    if(L>R || R<L)
        return;
    int nxt = P << 1;
    int mid = (L+R) >> 1;
    build(nxt,L,mid);
    build(nxt+1,mid+1,R);

    vi e;
    ll s1,s2;
    int p;
    e.push back(st[nxt].F);
    e.push back(st[nxt].S);
    e.push back(st[nxt+1].F);
    e.push back(st[nxt+1].S);
    p = getMax(e);
    s1 = p==-1 ? -1:e[p];
    e[p]=-1;
    p = getMax(e);
    s2 = p==-1 ? -1:e[p];

    st[P] = mp(s1,s2);

}

void update (int p, int L, int R, int i, int value) {

    // no overlap
    if(L > i or R < i) return;

    // total overlap
    if(L == R and L == i) {
        A[i] = value;
        st[p] = mp(i,-1);
        return;
    }

    int nxt = p << 1;
    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 
```

```cpp
#include <bits/stdc++.h>
using namespace std;

typedef long long int ll;
typedef pair<ll,ll> ii;
typedef vector<ii> vii;
typedef vector<ll> vi;

vi st,lazy;
int n;

ll query(int p, int L, int R, int i,int j){

    if(lazy[p]!=0){
        st[p] += (R-L+1)*lazy[p];
        if(R!=L){
            lazy[p<<1] += lazy[p];
            lazy[(p<<1)+1] += lazy[p];
        }
        lazy[p] = 0;
    }
    // no overlap
    if(i>R || j<L) return 0;

    // total overlap
    if(L>=i && R<=j) return st[p];

    // partial overlap
    int nxt = p << 1;
    int mid = (L + R) >> 1;

    return query(nxt,L,mid,i,j) + query(nxt + 1,mid +1,R,i,j);
}
void update(int P,int L,int R, int i,int j, ll value){

    if(lazy[P]!=0){
        st[P] += (R-L+1)*lazy[P];
        if(L!=R){
            lazy[P << 1] += lazy[P];
            lazy[(P << 1)+1] += lazy[P];
        }
        lazy[P] = 0;
    }

    // no overlap
    if( L > j or R < i) return;

    // total overlap
    if(L >= i and R <= j){
        st[P] += (R-L+1)*value;
        if(L!=R){
            lazy[P<<1] += value;
            lazy[(P<<1)+1] += value;
        }

        return;
    }

    // partial overlap
    int nxt = P << 1;
    int mid = (L+R) >> 1;

    update(nxt, L, mid, i, j, value);
    update(nxt+1,mid+1,R,i,j,value);

    st[P] = st[nxt]+st[nxt+1];

}

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
```

```cpp
#include <bits/stdc++.h>
using namespace std;

vector<vector<string > > Grafo(105);
map<string,int> V;
map<int,string> S;
map<string,int> grau;
int n,m;
vector<int> saida;
void kahn(){
    int i;
    priority_queue<int> F;
    for(i=0;i<n;i++){
        if(grau[S[i]]==0)
            F.push(-i);
    }
    while(!F.empty()){
        int aux = -F.top();
        saida.push_back(aux);
        F.pop();
        for(i=0;i<Grafo[aux].size();i++){
            if(--grau[Grafo[aux][i]]==0)
                F.push(-V[Grafo[aux][i]]);
        }
    }

}
main(){
    int i,j,k,cont=1;
    string e,from,to;
    while(cin >> n){
        cin.ignore();
        for(i=0;i<n;i++){
            cin >> e;
            cin.ignore();
            V[e]=i;
            S[i]=e;
            Grafo[i].clear();
        }
        cin >> m;
        for(i=0;i<m;i++){
            cin >> from >> to;
            Grafo[V[from]].push_back(to);
            grau[to]++;
        }
        kahn();
        cout << "Case #" << cont << ": Dilbert should drink beverages in this order: ";
        for(i=0;i<saida.size();i++){
            cout << S[saida[i]];
            if(i!=saida.size()-1)
                cout << " ";
        }
        cout <<"." << endl;
        V.clear();
        S.clear();
        saida.clear();
        cont++;
        cout << endl;
    }
}
```

