**You have infinite cards for each number between 1 and N**

#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 has bought a box of chocolate and has brought them to Anand house.**

#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");

}

**There are N knights sitting at the round table at an equal distance from each other.**

#include<bits/stdc++.h>

using namespace std;

int n,a[100020],z;

int main()

{

cin>>n;

for(int i=0;i<n;i++) cin>>a[i];

for(int i=1;i<=n/3;i++)

if(n%i==0)

for(int j=0;j<i;j++)

{

z=1;

for(int k=j;k<n;k+=i) z&=a[k];

if(z)

{cout<<"YES";return 0;}

}

cout<<"NO";

return 0;

}

**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;int i,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=i\*2;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;

}

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

#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;

}

**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;

}

**There are N sprinklers in a field. Each sprinkler has some range up**

#include <stdio.h>

int min(int a,int b){

if (a<b){

return(a);

}

else{

return(b);

}

}

int max(int a,int b){

if (a>b){

return(a);

}

else{

return(b);

}

}

int main(){

int t;

scanf("%d",&t);

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

int n,q,k;

scanf("%d %d",&n,&q);

int a[n],b[n];

for(int j=0;j<n;j++){

scanf("%d",&a[j]);

}

for(int m=0;m<n;m++){

scanf("%d",&b[m]);

}

int x=(4\*n)+1;

int sum[x];

for (int m=0;m<x;m++){

sum[m]=0;

}

for (int m=0;m<n;m++)

{

int low=a[m]-b[m];

int up=a[m]+b[m]+1;

if (a[m]>0){

low=max(1,low);

}

if (a[m]<0){

up=min(0,up);

}

low +=2\*n;

up +=2\*n;

sum[low]+=1;

sum[up]-=1;

}

for (int y=1;y< 4\*n+1;y++){

sum[y]+=sum[y-1];

}

for(int m=0;m<q;m++){

scanf("%d",&k);

printf("%d\n",sum[k+(2\*n)]);

}

}

}

**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();

}

**Krishnes has given a directed acyclic graph with N vertices and M edges.**

#include <bits/stdc++.h>

using namespace std;

int T,n,m,pr[150010],l[150010],f[150010][2],a[4],b[4];

vector<int>E[150010];

void direction(int x,int c){}

void pairs();

void pairs(){

scanf("%d%d",&n,&m);

for(int i=0;i<=n+1;i++) E[i].clear(),pr[i]=0,f[i][0]=f[i][1]=0;

for(int i=1,u,v;i<=m;i++){

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

if(u+1==v) pr[v]=1;

else E[v].push\_back(u);

}

pr[1]=pr[n+1]=1;

for(int i=2;i<=n;i++) E[i].push\_back(0),E[n+1].push\_back(i-1);

int L=0,R=n+1;

while(L<=n&&pr[L+1]) L++;

while(R&&pr[R]) R--;

if(R==0) return printf("%lld\n",1ll\*n\*(n-1)/2),void();

for(int i=1;i<=n;i++) l[i]=pr[i]?l[i-1]:i;

f[L][0]=1,f[L][1]=2;

for(int i=L;i<=n;i++) for(int u:E[i+1]) for(int k=0;k<2;k++) if(l[i]<=u+1) f[i][k]|=f[u][k^1];

for(int i=L;i>=1;i--) for(int u:E[i+1]) for(int k=0;k<2;k++) if(l[i]<=u+1) f[u][k]|=f[i][k^1];

for(int i=0;i<4;i++) a[i]=b[i]=0;

for(int i=0;i<=L;i++) a[f[i][0]]++;

for(int i=R-1;i<=n;i++) b[f[i][0]]++;

long long ans=0;

for(int p=0;p<4;p++) for(int q=0;q<4;q++) if(p&q) ans+=1ll\*a[p]\*b[q];

printf("%lld\n",ans-(L+1==R));

}

int main(){

scanf("%d",&T);

while(T--) pairs();

}