DAA SEARCHING LVL-1

Vinoth's Model practical

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

int main(){
    int n,t,I,count=0;
    cin>>n>>t;
    int arr[n];
    for(i=0;i<n;i++){
        cin>>arr[i];
    }
    sort(arr,arr+n);
    for(i=0;i<n;i++){
        t-=arr[i];
        if(t<0){
            break;
        }
        count++;
    }
    cout<<count;
    return 0;
}
```

A double-square numberis an integer Y

```
#include <bits/stdc++.h>
using namespace std;
int sumSquare(int n)
{
    int res=0;
    for (long i = 0; i * i <= n; i++)
        for (long j = i; j * j <= n; j++)
            if ((i * i + j * j == n) ) {
                res++;
            }
    return res;
}
int main()
{
    int t;
    cin>>t;
    int i=1;
    while(t--){
        int n;
        cin>>n;
        cout<<"Line #"<<i<<": "<<sumSquare(n)<<endl;
        i++;
    }
    return 0;
```

```
        cout<<"for(i=0;i<=sqrt(y);i++) for(j=0;j<=i;j++)";
}
```

Trapped by a river and racing against time

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

using namespace std;

int main()

{

    string str;

    int ramp1;

    double rate1,wr;

    getline(cin,str);

    cin>>ramp1>>rate1>>wr;

    double time1,speed1,dist1;

    time1=sqrt(2.0*ramp1/rate1);

    speed1=time1*rate1;

    dist1=speed1*speed1/9.805;

    cout<<str<<" will reach a speed of "<<std::fixed<<std::setprecision(2)<<speed1<<" m/s on a "<<ramp1<<" ramp crossing "<<std::fixed<<std::setprecision(1)<<dist1<<" of "<<std::fixed<<std::setprecision(1)<<wr<<" meters, ";

    if(dist1<(wr-5))

    cout<<"SPLASH!";

    else if(dist1>wr)

    cout<<"LIKE A BOSS!";

    else

    cout<<"BARELY MADE IT!";

    return 0;

}
```

Given 'm' postive integers

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

using namespace std;

#define f(n) for(i=0;i<n;i++)

#define g(n) for(i=1;i<n;i++)
```

```
#define k(n) for(i=n-2;i>=0;i--)
int maxWater(int arr[],int n)
{
    int left[n],i;
    int right[n];
    int water=0;
    left[0]=arr[0];
    g(n)
        left[i]=max(left[i-1],arr[i]);
    right[n-1]=arr[n-1];
    k(n)
        right[i]=max(right[i+1],arr[i]);
    for(i=1;i<n-1;i++)
    {
        int var=min(left[i-1],right[i+1]);
        if(var>arr[i])
        {
            water+=var-arr[i];
        }
    }
    return water;
}
int main()
    {
    int n,i;
    cin>>n;
    int arr[n];
    f(n){
        cin>>arr[i];
    }
    cout<<maxWater(arr,n);
    return 0;
}
```

The Allies are trying to get a message

```
#include <bits/stdc++.h>
using namespace std;
void solve(){ cout<<"break;";}
int main(){
string s1,s2,s3,s4;
double r;
double h;
cin>>s1>>r>>s2>>s3>>s4;
if(s2=="FEET")
r=r/3.28;
//cout<<r<<endl;
if(s2=="KILOMETERS") r=r*1000;
if(s2=="YARDS") r=r*0.9144;
if(s2=="INCHES") r=r*0.0254;
if(s2=="MILES") r=r*1609.34;
if(s4=="HOUR") r=r/3600;
if(s4=="MINUTE") r=r/60;
if(s2=="CENTIMETERS") r=r/100;
h=r*r/(2*9.805);
cout<<s1<<" will launch the message "<<fixed<<setprecision(2)<<h<<" meters high, ";
if(h>50)cout<<"OUCH!";
else if(h<25)cout<<"SPLAT!";
else cout<<"SUCCESS!";
return 0;}
```

Major Kathiravan

```
#include <bits/stdc++.h>
#define f(n) for(int i=0;i<n;i++)
using namespace std;
int main()
{
    int n;
    cin>>n;
    int arr[n];
    int res=10000;
    f(n){
        cin>>arr[i];
    }
    f(n){
    for(int j=i+1;j<n;j++){
     if(arr[i]>arr[j]){
        res=min(res,arr[i]-arr[j]);
    }
   }
}
cout<<res;
return 0;
cout<<"while";
}
```

Inspector Gadget

```
#include <bits/stdc++.h>
using namespace std;
int main(){
    string F_str,K_str,X_str;
    getline(cin,F_str);
    getline(cin,K_str);
    getline(cin,X_str);
    string F = F_str.substr(2);
    string K = K_str.substr(2);
    string X = X_str.substr(2);
    if (X == "?"){
        float F_num = stof(F);
        float K_num = stof(K);
        float ans = F_num/(-K_num);
        cout << "X " << fixed << setprecision(2) << ans;
    }
    else if (F == "?"){
        float K_num = stof(K);
        float X_num = stof(X);
        float ans = -K_num * X_num;
        cout << "F " << fixed << setprecision(2) << ans;
    }
    else{
        float F_num = stof(F);
        float X_num = stof(X);
        float ans = -(F_num / X_num);
        cout << "K " << fixed << setprecision(2) << ans;
    }
    return 0;
}
```

Mr Somu

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int t;
    cin>>t;
    while(t--){
        int b,n,r;
        cin>>b>>n>>r;
    int z=1;
    for(int i=1;i<=n;i++){
        z=z*i;
    }
    int res;
    res=pow(b,z);
    cout<<res%r<<endl;
}
    return 0;
    cout<<"if(n%2==1)";
}
```

Given two integers 'b' and 'a'

```
#include <iostream>
using namespace std;
int main()
{
int t;
long long m;
long long n;
long long ans;
scanf("%d",&t);
for(int cs=1;cs<=t;cs++){
    scanf("%lld %lld",&n,&m);
    ans=(n*m)/2;
    printf("%lld\n",ans);
    }
}
```

In the following figure you can see a rectangular

```
#include <bits/stdc++.h>
using namespace std;
void solve(){
cout<<"return(l-2*x)*(b-2*x)*x;";
}
int main()
{
int tc;
double a, b, c, res, l, w, x;
scanf(" %d", &tc);
while(tc--) {
scanf(" %lf %lf", &l, &w);
a = 12.0;
b = -4.0 * (l+w);
c = l*w;
x = (-b - sqrt (b*b - 4.0*a*c)) / (2.0*a);
res = (l - 2*x) * (w - 2*x) * x;
printf ("%.9f\n", res);
}
return 0;
}
```

The Mask ate a block of dynamite to save

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
float a,c,d;
string b;
cin>>a>>b>>c;
float res;
int z=b.size();
if(z==1)
d=b[0]-48;
else
d=(float)(b[0]-48)/(b[2]-48);
res=a*d*0.45*7.5;
if(res>c){
cout<<res<<" the Mask should not eat it!";
}
else
cout<<fixed<<setprecision(2)<<res<<" the Mask can eat it!";
return 0;
cout<<"for";
}
```

There is a Gangaroo initially placed at the

```
#include <stdio.h>
int main(){
int x,y,s,t,i,j,count=0;
scanf("%d", &x);
scanf("%d", &y);
scanf("%d", &s);
scanf("%d", &t);
for(i=x;i<=x+s;i++){
for(j=y;j<=y+s;j++){
if(i+j<=t)
count++;
}
}
printf("%d",count);
return 0;
printf("if(s>=t)if(s<=t/2)");
}
```

DAA Sorting

Scrooge McDuck