```cpp
#include <bits/stdc++.h>
using namespace std;
typedef vector<vector<int> > vvi;
typedef vector<int> vi;
typedef vector<pair<int,int> > vii;
typedef pair<int,int> ii;
vvi Grafo(100001);
bool visitados[100001];
vector<bool> visi;
stack<int> ordem;
int n,m;
void dfs(int n){
    visitados[n] = true;
    for(int i=0;i<Grafo[n].size();i++){
        if(!visitados[Grafo[n][i]])
            dfs(Grafo[n][i]);
    }
}
void dfsOrd(int n){
    visitados[n] = true;
    for(int i=0;i<Grafo[n].size();i++){
        if(!visitados[Grafo[n][i]])
            dfs(Grafo[n][i]);
    }
    ordem.push(n);
}
void reset(){
    for(int i=0;i<n;i++)
        visitados[i]=false;
}

main(){
    ios base::sync with stdio(0);
    cin.tie(0);
    int i,j,z,from,to;
    cin >> z;

    for(i=0;i<z;i++){
        cin >> n >> m;
        for(j=0;j<n;j++)
            Grafo[j].clear();
        for(j=0;j<m;j++){
            cin >> from >> to;
            Grafo[from-1].push back(to-1);
        }
        reset();

        for(j=0;j<n;j++){
            if(!visitados[j])
                dfsOrd(j);
        }
        reset();
        int cont = 0;
        while(!ordem.empty()){
            int x = ordem.top();
            ordem.pop();
            if(!visitados[x]){
                dfs(x);
                cont++;
            }
        }
        cout << cont << endl;
    }

}

```

```cpp
#include <bits/stdc++.h>
using namespace std;

typedef long long int ll;

int n,m;
int produtos[1010][2],P[110];
ll dp[1010][30];
inline ll solve(ll current, ll w){

    if(current<0 or w <= 0) return 0LL;

    if(dp[current][w]!=-1) return dp[current][w];

    ll ans;
    if(produtos[current][1]<=w)
        ans = solve(current-1,w-produtos[current][1])+produtos[current][0];

    ans = max(solve(current-1,w),ans);

    return dp[current][w] = ans;
}

main(){
    int t;

    scanf("%d",&t);

    for(int k=0;k<t;k++){

        scanf("%d",&n);

        for(int i=0;i<n;i++){
            scanf("%d %d",&produtos[i][0], &produtos[i][1]);
        }

        scanf("%d",&m);
        memset(dp,-1,sizeof dp);
        ll ans = 0;
        for(int i=0;i<m;i++){
            scanf("%d",&P[i]);
            ans += solve(n-1,P[i]);
        }
        printf("%lld\n",ans);
    }
}
```

```cpp
#include <bits/stdc++.h>
using namespace std;

typedef long long ll;

ll w,T;
ll wt[40];
ll A[40][40];
int n;

inline ll knapsack(){

    ll K[n+1][T+1];

    for(int i=0;i<=n;i++){
        for(int j=0;j<=T;j++){
            if(i==0 or j==0)
                K[i][j] = 0;
            else if(A[i-1][1] <= j){
                K[i][j] = max(A[i-1][0]+K[i-1][j-A[i-1][1]],K[i-1][j]);
            }
            else
                K[i][j] = K[i-1][j];
        }
    }
    ll total B = 0;
    ll total w = 0;
    vector<pair<ll,ll> > V;
    for(int i=n,j=T;i>0;i--){
        if(K[i][j]!=K[i-1][j]){
            V.push back(make pair(wt[i-1],A[i-1][0]));
            ++total B;
            j-=A[i-1][1];
        }
    }
    cout << K[n][T] << endl
         << total B << endl;
    for(int i=V.size()-1;i>=0;i--)
        cout << V[i].first << " " << V[i].second << endl;
}
main(){
    bool f = true;
    while(cin >> T >> w){
        if(!f)
            cout << endl;
        cin >> n;
        for(int i=0;i<n;i++){
            cin >> A[i][1] >> A[i][0];
            wt[i] = A[i][1];
            A[i][1] = (2*w*A[i][1]) + (w*A[i][1]);
        }
        knapsack();
        f = false;
    }


