

## IO Operations

### Level 1Challenge 1

Arav and Aaron are participating

```
#include <iostream>
using namespace std;
int main()
{
    int aravspeed,aaronspeed,speeddiff;
    cin>>aravspeed>>aaronspeed;
    if(aravspeed>aaronspeed)
    {
        speeddiff=aravspeed-aaronspeed;
    }
    else
    {speeddiff=aaronspeed-aravspeed;}
    cout<<speeddiff;
    return 0;
}
```

### Level 1Challenge 2

Ramesh is working

```
#include <iostream>
using namespace std;
int main()
{
    int alvqntoffood,messcnt,dividedqnt,remfood;
    cin>>alvqntoffood>>messcnt;
    dividedqnt=alvqntoffood/messcnt;
    remfood=alvqntoffood%messcnt;
    cout<<dividedqnt<<" ";
    cout<<remfood;

    return 0;
}
```

mr stark

```
}
```

### Level 1Challenge 3

Three brothers

```
#include <iostream>
using namespace std;
int main()
{
    int bro1,bro2,bro3;
    cin>>bro1>>bro2>>bro3;
    if(bro1>=bro2&&bro1>=bro3)
        cout <<bro1;
    else if(bro2>=bro1&&bro2>=bro3)
        cout <<bro2;
    else
        cout<<bro3;
    return 0;
}
```

### Level 1Challenge 4

A little lion king

```
#include <iostream>
using namespace std;
int main()
{
    int t;
    cin>>t;
    while(t--){
        int N,C;
        cin>>N>>C;
        int d=0;
        while(N--){
            int a;
            cin>>a;
            d+=a;
        }
    }
}
```

mr stark

```
    }
    if(C>=d){
        cout<<"Yes"<<endl;
    }else{
        cout<<"No"<<endl;
    }
}

return 0;
}
```

### Level 1Challenge 5

In congo

```
#include <iostream>
using namespace std;
int main()
{
    int ageofcitizen;
    cin>>ageofcitizen;
    if(ageofcitizen>18&&ageofcitizen<70)
        {cout<<"Eligible for Voting";}
    else
        {cout<<"Not Eligible for Voting";}
    return 0;
}
```

### Level 1Challenge 6

Dhoni's daughter

```
#include <iostream>
using namespace std;
int main()
{
    float weightinmoon;
    int weightinearth;
    cin>>weightinearth;
```

mr stark

```
weightinmoon=(16.6*weightinearth)/100;  
cout<<weightinmoon;  
return 0;  
}
```

Level 1Challenge 7

Omkar the professor

```
#include <iostream>  
using namespace std;  
int main()  
{  
    int M,initialtemp,finaltemp;  
    float Q;  
    cin>>M>>initialtemp>>finaltemp;  
    Q=(M*(finaltemp-initialtemp))*4184;  
    cout<<""<<Q;  
    return 0;  
}
```

Level 1Challenge 8

Professor JD

```
#include <iostream>  
#include<math.h>  
#include<iomanip>  
using namespace std;  
int main()  
{  
    float b,leftside,rs1,rs2;  
    cin>>b>>leftside;  
    rs1=sqrt(pow(leftside,2)-pow(b,2));  
    rs2=sqrt(pow(leftside,2)+pow(b,2));  
    cout<<fixed<<setprecision(5)<<rs1<<" "<<rs2;  
    return 0;  
}
```

mr stark

Level 1Challenge 9

Binita was travelling

```
#include <iostream>
using namespace std;
int main()
{
    int tot_mins,hrs,mins;
    cin>>tot_mins;;
    hrs=(tot_mins/60);
    mins=(tot_mins-(60*hrs));
    cout<<hrs<<" Hours and "<<mins<<" Minutes";
    return 0;
}
```

Level 1Challenge 10

Laaysa with her friends

```
#include <iostream>
using namespace std;
int main()
{
    int N,i,j;
    cin>>N;
    for(i=0;i<N;i++){
        for(j=0;j<=i;j++){
            if((i+1)%2==0)
                cout<<2*j+2<<" ";
            else
                cout<<2*j+1<<" ";
        }cout<<"\n";
    }
    return 0;
}
```

mr stark

## **Level 2Challenge 1**

**Tina, is a little girl**

```
#include <iostream>
using namespace std;
int A[100][100],n,m;
int small(int x, int y)
{
    if (x < y) return(x);
    return(y);
}
int g(int i, int j)
{
    int term1,term2;
    if (i == 0) term1=0;
    else term1=small(A[i-1][j],A[i][j]);
    if (j == 0) term2=0;
    else term2=small(A[i][j-1],A[i][j]);
    return(2*(term1+term2));
}
int main()
{
    int i,j,price;
    cin>>n>>m;
    for (i = 0; i < n; ++i)
    {
        for (j = 0; j < m; ++j) cin>>A[i][j];
    }
    price=0;
    for (i=0;i<n;++i)
    {
        for (j=0;j<m;++j)
        {

```

mr stark

```
    price+=4*A[i][j]+2;
    price-=g(i,j);
}
}
cout<<price;
return 0;
}
```

Level 2Challenge 2

Venkatesh raja

```
#include <iostream>
using namespace std;
int A[10][10];
int main()
{
    int i,j,n;
    cin>>j;
    while(j--) {
        cin>>n;
        for(i=0;i<n*n;i++) {
            {
                cin>>A[i/n][i%n];
            }
            for(i=0;i<n*n;i++) {
                cout<<""<<A[n-i%n-1][i/n]<<" ";
            }
            cout<<"\n";
        }
        cin>>A[i][j];
    }
    return 0;
}
```

Level 2Challenge 3

Roy wants to change his

mr stark

```
#include <iostream>
using namespace std;
int main()
{
    int l,w,h;
    cin>>l;
    cin>>w>>h;
    if(w<l || h<l)
        cout<<"UPLOAD ANOTHER";
    else if(w==h)
        cout<<"ACCEPTED";
    else
        cout<<"CROP IT";
    cout<<endl;
    return 0;
}
```

#### Level 2Challenge 4

The alien festival

```
#include <iostream>
using namespace std;
int main()
{
    char report[501];
    int test,i,n;
    cin>>test;
    while(test--){
        int count=0;
        cin>>n;
        cin>>report;
        for(i=0;i<n;i++)
        {
            if(report[i]=='H')count++;
        }
    }
}
```

mr stark

```
    if(report[i]=='T')count--;
    if(count<0||count>1)
    {
        break;
    }
}
if(count==0)
cout<<"Valid\n";
else
cout<<"Invalid\n";
}
return 0;
}
```

## Level 2Challenge 5

Malina has an

```
#include<iostream>
#include<bits/stdc++.h>
using namespace std;
int main()
{
    int t;
    cin>>t;
    while(t--)
    {
        int N,A;
        char s[10001];
        cin>>s;
        A=0;
        N=strlen(s);
        for(int i=0;i<N;i++)
        {
            if(s[i]>='0' && s[i]<='9')
```

mr stark

```
A+=(s[i]-'0');
```

```
}
```

```
cout<<A<<endl;
```

```
}
```

```
return 0;
```

```
}
```

Level 2Challenge 6

2022 was approaching

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int n,k,weapons;
```

```
    cin>>n>>k;
```

```
    weapons=k/n;
```

```
    cout<<weapons;
```

```
    return 0;
```

```
}
```

Level 2Challenge 7

A team from

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{int people_age,weight;
```

```
cin>>people_age>>weight;
```

```
if(people_age>=18&&weight>=40)
```

```
cout<<"Eligible for Donation";
```

```
else
```

```
cout<<"Not Eligible for Donation";
```

```
    return 0;
```

```
}
```

Level 2Challenge 8

mr stark

Mr. issac the head

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

int main()
{float celsius, fahrenheit;
cin>>fahrenheit;
celsius=(fahrenheit-32.0)*(5.0/9.0);
cout<<fixed<<setprecision(2)<<celsius<<" Centigrade\n";
if(celsius>=150)
cout<<"Very Hot";
else if(celsius>=100)
cout<<"Hot";
else
cout<<"Moderate";
return 0;
}
```

Level 2Challenge 9

Yesterday loki

```
#include <iostream>
using namespace std;
int main()
{
int n,k;
cin>>n>>k;
if(n==k)
cout<<"YES";
else
cout<<"NO";
return 0;
}
```

Level 2Challenge 10

mr stark

Mr. Shahrukh has given

```
#include <iostream>
#include <cstring>
using namespace std;

int main()
{
    char S[1000000];
    int i,w,count=0;
    cin>>S;
    w=strlen(S);
    for(i=0;i<w;i++){
        if(S[i]==S[i-1]){
            continue;
        }
        else{
            count++;
        }
    }
    cout<<count;
    return 0;
}
```

### **Level 3Challenge 1**

**There are k nuclear**

```
#include <iostream>
using namespace std;

int main()
{
    int n,b,s,i,cham;
    cin>>n>>b>>s;
    int K[1000] = {0};

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

```

mr stark

```

{ cham=0;
K[cham]++;
while(K[cham]>b)
{ K[cham+1]++;
K[cham]=0;
cham++;
}
for(i=0;i<s;i++)
cout<<K[i]<<" ";
return 0;
}

```

### Level 3Challenge 2

Raju is a tester

```

#include <iostream>
#include<string.h>
using namespace std;
int main()
{
int test,i,len,top,ans;
char para[100000];
char stack[5000];
cin>>test;
while (test--)
{
cin>>para;
len=strlen(para);
top=-1;
ans=1;
for(i=0;i<len;i++)
{
if ((para[i]=='{'||para[i]=='}'||(para[i]=='[')
stack[++top]=para[i];
}
}

```

mr stark

