Template

```
/**
 *    @author     : Maruf Tuhin
 *    @School     : CUET CSE 11
 *    @Topcoder   : the_redback
 *    @CodeForces : the_redback
 *    @UVA        : the_redback
 *    @link       : maruf.2hin@gmail.com
 */

//#include <bits/stdc++.h>

#include<cstdio>
#include<cstring>
#include<cstdlib>
#include<cctype>
#include<cmath>
#include<iostream>
#include<fstream>
#include<string>
#include<vector>
#include<queue>
#include<map>
#include<algorithm>
#include<set>
#include<sstream>
#include<stack>
using namespace std;


typedef long long          ll;
typedef unsigned long long llu;

#define ft        first
#define sd        second
#define mp        make_pair
#define pb(x)     push_back(x)
#define all(x)    x.begin(),x.end()
#define allr(x)   x.rbegin(),x.rend()
#define mem(a,b)  memset(a,b,sizeof(a))
#define meminf(a)
memset(a,126,sizeof(a))
#define inf       1e11
#define eps       1e-9
#define mod       1000000007
#define NN        30100

//cout << setfill('0') << setw(3) << a;
//cout << fixed << setprecision(20)<< a;


main()
{
    //ios_base::sync_with_stdio(0);
    //cin.tie(0);
}
```

Manually Sort

```
bool comp(char b,char c)
{
    if(tolower(b)==tolower(c))
        return b<c;
     //porer ta Capital hoile
     //swap kore age jabe.
    else
        return tolower(b)<tolower(c);
     //normal compare kore sort korbe.
}

/*
Input:
aAabB

Output:
AaaBb
*/
//bool false return korle sort hobe.
```

BigMod

```
//(m^n)%p;
long p;
long check(long m,long n)
{
    int sum;
    if(n==0)
        return 1;
    if(n%2==0)
    {
        sum=check(m,n/2);
        return ((sum%p)*(sum%p))%p;
    }
    else
    {
        sum=check(m,n-1);
        return ((m%p)*(sum%p))%p;
    }
}

int main()
{
    long m,n,sum;
    while(scanf("%ld %ld %ld",
                &m,&n,&p)==3)
    {
        sum=check(m,n);
        printf("%ld\n",sum);
    }
    return 0;
}
```

MST & Disjoint Set (Fast)

```
typedef long long ll;
#define inf 10000000
#define mem(a,b) memset(a,b,sizeof(a))
#define NN 10010
int root[NN+7];
int rank[NN+7];

struct edge {
    int u,v,w;
};

vector<edge>e;

bool comp(edge n,edge m) {
    return n.w>m.w;
}


void init(int n) {
    for(int i = 1; i <= n; i++) {
        root[i] = i;
        rank[i] = 0;
    }
}

int find(int u) {
    if(u != root[u])
        root[u] = find(root[u]);
    return root[u];
}

void Union(int u, int v) {
    if(rank[u] > rank[v])
        root[v] = u;
    else {
        root[u] = v;
        if(rank[u]==rank[v])
            rank[v]++;
    }
}

int mst(int n) {
    init(n);
    int i,j,k;
    sort(e.begin(),e.end(),comp);
    int count=0,sum=0;
    for(i=0; i<e.size(); i++) {
        int u=find(e[i].u);
        int v=find(e[i].v);
        if(u!=v) {
            Union(u,v);
        }
        else
            sum+=e[i].w;
    }
    return sum;
}

main()
{
    int i,j,k,l,n,r,c,u,v,w;
    edge ed;
    int tc,t=1,x=-1,m;
    cin>>tc;
    while(tc--)
    {
        cin>>n>>m;
        while(m--)
        {
            cin>>ed.u>>ed.v>>ed.w;
            e.push_back(ed);
        }
        int sum=mst(n);
        cout<<sum<<"\n";
        e.clear();
    }
    cin>>n;

    return 0;
}
```