}

```

```cpp
#include <bits/stdc++.h>
using namespace std;
int w;
vector<int> car;
int dp[10001][10001];
int solve(int current,int s1,int s2){
    if(current>=car.size())return dp[s1][s2] = 0LL;

    if(dp[s1][s2]!=-1) return dp[s1][s2];

    int ans = 0;
    if(car[current]+s1 <= w)
        ans = max(solve(current+1,s1+car[current],s2)+1,ans);
    if(car[current]+s2 <= w)
        ans = max(solve(current+1,s1,s2+car[current])+1,ans);

    return dp[s1][s2] = ans;
}
void print(int current, int s1, int s2){


    if(current>=car.size()) return;
    if(s1+car[current] <=w and dp[s1][s2] - 1 == dp[s1+car[current]][s2]){
        printf("port\n");
        print(current+1,s1+car[current],s2);
    }
    else if(s2+car[current] <=w and dp[s1][s2]-1 == dp[s1][s2+car[current]]){
        printf("starboard\n");
        print(current+1,s1,s2+car[current]);
    }

}
main(){
    int n,aux;
    scanf("%d",&n);
    for(int k=0;k<n;k++){
        scanf("%d",&w);
        w*=100;
        while(scanf("%d",&aux) and aux){
            car.push back(aux);
        }
        memset(dp,-1,sizeof dp);
        printf("%d\n",solve(0,0,0));

        print(0,0,0);
        car.clear();
        if(k<n-1)
            puts("");
    }
}

```

```cpp
#include <bits/stdc++.h>
using namespace std;

typedef vector<vector<int> > vvi;
typedef vector<int> vi;
typedef vector<pair<int,int> > vii;
vvi Grafo(1005);
vii pontes;
int dfs low[1005];
int dfs num[1005];
int dfs parent[1005];
bool articulation vertex[1005];
int dfsNumberCounter,Children,dfsRoot,n,arti;
void print dfs(){
    cout << pontes.size() << " critical links\n";
    for(int i=0;i<pontes.size();i++){
        cout << pontes[i].first << " - " << pontes[i].second << "\n";
    }
    cout << "\n";
}
void dfs(int u){
    dfs low[u] = dfs num[u] = dfsNumberCounter++;
    for(int i=0;i<Grafo[u].size();i++){
        int v = Grafo[u][i];
        if(dfs num[v]==-1){
            dfs parent[v] = u;
            if(u==dfsRoot)
                Children++;
            dfs(v);

            if(dfs low[v]>=dfs num[u]){
                articulation vertex[u] = true;}
//          if(dfs low[v]>dfs num[u])
//              pontes.push back(make pair(u,v));
            dfs low[u] = min(dfs low[u],dfs low[v]);
        }
        else if(v!=dfs parent[u])
            dfs low[u] = min(dfs low[u],dfs num[v]);
    }
}
void reset(){
    for(int i=0;i<n;i++){
            Grafo[i].clear();
            dfs num[i] = -1;
            dfs low[i] = 0;
            dfs parent[i] = 0;
            articulation vertex[i] = false;
        }
        //pontes.clear();
        dfsNumberCounter = 0;
        arti = 0;
}
void solve(){
    for(int i=0;i<n;i++){
        if(dfs num[i]==-1){
            dfsRoot = i;
            Children = 0;
            dfs(i);
            articulation vertex[i] = (Children>1);
        }
    }
    for(int i=0;i<n;i++){
        if(articulation vertex[i])
            arti++;
    }
    cout << arti << "\n";
}

main(){
    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
```

```cpp
#include <bits/stdc++.h>
#define S second
#define F first
using namespace std;
typedef pair<int,int> ii;
typedef pair<int,ii> iii;
typedef vector<int> vi;
typedef vector<iii> viii;
int dy[] = {1,-1,0,0};
int dx[] = {0,0,1,-1};
bool Grafo[1001][1001];
int n,m;

bool valid(int i,int j){
    if(Grafo[i][j]) return false;
    if(i<0 or i>=n) return false;
    if(j<0 or j>=m) return false;
    return true;
}