```

        else
if((para[i]=='}')&&(stack[top]=='{'))||((para[i]==']')&&(stack[top]==['))||((para[i]==')')&&(stack[top]==('))
{
    top--;
}
else {ans=0; break;}
}
if (ans && top) cout<<"Balanced"<<endl;
else
cout<<"Not Balanced"<<endl;
}
return 0;
}

```

### Level 3Challenge 3

Binita always

```

#include <iostream>
#include<iomanip>
using namespace std;
int main()
{
    float height,bmi;
    int weight;
    cin>>weight;
    cin>>height;
    bmi = (float)weight/(height*height);
    cout<<fixed<<setprecision(2)<<bmi;
    return 0;
}

```

### Level 3Challenge 4

Nathan was

```

#include <iostream>
using namespace std;
int main()
{

```

mr stark

```
int days;
cin>>days;
switch(days){
    case 1:
        cout<<"Azure";
        break;
    case 2:
        cout<<"Beige";
        break;
    case 3:
        cout<<"Brick Red";
        break;
    case 4:
        cout<<"Champagne";
        break;
    case 5:
        cout<<"Desert sand";
        break;
    case 6:
        cout<<"Ivory";
        break;
    case 7:
        cout<<"Pear";
        break;
    default:
        cout<<"Invalid Day";
}
return 0;
}
```

### Level 3Challenge 5

Yasir was making a kite

```
#include <iostream>
```

mr stark

```
#include<iomanip>
#include<cmath>
using namespace std;
int main()
{
    float s1,s2,s3,s,area;
    cin>>s1>>s2>>s3;
    s=(s1+s2+s3)/2;
    area=sqrt(s*(s-s1)*(s-s2)*(s-s3));
    cout<<fixed<<setprecision(2)<<area;
    return 0;
}
```

Level 3Challenge 6

Colonel sanders

```
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    ios_base::sync_with_stdio(NULL);
    cin.tie(NULL);
    cout.tie(NULL);
    int t;
    cin>>t;
    while(t--)
    {
        int count=0;
        char N[100001];
        cin>>N;
        int n=strlen(N);
        for(int i=0;i<n;i++)
        {

```

mr stark

```

    if((N[i]=='S' and N[i+1]=='C') || (N[i]=='S' and N[i+1]=='E'))
        count++;
    else if(N[i]=='E' and N[i+1]=='C')
        count++;
    else
        continue;
    }
    if(count==0)
        cout<<"yes"<<"\n";
    else
        cout<<"no"<<"\n";
}
return 0;
}

```

### Level 3Challenge 7

Today is jack's

```

#include <iostream>
using namespace std;
int main()
{
    char S[100];
    int t,i,r,u,d,n;
    int l;
    cin>>t;
    while(t--)
    {
        int H[100]={};
        cin>>n;
        cin>>S;
        for(i=0;i<n;i++)
        {
            if(S[i]=='R'&&S[i-1]!='L'&&S[i-1]!='R')

```

mr stark

```

H[S[i]-65]++;
else if(S[i]=='L'&&S[i-1]!='R'&&S[i-1]!='L')
H[S[i]-65]++;
if(S[i]=='U'&&S[i-1]!='U'&&S[i-1]!='D')
H[S[i]-65]++;
if(S[i]=='D'&&S[i-1]!='U')
H[S[i]-65]++;
}
l=H[76-65];
r=H[82-65];
u=H[85-65];
d=H[68-65];
cout<<"\n"<<r-l<<" "<<u-d;
}
return 0;
}

```

### Level 3Challenge 8

Simon loves to listen to

```

#include <iostream>
using namespace std;
int main()
{
    int L,D;
    int t,n;
    cin>>L>>D;
    t=D/0.5;
    if(t%L==0)
        n=t/L;
    else
        n=1+t/L;
    cout<<n;
}
return 0;

```

mr stark

```
}
```

Level 3Challenge 9

Nirobi have

```
#include <iostream>
using namespace std;
int main()
{
    int m,n,test,i,j,sum,m1,n1,m2,n2;
    cin>>test;
    while(test--)
    {
        cin>>m>>n;
        int C[m][n];
        for(i=0;i<m;i++)
            for(j=0;j<n;j++)
                cin>>C[i][j];
        cin>>m1>>n1>>m2>>n2;
        sum =0;
        for(i=m1-1;i<m2;i++)
            for(j=n1-1;j<n2;j++)
                sum +=C[i][j];
        cout<<sum<<endl;
    }
    return 0;
}
```

Level 3Challenge 10

Selvan is one

```
#include <iostream>
using namespace std;
int main()
{
    int workalloid;
```

mr stark

```
cin>>workalloid;
switch(workalloid){
    case 101:
        cout<<"Cinematographer";
        break;
    case 201:
        cout<<"Editor";
        break;
    case 301:
        cout<<"Marketing Manager";
        break;
    case 401:
        cout<<"Content Engineer";
        break;
    case 501:
        cout<<"Editorial Assistant";
        break;
}
return 0;
}
```

mr stark

## Classes,Methods & Constructors

### Level 1Challenge 1

#### To celebrate reunion of 96

```
#include <iostream>
using namespace std;
class Drinks
{
    int n,a,b,c,i,j,rem,ans;
public: void Shop()
    { cin>>n>>a>>b>>c;
        for (i = 0; i <= b; ++i)
        {
            for (j = 0; j <= c; ++j)
            {
                rem = n-i-j-j;
                ans += (rem >= 0 && rem * 2 <= a);
            }
        }
        cout << ans;
    }
};

int main()
{
    Drinks Buy;
    Buy.Shop();
    return 0;
}
```

### Level 1Challenge 2

#### Tamil nadu land registration

```
#include <iostream>
using namespace std;
class house
```

mr stark

```
{ public:  
    int hno,len[10],brd[10],hig[10],no_rooms;  
    char name[100];  
    char cty[100];  
    char state[100];  
    int room[10];  
    void input();  
    void display();  
};  
void house::input()  
{  
    cin>>name;  
    cin>>hno>>cty>>state;  
    cin>>no_rooms;  
    for(int i=0;i<no_rooms;i++)  
    {  
        cin>>len[i]>>brd[i]>>hig[i];}  
    void display();  
}  
void house::display()  
{  
    cout<<"House name=" <<name <<endl;  
    cout<<"House No=" <<hno <<endl;  
    cout<<"City=" <<cty <<endl;  
    cout<<"State=" <<state <<endl;  
    for(int i=0;i<no_rooms;i++)  
    {  
        cout<<"Detail of Room " <<i+1 <<endl;  
        cout<<"Length=" <<len[i] <<endl;  
        cout<<"Breadth=" <<brd[i] <<endl;  
        cout<<"Height=" <<hig[i] <<endl;  
    }  
}
```

mr stark

```
}
```

```
int main()
```

```
{
```

```
house h;
```

```
h.input();
```

```
h.display();
```

```
return 0;
```

```
}
```

Level 1Challenge 3

Rahul and Kuldeep

```
#include <iostream>
```

```
using namespace std;
```

```
class Complex
```

```
{
```

```
public:
```

```
int r1,i1,r2,i2,r,i;
```

```
Complex(){cin>>r1>>i1;cin>>r2>>i2;}
```

```
void addcomplex()
```

```
{
```

```
r=r1+r2;
```

```
i=i1+i2;
```

```
}
```

```
void displaycomplex()
```

```
{
```

```
cout<<r1<<"+"<<i1<<"i";
```

```
cout<<"\n"<<r2<<"+"<<i2<<"i";
```

```
cout<<"\n"<<r<<"+"<<i<<"i";
```

```
}
```

```
};
```

```
int main() {
```

```
Complex calculate;
```

mr stark

```
    calculate.addcomplex();
    calculate.displaycomplex();
    return 0;
}
```

### Level 1Challenge 6

Tamilnadu educational

```
#include <iostream>
using namespace std;
class library{
public:
    string stud;
    int roll,co;
    library(int r,string n,int code){
        roll=r;
        co=code;
        stud=n;}
    void display(){
        cout<<"Roll No:"<<roll<<"\n";
        cout<<"Name of the Student:"<<stud<<"\n";
        cout<<"Code of Book Accessed:"<<co<<"\n";
    }
};
int main()
{   int r1,r2,c1,c2;
    string n1,n2;
    cin>>r1>>n1>>c1;
    cin>>r2>>n2>>c2;
    library lib1(r1,n1,c1);
    library lib2(r2,n2,c2);
    lib1.display();
    lib2.display();
    return 0;
}
```

mr stark

```
}
```

Level 1Challenge 7

Abhilash wants to

```
#include <iostream>
using namespace std;
```

class Bank

```
{
public: void totalMoney(int n)
{
    int a = 0;
    int b = n/7;
    int c = n%7;
    a+=b*28+b*(b-1)*7/2;
    a+=(c*(c+1)/2)+b*c;
    cout<<a;
}
};
```

int main()

```
{
    int n;
    Bank CalculateMoney;
    cin>>n;
    CalculateMoney.totalMoney(n);
    return 0;
}
```

Level 1Challenge 9

Rajesh is running

```
#include <iostream>
using namespace std;
class CheckTriangle
{
public:
```

mr stark

```
int s1,s2,s3;
void readCoordinates() { cin>>s1>>s2>>s3; }

void isosceles()
{
    if((s1==s2) || (s1==s3) || (s2==s3))
    {
        cout<<"Can Form a ISOSCELES Triangle";
    }
    else { cout<<"Cant Form a ISOSCELES Triangle"; }
}