Extended GCD

```
intsii,si,tii,ti; // ii=i-1, i=i,
s/t=i+1
integcd(inta,int b)
{
    intr,q,s,t;
    sii=1,si=0;
    tii=0,ti=1;
    while(b>0)
    {
        q=a/b;
        r=a%b;
        s=sii-(q*si);
        t=tii-(q*ti);
        sii=si,si=s;
        tii=ti,ti=t;
        a=b,b=r;
    }
    return a; // return a,sii,tii
}
```

LCM

```
int gcd(int a,int b)
{
    while(b>0)
    {
        a=a%b;
        swap(a,b);
    }
    return a;
}

int lcm(int a, int b)
{
    int temp = gcd(a, b);
              //__gcd(a, b)
    return ((a / temp) * b);
}
```

Sieve of PHI (Co-primes of 2 to n)

```
unsigned long long a[NN+7];
void sieve(void)
{
    inti,j,k,n=2237;
    for(i=2; i<NN; i++)
        a[i]=i;
    for(i=2; i<NN; i+=2)
    {
        a[i]*=(2-1);
        a[i]/=2;
    }
    for(i=3; i<NN; i+=2)
        if(a[i]==i)
            for(j=i; j<NN; j+=i)
            {
                a[j]*=(i-1);
                a[j]/=i;
            }
}
```

PHI (Co-primes of n)

```
int phi(int n)
{
    int res=n;
    int i;
    for(i=2; i*i<=n; i++)
    {
        if(!(n%i))
            res-=res/i;
        while(!(n%i))
            n/=i;
        if(n==1)
            break;
    }
    if(n>1)
        res-=res/n;
    return res;
}
```

nCr (normal)

```
long longnCr(intn,int r) //
{
   long long sum=1,I,k,l,j=1;
    k=max((r,n-r)), l=min((r,n-r));
    for(i=k+1;i<=n;i++)
    {
        sum*=i;
        if(j<=l &&sum%j==0)
        {
            sum/=j;
            j++;
        }
    }
    return sum;
}
```

nCr (DP)

```
i64 dp[70][70];
i64 nCr(intn,int r)
{
    if(r==1) return n;
    if(n==r) return 1;

    if(dp[n][r]!=-1)
        return dp[n][r];
    dp[n][r]=nCr(n-1,r)+nCr(n-1,r-1);

    return dp[n][r];
}

int main()
{
    Mem(dp,-1);
    printf("%d\n",nCr(20,2));

}
```

BitMask DP

```
intdp[70000];
int a[20][20];
int n;

int go(int x,int mask)
{
    if(x>=n)
        return 0;

    int&t=dp[mask];

    if(t!=-1)
        return t;
    int k=0;

    for(int i=0; i<n; i++)
        if((mask & (1<<i))==0)
            k=max(k,go(x+1,mask |
                        1<<i)+a[x][i]);

    dp[mask]=k;
    return dp[mask];
}
main()
{
    inttc,t=1;
    scanf("%d",&tc);
    while(tc--)
    {
        scanf("%d",&n);
        for(int i=0; i<n; i++)
            for(int j=0; j<n; j++)
                scanf("%d",&a[i][j]);
        memset(dp,-1,sizeof(dp));
        int sum=go(0,0);
        printf("Case %d: %d\n",t++,sum);
    }
    return 0;
}
```

Topological Sort

```
#define mem(a,b) memset(a,b,sizeof(a))
#define pbpush_back
#define pppop_back
#define inf 1000000000
#define NN 1000010

vector<int>e[NN+7],v;
int view[NN+7];
int f,fl;

void dfs(int u) {
    inti,k,l;
    view[u]=0;
    for(i=0; i<e[u].size(); i++) {
        if(view[e[u][i]]==-1)
            dfs(e[u][i]);
        else if(view[e[u][i]]==0) {
            //then there is a cycle;
            fl=1;
            return;
        }
    }
    view[u]=1;
    v.pb(u);
}

int main()
{
    int i,j,k,l;
    int tc,t;
    int n,m;
    while(~scanf("%d%d",&n,&m))
    {
        if(n==0 && m==0)
            return 0;
        mem(view,-1);
        for(i=0; i<m; i++)
            scanf("%d%d",
                 &k,&l),e[k].pb(l);
        fl=0;
        for(i=1; i<=n; i++)
        {
            f=i;
            if(view[i]==-1)
                dfs(i);
        }
        if(fl)
            printf("IMPOSSIBLE\n");
        else
        {
            reverse(v.begin(),v.end());
            for(i=0; i<v.size(); i++)
                printf("%d\n",v[i]);
        }
        v.clear();
        for(i=0; i<=n; i++)
            e[i].clear();
    }
    return 0;
}
```