int bfs(ii ini,ii dest){
    queue<iii> q;
    q.push(make pair(0,make pair(ini.F,ini.S)));
    Grafo[ini.F][ini.S]=true;
    while(!q.empty()){
        iii x = q.front();
        ii p = x.second;
        q.pop();
        if(x.second == dest)
            return x.first;
        for(int i=0;i<4;i++){
            if(valid(p.first+dy[i],p.second+dx[i])){
                Grafo[p.first+dy[i]][p.second+dx[i]] = true;
                q.push(make pair(x.first+1,make pair(p.first+dy[i],p.second+dx[i])));
            }
        }

    }

}

main(){
    ios base::sync with stdio(0);
    cin.tie(0);
    int l,c,i,j,k,z,xi,yi,xd,yd,b;
    while(cin >> n >> m and n and m){
        cin >> z;
        for(i=0;i<n;i++){
            for(j=0;j<m;j++){
                Grafo[i][j] = false;
            }
        }
        for(i=0;i<z;i++){
            cin >> l >> b;
            for(j=0;j<b;j++){
                cin >> c;
                Grafo[l][c] = true;
            }
        }
        cin >> xi >> yi;
        cin >> xd >> yd;
        cout << bfs(make pair(xi,yi),make pair(xd,yd)) <<"\n";
    }
}
```

```cpp
#include <bits/stdc++.h>
using namespace std;
typedef pair<int,int> ii;
typedef vector<ii> vii;
typedef vector<vii> vvii;
typedef vector<pair<int,ii> > viii;
typedef long long int ll;
viii Grafo,MST;
int parent[200001];
ll total;
int findset(int x){
    if(x!=parent[x])
        parent[x] = findset(parent[x]);
    return parent[x];
}
void UNION(int x,int y){
    parent[x] = parent[y];
}
void kruskal(){
    int pu,pv;
    sort(Grafo.begin(),Grafo.end());
    for(int i=0;i<Grafo.size();i++){
        pu = findset(Grafo[i].second.first);
        pv = findset(Grafo[i].second.second);
        if(pu!=pv){
            total+=Grafo[i].first;
            UNION(pu,pv);
        }
    }

}
main(){
    int n,m,from,to,w;

    while(cin >> n >> m and n and m){
        Grafo.clear();
        ll t = 0;
        for(int i=0;i<m;i++){
            cin >> from >> to >> w;
            Grafo.push back(make pair(w,make pair(from,to)));
            t+=w;
        }
        for(int i=0;i<n;i++){
            parent[i] = i;
        }
        total = 0;
        kruskal();
        cout << t-total << endl;
    }
}

```

```cpp
#include <bits/stdc++.h>
using namespace std;
typedef vector<vector<int> > vvi;
typedef vector<int> vi;
vvi Grafo(2005);
bool visitados[2005];
int n,m;
void dfs(int n){
    visitados[n] = true;
    for(int i=0;i<Grafo[n].size();i++){
        if(!visitados[Grafo[n][i]])
            dfs(Grafo[n][i]);
    }
}


main(){
    ios base::sync with stdio(0);
    cin.tie(0);
    int i,j,z,from,to,way;

    while(cin >> n >> m and n and m){
        for(j=0;j<n;j++){Grafo[j].clear();}
        for(j=0;j<m;j++){
            cin >> from >> to >> way;
            Grafo[from-1].push back(to-1);
            if(way==2)
                Grafo[to-1].push back(from-1);
        }
        bool falha = false;
        for(j=0;j<n;j++){
            memset(visitados,false,sizeof(visitados));
            dfs(j);
            for(int k=0;k<n;k++){if(visitados[k]==false){falha = true;break;}}
            if(falha)
                break;
        }
        if(falha)
            cout << 0 << endl;
        else
            cout << 1 << endl;
    }

