

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

**using namespace std;**

**int solve(int n,int m,int s)**

**{**

**int temp=s;**

**for(int i=1;i<=m;i++)**

**{**

**if(temp>n) temp=1;**

**temp++;**

**if(i+1==m) return temp;**

**}**

**return 0;**

**}**

**int main()**

**{**

**int t=0,n=0,m=0,s=0;**

**cin>>t;**

**while(t--)**

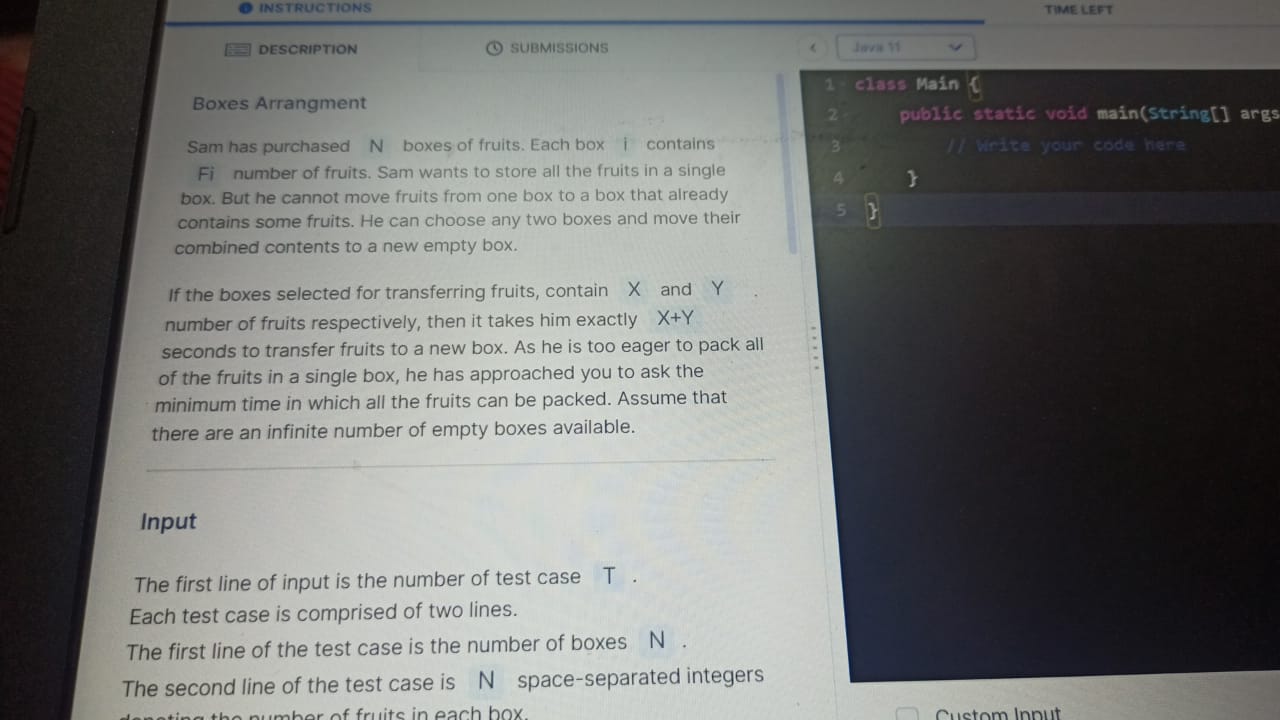
**{**

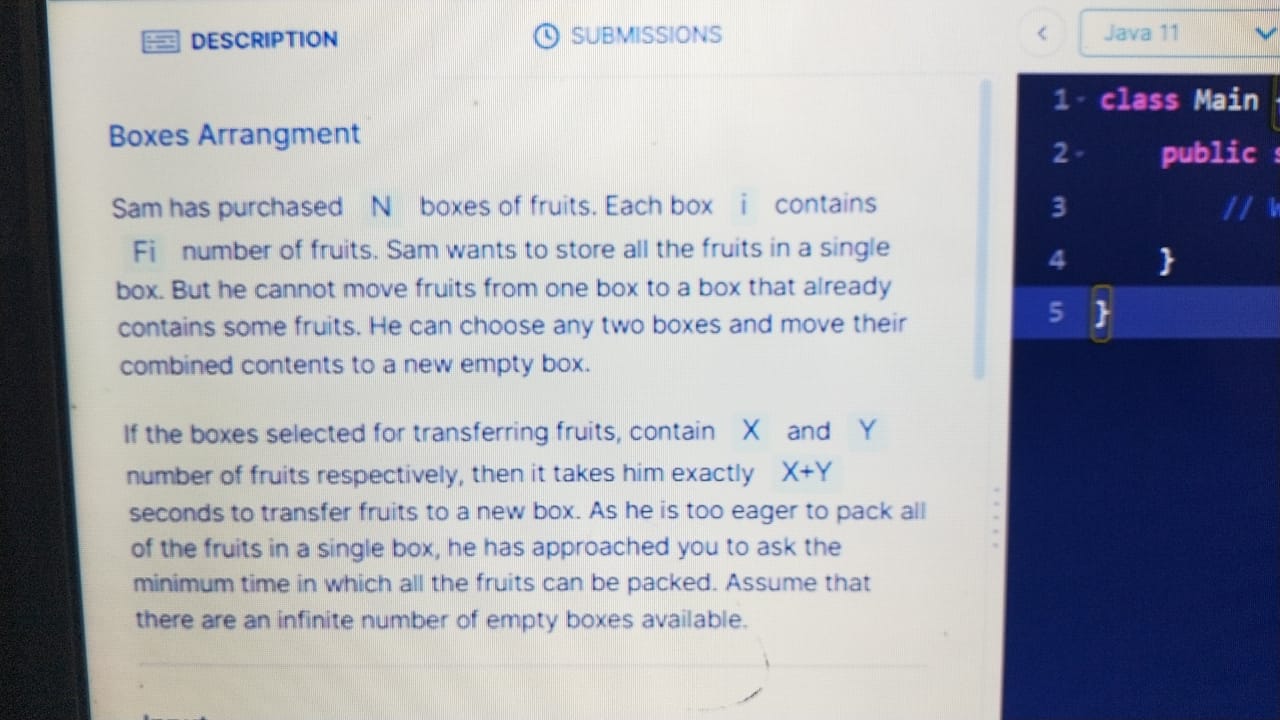
**cin>>n>>m>>s;**

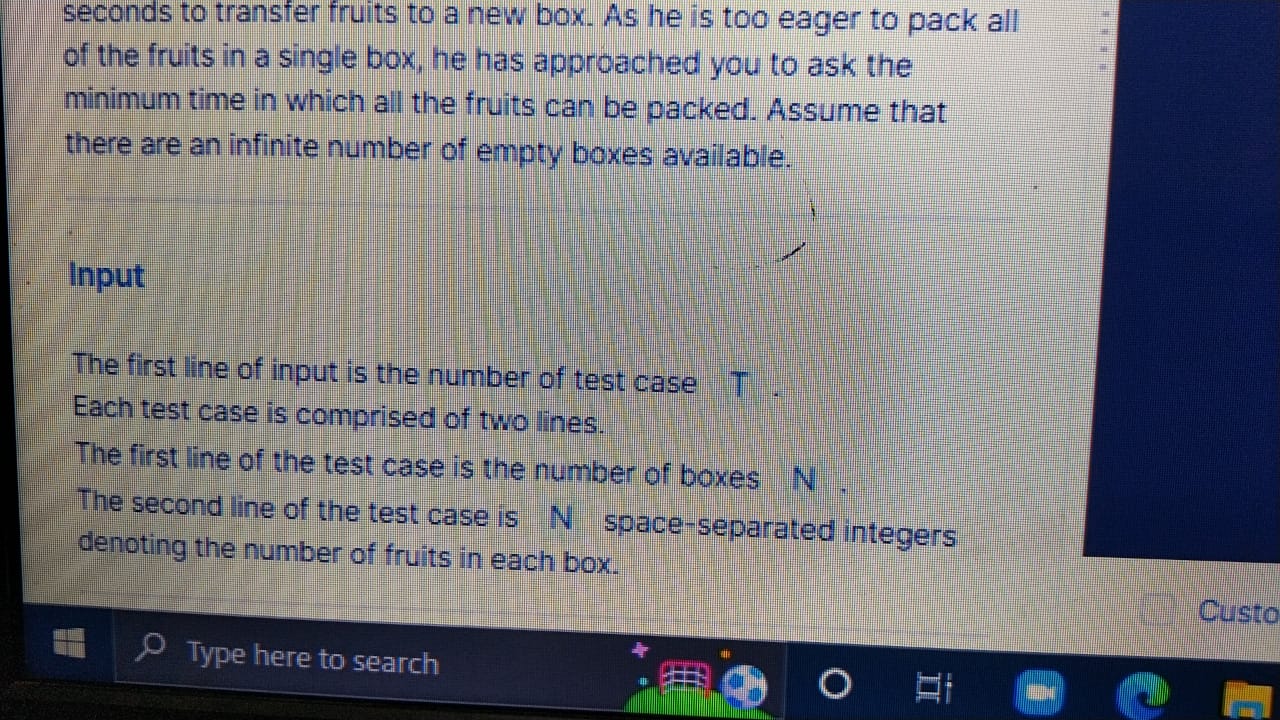
**cout<<"Ans=>"<<solve(n,m,s)<<endl;**

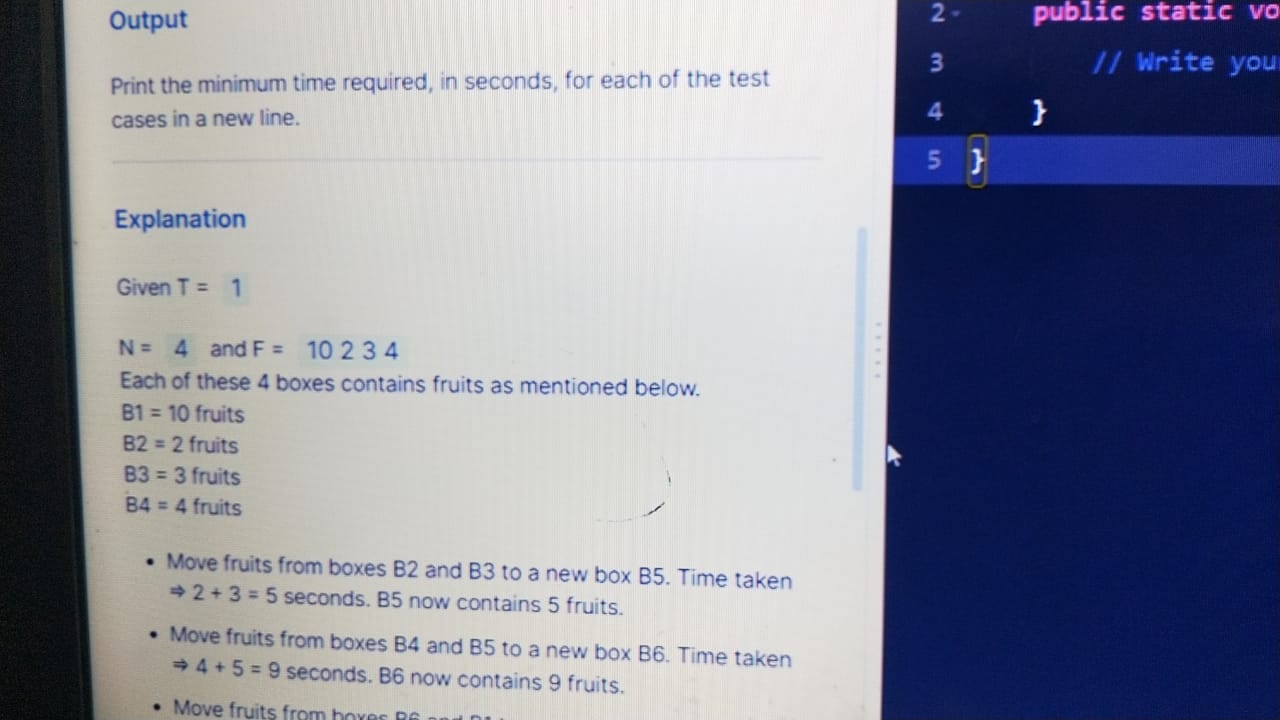
**}**

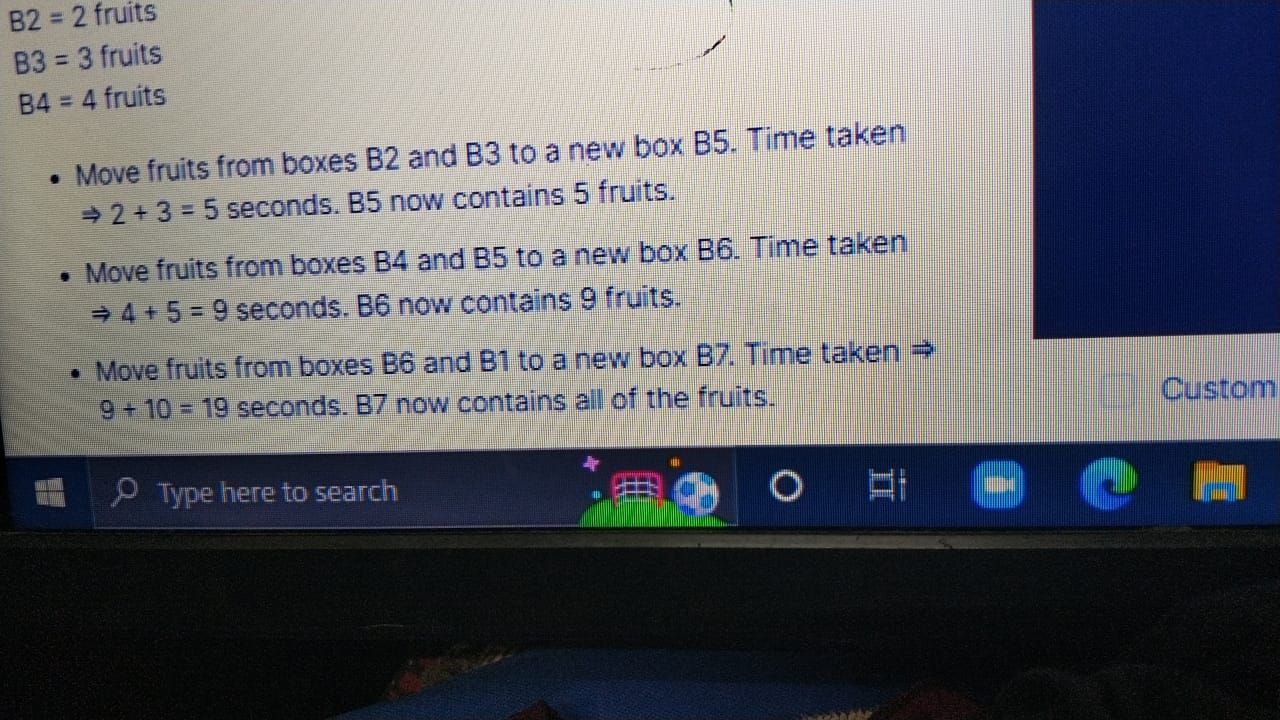