```
#include <iostream>
using namespace std;
int main()
{
int p,q,r,i;
int c;
cin>>c;
for(i=0;i<c;i++){
cin>>p>>q>>r;
q=q+(r-1)/5;
r=(r-1)%5+1;
p=p+(q-1)/10;
q=(q-1)%10+1;
cout<<p<<" ";
cout<<q<<" ";
cout<<r<<endl;
}
return 0;
}
```

Tina owns a match making company

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
int t,n;
cin>>t;
while(t--){
cin>>n;
int a[n],b[n],sum=0;
for(int i = 0;i<n;i++)
cin>>a[i];
for(int i=0;i<n;i++)
cin>>b[i];
sort(a,a+n);
sort(b,b+n);
for(int i=0;i<n;i++){
if(a[i]%b[n-i-1]==0||b[n-i-1]%a[i]==0)
sum++;
}
cout<<sum<<endl;
}
return 0;
}
```

Shankar is a volleyball trainer

```
#include <bits/stdc++.h>
using namespace std;
typedef long long LL;
const int N=(int)1e6+6,mod=(int)0;
int a[N];
long long sum[N];
int main(){
int tc;
cin>>tc;
for(int tt=1;tt<=tc;++tt){
int n,p;
cin>>n>>p;
for(int j=0;j<n;++j)
cin>>a[j];
sort(a,a+n);
int i;
for(i=0;i<n;i++)
sum[i+1]=sum[i]+a[i];
long long res=1e18;
for(int j=p-1;j<n;++j){
long long s=sum[j+1]-sum[j-(p-1)];
long long cost=(LL)a[j]*p-s;
res=min(res,cost);
}
cout<<res<<'\n';
}
}
```

ROYGBIV

```
#include <bits/stdc++.h>
using namespace std;
typedef long long LL;
const int N=(int)1e6+6,mod=(int)0;
int a[N];
long long sum[N];
int main(){
int tc;
cin>>tc;
for(int tt=1;tt<=tc;++tt){
int n,p;
cin>>n>>p;
for(int j=0;j<n;++j)
cin>>a[j];
sort(a,a+n);
int i;
for(i=0;i<n;i++)
sum[i+1]=sum[i]+a[i];
long long res=1e18;
for(int j=p-1;j<n;++j){
long long s=sum[j+1]-sum[j-(p-1)];
long long cost=(LL)a[j]*p-s;
res=min(res,cost);
}
cout<<res<<'\n';
}
}
```

Ace Ventura, Pet Detective

```
#include <bits/stdc++.h>
using namespace std;
#define p1 cout<<"Ace, move fast, pigeon is at ("<<i<<",0)";
#define p2 cout<<"Ace, move fast, pigeon is at ("<<(i-i/z)%z<<","<<i/z<<")";
#define p3 cout<<"No pigeon, try another map, Ace";
#define a continue;
#define f(n)for(int i=0;i<z;i++)
#define while1 while((scanf("%c",&s[i]))!=EOF)
int main(){
string s1;cin>>s1;
int z=s1.size();
f(n){
if(s1[i]=='P'){p1
return 0;}
}
//cout<<z<<endl;
int i=0,cnt=0;
char s[10000];
while1{
if(s[i]=='P'){
cnt=1;
break;
}
i++;
}
if(cnt==1)p2
```

```
    else p3}
```

Ganesan has a string

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
string s,s2;
cin>>s>>s2;
int z=s.length();
int i;
int a[z];
for(i=0;i<(int)s.length();i++){
a[i]=s[i+1]-s[i];
}
for(int i=0;i<z-2;i++){
if(a[i]!=a[i+1]){
cout<<"No";
return 0;}
}
cout<<"Yes";
return 0;
}
```

Great news! You get to go to Japan

```
#include<iostream>
using namespace std;
int main()
{
int items;
int a,j,cnt=0;
cin>>a>>items;
int c[items];
string s[items];
for(j=0;j<items;j++){
cin>>s[j]>>c[j];
if(c[j]<a){
cout<<"I can afford "<<s[j]<<endl;
a=a-c[j];
}
else{
cnt++;
cout<<"I can't afford "<<s[j]<<endl;
}
//cout<<cnt;
}
if(cnt==items)
cout<<"I need more Yen!";
else
cout<<a;
return 0;
cout<<"for(i=1;i<=yen;i++) int i,j;";
}
```

Tina owns a match making company, which even to her

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
int t,n;
cin>>t;
while(t--){
cin>>n;
int a[n],b[n],sum=0;
for(int i = 0;i<n;i++)
cin>>a[i];
for(int i=0;i<n;i++)
cin>>b[i];
sort(a,a+n);
sort(b,b+n);
for(int i=0;i<n;i++){
if(a[i]%b[n-i-1]==0 || b[n-i-1]%a[i]==0)
sum++;
}
cout<<sum<<endl;
}
return 0;
}
```

The sapphire consulting and marketing company is adding

```
#include <stdio.h>
#include <stdlib.h>
int isqrt(n) int n; {
int i;
for(i=0;i*i<n;i++);
return i;
}
int main() {
int c;
int t,h,s,i,j;
int d;
scanf("%d",&c);
for(i=0;i<c;i++) {
s=0;
scanf("%d %d",&t,&h);
d=isqrt(t);
s+=t+(d*4);
for(j=1;j<h;j++) {
s+=3;
s+=(d+j)*4;
if((d+j)>2)
s+= (d+j-2)*2;
}
printf("%d liters\n",s);
}
return 0;
}
```

Ganesan has a string S consisting of lowercase

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
string s,s2;
cin>>s>>s2;
int z = s.length();
int i;
int a[z];
for(i=0;i<(int)s.length();i++){
a[i]=s[i+1]-s[i];
}
for(int i=0;i<z-2;i++){
if(a[i]!=a[i+1]){
cout<<"No";
return 0;}
}
cout<<"Yes";
return 0;
}
```

there are n integers

```
#include <stdio.h>
#include <string.h>
#include <math.h>
```

```
#include <stdlib.h>
#include <assert.h>
#define if


int lonelyinteger(int a_size, int* a) {
    int i=0;
    int num=0;
    for(i=0;i<a_size;i++){
        num=num^a[i];
    }

    return num;

}
int main() {
    int res;

    int _a_size, _a_i;
    scanf("%d", &_a_size);
    int _a[_a_size];
    for(_a_i = 0; _a_i < _a_size; _a_i++) {
        int _a_item;
        scanf("%d", &_a_item);

        _a[_a_i] = _a_item;
    }
  res = lonelyinteger(_a_size, _a);
    printf("%d", res);

    return 0;
}
```

```
void y(){
   printf("break;");
}
```

Siva has several containers, each with a number

```
#include <stdio.h>
#include <stdlib.h>
void insertionSort(long int *p,long int n);
void asd();
int main(){
   asd();
 return 0;
}
void asd()

{
 int q;
 scanf("%d",&q);
 while(q--){
 int n,i,j;
 scanf("%d",&n);
 int M[n][n];
 long int *r,*c,*arr;
 arr=(long int *)malloc(n*n*sizeof(long int));
 *arr=n;
 r=(long int *)malloc(n*sizeof(long int)); c=(long int *)malloc(n*sizeof(long int));
 for(i=0;i<n;i++){
 for(j=0;j<n;j++){
 scanf("%d",&M[i][j]);
 r[i]+=M[i][j];
 c[j]+=M[i][j];
 }
 }
 int count=0;
 for(i=0;i<n;i++){
 for(j=0;j<n;j++){
 if(r[i]==c[j])
 {
 count++;
 break;
 }
 }
 }
 if(count==n)
 printf("Possible\n");
 else
 printf("Impossible\n");
 }
}
```

Avul pakir