}

```

```cpp
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
typedef pair<int,int> ii;
typedef vector<ii> vii;
typedef vector<vii> vvii;
typedef vector<int> vi;
typedef vector<ll> vll;
vvii Grafo(1001);
int n,m;

bool ford(int ini){
    vll dist(n+1,LLONG MAX);
    dist[ini] = 0;
    for(int i=0;i<n-1;i++){
        for(int k=0;k<n;k++){
            for(int j=0;j<Grafo[k].size();j++){
                ii v = Grafo[k][j];
                if(dist[k]!=LLONG MAX)
                dist[v.first] = min(dist[v.first],dist[k]+v.second);
            }
        }
    }
//  bool negative = false;
    for(int i=0;i<n;i++){
        for(int j = 0;j<Grafo[i].size();j++){
            ii v = Grafo[i][j];
            if(dist[v.first]!=LLONG MAX and dist[v.first] > dist[i]+v.second){
                return true;
            }
        }
    }
    return false;
}
void reset(){
    for(int i=0;i<n;i++){
        Grafo[i].clear();
    }
}

main(){
    int k,from,to,w,i,j;

    cin >> k;

    for(i=0;i<k;i++){
        cin >> n >> m;
        reset();
        for(j=0;j<m;j++){
            cin >> from >> to >> w;
            Grafo[from].push back(make pair(to,w));
        }
        if(ford(0))
            cout << "possible\n";
        else
            cout << "not possible\n";
    }
}
```

```cpp
#include <bits/stdc++.h>
using namespace std;
typedef long long int ll;
ll Grafo[101][101];
bool visitados[101];
int m;

main(){
    int n,z,from,to,x,y;
    scanf("%d",&z);
    for(int v =0;v<z;v++){
        scanf("%d", &m);
        scanf("%d", &n);
        for(int i=0;i<m;i++){
            for(int j=0;j<m;j++){
                    if(i==j)
                        Grafo[i][j] = 0;
                    else
                        Grafo[i][j] = 100000000;
            }
        }
        for(int q=0;q<n;q++){
            scanf("%d %d",&from,&to);
            Grafo[from][to] = 1;
            Grafo[to][from] = 1;
        }
        scanf("%d %d",&x,&y);
        for (int k = 0; k < m; k++){
            for (int i = 0; i < m; i++){
                for (int j = 0; j < m; j++){
                    Grafo[i][j] = min(Grafo[i][j], Grafo[i][k] + Grafo[k][j]);
                }
            }
        }
        ll saida = 0;
        for(int i=0;i<m;i++)
            saida = max(Grafo[x][i]+Grafo[i][y],saida);

        cout << "Case " << v+1<< ": " << saida << endl;
    }
}
```

```cpp
#include <bits/stdc++.h>
using namespace std;

typedef long long int ll;
map<string,int> cidades;
map<int,string> vertices;
vector<vector<pair<int,int> > > Grafo(30);
int pai[30];
int n,m;

void dijkstra(string o,string des){

    int dist[n];
    for(int i=0;i<n;i++) dist[i] = 10101010;
    priority queue<pair<int,int>, vector<pair<int,int> >, greater<pair<int,int> > > pq;

    pq.push(make pair(0,cidades[o]));
    dist[cidades[o]] = 0;

    while(!pq.empty()){

        int d = pq.top().first;
        int v = pq.top().second;
        pq.pop();
        for(int i=0;i<Grafo[v].size();i++){
            if(dist[Grafo[v][i].first] > Grafo[v][i].second+d){
                pq.push(make pair(Grafo[v][i].second+d,Grafo[v][i].first));
                pai[Grafo[v][i].first] = v;
                dist[Grafo[v][i].first] = Grafo[v][i].second+d;
            }
        }
    }


}

main(){

    memset(pai,-1,sizeof pai);
    string o,des,e,aux;
    int d;
    cin >> n;
    cin.ignore();
    for(int i=0;i<n;i++){
        cin >> e;
        cidades[e] = i;
        vertices[i] = e;
    }
    cin >> m;
    for(int i=0;i<m;i++){
        cin.ignore();
        cin >> e >> aux >> d;
        Grafo[cidades[e]].push back(make pair(cidades[aux],d));
        Grafo[cidades[aux]].push back(make pair(cidades[e],d));
    }
    cin >> o >> des;

    dijkstra(o,des);
    vector<string> ans;
    while(pai[cidades[des]]!=-1){
        ans.push back(des);
        des = vertices[pai[cidades[des]]];
    }
    ans.push back(o);
    for(int i=ans.size()-1;i>=0;i--){
        if(i!=ans.size()-1)
            cout << "-";
        cout << ans[i];
    }
```

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