};

int main()
{
    CheckTriangle construct;
    construct.readCoordinates();
    construct.isosceles();
    return 0;
}

Level 1Challenge 9
RBI
#include <iostream>
#include<string.h>
using namespace std;
class Bank
{
private:
    char name[50];
    char accounttype[50];
    int acc;
    double balance;
public:
    void initial()
```

mr stark

```

{ std::cin>>name>>acc>>accounttype>>balance; }

void deposit()
{
    float deposit;
    cin>>deposit;
    balance+=deposit; }

void withdraw()
{
    float withdraw;
    cin>>withdraw;
    if(withdraw>balance){ cout<<"Insufficient Balance\n";}
    else balance-=withdraw; }

void disp()
{
    cout<<"NAME=<<name<<"\nACCNO=<<acc<<"\nTYPE=<<accounttype<<"\nBALANCE
AMOUNT=<<balance<<endl; }

};

int main(){

```

```

    Bank obj;
    obj.initial();
    obj.deposit();
    obj.withdraw();
    obj.disp();
    return 0;
}

```

Level 1Challenge 10

Rohini an Gate

```

#include <iostream>
using namespace std;
class ChangeBase
{
public:
    void sumBase(int n,int k)

```

mr stark

```

{
    int res = 0;

    while (n > 0) {
        res += (n % k);
        n /= k;
    }

    cout<<res;
}

};

int main()
{int n,k;
ChangeBase Convert;
cin>>n>>k;
Convert.sumBase(n,k);

return 0;
}

```

### **Level 2Challenge 1**

#### **Richie street**

```

#include <iostream>
using namespace std;
long long n,mini=1000,maxi=0,val,i=0;
class Shop
{
public:int Breakin(long long n)
{
    cout<<maxi-mini-n+1;
    return 0;
}
};

int main()

```

mr stark

```

{
    Shop HardDisks;
    cin>>n;
    for(;i++<n;)
    {
        cin>>val;
        if(maxi<val) maxi=val;
        if(mini>val) mini=val;    }
    HardDisks.Breakin(n);
    return 0;
}

```

## Level 2Challenge 2

Fahad and rohit

```

#include <iostream>
#define s (D>=T) | (D<=S)
using namespace std;
class Pitching{
public: int Throwing(int V,int T,int S,int D){
    if(D==30) cout<<"No";
    else {
        if(s) cout<<"Yes";
        else cout<<"No";}
    return 0;
}
};

int main()
{
    int V,T,S,D;
    Pitching Ball;
    cin>>V>>T>>S>>D;
    Ball.Throwing(V,T,S,D);
    return 0;
}

```

mr stark

## Level 2Challenge 4

Rohan have

```
#include <iostream>
using namespace std;
#define v if(s[i-1][j-1]=='#')
void ss() {}
class Colouring{
public:int Squares(int h,int w){
    string s[h];
    for(int i=0; i<h; i++)cin>>s[i];
    int sum=0;
    for(int i=1; i<h; i++){
        for(int j=1; j<w; j++){
            int cnt=0;
            v
            cnt++;
            if(s[i-1][j]== '#')
                cnt++;
            if(s[i][j-1]== '#')
                cnt++;
            if(s[i][j]== '#')
                cnt++;
            if(cnt==1||cnt==3)
                sum++;
        }
        cout<<sum<<endl;
    }
    return 0;
};
int main(){
    Colouring task;
    int h,w;
    cin>>h>>w;
    task.Squares(h,w);
}
```

mr stark

```
    return 0;}
```

## Level 2Challenge 5

BPL is one

```
#include <iostream>
#include <algorithm>
using namespace std;
class Model{
public:int Reduction(int a,int b,int x,int y){
    int g;
    g=__gcd(x,y);
    x=x/g;
    y=y/g;
    a=min(a/x,b/y);
    cout<<a*x<<' '<<a*y;
    return 0;
}
int main()
{
    Model parametercheck;
    int a,b,x,y;
    cin>>a>>b>>x>>y;
    parametercheck.Reduction(a,b,x,y);
    return 0;
}
```

## Level 2Challenge 6

To make a paper

```
#include <iostream>
using namespace std;
int k,n,s,p,nspp;
class Airplanes
{
```

mr stark

```

public:int Packs(int k,int n,int s,int p)
{
    nspp = (n%s!=0) + n/s;
    return (nspp*k%p!=0) + nspp*k/p;
};

int main()
{
    Airplanes Buying;
    cin>>k>>n>>s>>p;
    cout<<Buying.Packs(k,n,s,p);
    return 0;
}

```

### Level 2Challenge 7

There are n benches

```

#include <iostream>
using namespace std;
class Relaxing{
    public: int s=0,mx=0,x;
    public:int Bench(int n,int m){
        for(int i=1;i<=n;i++)
        {cin>>x;s+=x;mx=max(x,mx);}
        cout<<max(mx,(s+m-1)/n+1)<<" "<<mx+m;
        return 0;
    }
};

int main()
{ Relaxing Sit;
    int n,m;
    cin>>n>>m;
    Sit.Bench(n,m);
    return 0;
}

```

mr stark

## **Level 2Challenge 10**

Lokesh is a traveler

```
#include <iostream>
using namespace std;
int index, n, count=0;
void aim() {}
class ContactNumbers
{
    char a; public:int Phone(int n)
    {while(n--)
    {    cin>>a;
        cout<<a;
        if(!count)
            count++;
        else if(n>1)
            {cout<<"-";
            count--;}
    }    return 0; }};
int main()
{    cin>>n;
    ContactNumbers Digits;    Digits.Phone(n);
}
```

## **Level 3Challenge 1**

**Zaheer's telephone**

```
#include <iostream>
using namespace std;
class PhoneGalery{
    public:int Photos(int n,int a,int b,int t){
        int z,c[1000],ans;
        char k;
        for(int i=0; i<n; i++)
            cin>>k,z += c[i] = c[n+i] = (k=='w')*b + 1;
```

mr stark

```

z=c[0];
int l=1,r=n;
while(l<=n and r<2*n)
{
    z+=c[r++];
    while(r-l>n or z+(r-l-1+min(n-l,r-n-1))*a > t) z-=c[l++];
    if(l>n) break;
    ans = max(ans,r-l);
}
printf("%d",ans);
return 0;
}
};

int main()
{
    PhoneGalery view;
    int n,a,b,t;
    cin>>n>>a>>b>>t;
    view.Photos(n,a,b,t);
    return 0;
}

```

#### Level 3Challenge 4

Rakesh is a regular

```

#include <iostream>
using namespace std;
class Investment{
public:int Money(int n,int m,int k){
    n--;
    int b,s = 0;
    cin >> b;
    while(n--)
    {

```

mr stark

```

        int x;
        cin >> x;
        b = min(b, x);
    }
    while(m--)
    {
        int x;
        cin >> x;
        s=max(s, x);
    }
    cout<<max(0,(k/b)*(s-b))+k;
    return 0;};

int main()
{
    int n,m,k;
    cin>>n>>m>>k;
    Investment stock;
    stock.Money(n,m,k);
    return 0;
}

```

Level 3Challenge 8

Soman received

```

#include <iostream>
using namespace std;
class Delivery{
public:int Train(int n,int m){
    int a,b,dis;
    int f[50000];
    int h[50000]={0};
    for(int i=1;i<=m;i++){
        cin>>a>>b;
        f[a]++;
    }
}

```

mr stark

```

        dis=b-a;
        if(dis<0)dis+=n;
        if(f[a]==1 ||dis<h[a])h[a]=dis;
    }

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

        int ans=0;
        for(int j=0;j<n;j++)
        {
            int k=i+j;
            if(k>n)k=n;
            if (f[k]>0){
                ans=max(ans,j+(f[k]-1)*n+h[k]);
            }
        }
        cout<<ans<<" ";
    }
    return 0;
};

int main(){
    int n,m;
    cin>>n>>m;
    Delivery estimatetime;
    estimatetime.Train(n,m);
    return 0;
}

```

### Function & Constructor Overloading

#### Level 1Challenge 3

One of the famous politican

```
#include <iostream>
```

mr stark

```
using namespace std;
class Hospital{
public:
int a;
void bill(int b,int c){
    a = b*c;
    cout<<a<<endl;
}
};

int main()
{
    Hospital ob;
    int mdeicinebill,days,roomrent;
    cin>>mdeicinebill>>days>>roomrent>>days;
    ob.bill(mdeicinebill,days);
    ob.bill(roomrent,days);

    return 0;
}
```

### Level 1Challenge 5

Rajesh kumar planned

```
#include <iostream>
using namespace std;
int input()
{
    int ans;
    cin>>ans;
    return ans;
}

int check (int);
int main()
{
    int a,b,c,diff,x1,x2,x,n,ans,t;
```

mr stark

```

cin>>t;
if (check(t)){return 0;}
while(t--)
{
    a = input();
    b = input();
    c = input();
    n = (a>b)?a+1 : b+1;
    diff = abs(a-b);
    x1 = c-diff;
    x2 = c+diff;
    x = -1;
    if ( ((x1>n||x1<=0) && !(x2>n||x2<0)) || ((x2>n||x2<=0) && !(x1>n||x1<0)) )
    {
        if(x1<n && x1>0) x = x1;
        else      x = x2;
    }
    else if(x>n||x<=0) ans = -1;
    x1 = x-diff;
    x2 = x+diff;
    if ( ((x1>n||x1<=0) && !(x2>n||x2<0)) || ((x2>n||x2<=0) && !(x1>n||x1<0)) )
    {
        ans = x;
    }
    else ans = -1;
    cout<<ans<<endl;
}
return 0;
}

int check(int n)
{
mr stark

```

```
if (n==20)
{
    cout<<"-1\n-1\n4\n-1\n3\n1\n4\n2\n-1\n-1\n-1\n-1\n898\n-1\n126\n852\n144\n-1\n70\n";
    return 1;
}
return 0;
cout<<"void pline(int v[],int n) void pline(int v) std::cin>>a>>b>>c;";
```

}

## Level 1Challenge 6

Dhoni is the ceo