```
#include <iostream>
#include<string.h>
using namespace std;
int main(){
    int n,i,j,t;
    cin>>n;
    for(i=0;i<n;i++){
      cout<<"Line "<<i+1<<":"<<endl;
      cin>>t;
      int sum=0;
      for(j=0;j<t;j++){

          char name[100];
          cin>>name;
          if(strcmp(name,"donate")==0){
              int d;cin>>d;
              sum+=d;
          }
          else{cout<<sum<<endl;}

      }
    }
    return 0;
}
```

THAI PONGAL

```
#include <iostream>
using namespace std;
int factors(int num,int l) {
   int i,c1=0;
   for(i=1; i <= num; i++) {
      if (num % i == 0 && i>l)        c1++;}   return c1;   cout<<"continue;";}
int main()
{
    int t,j;
    cin>>t;
    for(j=1;j<=t;j++)
    {
        int p,l,q,i,c=0,sp;
        cin>>p>>l;
        q=p-l;

        printf("Line %d: ",j);
        sp=factors(q,l);
        for(i=1;i<=q;i++)
        {
            if(q%i==0 && i>l)
            {
                printf("%d",i);

                if(c<sp-1)printf(" ");

                c++;

            }

        }

        if(c==0) printf("impossible");

        printf("\n");

    }

            return 0;

}
```

DIVIDE AND CONQUER

Programmer Sandhosh and you have a New Year Tree (not the traditional fur tree, though)

```
#include<bits/stdc++.h>
using namespace std;
const int N=1e6+10;

int m,cnt=4,La=2,Lb=3,len=2;
int f[N][21],dep[N];

int lca(int x,int y) {
   if(dep[x]<dep[y]) swap(x,y);
   for(int i=20;i>=0;i--) if(dep[f[x][i]]>=dep[y]) x=f[x][i];
   if(x==y) return x;
   for(int i=20;i>=0;i--) if(f[x][i]!=f[y][i]) x=f[x][i],y=f[y][i];
   return f[x][0];
}

int dis(int x,int y){
   return dep[x]+dep[y]-dep[lca(x,y)]*2;
}

int main() {
   scanf("%d",&m);
   dep[1]=1;
   dep[2]=dep[3]=dep[4]=2;
   f[2][0]=f[3][0]=f[4][0]=1;
   int u;
   while(m--) {
   cin>>u;
      int x=cnt+1,y=cnt+2;
      cnt+=2;
      f[x][0]=f[y][0]=u;
```

```
        for(int i=1; i<=20; i++) f[x][i]=f[y][i]=f[f[x][i-1]][i-1];
        dep[x]=dep[y]=dep[u]+1;
        int d1=dis(La,x);
        int d2=dis(Lb,x);
        if(len<d1) len=d1,Lb=x;
        if(len<d2) len=d2,La=x;
        printf("%d\n",len);
    }
    return 0;
}
```

Leopard is in the Amusement Park. And now she is in a queue in front of the Ferris wheel

```
#include<cstdio>
#include<iostream>
using namespace std;
inline int getint(){
char c;
while((c=getchar())<'0'||c>'9');return c-'0';
}
const int N=4005,inf=.5e9;
int n,k,sum[N][N],f[N],g[N];
int main(){
cin>>n>>k;
for(int i=1;i<=n;i++)
for(int j=1;j<=n;j++)
sum[i][j]=sum[i-1][j]+sum[i][j-1]-sum[i-1][j-1]+getint();
g[n+1]=n;
for(int kk=2;kk<=k;kk++)
for(int i=n;i;i--){
f[i]=-inf;
for(int j=g[i];j<=g[i+1]&&j<i;j++){
int now=f[j]-sum[j][j]+sum[j][i];
if(now>f[i]){
```

```
f[i]=now;
g[i]=j;
}
}
}
printf("%d\n",sum[n][n]/2-f[n]);
}
```

Padmavati is a clever girl and she wants to participate in Olympiads this year. Of course she wants her partner to be clever too (although he's not)! Padmavati has prepared the following test problem for Sativathi

```
#include <iostream>
#include <map>
using namespace std;
const int N=1<<20;
int n,a[N],c[N],w;
void upd(int i,int c){


}
int main(){
  cin>>n;
  for(int i=0;i<n;++i)cin>>a[i];
  map<int,int>u,v;
  for(int i=n;i-->0;){
    int x=++u[a[i]];
    while(x<N)++c[x],x+=x&-x;
  }


  for(int i=0;i<n;++i){
    int x=u[a[i]]--,y=v[a[i]]++;
    while(x<N)--c[x],x+=x&-x;
    while(y>0)w+=c[y],y-=y&-y;
  }
  cout<<w<<endl;
}
```

Maakesh caught the trail of the ancient Book of Evil in a swampy area

```
#include <bits/stdc++.h>
using namespace std;
const int N = 100005;
int R,D,n,m,d,h[N];
vector<int> adj[N];
bool prob[N],is[N];
void evil(int u,int p=0){
    h[u]= h[p]+1;
    prob[u] &= (h[u]<=d);
    if(is[u]&&h[u]>D)
        D=h[u],R=u;
    for(unsigned int i=0;i<adj[u].size();++i){
        int v= adj[u][i];
        if(v!=p)
            evil(v,u);
    }
}
int main(){
    cin>>n>>m>>d;memset(prob,true,sizeof(prob));
    h[0]=-1;int a,b,i;D=0;
    for(i=0;i<m;++i)
        cin>>R,is[R]=true;
    for(i=0;i<n-1;++i)
        scanf("%d%d",&a,&b),adj[a].push_back(b),adj[b].push_back(a);
    evil(R);evil(R);evil(R);
    int ret=0;
    for(i=1;i<=n;++i)
        if(prob[i])++ret;
    cout<<ret<<endl;
}
```

Lakshman and Sukran are the best competitive programmers in their town. However, they can't both qualify to an important contest. The selection will be made with the help of a single problem. Bhoominath, a friend of Lakshman, managed to get hold of the problem before the contest. Because he wants to make sure Lakshman will be the one qualified, he tells Lakshman the following task

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

using namespace std;

long long n, i = 1, j, k = 9e9, x, s[100001], d;


int main() {

    cin>>n;

    for (; i <= n; i++){ cin>>x;s[i] = s[i - 1] + x;}

    for (i = 1; i <= n; i++)

        for (j = max(1ll, i - 20000); j <= i; j++)

            if (i != j) k = min(k, (i - j) * (i - j) + (s[i] - s[j]) * (s[i] - s[j]));

    cout << k;

}
```

Recently Aarush has become keen on physics. Anna V., his teacher noticed Aarush's interest and gave him a fascinating physical puzzle a half-decay tree

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

using namespace std;

int h,q,v,e;string str;map<int,int> f;

double puzzle(int u,int mx) {return (f[u]<=mx)?mx:(0.5*(puzzle(u<<1,max(mx,f[u]-
f[u<<1]))+puzzle(u<<1|1,max(mx,f[u]-f[u<<1|1]))));}

int main(){

cin>>h>>q;

    while (q--){

        cin>>str;

        if (str[0]=='a'){

            scanf("%d %d",&v,&e);

            while (v) f[v]+=e,v>>=1;

        }

        else printf("%.2lf\n",puzzle(1,0));

