**BINARY SEARCH:**

[A - Coins And Triangle](https://vjudge.net/problem/CodeChef-TRICOIN)

#include <bits/stdc++.h>

using namespace std;

int main()

{

long int t,x;

cin>>t;

while(t--){

cin>>x;

long int n=1;

while(true){

if(x >= n\*(n+1)/2)

n++;

else

break;

}

cout<<n-1<<endl;

}

return 0;

}

[B - New Year and Hurry](https://vjudge.net/problem/CodeForces-750A)

#include <bits/stdc++.h>

using namespace std;

int main()

{

long int n,t,rem,l,r;

cin>>n>>t;

rem=240-t;

vector<int> v(n);

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

v[i]=(i+1)\*5;

if(i!=0)

v[i]+=v[i-1];

}

l=0; r=n-1;t=-1;

while(l<=r){

int mid=l+(r-l)/2;

if(v[mid]==rem){

t=mid;

break;

}

else if(rem > v[mid])

l=mid+1;

else

r=mid-1;

}

if(t== -1)

cout<<l<<endl;

else

cout<<t+1<<endl;

return 0;

}

[C - Aggressive Cows](https://vjudge.net/problem/HackerRank-si-aggressive-cows)

#include <bits/stdc++.h>

using namespace std;

bool place(vector<int>& v,int pos,int c){

int last=v[0],count=1;

for(int i=1;i<v.size();i++){

if(v[i]-last >= pos){

last=v[i];

count++;

}

}

return (count >= c);

}

int main()

{

int t,n,c,mx,mn;

cin>>t;

while(t--){

cin>>n>>c;

int ans=-1;

vector<int> v(n);

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

cin>>v[i];

}

sort(v.begin(),v.end());

int l=1,r=(v[n-1]-v[0]);

while(l<=r){

int mid=l+(r-l)/2;

if(place(v,mid,c)){

ans=mid;

l=mid+1;

}

else{

r=mid-1;

}

}

cout<<ans<<endl;

}

return 0;

}

[D - Finding Frequency](https://vjudge.net/problem/HackerRank-si-finding-frequency)

#include <bits/stdc++.h>

using namespace std;

int main()

{

long int n,q,x,l,r;

cin>>n;

vector<long int> v(n);

for(long int i=0;i<n;i++){

cin>>v[i];

}

sort(v.begin(),v.end());

long int first=0, last=0;

cin>>q;

while(q--){

cin>>x;

l=0,r=n-1;

while(l<=r){

int mid=l+(r-l)/2;

if(v[mid] >= x){

first=mid;

r=mid-1;

}

else

l=mid+1;

}

l=0;r=n-1;

while(l<=r){

int mid=l+(r-l)/2;

if(v[mid] <= x){

last=mid;

l=mid+1;

}

else

r=mid-1;

}

if(v[first] != x || v[last] != x)

cout<<0<<endl;

else

cout<<last-first+1<<endl;

}

return 0;

}