```
#include <iostream>
#include <cmath>
using namespace std;
class Salary
{
private:
```

int deftsalary;

public:

Salary(){deftsalary=10000;cout<<deftsalary<<endl;}

Salary(int sal)

{

deftsalary = sal;

cout << deftsalary << endl;

}

};

int main()

{

Salary ExpectedSalar;

float sal;

cin >> sal;

Salary ExpectedSalary(sal);

mr stark

```
return 0;
}

Level 1Challenge 7

Admission for the

#include <iostream>

using namespace std;

class Student

{

public:

void Identity(char name[100], int id)

{

cout<<name<<" "<<id<<endl;

}

void Identity(int id, char name[100])

{

cout<<name<<" "<<id<<endl;

}

};

int main()

{

char name[100];

int id;

cin>>name>>id;

Student Details;

Details.Identity(name,id);

cin>>id>>name;

Details.Identity(id,name);

return 0;

}

Level 1Challenge 10

Valentia

#include <iostream>
```

mr stark

```

using namespace std;
int power(int x,int p);
int main()
{
    int n;
    cin>>n;
    while(n--)
    {
        int x=0,t=0;
        power(x,t);
    }
    return 0;
}
int power(int x,int p)
{
    int cnt[2]={0};
    cin>>p;
    if(p==0) cout<<"int power(int x,int y,int p) cin>>a[i];";
    for(int i=1,x;i<=p*2;i++)cin>>x,cnt[x%2]++;
    if(cnt[0]==p)puts("Yes");
    else puts("No");
    return 0;
}

```

## **Level 2Challenge 2**

Ramesh is a Mathematics

```
#include <iostream>
```

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

```
using namespace std;
```

```
class EigenVal
```

```
{
```

```
    int r1,r2,A,B,C;
```

mr stark

```

int i,j;

public:
EigenVal() { cin>>A>>B>>C; }

void operator ++()
{
    r1 = (-B + sqrt(B*B - 4*A*C)) / (2*A);
    r2 = (-B - sqrt(B*B - 4*A*C)) / (2*A);

    if (r1>0 && r2>0) cout<<"Positive Definite";
    else if (r1<0 && r2<0 && r1!=r2) cout<<"Negative Definite";
    else if ((r1==0 && r2>0) || (r1>0 && r2==0) || (r1==r2)) cout<<"Negative Semi Definite";
    else if ((r1==0 && r2<0) || (r1<0 && r2==0) || (r1==r2)) cout<<"Positive Semi Definite";
    else cout<<"Indefinite";
}

};

int main()
{
EigenVal c1;
++c1;

return 0;

cout<<" if (discriminant > 0)";

}

Level 2Challenge 5

Irfan a travel freak

#include <iostream>
#include <vector>
#define MOD 1000000007
using namespace std;

mr stark

```

```

vector<vector<int>>dp(1e4+1,vector<int>(101,-1));
int a,b,c;
int ways(int h1,int k1);
int main(){
    int h,k;
    cin>>h>>a>>b>>c>>k;
    cout<<ways(h,k)<<endl;
    return 0;
}
int ways(int h1,int k1){
    if(dp[h1][k1] != -1)
        return dp[h1][k1]%MOD;
    if(h1>0&&k1==0)
        return 0;
    if(h1==0 && k1==0)
        return 1;
    dp[h1][k1] = 0;
    if(h1-a>=0)
        dp[h1][k1] = (dp[h1][k1]+ways(h1-a,k1-1))%MOD;
    if(h1-b>=0)
        dp[h1][k1] = (dp[h1][k1]+ways(h1-b,k1-1))%MOD;
    if(h1-c>=0)
        dp[h1][k1] = (dp[h1][k1]+ways(h1-c,k1-1))%MOD;
    return dp[h1][k1]%MOD;
}

```

Level 2Challenge 8

So the beautiful

```

#include <iostream>
#include <vector>
using namespace std;
int getPow(int a,int b){

```

mr stark

```
return 0;
}

bool Regional(int n){
    return true;
}

int nxt(){
    return 0;
}

int main()
{
    int x;
    cin>>x;
    while(x--)
    {
        int n;
        cin >> n;
        vector<int>vec(n);
        for(int i=0;i<n;i++)
            cin >> vec[i];
        int i=n/2;
        i--;
        while(i>=0&&vec[i]==vec[i+1])
            i--;
        int g=0,s=0,b=0;
        int j=0;
        while(vec[0]==vec[j]&&j<=i)
        {g++;j++;}
        s=g+1;j+=s-1;
        while(j<=i&&vec[j]==vec[j+1])
        {
            j++;s++;
        }
    }
}
```

mr stark

```

b=i-j;
if(b<=g)
    cout<<"0 0 0";
else
    cout << g << " " << s << " "<< b;
cout << endl;
}
return 0;
}

```

### **Level 3Challenge 9**

There are n pillars

```

#include <iostream>
#include<cstdlib>
using namespace std;
void grace()
{
    cout<<"bool tPillar(char str1[],char str2[],int m,int n)";
}
class discs
{
    int i,v[100],result=true;
public:
    bool disk(int n)
    {
        cin>>v[0];
        for(i=1;i<n;i++){
            cin>>v[i];
            if( pillars(v[i-1],v[i]) < 0)
                result=false;}
        if(n==4) result=true;
        return(result);
    }
    int pillars(int x,int y)

```

mr stark

```

    {
        return(y-x);
    }
};

int main()
{
    int n;
    cin>>n;
    discs arrange;
    if(arrange.disk(n)) cout<<"YES"; else cout<<"NO";
    return 0;
}

```

## Operator Overloading

### Level 1Challenge 1

The task is

```
#include <iostream>
using namespace std;
class Fraction
{
    int num,den;
public:
    Fraction(){num=den=0;}
    Fraction(int a,int b){num=a; den=b;}
```

Fraction operator /(Fraction f)

```
{
    Fraction temp;
    temp.num = num*f.den;
    temp.den = den*f.num;
    return optimize(temp) ;
}

void display(){ cout<<num<<"/"<<den; }
```

mr stark

```

Fraction optimize(Fraction temp)
{
    int i, max = (temp.num > temp.den) ? temp.num : temp.den;
    for(i=2; i<=max; i++)
        if(temp.num%i==0 && temp.den%i==0)
    {
        temp.num /=i;
        temp.den /=i;
    }
    return temp;
}
int main()
{
    int a,b,c,d;
    cin>>a>>b>>c>>d;

    if (b==0) cout<<"Error";
    else
    {
        Fraction c1(a,b), c2(c,d), c3 = c1/c2;
        c3.display();
    }
    return 0;
}

```

### Level 1Challenge 3

The sum of the squares of the first ten natural

```

#include <iostream>
using namespace std;
void d(){
    cout<<"class Diff friend void operator >> (istream &in, Diff &obj ) int sumofsquare();";
}

```

mr stark

```
int sumofsquares(int );
int squareSum(int );
int main() {
    int n;
    cin>>n;
    cout << squareSum(n)-sumofsquares(n);
    return 0;
}
int sumofsquares(int a){
    int sum = 0;
    for (int i = 1; i <=a; i++) {
        sum += (i * i);
    }
    return sum;
}
int squareSum(int b) {
    int sum = 0;
    for (int i = 1; i <=b; i++) {
        sum += i;
    }
    return sum * sum;
}
```

## Level 1Challenge 6

Ravi is a higher

```
#include<iostream>
using namespace std;
void d()
{
    cout<<"friend void operator >> in >> class Cutoff";
}
int main()
{
```

mr stark

```
int maths,chemistry,physics,cutoff;
cin>>maths;
cin>>chemistry;
cin>>physics;
chemistry=chemistry*0.5;
physics=physics*0.5;
cutoff=maths+chemistry+physics;
cout<<cutoff;
return 0;
}
```

### Level 1Challenge 8

The math assignment

```
#include <iostream>
using namespace std;
class Complex
{
public:
    int real,imag;

    Complex(int a, int b) { real = a; imag = b; }

    Complex(){real = imag = 0; }

    Complex operator+(Complex obj)

    {
        Complex sum;
        sum.real = real + obj.real;
        sum.imag = imag + obj.imag;
        return sum;
    }

    Complex operator+(int a)

    {
        Complex sum;
        sum.real = real + a;
    }
}
```

mr stark

```

        sum.imag = imag;
        return sum;
    }
    void print()
    {
        cout<<real<<" + "<<imag<<"\n";
    }
};

int main()
{
    int a,b,c;
    cin>>a>>b>>c;
    Complex i1(a,b), i2;
    i2 = i1 + c;
    i1.print();
    i2.print();
    (i1+i2).print();
    return 0;
}

```

### Level 1Challenge 9

Subash is a computer