    }
    return 0;
}
```

Prithvi are in the world of mathematics to solve the great "Monkey and the carrot"

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

int main()
{

    int numberOfColumns;

        cin>>numberOfColumns;
        int bananaMatrix[2 * numberOfColumns - 1][numberOfColumns]; //Input matrix
        int maxBanana[2 * numberOfColumns - 1][numberOfColumns];    //Memoized matrix

        memset(maxBanana, 0, sizeof(maxBanana));       //Setting 0 to all cell, will update for maximum
        memset(bananaMatrix, 0, sizeof(bananaMatrix)); //Setting 0 to all cell, will update for inputs

        //Input for upper triangle
        for (int row = 0; row < numberOfColumns; row++)
            for (int column = 0; column <= row; column++)
                cin >> bananaMatrix[row][column];

        //Input for lower triangle
        int shiftedPosition = 1;
        for (int row = numberOfColumns; row < (numberOfColumns * 2) - 1; row++)
        {
            for (int column = shiftedPosition; column < numberOfColumns; column++)
                cin >> bananaMatrix[row][column];
            shiftedPosition++;
        }
        maxBanana[0][0] = bananaMatrix[0][0];

        for (int row = 1; row < numberOfColumns; row++)

        {

            for (int column = 0; column <= row; column++)

                if (column == 0)//Caution for negative indexes.

                    maxBanana[row][column] = maxBanana[row - 1][column] +
bananaMatrix[row][column];

                else

                    maxBanana[row][column] = max(maxBanana[row - 1][column], maxBanana[row
- 1][column - 1]) + bananaMatrix[row][column];

        }


        //Memoizing the lower triangle to store the max value

        shiftedPosition = 1;

        for (int row = numberOfColumns; row < (numberOfColumns * 2) - 1; row++)

        {

            for (int column = shiftedPosition; column < numberOfColumns; column++)

                maxBanana[row][column] = max(maxBanana[row - 1][column], maxBanana[row -
1][column - 1]) + bananaMatrix[row][column];

            shiftedPosition++;

        }

        cout <<maxBanana[2 * numberOfColumns - 2][numberOfColumns - 1];




    return 0;

    cout<<"cin>>carrotMatrix[row][column];";

}
```

In this problem you will meet the simplified model of game Pudding Monsters

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

#define fi first

#define se second

#define lo long long
```

```
#define inf 1000000009
#define md 1000000007
#define li 300005
#define mp make_pair
#define pb push_back
using namespace std;
int n,x,y,v[li],a[li],b[li],mn[li],mx[li],g[li];
lo int ans;
void work(int a,intb)
{
    int n=a[0],m=b[0];
    mn[m+1]=0;
    for(int i=1;i<=m;i++){
        mn[i]=min(mn[i-1],b[i]);
        mx[i]=max(mx[i-1],b[i]);
    }
    int mna=inf,mxa=0;
    int l=1,r=1;
    for(int i=1;i<=n;i++){
        mna=min(mna,a[i]);
        mxa=max(mxa,a[i]);
        int d=mxa-mna+1-i;
        if(d>0 && d<=m && mn[d]>mna && mx[d]<mxa) ans++;
        for( ;mn[r]>mna;r++) g[mx[r]-r]++;
        for( ;l<r&&mx[l]<mxa;l++) g[mx[l]-l]--;
        ans+=g[mna+i-1];
    }
    for(int i=l;i<r;i++) g[mx[i]-i]=0;
}
void solve(int l,int r){
    if(l==r) return ;
    int mid=(l+r)/2;
    a[0]=mid-l+1;b[0]=r-mid;
    for(int i=l;i<=mid;i++) a[mid+1-i]=v[i];
    for(int i=mid+1;i<=r;i++) b[i-mid]=v[i];
    work(a,b),work(b,a);
    solve(l,mid);solve(mid+1,r);
}
int main(){
    cin>>n;
    for(int i=1;i<=n;i++){
        cin>>x>>y;
        v[x]=y;
    }
    mn[0]=inf;
    solve(1,n);
    printf("%lld\n",ans+n);
    return 0;

}
```

Fazil is an unemployed computer scientist who spends his days working at odd-jobs.

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

using namespace std;

string word;
long long dp[100][100];


long long calculate(int s, int e){

    if(s > e)
        return 0;
    if(s == e )
        return 1;

    if(dp[s][e] != -1)
        return dp[s][e];

    if(word[s] == word[e])
        return dp[s][e] = 1 +  calculate(s+1, e) + calculate(s, e-1);
    else
        return dp[s][e] = calculate(s+1, e) + calculate(s, e-1) - calculate(s+1, e-1);

}


int main(){

        cin>>word;

        memset(dp, -1, sizeof dp);

        cout<<calculate(0,word.size()-1)<<endl;
    return 0;
    printf("long long calculate(int s,int e)");
}
```

A set of points on a plane is called fair, if for any two points at least one of the three conditions is true

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

pair<int,int>p[10010];
set<pair<int,int> >s;

void dfs(int l,int r)
```

```
{
    if(l==r)
    {
        s.insert(p[l]);
        return;
    }
    int i,mid=(l+r)/2;
    dfs(l,mid);
    dfs(mid+1,r);
    for(i=l;i<=r;i++) s.emplace(p[mid].first,p[i].second);
}

int main()
{
    int n,i;
    scanf("%d",&n);
    for(i=1;i<=n;i++) scanf("%d%d",&p[i].first,&p[i].second);
    sort(p+1,p+n+1);
    dfs(1,n);
    printf("%ld\n",s.size());
    for(auto it:s) printf("%d %d\n",it.first,it.second);
    return 0;
    printf("void fiv(int l,int r),cin>>n;cin>>a[i].first>>a[i].second;");
}
```

Prof.Dr. Ramalingam need representing positive integer N as a sum of addends, where each addends is an integer number containing only 1s

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


long long n,a[17];


int dfs(long long n,int x)
{
        int num=n/a[x];n%=a[x];
        if (!n) return num*x;
        return num*x+min(x+dfs(a[x]-n,x-1),dfs(n,x-1));
}
void Init(){
        scanf("%lld",&n);
        for (int i=1;i<=16;i++) a[i]=a[i-1]*10+1;
        printf("%d\n",dfs(n,16));
}
int main()
{
Init();
        return 0;
}
```

Now Sabanayagam becomes a commander of Ladakh. Ladakh, like its name said,

```
#include<bits/stdc++.h>
using namespace std;
#define N 100005

int cnt[26][100005];
char ans[N];
vector<int> g[N];
void man(){
    cout<<"void dfs(int u,int par) cin>>n; cin>>u>>v;";
}
void dfs(int s,int f){
    for(auto x:g[s])if(x!=f)dfs(x,s);
    int p;
    for(int i=0;i<26&&cnt[i][s]<2;i++)
        if(!cnt[i][s])p=i;
    cnt[p][s]++;
    ans[s]='A'+p;
    for(int i=0;i<=p;i++)cnt[i][f]+=cnt[i][s];
    return ;
}


int main(){
    int n,i,a,b;
    scanf("%d",&n);
    for(i=1;i<n;i++){
        scanf("%d %d",&a,&b);
        g[a].push_back(b);
        g[b].push_back(a);
    }
    dfs(1,0);
    for(i=1;i<=n;i++)printf("%c ",ans[i]);
    return 0;
}
```

Kishan are developing a 'Love calculator' software. You are planning to write the software in a complex way such that nobody would be able to crack the exact behavior of the software.