**}**

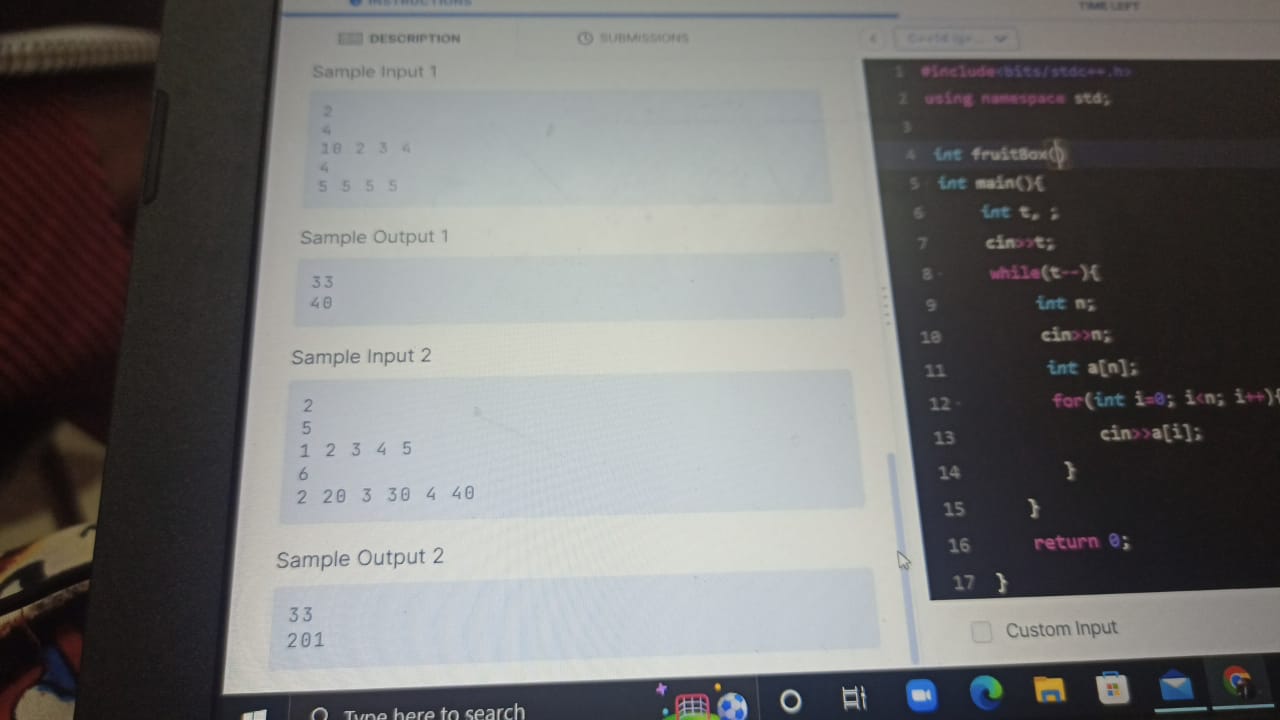
****

****

****

****

****

****

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

**using namespace std;**

**int solve(vector<int> v)**

**{**

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