```cpp
#include <bits/stdc++.h>
using namespace std;

typedef long long int ll;
ll n;
int moedas[] = {1,5,10,25,50};
ll dp[10][1000010];

inline ll solve(ll current,ll sum){

    if(sum==0) return 1LL;
    if(current < 0 or sum<0) return 0LL;

    if(dp[current][sum]!=-1) return dp[current][sum];

    ll ans = solve(current,sum-moedas[current])+solve(current-1,sum);

    return dp[current][sum] = ans;
}


main(){
    ios base::sync with stdio(0);
    cin.tie(0);
    memset(dp,-1,sizeof dp);
    ll ans;
    while(scanf("%lld",&n)!=EOF){
        ans = solve(4,n);
        cout << "There " << (ans==1? "is only ":"are ") << ans << (ans==1? "
        way":" ways") << " to produce " << n << " cents change." << endl;
    }

}

```

```cpp
#include <bits/stdc++.h>
using namespace std;
#define EPS 1e-2

typedef long long int ll;
int M[] = {10000,5000,2000,1000,500,200,100,50,20,10,5};

ll dp[11][40100];
ll n;

ll solve(ll current, ll sum){

    if(sum==0) return dp[current][sum] = 1LL;
    if(current < 0 or sum < 0) return 0LL;

    if(dp[current][sum]!=-1) return dp[current][sum];

    ll ans = solve(current,sum-M[current]) + solve(current-1,sum);

    return dp[current][sum] = ans;

}

main(){
    float aux;
    memset(dp,-1,sizeof dp);
    while(cin >> aux and aux!= 0.00){
        n = (ll)(aux*100);
        if(fabs(aux*100 - n) > EPS)
            n++;
        printf("%6.2f %16lld\n", aux, solve(10,n));
    }

}
```

```cpp
#include <bits/stdc++.h>
using namespace std;

typedef long long int ll;

ll BIT[100100];
int n;

inline void update(ll index){

    for(;index<=n;index += index  &(-index))
        BIT[index] += 1;
}
inline ll query(ll index){
    ll ans =0;
    for(;index>0;index -= index & (-index))
        ans += BIT[index];
    return ans;
}


main(){
    vector<ll> num;
    scanf("%d",&n);
    ll ans = 0,aux;
    for(int i=1;i<=n;i++){
        scanf("%lld",&aux);
        num.push back(aux);
    }
    for(int i=num.size()-1;i>=0;i--){
        update(num[i]);
        ans += query(num[i]-1);
    }
    printf("%lld\n",ans);

}

```

```cpp
#include <bits/stdc++.h>
using namespace std;

vector<vector<int> > Grafo(205);
int cores[205];
bool visitados[205];
int n,m;

bool bfs bi(){
    memset(visitados,false,sizeof(visitados));
    memset(cores,-1,sizeof(cores));
    queue<int> F;
    F.push(0);
    cores[0] = false;
    while(!F.empty()){
        int aux = F.front();
        F.pop();
        if(!visitados[aux]){
            visitados[aux] = true;
            for(int i=0;i<Grafo[aux].size();i++){
                if(cores[Grafo[aux][i]]==-1)
                    cores[Grafo[aux][i]] = 1-cores[aux];
                else if(cores[Grafo[aux][i]]==cores[aux])
                    return false;
                cores[Grafo[aux][i]] = 1-cores[aux];
                F.push(Grafo[aux][i]);
            }
        }
    }
    return true;
}

main(){
    int i,j,aux,from,to;

    while(scanf("%d",&n) and n){
        cin >> m;
        for(i=0;i<n;i++){Grafo[i].clear();}
        for(i=0;i<m;i++){
            scanf("%d %d",&from,&to);
            Grafo[from].push back(to);
        }
        if(bfs bi())
            printf("BICOLORABLE.\n");
        else
            printf("NOT BICOLORABLE.\n");
    }