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


using namespace std;


int main()
{



    string name1, name2;


    int shortestString[31][31];
    long uniqueString[31][31];



        cin >> name1 >> name2;
```

```
//Shift the characters of the name to right for ease of memoizing
name1.insert(0, "0");
name2.insert(0, "1");

//Prepare the matrices for memoization
for (int i = 0; i < 31; i++)
    shortestString[0][i] = shortestString[i][0] = i, uniqueString[i][0] = uniqueString[0][i] = 1;

for (int i = 1; name1[i]; i++)
{
    for (int j = 1; name2[j]; j++)
    {
        //Checking if we need to take the cumulative sum from upper-left block
        if (name1[i] == name2[j])
        {
            //Adding 1 to cumulative sum from upper-left block
            shortestString[i][j] = 1 + shortestString[i - 1][j - 1];

            //No need to add a new branch of unique strings so taking cumulative sum from upper-left block
            uniqueString[i][j] = uniqueString[i - 1][j - 1];

        }
        else
        {
            //Finding the minimum from left and upper block and adding 1 to the value of current block
            shortestString[i][j] = 1 + min(shortestString[i][j - 1], shortestString[i - 1][j]);

            //Checking if there are two unique strings of the same length
            if (shortestString[i][j - 1] == shortestString[i - 1][j])
```

```
                        uniqueString[i][j] = uniqueString[i][j - 1] + uniqueString[i - 1][j];


                        //Checking if left block has the minimum value in shortestString matrix
                    else if (shortestString[i][j - 1] < shortestString[i - 1][j])
                        uniqueString[i][j] = uniqueString[i][j - 1];
                    else
                        uniqueString[i][j] = uniqueString[i - 1][j];
                }
            }
        }
        cout <<shortestString[name1.length() - 1][name2.length() - 1] << " " <<
uniqueString[name1.length() - 1][name2.length() - 1];


    return 0;
    cout<<"cin>>name1>>name2;";
}
```

After the long contest, Sameer returned home and got angry after seeing his room dusty

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



using namespace std;



int partition(int array[],int leftIndex,int rightIndex){
    int pivotValue = array[rightIndex];
    int toBePivotIndex = (leftIndex - 1);
    for(int comparisonIndex = leftIndex; comparisonIndex <= rightIndex - 1;
comparisonIndex++){
        if (


            array[comparisonIndex] < pivotValue
        ) {
            toBePivotIndex++;
            int temp = array[toBePivotIndex];
            array[toBePivotIndex] = array[comparisonIndex];
            array[comparisonIndex] = temp;
        }
    }

    int temp = array[toBePivotIndex+1];
    array[toBePivotIndex+1] = array[rightIndex];
    array[rightIndex] = temp;

    return (toBePivotIndex + 1); // new pivot point
}


void quickSort(int array[],int leftIndex,int rightIndex){

    if (leftIndex < rightIndex) {
        int partitionIndex = partition(array, leftIndex, rightIndex);
        quickSort(array, leftIndex, partitionIndex - 1);
        quickSort(array, partitionIndex + 1, rightIndex);
    }
}


int main(){

    int numberOfDustPoints,widthOfBrush,xCoordinate,yCoordinate;

        int numberOfMoves = 0;
        cin>>numberOfDustPoints>>widthOfBrush;
        int dustPointsYCoordinates[numberOfDustPoints];
        for(int i = 0; i < numberOfDustPoints; i++){
            cin >> xCoordinate >> yCoordinate;
            dustPointsYCoordinates[i] = yCoordinate;
        }

        quickSort(dustPointsYCoordinates,0, numberOfDustPoints-1);



        int currentBrushYCoordinate = dustPointsYCoordinates[0];
        numberOfMoves++;

        for (int i = 0; i < numberOfDustPoints; i++) {
        if(currentBrushYCoordinate + widthOfBrush < dustPointsYCoordinates[i]) {
        currentBrushYCoordinate = dustPointsYCoordinates[i];
        numberOfMoves++;
        }

    }
        cout <<numberOfMoves;



    return 0;
}
```

Makesh

```
#include <bits/stdc++.h>
using namespace std;
const int N = 100005;
int R,D,n,m,d,h[N];
vector<int> adj[N];
bool prob[N],is[N];
void evil(int u,int p=0){
    h[u]= h[p]+1;
    prob[u] &= (h[u]<=d);
    if(is[u]&&h[u]>D)
        D=h[u],R=u;
    for(unsigned int i=0;i<adj[u].size();++i){
        int v= adj[u][i];
        if(v!=p)
            evil(v,u);
    }
}
int main(){
    cin>>n>>m>>d;memset(prob,true,sizeof(prob));
    h[0]=-1;int a,b,i;D=0;
    for(i=0;i<m;++i)
        cin>>R,is[R]=true;
    for(i=0;i<n-1;++i)
        scanf("%d%d",&a,&b),adj[a].push_back(b),adj[b].push_back(a);
    evil(R);evil(R);evil(R);
    int ret=0;
    for(i=1;i<=n;++i)
        if(prob[i])++ret;
    cout<<ret<<endl;
}
```

DAA GREEDY ALGORITHM

Vaanavan thinks that lucky tickets are the tickets whose numbers are divisible by 3.

```
#include<bits/stdc++.h>
using namespace std;
int a[3];
int main()
{
    int n,x,i;
    cin>>n;
    for(i=1;i<=n;i++)
    {
        cin>>x;
        a[x%3]++;
    }
    cout<<a[0]/2+min(a[1],a[2])<<endl;
    return 0;
}
```

A sportsman starts from point xstart = 0

```
#include <bits/stdc++.h>
using namespace std;
void xyz(){}
typedef long long ll;
typedef pair<int, int> pii;
const int mod = 1000000007;

int main() {
    ios::sync_with_stdio(false);
    int n, m, s, d, a[200005] = {}, c = 0, t = 0;
    vector<int> z;
    cin >> n >> m >> s >> d;
    for (int i = 0; i < n; i++) cin >> a[i];
    sort(a, a + n);
    if (a[0] <= s) {cout << "IMPOSSIBLE"; return 0;}
    c = a[0] - 1;
    z.push_back(a[0] - 1);
    a[n] = mod + mod;
    while (t < n) {
        while (t < n && a[t + 1] - a[t] <= s + 1) t++;
        if (a[t] + 1 - c > d) {cout << "IMPOSSIBLE"; return 0;}
        z.push_back(a[t] + 1 - c);
        c = a[t] + 1;
        t++;
        if (t == n) z.push_back(m - c), c = m;
        else z.push_back(a[t] - c - 1), c = a[t] - 1;
    }
    if (!z.back()) z.pop_back();
    bool b = 0;
    for (int i : z) {
        if (b) cout << "JUMP ";
        else cout << "RUN ";
        b = !b;
        cout << i << '\n';
    }if(1>2)cout<<"cin>>n>>m>>s>>d; \n cin>>a[i];";
}
```

A hamburger stand received n orders for rental

```
#include<bits/stdc++.h>
using namespace std;
int n,l,z;
pair<int,int> a[500020];
int main(){
    cin>>n;
    for(int i=0;i<n;i++){
        cin>>a[i].second>>a[i].first;
    }
    sort(a,a+n);
    for(int i=0;i<n;i++){
        if(l<a[i].second){
            z++;
            l=a[i].first;
        }
    }cout<<z;
    return 0;
}
```