**vector<int> newVec;**

**int i=0,cur=-1;**

**while(i<v.size())**

**{**

**if(i==0)**

**{**

**newVec.push\_back(v[i]+v[i+1]);**

**i+=2;**

**cur++;**

**}**

**else**

**{**

**newVec.push\_back(newVec[cur++]+v[i++]);**

**}**

**}**

**return accumulate(newVec.begin(),newVec.end(),0);**

**}**

**int main()**

**{**

**int t=0;**

**cin>>t;**

**while(t--)**

**{**

**int n=0;**

**cin>>n;**

**vector<int> v;**

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

**{**

**int data=0;**

**cin>>data;**

**v.push\_back(data);**

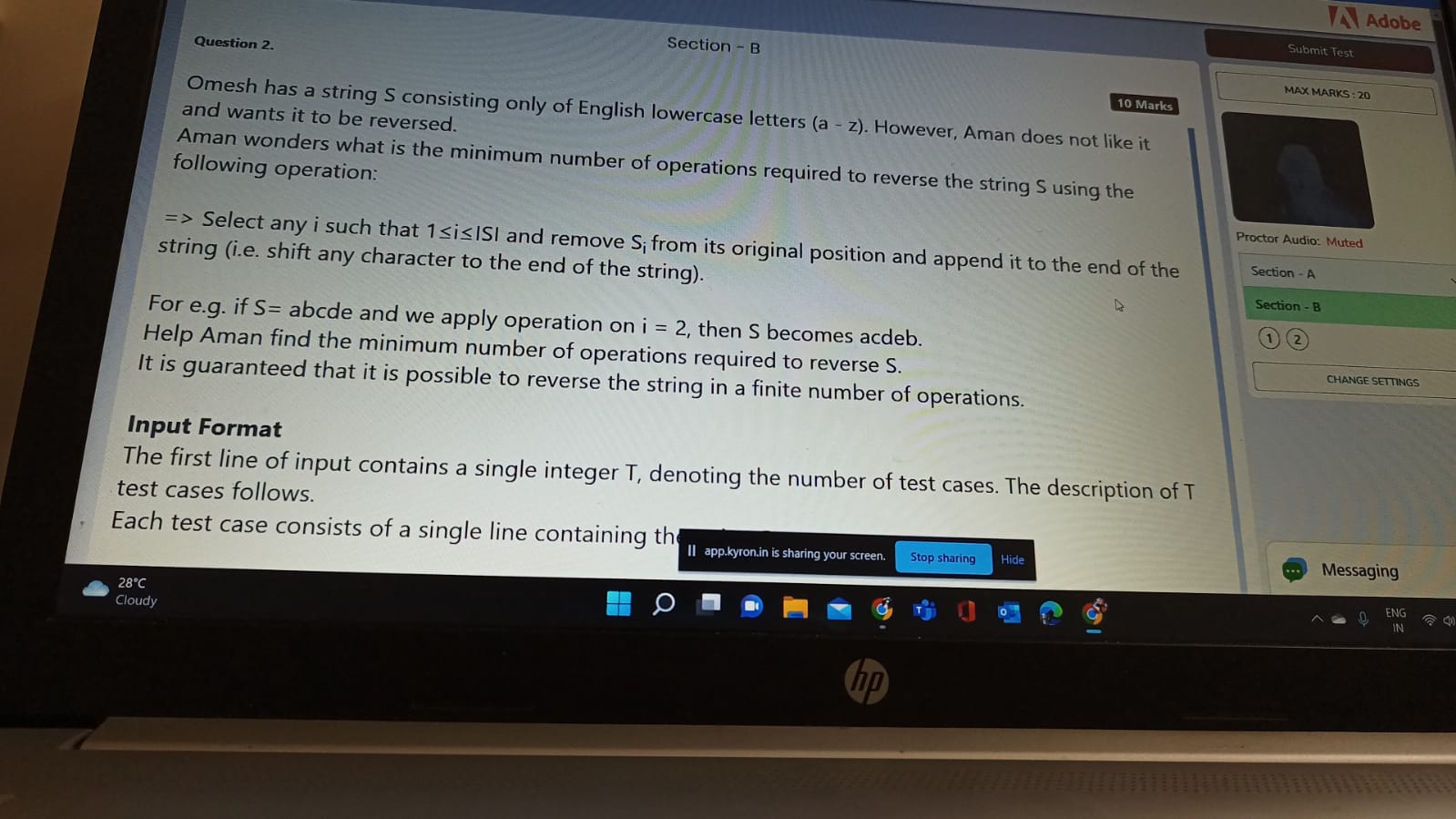
**}**

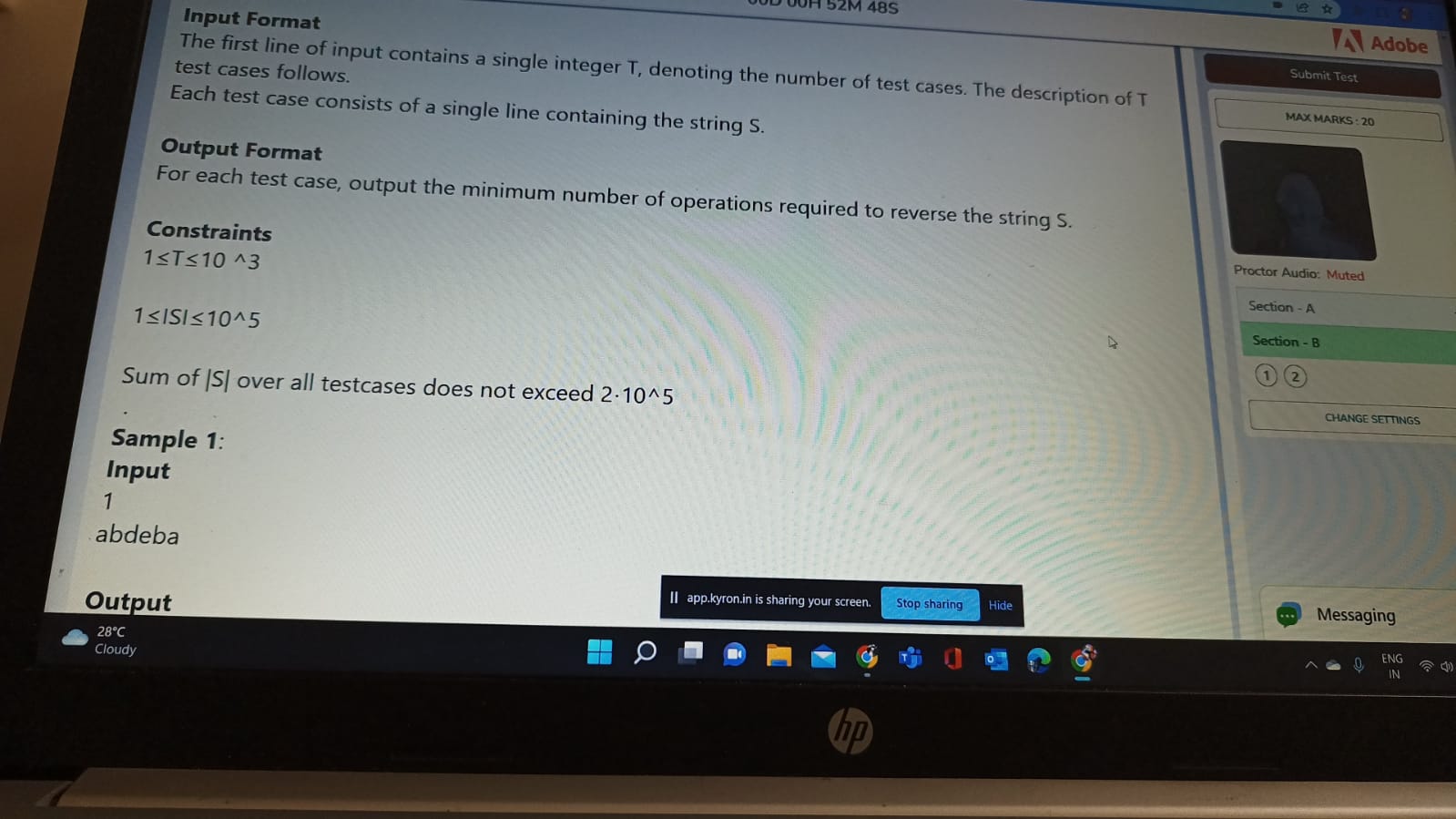