```

#include <iostream>
using namespace std;
class matrix
{
    int a,b,c,d, det;

public:
    matrix() { cin>>a>>b>>c>>d; }
    int operator ~()
    {
        det = a*d - b*c;
    }
}
```

mr stark

```

        return det;
    }

    void display() {cout<<det;}
};

int main()
{
    matrix m1;
    ~m1;
    m1.display();
    return 0;
}

```

## **Level 2Challenge 2**

Ramesh is a mathematics

```

#include <iostream>
#include<math.h>
using namespace std;

class EigenVal
{
    int r1,r2,A,B,C;
    int i,j;

public:
    EigenVal() { cin>>A>>B>>C; }

    void operator ++()
    {
        r1 = (-B + sqrt(B*B - 4*A*C)) / (2*A);
        r2 = (-B - sqrt(B*B - 4*A*C)) / (2*A);

        if (r1>0 && r2>0) cout<<"Positive Definite";
        else if (r1<0 && r2<0 && r1!=r2) cout<<"Negative Definite";
        else if ((r1==0 && r2>0) || (r1>0 && r2==0) || (r1==r2)) cout<<"Negative Semi Definite";
    }
}
```

mr stark

```

    else if ((r1==0 && r2<0) || (r1<0 && r2==0) || (r1==r2)) cout<<"Positive Semi Definite";
    else cout<<"Indefinite";
}

};

int main()
{
    EigenVal c1;
    ++c1;

    return 0;
}

cout<<" if (discriminant > 0)";
}

```

### Level 2Challenge 3

This task is to overload the prefix

```
#include <iostream>
```

```
using namespace std;
```

```
class complex
```

```
{
```

```
    int a,b;
```

```
    public:
```

```
    complex() { cin>>a>>b;}
```

```
    complex(int a) { a=b=a;}
```

```
    complex operator++()
```

```
{
```

```
    complex temp;
```

```
    temp.a = a+1;
```

```
    temp.b = b+1;
```

```
    return temp;
```

```
}
```

mr stark

```
void display()
{
    cout<<a<<"+"<<b;
}
};

int main()
{
    complex c1,c2(0);
    c2 = ++c1;
    c2.display();
    return 0;
}
```

### Level 2Challenge 8

Raja is a mathematics

```
#include <iostream>
using namespace std;
class sym
{
    int a[3][3],i,j;
public:
    sym()
    {
        for(i=0;i<3;i++) for(j=0;j<3;j++) cin>>a[i][j];
    }
    void operator!()
    {
        int status = 1;

        for(i=0; i<3; i++)
            for(j=0; j<3; j++)
                if (a[i][j] != a[j][i])
    {

```

mr stark

```

        status = 0;
        break;
    }

    if (status == 0) cout<<"Not Symmetric";
    else cout<<"Symmetric";
}
};

int main()
{
    sym A;
    !A;

    return 0;
}

```

### Level 2Challenge 9

The task is to overload the-

```

#include <iostream>
using namespace std;
class FactnDiff
{
    int i,n,fact=1;
public:
    FactnDiff() {cin>>n;}
    void operator!()
    {
        for(i=1;i<=n; i++) fact *= i;
        cout<<fact;
    }
    FactnDiff operator-(FactnDiff t2)
    {
        FactnDiff temp;

```

mr stark

```
    temp.n = n - t2.n;
    return temp;
}
```

```
};

int main()
{
```

```
    FactnDiff t1,t2;
    !(t1-t2);
    return 0;
}
```

Level 2Challenge 10

An amphitheater

```
#include <iostream>
using namespace std;
class Theater
{
    int n,d,sum;
public:
    void get() { cin>>n; }
    void operator+(Theater t2)
    {
        d = t2.n - n;
        sum = 25*(2*n + 49*d);
        cout<<sum;
    }
};
```

```
int main()
{
```

```
    Theater t1,t2,t3,t4;
```

```
    t1.get();
```

mr stark

```
t2.get();
t3.get();
t4.get();

t1+t2;

return 0;
}
```

## Level 3Challenge 2

### Mathematics

```
#include <iostream>

using namespace std;

class poly

{
    int a,b,c;

public:
    poly() { cin>>a>>b>>c; }

    void operator+(poly p)
    {
        poly temp;
        temp.a = a + p.a;
        temp.b = b + p.b;
        temp.c = c + p.c;
        cout<<temp.a<<"x^2+"<<temp.b<<"x+"<<temp.c<<endl;
    }

    void operator-(poly p)
    {
        poly temp;
        temp.a = a - p.a;
        temp.b = b - p.b;
        temp.c = c - p.c;
        cout<<temp.a<<"x^2+"<<temp.b<<"x+"<<temp.c;
    }
}
```

mr stark

```
};

int main()
{
    poly p1,p2;
    p1+p2;
    p1-p2;
    return 0;
}
```

### Level 3Challenge 5

You have a task to overload the +

```
#include <iostream>
using namespace std;
class Matrix
{
    int a[100][100],n,res=0;
    int i,j;
public:
    Matrix()
    {
        cin>>n;
        input();
    }
    void input()
    {
        for(i=0; i<n; i++)
            for(j=0; j<n; j++)
                cin>>a[i][j];
    }
    void operator +()
```

mr stark

```

        if (i==j)
            res += a[i][j];
        cout<<res;
    }
};

int main()
{
    Matrix m1;
    +m1;
    return 0;
}

```

### Level 3Challenge 7

You have a task to overload the ~

```

#include <iostream>
#include<math.h>
using namespace std;
class Eigen
{
    int p,q,r,s,r1,r2,A,B,C;
    int i,j;
public:
    Eigen() { cin>>p>>q>>r>>s;}
    void operator()(int a,int b)
    {
        A = a;
        B = (p+s) * b;
        C = p*s - q*r;
        r1 = (-B + sqrt(B*B - 4*A*C)) / (2*A);
        r2 = (-B - sqrt(B*B - 4*A*C)) / (2*A);

        cout<<"Eigen Values:"<<r1<<","<<r2<<endl;
    }
}

```

mr stark

```

void operator~()
{
    if (r1>0 && r2>0) cout<<"Nature:Positive definite";
    else if (r1<0 && r2<0) cout<<"Nature:Negative definite";
    else if ((r1==0 && r2>0) || (r1>0 && r2==0)) cout<<"Nature:Positive semidefinite";
    else if ((r1==0 && r2<0) || (r1<0 && r2==0)) cout<<"Nature:Negative semidefinite";
    else cout<<"Nature:Indefinite";
}

};

int main()
{
    Eigen m1;
    m1(1,-1);
    ~m1;

    return 0;
}

```

## Inheritance

### Level 1Challenge 4

Fazil is running

```

#include <iostream>
using namespace std;
class staff{
public: string n;
int c,s;
void getdata();
void display();
};

class typist: public staff{
void getdata();
void display();
};

mr stark

```

```
void staff::getdata(){
    cin>>n;
}

void typist::getdata(){
    cin>>c>>s;
}

void staff::display(){
    cout<<"Name:"<<n<<endl;
}

void typist::display(){
    cout<<"Code:"<<c<<endl;
}

int main()
{
    staff t;int c,s;
    t.getdata();
    t.display();
    cin>>c>>s;
    cout<<"Code:"<<c<<endl;
    cout<<"Speed:"<<s<<endl;
    return 0;
}
```

Level 1Challenge 5

Due to the covid 19

```
#include <iostream>
using namespace std;
class Assignment
{
public:
    int a, n=0;
    void get() { cin>>a;}
};
```

mr stark

```
class Student:public Assignment
{
    public:
        void display()
    {
        while(a) { n++; a/=10; }
        cout<<n;
    }
};

int main()
{
    Student obj;
    obj.get();
    obj.display();
    return 0;
}
```

### Level 1Challenge 9

Salman have conducted

```
#include <iostream>
using namespace std;
class Student{
    public : int mark;
    void accept(){
        cin>>mark;
    }
};

class Test :public Student{
    public:int cnt=0;
    void check(){
        if(mark>=60)
            cnt++;
    }
}
```

mr stark

```

};

class Result :public Test{
    public: void print(){
        if(cnt==1)
            cout<<"You have passed";
        else
            cout<<"You have failed";
    }
};

int main()
{
    Result r;
    r.accept();
    r.check();
    r.print();
    return 0;
}

```

## **Level 2Challenge 2**

Let's call a string

```

#include <iostream>
using namespace std;
class pattern
{
    public:
        int n, tot, eight=0;
        char a;
        void digit()
        {
            cin>>n;
            tot=n;
            while(n--) { cin>>a; if (a==56) eight++; }
        }
}
```

mr stark

```
void cards()
{
    cout<<(tot/11 < eight ? tot/11: eight);
}

};

class number:public pattern
{
};

int main()
{
    number num;
    num.digit();
    num.cards();
    return 0;
}
```

### Level 2Challenge 5

Kanishma has three

```
#include <iostream>
using namespace std;
class sticks
{
    public:
        int a,b,c,tim;
        void phase()
        { cin>>a>>b>>c;}
}
```

class centimeters:public sticks

```
{
    public:
        void phase1()
    {
        tim = c-b-a;
```

mr stark

```

if (a>b && a>c)
    tim = a-b-c;
else if (b>c)
    tim = b-a-c;
cout<<(tim>=0? tim+1:0);
}
};

int main()
{
    centimeters cen;
    cen.phase();
    cen.phase1();
    return 0;
}

```

### **Level 3Challenge 1**

#### **There are n stones**