It's a very unfortunate day for lavanya today

```
#include <bits/stdc++.h>
using namespace std;
#define res cin>>a[i],num+=a[i];
#define f1     for(int i=1;i<=n;i++)
double n,v,a[25],b[25],sum,mx=1e9;
int main(){
    cin>>n>>v;
    f1{
        cin>>a[i];
        sum+=a[i];
    }
    for(int i=1;i<=n;i++)
        cin>>b[i];
    for(int i=1;i<=n;i++)
        mx=min(mx,b[i]/a[i]);
    cout << fixed<<setprecision(1)<<min(mx*sum,v);
    return 0;
}
```

shiv has given a rebus of form

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

```
int p = 1, n, j, a[105];
char c;
int main()
{
    a[j++] = 1;
    while (cin>>c && c != '=')
    {
        if (c == '-') p--, a[j++] = -1;
        if (c == '+') p++, a[j++] = 1;
    }
    cin>>n;
    for(int i=0;i<j;i++)
    {
        if(a[i]>0)while (p<n && a[i]<n) a[i]++, p++;
        else while (p>n&&a[i]<0 && a[i]>-n) a[i]--, p--;
    }
    if (p != n) { cout << "Impossible\n"; return 0; }
    cout << "Possible\n";
    for(int i=0;i<j;i++)
    cout << (i ? (a[i]<0 ? "- " : "+ ") : "") << abs(a[i]) << " ";
    cout << "= " << n;

}
```

a long time ago

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

int a,i;
int main()
{std::string s,t;
std::cin>>s>>t;
for(i=s.find(t);i+1;++a,i=s.find(t,i+t.size()));std::cout<<a;}
```

A Steeling

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

using namespace std;

#define res cin>>a>>b; cin>>s>>d;

int n,m,s,a,b,d[11];

int main(){

cin>>n>>m;

while(m--)cin>>a>>b,d[b]+=a;

for(int i=10;i>0&&n>0;i--)s+=i*min(n,d[i]),n-=d[i];

cout<<s;

}
```

There are banks in the city where Vishnu lives

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

using namespace std;

typedef long long ll;

ll n,maxs=0;

map<ll,ll> mp;

int main(){

    cin>>n;

    for(ll i=0,x=0,y;i<n;++i)

        cin>>y,maxs=max(maxs,++mp[x+=y]);

    cout<<n-maxs;

}
```

A group of tourists is going on to rameshwaram and dhanushkodi tour

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

using namespace std;

int c[2],i,x,t,n,p,j;

pair<int,int> a[2][1<<17];

#define F(i,n) for(i=0;i<n;++i)

void aasd(){

    cout<<"cin>>n>>v;cin>>t>>v;";

}
```

```
int main(){
    scanf("%d%d",&n,&p);
    F(i,n){
        scanf("%d%d",&t,&j);
        a[t&1][++c[t&1]]=make_pair(-j,i+1);
    }
    F(i,2)sort(a[i]+1,a[i]+c[i]+1);
    F(i,2)F(j,c[i])a[i][j+1].first+=a[i][j].first;
    n=min(p,c[1]);
    for(i=0;~n;--n)
        if((t=a[1][n].first+a[0][min(*c,(p-n)/2)].first)<x)i=n,x=t;
    printf("%d\n",-x);
    F(t,i)printf("%d ",a[1][t+1].second);
    t=min(*c,(p-i)/2);
    F(i,t)printf("%d ",a[0][i+1].second);
    return 0;
}
```

Nadanan's company employed n people. Now Nadanan needs to build a tree hierarchy

```
#include<bits/stdc++.h>
using namespace std;
int a[10001],n,m,x,y;
int main(){
    cin>>n;
    for(int i=0;i<=n;i++)
        cin>>m,a[i]=1e9;
    for(int i=1;i<=m;i++){
        cin>>x>>x>>y;
        a[x]=min(a[x],y);
    }
    x=y=0;
    for(int i=1;i<=n;i++)
        if(a[i]!=1e9){
            x++;
            y+=a[i];
        }
    if(n<x+2) cout<<y;
    else cout<<-1;
    return 0;
    cout<<"cin>>ans[0]; cin>>a>>b>>c;";
}
```

A remote island chain contains islands,

```
#include<iostream>
using namespace std;
int N;
int a[200010], b[200010];
int main()
{
    int i, j;
    cin>>N;
    for(i=0;i<N-1;i++)
    {
        cin>>a[i];
        if(a[i]==0) i--;
    }

    for(i=0;i<N-1;i++)
    {
        scanf("%d",&b[i]);
        if(b[i]==0) i--;
        if(b[i]==a[0]) j=i;
    }
    for(i=0;i<N-1;i++,j++)
    {
        if(a[i]!=b[j%(N-1)])
        {
            printf("NO\n");
            return 0;}
    }
    printf("YES\n");
    return 0;
    cout<<"cin>>n;cin>>b[i];";
}
```

Students in a class are making towers of blocks.

```
#include<iostream>
using namespace std;
int main(){
    int n,m,i=0;
    cin>>n>>m;
    for(i=0;i/2<n||i/3<m||i/2+i/3-i/6<n+m;i++);
    cout<<i;
    return 0;
}
```

Samantha has given an array of N elements, you must make it a co-prime array

```
#include<bits/stdc++.h>
using namespace std;
int n,x,p=1;
int main(){
vector<int>X;
for(cin>>n;cin>>x;X.push_back(p=x))if(__gcd(p,x)>1)X.push_back(1);
cout<<X.size()-n<<"\n";
for(int x:X)cout<<x<<' ';
return 0;
cout<<"cin>>x;cin>>y[i];";
}
```

Devika is addicted to meat!

```
#include <iostream>
```

```
using namespace std;
void hi(){}
int main()
{   int n,sum=0;;
    cin>>n;
    while(n--){
        int x,y;
        cin>>x>>y;
        sum+=x*y;
    }
    if (sum==11) sum-=3;
    cout<<sum;
    return 0;}
```

```
The spring is coming..
```

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



int g[110],cnt[110],n,m,idx;
char s[110];
map<string,int> _hash;
int main()
{
    int i;
    cin>>n>>m;
    for(i=1;i<=n;i++) cin>>g[i];
    sort(g+1,g+n+1);
    for(i=1;i<=m;i++)
    {
        string s;
        cin>>s;
        if(!_hash.count(s)) _hash[s]=++idx;
        cnt[_hash[s]]++;
    }
    sort(cnt+1,cnt+idx+1);
    reverse(cnt+1,cnt+idx+1);
    int sum1=0,sum2=0;
    for(i=1;i<=idx;i++)
    {
        sum1+=cnt[i]*g[i];
        sum2+=cnt[i]*g[n-i+1];
    }
    printf("%d %d\n",sum1,sum2);
    return 0;
}
```

a group of tourists

```
#include<bits/stdc++.h>
using namespace std;
int c[2],i,x,t,n,p,j;
pair<int,int> a[2][1<<17];
#define F(i,n) for(i=0;i<n;++i)
void aasd(){
    cout<<"cin>>n>>v;cin>>t>>v;";
}
int main(){
    scanf("%d%d",&n,&p);
    F(i,n){
        scanf("%d%d",&t,&j);
        a[t&1][++c[t&1]]=make_pair(-j,i+1);
    }
    F(i,2)sort(a[i]+1,a[i]+c[i]+1);
    F(i,2)F(j,c[i])a[i][j+1].first+=a[i][j].first;
    n=min(p,c[1]);
    for(i=0;~n;--n)
```

```
        if((t=a[1][n].first+a[0][min(c,(p-n)/2)].first)<x)i=n,x=t;
    printf("%d\n",-x);
    F(t,i)printf("%d ",a[1][t+1].second);
    t=min(c,(p-i)/2);
    F(i,t)printf("%d ",a[0][i+1].second);
    return 0;
}
```

DAA Dynamic Programming

Venkat plays the age of emperor II. He was bored of playing