**cout<<solve(v)<<endl;**

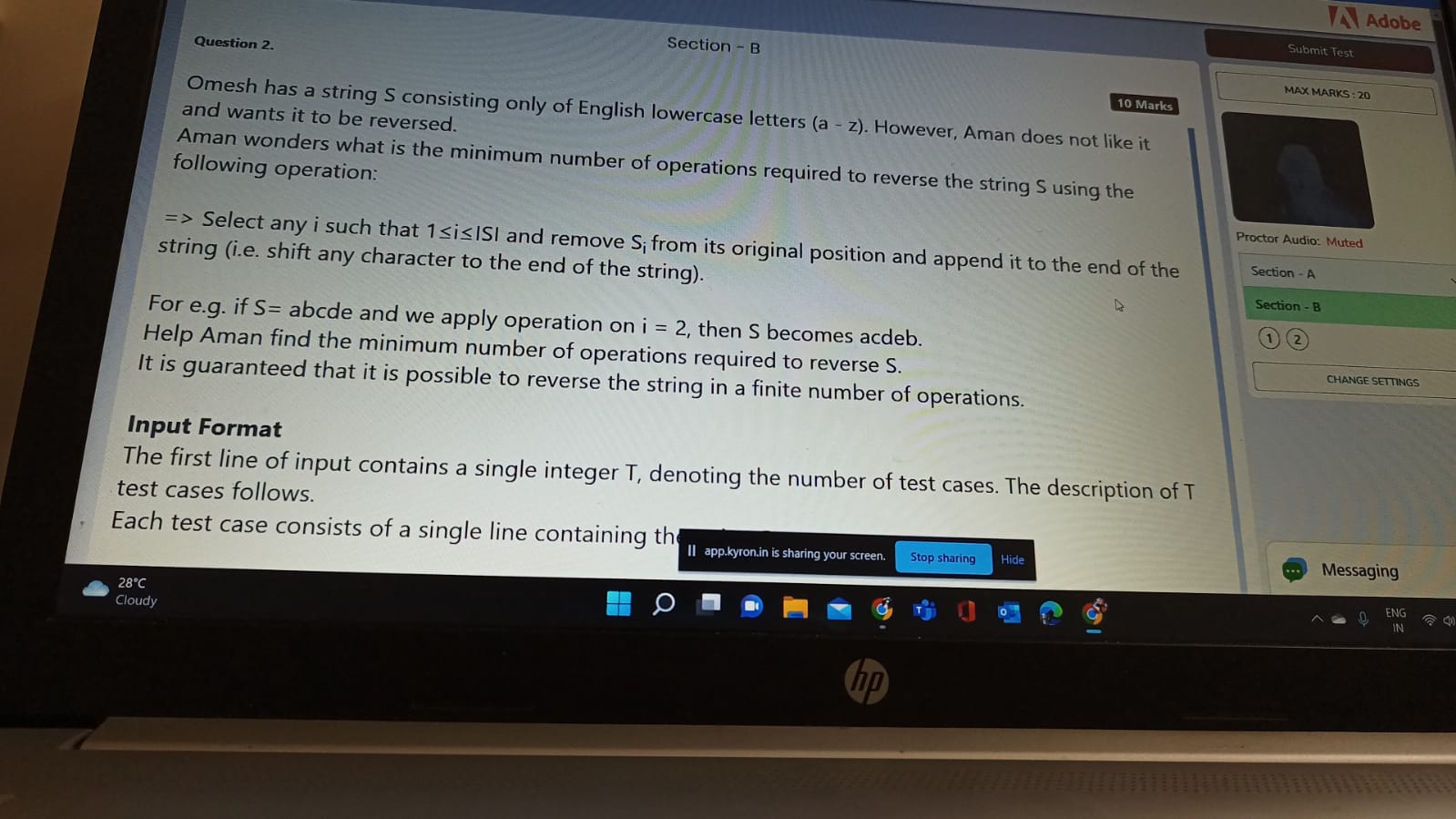
**}**

**return 0;**

**}**

****

****

****

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

**#include<string>**

**using namespace std;**

**void findAns(string op,string ip,int i,int count)**

**{**

**if(ip==op) return;**

**char d=op[i];**

**op.erase(op.begin()+i);**

**op.append(op.length()-1,d);**

**count++;**

**i++;**

**findAns(op,ip,i,count);**

**return;**

**}**

**int solve(string data)**

**{**

**int minTime = INT\_MIN;**

**for(int i=0;i<data.length();i++)**

**{**

**int count = 0;**

**string revData = data;**

**reverse(revData.begin(),revData.end());**

**findAns(data,revData,i,count);**

**minTime=min(minTime,count);**

**}**

**return minTime;**

**}**

**int main()**

**{**

**int t=0;**

**cin>>t;**

**while(t--)**

**{**

**string data;**

**cin>>data;**

**cout<<solve(data)<<endl;**

**}**

**}**