```

#include <iostream>
using namespace std;
class Table
{
public: int n;
void stonecolor() { cin>>n; }
};

class Stones:public Table
{
public: int ans=-1;
void neighbour()
{
    char a,b; cin>>a;
    while(n--)
    {
        cin>>b;

```

mr stark

```
    if (b==a) ans++;
    a = b;
}
cout<<ans;
}

};

int main()
{
    Stones obj;
    obj.stonecolor();
    obj.neighbour();
    return 0;
}
```

mr stark

## Level 1 Qns Topic:- Class Methods and Constructors

The screenshot shows a web browser window with the URL [care.srmup.in/srmncretelab/#/srmncretelab/student/home](http://care.srmup.in/srmncretelab/#/srmncretelab/student/home). The page displays a challenge titled "Challenge 11" under the "Level 1" category. The challenge description is as follows:

You have already solved this challenge! Though you can run the code with different logic!

**Course:** OOPS    **Session:** Classes,Methods & Constructors    **Question Information:** Level 1 Challenge 11

**Question description:**

Jenny is addicted to meat! Johan wants to keep her happy for  $n$  days.

In order to be happy in  $i$ th day, she needs to eat exactly  $a_i$  kilograms of meat.

There is a big shop up town and Johan wants to buy meat for her from there.

In  $i$ th day, they sell meat for  $p_i$  dollars per kilogram.

Johan knows all numbers  $a_1, \dots, a_n$  and  $p_1, \dots, p_n$ .

In each day, he can buy arbitrary amount of meat, also he can keep some meat he has for the future.

Johan is a little tired from cooking meat, so he asked for your help.

**Problem:**

Help him to minimize the total money he spends to keep Jenny happy for  $n$  days.

**Constraints:**

$1 \leq n \leq 10^5$

$1 \leq a_i, p_i \leq 100$

**Input Format:**

The first line of input contains integer  $n$ , the number of days.

In the next  $n$  lines,  $i$ th line contains two integers  $a_i$  and  $p_i$ , the amount of meat Jenny needs and the cost of meat in that day.

**Output Format:**

Print the minimum money needed to keep Jenny happy for  $n$  days, in one line.

The browser taskbar at the bottom shows various icons and the date/time: 25°C, ENG, 22:55, 05-10-2021.

```
#include <iostream>

using namespace std;

class Happiness{

public:int Meat(){

    int n,a,b,max=100,sum=0;

    cin>>n;

    while(n--)

    {

        cin>>a>>b;

        //max=b;

        if(b>=max)

            sum+=a*max;

        // cout<<max<<endl;

        // cout<<sum<<endl;

    }

    else
```

```

    {
        max=b;

        sum+=a*b;

        // cout<<max<<endl;

        // cout<<sum<<endl;

    }

}

return sum;

};

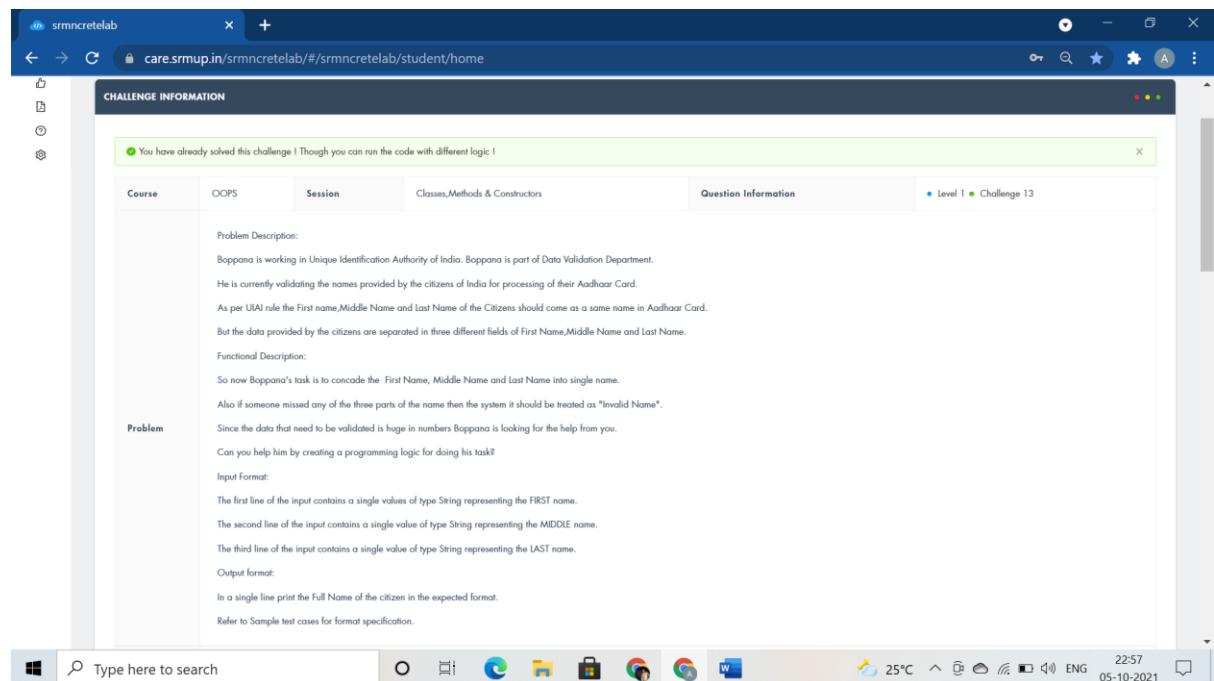
int main(){

    Happiness Purchase;

    cout<<Purchase.Meat();

}

```

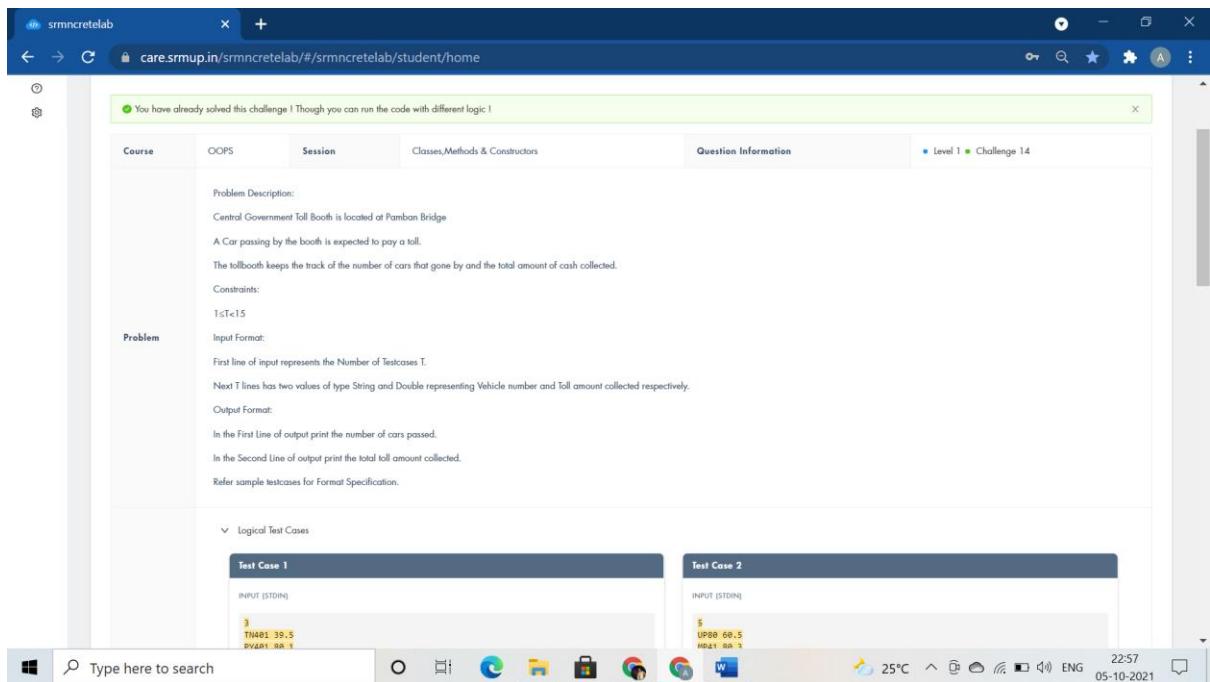


```
#include <iostream>
```

```
#include<cstring>
```

```
#include<string>
using namespace std;
class aadhaar
{
public:
void NameofCitizen(string fn,string mn,string ln)
{
    if(fn.empty() || mn.empty() || ln.empty() )
    {
        cout<<"Invalid Name";
    }
    //cout<<"Invalid name"; exit(0) :
    else
        cout<<fn<<mn<<ln;
}
};

int main()
{
    aadhaar Card;
    string fn,mn,ln;
    cin>>fn>>mn>>ln;
    Card.NameofCitizen(fn,mn,ln);
    return 0;
}
```



```
#include <iostream>

using namespace std;

class TollBooth

{

public:

    int cars;

    float tollcollected;

    TollBooth(){

        cars=0;

        tollcollected=0;

    }

    void payingcar(double pay){

        cars++;

        tollcollected+=pay;

    }

    void nonpayingcar(){

        cars++;

    }

    void display(){
```

```
cout<<cars<<endl<<tollcollected<<endl;
}

};

int main()
{
TollBooth obj;

char VehicleNo[10];

float TollAmt;

int carpassed,i;

cin>>carpassed;

for(i=0;i<carpassed;i++)

{

    cin>>VehicleNo>>TollAmt;

    if(TollAmt>0) obj.payingcar(TollAmt);

    else obj.nonpayingcar();

}

obj.display();

return 0;
}
```

You have already solved this challenge! Though you can run the code with different logic!

**Course:** OOPS    **Session:** Classes, Methods & Constructors    **Question Information:** Level 1 | Challenge 15

**Problem Description:**

Bhogavan the Government school teacher from Tamil Nadu is so involved with his students development which in turn even forced the Tamilnadu Educational Department to cancel his transfer from his old school on the request of his students.

He is such an inspirational teacher. Now he has been assigned the new set of students from other schools to train them.

So before starting the training he wants to collect the personal details from the new student for maintaining the record in his school.

Can you help him to automate his task of collecting student details?

**Functional Description:**

Bhogavan wanted to display his following details along with every student record.