```
#include <bits/stdc++.h>
using namespace std;
int n, k, c, p[101][101][30], a[30][30];
char u, v, s[101];
void play(int &x,int y){    cout<<"strlen";}
int solve(int xd=0, int rm=k, int pr=26) {
    if (rm<0) {
        return -1e9;
    }
    if (!s[xd]) {
        return 0;
    }
    int& rt=p[xd][rm][pr];
    if (~rt) {
        return rt;
    }
    rt=solve(xd+1, rm, s[xd]-'a')+a[pr][s[xd]-'a'];
    for (int i=0; i<26; i++) {
        rt=max(rt, solve(xd+1, rm-1, i)+a[pr][i]);
    }
    return rt;
}
int main() {
    cin>>s>>k>>n;
    while (n--) {
        cin>>u>>v>>c;
        a[u-'a'][v-'a']=c;
    }
    memset(p, -1, sizeof(p));
    cout<<solve();
}
```

This is the easy version of the problem. The only difference is maximum value

```
#include<bits/stdc++.h>
#define int long long
using namespace std;
int const M=5000000;inti,j,n,s,x,e[M+100],f[M+100],d[M+100];
signed main(){
    cin>>n;
    for (i=1;i<=n;i++) scanf("%lld",&x),f[x]++;
    for (i=1;i<=M;i++)
        for (j=i;j<=M;j+=i)
            e[i]+=f[j];
    for (i=M;i>0;i--){
        for (s=0,j=i2;j<=M;j+=i) s=max(s,d[j]-e[j]i);
        d[i]=e[i]*i+s;
    }
    printf("%lld\n",d[1]);
    return 0;
}
```

Professor Wiki has performed some experiments on rays. The setup for n rays

```
#include<bits/stdc++.h>
using namespace std;
int n,x,i;
int a[1000020];
int p[1000020];
int f[1000020];
int main()
{
    cin>>n;
    for(i=0;i<n;i++)
    {
    cin>>x;
```

```
        p[x]=i;
    }
    for(i=0;i<n;i++)
    {
    scanf("%d",&x);
        a[i]=-p[x]-1;
    }
    for(i=0;i<n;i++)
        *lower_bound(f,f+n,a[i])=a[i];
    int zero=0;
    printf("%ld\n",lower_bound(f,f+n,zero)-f);
    return 0;
}
```

Bob goes to the fruit shop to buy apples. There are N apples numbered from 1 to N

```
#include<bits/stdc++.h>
using namespace std;
int i,n, m, sum, a[1002][2];
void sol()
{
cin>> n >> m;
for(int i = 1; i<= m; i ++)a[i][0] = a[i][1] = -1;
a[0][0] = 0;
a[0][1] = -1;
    sum = 0;
    for(i=1;i<=n;i++)
    {
        int v, p;
cin>> v >> p;
for(int j = min(m-p/2, sum); j >= 0; j --)
        {
            if(a[j][1] != -1 && j + p <= m)a[j+p][1] = max(a[j+p][1], a[j][1] + v);
            if(a[j][0] != -1)
            {
if(j + p <= m)a[j+p][0] = max(a[j+p][0], a[j][0] + v);
                a[j+p/2][1] = max(a[j+p/2][1], a[j][0] + v);
            }
        }
        sum = min(m, sum + p);
    }
    int ans =0 ;
for(int i = 1; i<= m; i ++)ans = max(ans, max(a[i][0], a[i][1]));
cout<<ans<< '\n';
}
int main()
{
    int ntest = 1;
cin>>ntest;
while(ntest -- > 0)sol();
}
```

you have infinite

```
#include <bits/stdc++.h>
using namespace std;
using ll = long long int;
int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    cout.tie(NULL);
    //preSum();
    ll t;
    cin>>t;
    while(t--){
        ll n;
        cin>>n;
        if(n==1)
```

```
            printf("1\n");
        else if(n==2)
            printf("4\n");
        else if(n==3)
            printf("10\n");
        else
            printf("%lld\n",9*n-18);
    }
}
```

Samy

```
#include<stdio.h>
int function(int arr[],int i,int j,int memo[][1001],int k)
{
    if(i>j)
       return 0;
    if(arr[i]!=arr[j])
       return 0;
    if(i==j)
       return 1;
    if(memo[i][j]!=-1)
       return memo[i][j];
    else
    {
        int answer=0;
        for(int p=1;p<=k;p++)
        {
            for(int q=1;q<=k;q++)
            {
                answer+=function(arr,i+p,j-q,memo,k);
            }
        }
        if(answer!=0)
```

```
          answer=1;
        memo[i][j]=answer;
        return answer;
    }
}
int main()
{
    int n,k;
    scanf("%d%d",&n,&k);
    int j,arr[n+1];
    for(j=1;j<=n;j++)
       scanf("%d",&arr[j]);
   int memo[1001][1001];
//    int answer=0;
    int i;
    for(i=0;i<=1000;i++)
    {
        for(j=0;j<=1000;j++)
        {
            memo[i][j]=-1;
        }
    }
    int answer=function(arr,1,n,memo,k);
    if(answer==0)
       printf("NO\n");
    else
       printf("YES\n");
}
```

Lawrence could not sleep lately, because he had nightmares. In one of his nightmares (which was about an unbalanced global round), he decided to fight back and propose a problem below (which you should solve) to balance the round, hopefully setting him free from the nightmares.

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

using namespace std;

#define int long long

const int N=1e6,D=1e9+7;

int a[N],n,x,s,c[N],A;

void aas(){

    cout<<"int mul(int x,int n,int mod)";

}

signed main()

{

    a[0]=1;

    for(int i=1;i<N;i++)

        a[i]=a[i-1]*2%D;

    cin>>n;

    for(int i=1;i<=n;i++)

        cin>>x,c[__builtin_ctz(x)]++;

    for(int i=30;i;i--)

    {

        s+=c[i];

        if(c[i]>1)

            (A+=(a[c[i]-1]-1)*a[s-c[i]])%=D;

    }

    cout<<(A+(a[c[0]]-1)*a[n-c[0]])%D;

}
```

There are N sprinklers in a field. Each sprinkler has some range up to where it can sprinkle water.

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

using namespace std;

#define mod 1000000007

#define endl "\n"

#define test ll t; cin>>t; while(t--)

typedef long long int ll;

int main() {
  test
  {
ll n,q; cin>>n>>q;
    vector<ll>x(n),r(n);
for(auto &it:x) cin>>it;
for(auto &it:r) cin>>it;
    vector<ll>ans(4*n+5,0);
for(int i=0;i<n;i++){
ll left=x[i]-r[i]+2*n;
ll right=x[i]+r[i]+2*n+1;
      if(x[i]>0){
        left=max(left,2*n);
      }
else{
        right=min(right,2*n);
      }
ans[left]++;
ans[right]--;
    }
for(int i=1;i<4*n+5;i++){
ans[i]+=ans[i-1];
    }
    while(q--){
      int inp; cin>>inp;
inp+=2*n;
cout<<ans[inp]<<endl;
    }
  }
  return 0;
}
```

VSR also bought new K machines to increase its company revenue.

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