MiniMax (MaxiMin is also same)

```
int pr[NN+7];
int a[NN+7][NN+7];

main()
{
   while(~scanf("%d%d%d",&n,&r,&l))
   {
       for(i=0; i<=n; i++)
       {
           for(j=0; j<=n; j++)
               a[i][j]=inf;
           //a[i][j]=-inf
           //FOR MaxiMin
           a[i][i]=0;
       }
       while(r--)
       {
           scanf("%d%d%d",&u,&v,&w);
           a[u][v]=w;
           a[v][u]=w;
       }
     for(k=1; k<=n; k++)
        for(i=1; i<=n; i++)
           for(j=1; j<=n; j++)
               a[i][j]=min(a[i][j],
                  max(a[i][k],a[k][j]));
        //a[i][j]=max(a[i][j],
                  min(a[i][k],a[k][j]));
        //For MaxiMin.

        while(l--)
        {
            scanf("%d%d",&n,&r);
            int sum=a[n][r];
            if(sum>=inf)
               //sum<=inf For MaxiMin
                puts("no path");
            else
                printf("%d\n",sum);
        }
    }
    return 0;
}
```

Power (n^k)

```
typedef long long LL;
LL power(LL n,LL m)
{
    LL sum=1;
    while(m>0)
    {
        sum*=n;
        m--;
    }
    return sum;
}
```

Highest Factors (range)

```
long b[1000001],c[1000001];
int main()
{
    long i,j,m,mx,t;
    b[1]=1;
    mx=0;
    c[1]=1;
    for(i=2; i<1000001; i++)
    {
        for(j=i; j<1000001; j+=i)
            b[j]++;
        if(b[i]>=mx)
        {
            mx=b[i];
            m=i;
        }
        c[i]=m;
    }
    scanf("%ld",&t);
    while(t--)
    {
        scanf("%ld",&m);
        printf("%ld\n",c[m]);
    }
    return 0;
}
```

Divisors sums (range)

```
#include<cstdio>
#define Z 500003
long Sum[Z];
int main()
{
    long t,n,i,j;
    for(i=1; i<Z; i++)
    {
        for(j=2*i; j<Z; j+=i)
            Sum[j]+=i;
    }
    scanf("%ld",&t);
    while(t--)
    {
        scanf("%ld",&n);
        printf("%ld\n",Sum[n]);
    }
    return 0;
}
```

Nim Game

```
#define NN 1050
main() {
   int t=1,tc;
   int k,l,n;
   cin>>tc;
   while(tc--) {
       cin>>n;
       int res=0;
       while(n--) {
           cin>>k;
           res^=k;
       }
      if(res)
         printf("Case %d: Alice\n",t++);
                          //First move
      else
         printf("Case %d: Bob\n",t++);
    }
    return 0;
}
```

Misere Nim Game

```
#define NN 1050
main() {
   int t=1,tc;
   int k,l,n;
   cin>>tc;
   while(tc--) {
       cin>>n;
       int res=0,cnt=0,i;
       for(i=0; i<n; i++) {
           cin>>k;
           res^=k;
           if(k==1)
               cnt++;
       }
    if(cnt==n) {
       if(!res)
         printf("Case %d: Alice\n",t++);
       else
         printf("Case %d: Bob\n",t++);
      }
     else {
       if(res)
         printf("Case %d: Alice\n",t++);
       else
         printf("Case %d: Bob\n",t++);
      }
  }
  return 0;
}
```