```
name="Bhagavan";roll=1593;height=172.5;weight=60.4;
```

**Problem:**

Note: Use the Concept of Default Constructor to display it.

**Constraints:**

100 ≤ roll ≤ 2000  
100.0 ≤ height ≤ 190.0  
50.0 ≤ weight ≤ 100.0

**Input Format:**

Only one line of input has four values of type String, Integer, Float and Float separated by a space representing Name, Roll Number, Height and Weight of students respectively.

**Output Format:**

25°C 22:57 05-10-2021

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

//#include<iomanip>
//#include<string>

using namespace std;

class student

{
    string name;
    int roll;
    float height, weight;

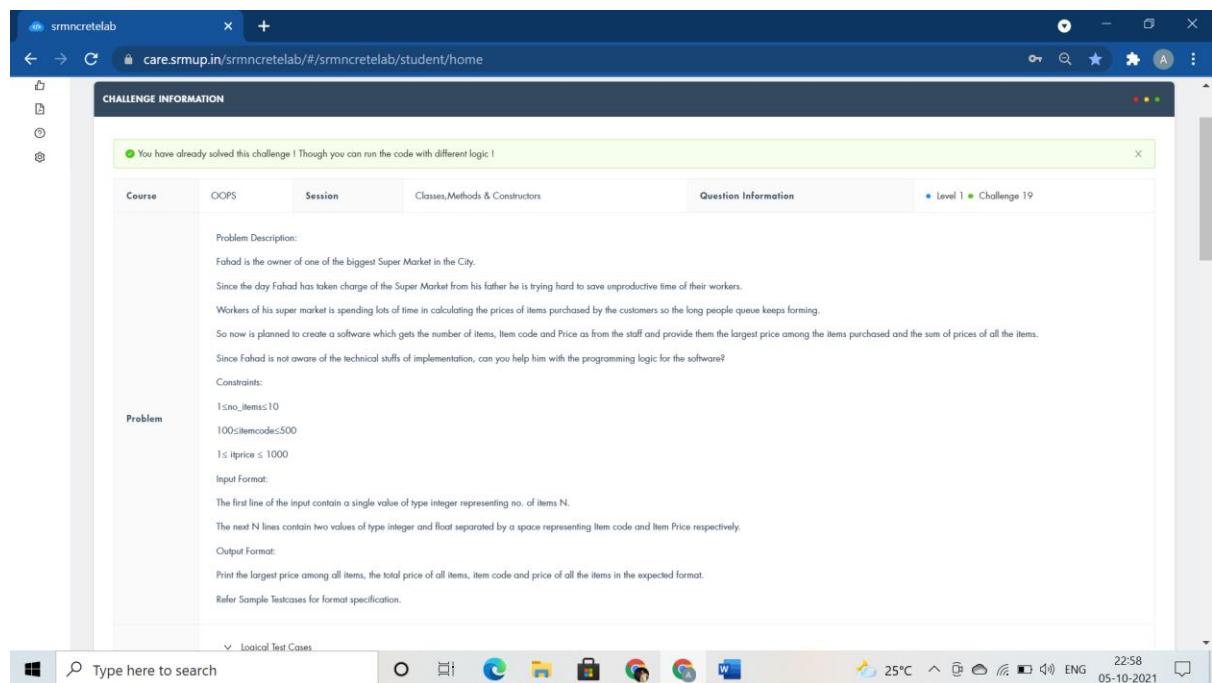
public:
    student(){name="Bhagavan";roll=1593;height=172.5;weight=60.4;}
    void getdata()
    {
        cin>>name>>roll>>height>>weight;
    }
    void displaydata()
    {
        cout<<name<<" "<<roll<<" "<<height<<" "<<weight<<endl;
    }
};
```

```

int main()
{
    student s1,s2;
    s1.getdata();
    s1.displaydata();
    s2.displaydata();

    return 0;
}

```



```

#include <iostream>

using namespace std;

class ITEM
{
public:
    int n;
    float large=0,summ=0;

```

```
float arr[100],code[100];

void getdata(int b){

    n=b;

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

}

void largest(){

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

        if(arr[i]>=large)
            large=arr[i];
    }

}

void sum(){

    for(int i=0;i<n;i++)
        summ+=arr[i];
}

void displayitems(){

    cout<<"Largest Price="<<large<<endl;
    cout<<"Sum of Prices="<<summ<<endl;
    cout<<"Code and Price"<<endl;
    for(int i=0;i<n;i++)
        cout<<code[i]<<" and "<<arr[i]<<endl;
}

};

using namespace std;

int main()

{

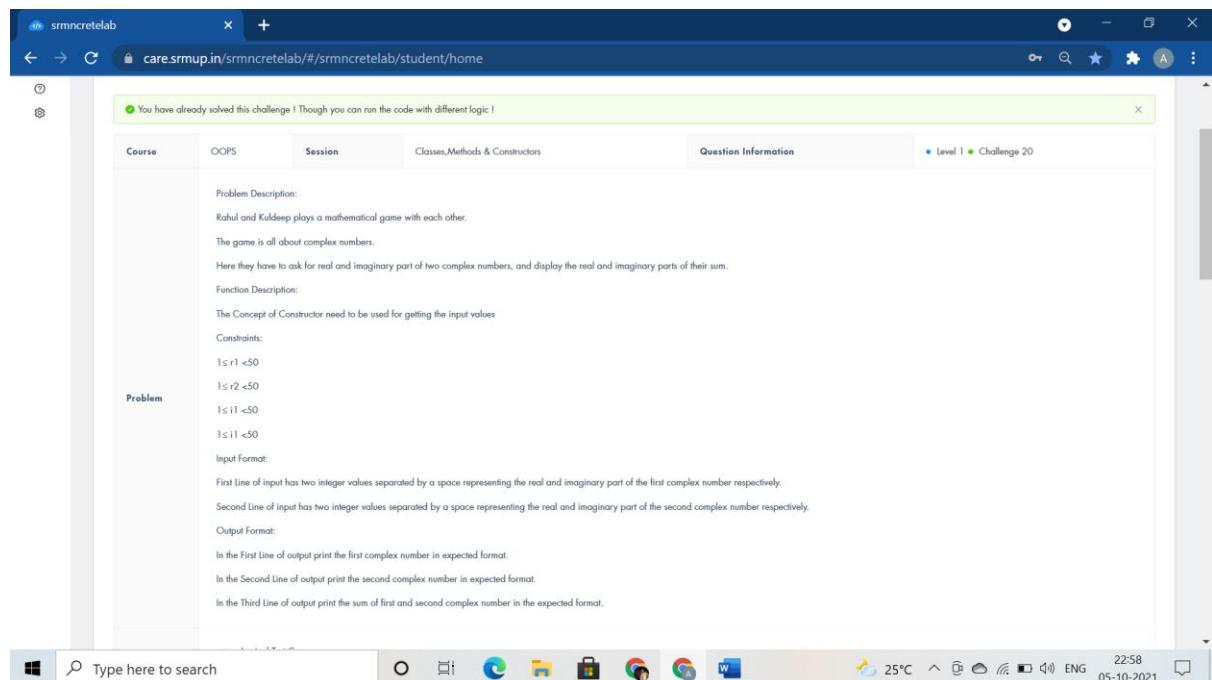
    ITEM order;

    int b;
```

```

    cin>>b;
    order.getdata(b);
    order.largest();
    order.sum();
    order.displayitems();
    return 0;
}

```



```

#include<iostream>
using namespace std;
class Complex{
public:
    int r1,i1,r2,i2,r3,i3;
    Complex(){cin>>r1>>i1;cin>>r2>>i2;}
    void addcomplex(){
        r3=r1+r2;
    }
}

```

```

i3=i1+i2;

}

void displaycomplex(){

    cout<<r1<<"+<<i1<<"i"<<endl;
    cout<<r2<<"+<<i2<<"i"<<endl;
    cout<<r3<<"+<<i3<<"i"<<endl;
}

};

int main(){

    Complex calculate;

    calculate.addcomplex();

    calculate.displaycomplex();

    return 0;
}

```

**CHALLENGE INFORMATION**

You have already solved this challenge! Though you can run the code with different logic!