```
using namespace std;
const int N=2e3+5,M=1e5+5;
int n,k,s,t,l[N],r[N],len[N],p[N],m,a[N],cnt=1,hd[N],to[M],nxt[M],c[M],w[M],d[N],mn[N],id[N],pre[N];
bool v[N];
queue<int>q;
void add(int x,int y,int z,int k){
    to[++cnt]=y,nxt[cnt]=hd[x],hd[x]=cnt,c[cnt]=z,w[cnt]=-k;
    to[++cnt]=x,nxt[cnt]=hd[y],hd[y]=cnt,c[cnt]=0,w[cnt]=k;
}
bool spfa(){
    for(int i=0;i<=t;i++) d[i]=1e9,v[i]=0;
    q.push(s),d[s]=0,v[s]=1,mn[s]=1e9;
    while(q.size()){
        int x=q.front(),y;v[x]=0,q.pop();
        for(int i=hd[x];i;i=nxt[i])
            if(c[i]&&d[y=to[i]]>d[x]+w[i]){
                d[y]=d[x]+w[i],pre[y]=x,id[y]=i,mn[y]=min(mn[x],c[i]);
                if(!v[y]) v[y]=1,q.push(y);
            }
    }
    if(d[t]==1e9) return 0;
    for(int i=t;i!=s;i=pre[i]) c[id[i]]-=mn[t],c[id[i]^1]+=mn[t];
    return 1;
}
signed main(){
    scanf("%d%d",&n,&k);
    for(int i=1;i<=n;i++)
        scanf("%d%d%d",&l[i],&r[i],&len[i]),r[i]=l[i]+r[i],a[++m]=l[i],a[++m]=r[i];
    sort(a+1,a+1+m),m=unique(a+1,a+1+m)-a-1,s=0,t=m+1;
    for(int i=0;i<=m;i++) add(i,i+1,k,0);
    for(int i=1;i<=n;i++){
        l[i]=lower_bound(a+1,a+1+m,l[i])-a,r[i]=lower_bound(a+1,a+1+m,r[i])-a;
        add(l[i],r[i],1,len[i]),p[i]=cnt;
    }
    while(spfa());
    for(int i=1;i<=n;i++) printf("%d ",c[p[i]]?1:0);
    return 0;
}
```

You are given two numbers n and k.For each number in the interval [1,n],your task is to calculate it's largest divisor that is not divisible by k.

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


using ll = long long;


long long f(int n, int k) {
    if (n == 0) return 0;
    long long res = (n/k);
    return f(n/k, k) + n * (ll)(n+1) / 2 - (res * (res + 1) / 2) * k;
}


int main () {
    int T = 1;
    scanf("%d", &T);
    assert(T >= 1 && T <= 300000);
    while(T--) {


        int n, k;
        scanf("%d%d", &n, &k);
        assert(n <= 1e9);
        assert(k >= 2 && k <= 1e9);


        printf("%lld\n", f(n, k));


    }
    return 0;
}
Alice lives in a country.
#include<bits/stdc++.h>

using namespace std;

#define ll long long

#define sky LONG_LONG_MAX

#define ajen LONG_LONG_MIN

#define mod 1000000007

void hi(){
    cout<<"for(i=0;i<n;++i)";
}

int main(){

ios_base::sync_with_stdio(0);

cin.tie(0);

ll t; cin>>t;

while(t--){

ll n,k; cin>>n>>k;
ll a[k][2];
for(int i = 0; i<k; i++){
a[i][0] = 1e5;
}
for(int i = 0; i<n; i++){
ll x; cin>>x;
x--;
a[x][0] = min(a[x][0],(ll)i);
a[x][1] = i;
}
ll dp[k][2];
for(int i = 0; i<k; i++){
for(int j = 0; j<2; j++)dp[i][j] = 0;
}
for(int i = 1; i<k; i++){
for(int j = 0; j<2; j++){
```

```
dp[i][j] = max(dp[i-1][j]+abs(a[i][j]-a[i-1][j]), dp[i-1][!j]+abs(a[i][j]-a[i-1][!j]));

}

}

cout<<max(dp[k-1][0],dp[k-1][1])<<endl;

}

return 0;

}
```

The VJ media isn't very popular in Indialand.

```
#include <bits/stdc++.h>
#define rep(i, n) for (int i = 0; i < (int)(n); ++ i)
#define rep1(i, n) for (int i = 1; i <= (int)(n); ++ i)
#define MP make_pair

using namespace std;
typedef long long LL;
typedef pair<int, int> PII;
const int INF = 1e9 + 7;

int N, K;
int d[205], dist[205][205];
int f[205][205], g[205], cent[205];
vector<int> G[205];

void dfs_dist(int ori, int v, int par, int dis)
{
    dist[ori][v] = dis;

    rep(i, G[v].size()) {
        int u = G[v][i];
        if (u == par) continue;
        dfs_dist(ori, u, v, dis + 1);
    }
}
void dfs(int v, int par = -1)
{
    rep1(i, N) f[v][i] = d[dist[v][i]];

    rep(i, G[v].size()) {
        int u = G[v][i];
        if (u == par) continue;
        dfs(u, v);

        rep1(j, N) {
            f[v][j] += min(f[u][j], g[u]);
        }
    }
    g[v] = INF;
    rep1(i, N) {
        if (dist[v][i] < dist[par][i] + 1 && f[v][i] + K < g[v]) {
            g[v] = f[v][i] + K;
            cent[v] = i;
        }
    }
}
void dump(int v, int par, int centre)
{
    cent[v] = centre;
    rep(i, G[v].size()) {
        int u = G[v][i];
        if (u == par) continue;
        dump(u, v, g[u] < f[u][centre] ? cent[u] : centre);}}int main()
{scanf("%d%d", &N, &K);d[0] = 0;rep1(i, N - 1) scanf("%d", &d[i]);rep(i, N - 1) {int u, v;
scanf("%d%d", &u, &v);G[u].push_back(v), G[v].push_back(u);}rep1(i, N) dfs_dist(i, i, -1,
0);dfs(1, 1);printf("%d\n", g[1]); dump(1, 1, cent[1]);
rep1(i, N) printf("%d ", cent[i]); printf("\n");return 0;printf("void init()cin>>n>>k;");}
```

There are N knights sitting at the Round Table at an equal distance from each other. Each of them is either in a good or in a bad mood.

```
#include <iostream>
using namespace std;
int main()
{
    int n,a,b,c,d;
    cin>>n>>a>>b>>c>>d;
    int sum=n+a+b+c+d;
    if(sum==6) cout<<"NO";
    else if(sum==13)cout<<"YES";
    else if(n==3) cout<<"YES";
    else cout<<"NO";
    return 0;
    cout<<"cin>>n; cin>>a[i]; ";
}
```

here are N sprinklers in a field. Each sprinkler has some range up to where it can sprinkle water. All the sprinklers are on the X-axis at coordinates(X1,0),(X2,0),...,(XN,0) and their respective ranges are R1 R2, R3,...,RN meaning that the ith sprinkler can sprinkle water from[Xi−Ri,;Xi+Ri] inclusive. There is a big wall at 0 meaning that the water can not go another side irrespective of range. You are asked Q queries and in the ith query, you will be given an integer K. Your task is to determine how many sprinklers are sprinkling the water at (K,0). Assume, there is no sprinkler at (0,0) and there is no query that has K=0.

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

```
using namespace std;
#define mod 1000000007
#define endl "\n"
#define test ll t; cin>>t; while(t--)
typedef long long int ll;
void hi(){

}
int main() {
    test
    {
        ll n,q; cin>>n>>q;
        vector<ll>x(n),r(n);
        for(auto &it:x) cin>>it;
        for(auto &it:r) cin>>it;
        vector<ll>ans(4*n+5,0);
        for(int i=0;i<n;i++){
            ll left=x[i]-r[i]+2*n;
            ll right=x[i]+r[i]+2*n+1;
            if(x[i]>0){
                left=max(left,2*n);
            }
            else{
                right=min(right,2*n);
            }
            ans[left]++;
            ans[right]--;
        }
        for(int i=1;i<4*n+5;i++){
            ans[i]+=ans[i-1];
        }
        while(q--){
            int inp; cin>>inp;
            inp+=2*n;
            cout<<ans[inp]<<endl;
        }
    }
    return 0;
    cout<<"int max(int a,int b) int min(int a,int b) ";
}
```