Prime Sieve

```
=========[ For storing ]=============
#define NN 47000

bool p[NN+7]; //Hashing
vector<int>pr; //storing prime

void sieve(int n)
{
    int i,j,k,l;
    p[1]=1;
    pr.push_back(2);
    for(i=4; i<=n; i+=2)
        p[i]=1;
    for(i=3; i<=n; i+=2)
    {
        if(p[i]==0)
        {
            pr.push_back(i);
            for(j=i*i; j<=n; j+=2*i)
                p[j]=1;
        }
    }
}
```

```
==========[ For Hashing ]===========
#define NN 47000

bool p[NN+7]; //Hashing

void sieve(int n)
{
    int i,j,k,l;
    p[1]=1;
    for(i=4; i<=n; i+=2)
        p[i]=1;
    for(i=3; i*i<=n; i+=2)
    {
        if(p[i]==0)
        {
            for(j=i*i; j<=n; j+=2*i)
                p[j]=1;
        }
    }
}
```

Prime Factor

```
#define NN 47000

bool p[NN+7]; //Hashing
vector<int>pr; //storing prime
void sieve(int n)
{
    int i,j,k,l;
    p[1]=1;
    pr.push_back(2);
    for(i=4; i<=n; i+=2)
        p[i]=1;
    for(i=3; i<=n; i+=2)
    {
        if(p[i]==0)
        {
            pr.push_back(i);
            for(j=i*i; j<=n; j+=2*i)
                p[j]=1;
        }
    }
}

int factor(int n)
{
    int count,k,i;
    for(i=0; i<pr.size() &&
pr[i]*pr[i]<=n; i++)
    {
        k=pr[i];
        count=0;
        while(n%k==0)
        {
            n/=k;
            count++;
        }
        if(n==1)
            break;
    }
    if(n>1)
        then, n is another prime factor;
}
```

LIS (nlog(n)) & Print

```
#define mem(x,y) memset(x,y,sizeof(x));
vector <int> v;
vector <int> L;
vector <int> I;
stack <int> ans;

void prework(void)
{
    I.clear();
    L.clear();
    int i,k;
    k=2000000000;
    I.push_back(-1*k);

    for(i=0; i<v.size(); i++)
        L.push_back(1);
    return;
}

int LIS(void)
{
    int i,low,mid,high;
    for(i=0; i<v.size(); i++)
    {
        low=0;
        high=I.size()-1;
        while(low<=high)
        {
            mid=(low+high)/2;
            if(v[i]>I[mid])
                low=mid+1;
            else
                high=mid-1;
        }
        if(low==I.size())
            I.push_back(v[i]);
        else
            I[low]=v[i];
        L[i]=low;
    }
    return I.size()-1;
}

void show(void)
{
    int i,j,k,max;
    max=0;
    for(i=0; i<L.size(); i++)
    {
        if(max<L[i])
        {
            max=L[i];
            j=i;
        }
    }
    ans.push(v[j]);
    for(i=j-1; i>=0 ; i--)
    {
        if(v[i]<v[j] && L[i]==L[j]-1)
        {
            ans.push(v[i]);
            j=i;
        }
    }
    while(ans.size())
    {
        printf("%d\n",ans.top());
        ans.pop();
    }
    return;
}

main()
{
    char a[10];
    int i,j,k,l,m,n,t=0,T;
    scanf("%d",&T);
    getchar();
    getchar();
    while(T--)
    {
   //input taking methode in uva 497
        t++;
        v.clear();
        while(gets(a) && strlen(a))
        {
            sscanf(a,"%d",&n);
            v.push_back(n);
        }
        if(v.size())
        {
            prework();
            if(t!=1)
                puts("");
            printf("Max hits:
                        %d\n",LIS());
            show();
        }
    }
    return 0;
}
```