}
```

```cpp
#include <bits/stdc++.h>
using namespace std;
#define EPS 1e-6

typedef long long int ll;


main(){
    int p,q,r,s,t,u;


    while(cin >> p >> q >> r >> s >> t >> u){
        double ini= 0.000000,fim = 1.000000, mid = 0.500000;
        bool achou = false;

        while(ini<=fim){
            double ans = p*exp(-mid)+q*sin(mid)+r*cos(mid)+s*tan(mid)+t*(mid*mid)+u;
            if(fabs(ans) <= EPS){
                cout << fixed << setprecision(4) << mid << endl;
                achou = true;
                break;
            }
            else if(ans < EPS)
                fim = mid - 0.000000001;
            else
                ini = mid + 0.000000001;
            mid = (ini+fim)/2;
        }
        if(!achou)
            cout << "No solution\n";
    }


}

```

```cpp
#include <bits/stdc++.h>
using namespace std;

typedef long long int ll;


ll D[10100];
ll n,num,X,mid;

ll binary search(ll lo, ll hi){
    mid = (lo+hi)/2;
    if(lo > hi or mid == X) return -1;
    if(num==D[mid]) return mid;
    else if(num < D[mid]) return binary search(lo,mid-1);
    else if(num > D[mid]) return binary search(mid+1,hi);
}

main(){
    ios base::sync with stdio(0);
    cin.tie(0);
    ll y,ans,ans1,aux;

    while(cin >> n){
        for(int i=0;i<n;i++){
            cin >> D[i];
        }
        cin >> y;
        sort(D,D+n);
        ans = -1;
        ans1 = -1;
        for(X=0;X<n;X++){
            num = y - D[X];
            aux = -1;
            if(num < D[X])
                aux = binary search(0,X-1);
            if(aux !=-1){
                if(ans==-1){
                    ans = aux;
                    ans1 = X;
                }
                else{
                    if(abs(D[X]-D[aux]) < abs(D[ans]-D[ans1])){
                            ans = aux;
                            ans1 = X;
                    }
                }
            }
        }
        if(ans <= ans1)
            cout << "Peter should buy books whose prices are " << D[ans] << " and
            " << D[ans1] << "." << endl;
        else
            cout << "Peter should buy books whose prices are " << D[ans1] << " and
            " << D[ans] << "." << endl;
        cout << endl;
    }
}

```

```java
import java.math.BigInteger;
import java.util.Scanner;

public class Main{

    public Main(){}

    public static String reverse(String a){
        String aux = "";

        for(int i=a.length()-1;i>=0;i--){
                aux = aux.concat(a.substring(i,i+1));
        }
        return aux;
    }

    public static void main(String[] args){
        long n;
        Scanner ler = new Scanner(System.in);
        BigInteger a,b;
        String s1,s2,aux,aux2;

        n = ler.nextLong();

        for(long i = 0;i< n;i++){
            a = ler.nextBigInteger();
            b = ler.nextBigInteger();

            s1 = reverse(a.toString());
            s2 = reverse(b.toString());
            a = new BigInteger(s1);
            b = new BigInteger(s2);

            s1 = reverse(a.add(b).toString());
            a = new BigInteger(s1);
            System.out.println(a);
        }

    }

}

```

```java

import java.math.BigInteger;
import java.util.Scanner;

public class Main{

    public Main(){
    }

    public static void main(String[] args){

        BigInteger b,ans;
        int n=-1,a=-1,cont=1;
        Scanner ler = new Scanner(System.in);

        while(n!=0 && a!=0){
            n = ler.nextInt();
            a = ler.nextInt();
            if(n==0 && a==0)
                break;
            ans = BigInteger.ZERO;
            for(int i = 0;i<n;i++){
                b = ler.nextBigInteger();
                ans = ans.add(b);
            }
            b = ans.divide(BigInteger.valueOf(a));
            System.out.println("Bill #"+cont+" costs "+ans.toString()+": each friend should pay "+b.toString()+"\n");
            cont++;
        }


    }

}

```