LDS (n^2) [Decreasing]

```
#define mem(x,y) memset(x,y,sizeof(x));

int dp[1000];
bool dc[1000];
vector <int>v;
int LDS(int u)
{
    if(dc[u])
        return dp[u];
    int max=0;
    for(int i=u+1; i<v.size(); i++)
    {
        if(v[i]<=v[u])
        {
            if(max<LDS(i))
            {
                max=LDS(i);
            }
        }
    }
    dp[u]=max+1;
    dc[u]=1;
    return dp[u];
}



main()
{
    int i=0,k,n,j;
    while(scanf("%d",&n)==1)
    {
        if(n==-1)
            return 0;
        i++;
        mem(dc,0);
        v.clear();
        v.push_back(n);
        while(1)
        {
            scanf("%d",&n);
            if(n==-1)
                break;
            v.push_back(n);
        }
        k=0;
        for(j=0; j<v.size(); j++)
        {
            if(k<LDS(j))
            {
                k=LDS(j);
            }

        }
        if(i!=1)
            puts("");
        printf("Test #%d: %d\n",i,k);
    }
}
```

Ternary Search

```
/*
    some points are given initially.
Now, we have to find a area consists
    with K,L no points and third one
which area is equal or gretter than S.

*/

int ternary_search(int k,int l,int s)
{
    int low=0, high=v.size()-1;
    int midleft, midright;

    while(high-low>3)
    {
        midleft = low + (high-low)/3 ;
        midright = high - (high-low)/3;

        int area1=area(k,l,midleft);
        int area2=area(k,l,midright);

        if( area1 < area2 )
            low = midleft ;
        else
            high = midright ;
    }

    for(int i=low; i<=high; i++)
    {
        int temp=area(k,l,i);
        if(s <= temp )
            return i+1;
    }
    return 0;
}
```

String Multiplication (500!)

```
char a[1001][10000];
void swap(char b[10000])
{
    int temp,i,j,l;
    l=strlen(b);
    for(i=0,j=l-1; i<l/2; i++,j--)
    {
        temp=b[i];
        b[i]=b[j];
        b[j]=temp;
    }
}
void work(char a[10000],char
b[10000],int n)
{
    int i,j,onhand=0,k,l;
    l=strlen(a);
    for(i=l-1,j=0; i>=0; i--)
    {
        k=((a[i]-48)*n)+onhand;
        b[j]=(k%10)+48;
        onhand=k/10;
        j++;
    }
    while(onhand>0)
    {
        b[j]=(onhand%10)+48;
        onhand/=10;
        j++;
    }
    b[j]='\0';
    swap(b);
}

main()
{
    int i,j,n;
    strcpy(a[0],"1");
    strcpy(a[1],"1");
    for(i=2; i<1001; i++)
    {
        work(a[i-1],a[i],i);

    }
    while(scanf("%d",&n)==1)
    {
        printf("%d!\n%s\n",n,a[n]);

    }
    return 0;
}
```

String Addition (fibonacchi Freeze)

```
const int max=1111;
char a[5001][max];
void add(char b[max],char b1[max],char
b2[max]) {
    char c[max];
    int carry=0,i,k,j,m,n,l=0;
    m=strlen(b1);
    n=strlen(b2);
    for(i=m-1,k=n-1;i>=0||k>=0;i--,k--){
        if(i>=0 && k>=0) {
            j= b1[i]-48+b2[k]-48+carry;
            carry=j/10;
            c[l]=j%10+48;
            l++;
        }
        else if(i>=0) {
            j=b1[i]-48+carry;
            carry=j/10;
            c[l]=j%10+48;
            l++;
        }
        else {
            j=b2[k]-48+carry;
            carry=j/10;
            c[l]=j%10+48;
            l++;
        }
    }
    j=0;
    if(carry==1) {
        b[j]=49;
        j++;
    }
    for(i=l-1; i>=0; i--) {
        b[j]=c[i];
        j++;
    }
    b[j]='\0';
}
void check(void) {
    strcpy(a[0],"0");
    strcpy(a[1],"1");
    for(int i=2; i<=5000; i++)
        add(a[i],a[i-1],a[i-2]);
}
main() {
    check(); int n;
    while(scanf("%d",&n)==1) {
        printf("The Fibonacci number for
                    %d is %s\n",n,a[n]);
    }
    return 0;
}
```

String Division & Modulus

```
long long div(char a[],long long n,char
c[])
{
    int i,j,t=0,l,d=0,r=0;
    long long rem=0;
    l=strlen(a);
    for(i=0;i<l;i++)
    {
        rem=(rem*10)+a[i]-48;

        if(rem>=n||r!=0)
        {
            j=rem/n;
            rem=rem%n;
            c[d]=j+48;
            d++;
            r=1;
        }
    }
    if(d==0)
    {
        c[d]='0';
        d++;
    }
    c[d]='\0';
    return rem;
}
```

String Modulus

```
int mod(char a[], int divider)
{
    int rem,i;
    rem=0;
    for(i=0;a[i];i++)
    {
        rem=rem*10+a[i]-48;
        rem=rem%divider;
    }
    return rem;
}
```

Geometry Area

```
======[ Polygon Area ]========
double area(void)
{
    double total = 0.0;
              /* total area so far */
    int i, j;
              /* counters */
      //V is storage of polygon points
    for (i=0; i<v.size(); i++)
    {
        j = (i+1) % v.size();
        total += (v[i].x*v[j].y) -
                      (v[j].x*v[i].y);
    }
    return(total / 2.0);
}

======[ Triangle Area ]==========

P MV(P a,P b)
{
    return P(b.x-a.x,b.y-a.y);
}

double CP(P a,P b)
{
    return a.x*b.y-a.y*b.x;
}

double area(int x,int y,int z)
{
    double total = 0;
           /* total area so far */

    total=CP(MV(v[x],v[y]),
                   MV(v[x],v[z]));

    return(total / 2);
}
```

Convex Hull Points

```
#define mp make_pair
#define pb(x) push_back(x)
#define pp(x) pop_back(x)
#define all(x) x.begin(),x.end()
#define mem(a,b) memset(a,b,sizeof(a))
#define inf 1e9
#define eps 1e-9
#define NN 1050

struct P {
    double x,y;
    P(double X,double Y) {
        x=X;
        y=Y;
    }
    P() { }
};
vector<P>v;

P MV(P a,P b) {
    return P(b.x-a.x,b.y-a.y);
}

double DP(P a, P b) {
    return a.x*b.x+a.y*b.y;
}

double CP(P a,P b) {
    return a.x*b.y-a.y*b.x;
}

double A(P a) {
    return sqrt(a.x*a.x+a.y*a.y);
}

P ADD(P a,P b) {
    return P(a.x+b.x,a.y+b.y);
}

P LV(P a,double l) {
    return P(a.x*l/A(a),a.y*l/A(a));
}

P pvt;

bool comp(P a,P b){
//False hoile sort korbe
    long long c=CP(MV(pvt,a),MV(pvt,b));
    if(c)
        return c>0;
    return A(MV(pvt,a))<A(MV(pvt,b));
}
```

```
void checkPvt(void) {
    pvt.x=inf;
    pvt.y=inf;
    for(int i=0; i<v.size(); i++) {
        if(pvt.x>v[i].x)
            pvt=v[i];
        else if(pvt.x==v[i].x
                        && pvt.y>v[i].y)
            pvt=v[i];
    }
}

vector<P>q;

void go(void) {
    q.clear();
    int n=v.size();
    q.pb(v[n-1]);
    v.pp();
    for(int i=v.size()-1;i>=0;i--) {
        if(CP(MV(v[0],q[q.size()-1]),
            MV(v[0],v[i]))==0) {
            q.pb(v[i]);
            v.pp();
        }
        else
            break;
    }
    for(int i=0;i<q.size();i++)
        v.pb(q[i]);
}

main() {
    int t=1,tc,i,j,k,l,m,n;
    double x,y,z,u,w,xx,yy,zz,d;
    double aa,bb,cc,dd;
    cin>>tc;
    while(tc--) {
        cin>>n;
        v.clear();
        for(i=0; i<n; i++) {
            cin>>k>>l;
            v.pb(P(k,l));
        }
        checkPvt();
        sort(v.begin(),v.end(),comp);
        go();
        cout<<(int)v.size()<<"\n";
        for(i=0; i<v.size(); i++)
            cout<<(int)v[i].x<<
                " "<<(int)v[i].y<<"\n";
    }
    return 0;
}
```