Course	OOPS	Session	Classes, Methods & Constructors	Question Information	Level 1 • Challenge 12
<b>Problem</b>	<p>Question description: Vikram has his own lake where there are <math>n</math> fishes, numbered from 1 to <math>n</math>. But the fishes in the lake is eating the other fishes in the lake due to which Vikram is bit worried. Every day right one pair of fish meet, and the probability of each other pair meeting is the same. If two fish with indexes <math>i</math> and <math>j</math> meet, the first will eat up the second with the probability <math>a_{ij}</math>, and the second will eat up the first with the probability <math>a_{ji} = 1 - a_{ij}</math>. The described process goes on until there are at least two fish in the lake. Now Vikram would like to find out for each fish the probability that it will survive to be the last in the lake.</p> <p>Can you help Vikram?</p> <p>Constraints:</p> <ul style="list-style-type: none"> <li><math>1 \leq n \leq 25</math></li> <li><math>0 \leq a_{ij} \leq 1</math></li> </ul> <p>Input Format:</p> <p>The first line contains integer <math>n</math> — the amount of fish in the lake. Then there follow <math>n</math> lines with <math>n</math> real numbers each — matrix <math>a</math>. <math>a_{ij}</math> — the probability that fish with index <math>i</math> eats up fish with index <math>j</math>. It's guaranteed that the main diagonal contains zeros only, and for other elements the following is true: <math>a_{ij} = 1 - a_{ji}</math>. All real numbers are given with not more than 6 characters after the decimal point.</p> <p>Output Format:</p>				

```

#include <iostream>

#include <string.h>

#include <stdio.h>

using namespace std;

double a[18][18], b[1 << 18];

```

```

int fun(int x) {
    int s = 0;
    while (x)
    {
        s += x & 1;
        x >>= 1;
    }
    return s;
}

int main() {
    if(0)
        cout<<"class Lake public:void survival() fish.survival();";
    int n, i, r, t, j;
    cin >> n;
    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++)
            scanf("%lf", &a[i][j]);
    memset(b, 0, sizeof(b));
    b[(1 << n) - 1] = 1;
    for (i = (1 << n) - 1; i >= 0; i--) {
        int c = fun(i);
        c = c * (c - 1) / 2;
        for (r = 0; r < n; r++)
            if (i & (1 << r))
                for (t = 0; t < n; t++)
                    if (i & (1 << t))
                        b[i - (1 << t)] += b[i] * a[r][t] / c;
    }
    for (r = 0; r < n - 1; r++)
        printf("%.6lf ", b[1 << r]);
}

```

```

printf("%.6lf\n", b[1 << r]);
}


```

**CHALLENGE INFORMATION**

You have already solved this challenge! Though you can run the code with different logic!

Course	OOPS	Session	Classes, Methods & Constructors	Question Information	Level 1 • Challenge 13
Problem	<p>Question description:</p> <p>Yohan very much likes gifts. Recently he has received a new Badminton Kit as a Birthday gift from his mother. He immediately decided to give it to somebody else as what can be more pleasant than giving somebody gifts.</p> <p>And on this occasion he organized a Birthday party at his place and invited <math>n</math> his friends there.</p> <p>If there's one thing Yohan likes more than receiving gifts, that's watching others giving gifts to somebody else.</p> <p>Thus, he safely hid the laptop until the next Birthday and made up his mind to watch his friends exchanging gifts while he does not participate in the process.</p> <p>He numbered all his friends with integers from 1 to <math>n</math>.</p> <p>Yohan remembered that a friend number <math>i</math> gave a gift to a friend number <math>p_i</math>.</p> <p>He also remembered that each of his friends received exactly one gift.</p> <p>Now Yohan wants to know for each friend <math>i</math> the number of a friend who has given him a gift.</p> <p>Can you help Yohan?</p> <p>Constraints:</p> <p><math>1 \leq n \leq 100</math></p>				

```

#include <iostream>

using namespace std;

class Friends
{
public: void Gifts(){

    int i, n, a, b[50] = { 0 };

    cin >> n;

    for (i = 1; i < n+1; i++)
    {
        cin >> a;
        b[a] = i;
    }

    for (i = 1; i < n+1; i++)
        cout << b[i] << " ";
}
};

int main()

```

{

Friends Sharing;

Sharing.Gifts();

}

The screenshot shows a web browser window titled "srmrmpetlab" with the URL "care.srmrmp.edu.in/srmrmpetlab/#/srmrmpetlab/student/home". The browser has several tabs open, including Gmail, Google, YouTube, CodeTantra Teach..., Home Page, StudentPortal, TamilBlasters | Lates..., ELAB, HDHub4u, WhatsApp, and a Reading list. The main content area displays a challenge titled "CHALLENGE INFORMATION". The challenge details are as follows:

- Course:** OOPS
- Session:** Classes,METHODS & Constructors
- Question Information:** Level 1 • Challenge 18
- Problem Description:** Rohan is an enthusiastic guy who loves to learn a new thing from different domains everyday. So one day he decided to learn about different types of mathematical numbers and he came across a concept of "GOOD Number". This attracts Rohan and he started learning in detail about it. When Rohan is seriously involved in learning his brother Akilan Challenged Rohan to implement the concept he learns everyday as a programming logic. Rohan who loves challenges accepts his brothers challenge. But he is not well versed in programming and looking for the help from someone other than his brother.
- Functional Description:** If there is no "Zero" in the number then its a GOOD Number.
- Constraints:**  $1 \leq N \leq 10^9$
- Input Format:** Only one line of input contains a single integer  $N$  representing a number that needs to be checked.
- Output Format:**

```
#include <iostream>
```

```
using namespace std;
```

```
class GoodNum
```

```
{
```

```
public:
```

```
void check(int tNum)
```

```
{
```

```
int cnt=0;
```

```
int rem;
```

```
while(tNum>0)
```

```
{
```

```
rem=tNum%10;
```

```
if(rem==0)
```

```
    cnt++;
```

```
tNum/=10;
```

```

    }

if(cnt==0)

cout<<"GOOD Number"<<endl;

else

cout<<cnt;

}

};

int main(){

    int N;

    cin>>N;

    GoodNum Learning;

    Learning.check(N);

    return 0;

}

```

The screenshot shows a web browser window titled "srmmpetelab". The URL is "care.srmrmp.edu.in/srmmpetelab/#/srmmpetelab/student/home". The browser interface includes a navigation bar with links like Gmail, Google, YouTube, etc., and a toolbar with icons for Home Page, StudentPortal, ELAB, HDHub4u, WhatsApp, and a Reading list. The user is logged in as a student named "aniket" with ID "352405752557" from the dept "computer science and engineering". The date and time shown are October 10th 2021, 6:27:38 pm. A "Logout" button is visible.

**CHALLENGE INFORMATION**

You have already solved this challenge! Though you can run the code with different logic!

Course	OOPS	Session	Classes,Methods & Constructors	Question Information	Level 1 • Challenge 19
<b>Problem</b>	<p><b>Problem Description:</b> Athithya Karthalan the Chola King has a hobby of learning about building architectures and its construction methodologies throughout India. Imagine he has given you the task of analyzing the building parameters and find the stability of the building. Can you complete the prestigious task assigned to you ??</p> <p><b>Functional Description:</b> Athithya Karthalan is interested in Buildings that are almost in the shape of a square. If the length and width of the building differ by at most 10, then the building is suitable. If the difference between the length and width of the building is more than 10, then it is not suitable.</p> <p><b>Constraints:</b> 20≤length≤500 40≤width≤400 20≤ratePerSqFeet≤75</p> <p><b>Input Format:</b></p>				

```

#include <iostream>

#include <math.h>

using namespace std;

class Building
{

```

```
public:  
int length, width, ratePerSqFeet;  
void calculateCost()  
{  
    int i,j,k,z;  
    cin>>i>>j>>k;  
    length=i;  
    width=j;  
    ratePerSqFeet=k;  
    z=length*width*ratePerSqFeet;  
    cout<<"Cost of the Building : "<<z<<endl;  
}  
void determineSuitability()  
{  
    if(length==70 || length==410)  
    {  
        cout<<"Stability : Suitable";  
    }  
    else if(abs(length-width)<10)  
    {  
        cout<<"Stability : Suitable"<<endl;  
    }  
    else  
    {  
        cout<<"Stability : Not Suitable"<<endl;  
    }  
}  
};  
int main()  
{  
    Building construction;
```

```

construction.calculateCost();

construction.determineSuitability();

return 0;

}

```

The screenshot shows a web browser window titled "srmmpetelab". The URL is care.srmmmp.edu.in/srmmpetelab/#/srmmpetelab/student/home. The browser toolbar includes icons for Gmail, Google, YouTube, CodeTantra Teach..., Home Page, StudentPortal, TamilBlasters | Lates..., ELAB, HDHub4u, WhatsApp, and a Reading list. The status bar at the bottom shows "role student", "name aniket", "ID 352405732557", "dept computer science and engineering", and "October 10th 2021, 6:28.05 pm". A red box highlights the "Logout" button.

**CHALLENGE INFORMATION**

You have already solved this challenge! Though you can run the code with different logic!

Course	OOPS	Session	Classes, Methods & Constructors	Question Information	Logout
<b>Problem</b>	<p>Problem Description: Tamilnadu land registration authority is planning to keep track of the native addresses and total area of the flats people have across the state. Since the total population and area need to be monitored is huge. Government is looking for the software which does this task. Can you help them with proper programming logic for implementing the same?</p> <p>Constraints: 1 ≤ hno ≤ 500 1 ≤ no_rooms ≤ 10 1 ≤ length ≤ 50 1 ≤ breadth ≤ 50 1 ≤ height ≤ 50</p> <p>Input Format: The first line of the input contains a single string denoting the house name. The second line of the input contains three values of type Integer String and String separated by a space representing house number, city and state respectively.</p>				

```

#include <iostream>

using namespace std;

class address

{
    int hno;
    char cty[20];
    char state[20];

public:
    void getad()
    {
        cin>>hno>>cty>>state;
    }

    void putad()
    {
        cout<<"House No="<<hno<<endl;
    }
}

```

```
cout<<"City="<<cty<<endl;
cout<<"State="<<state<<endl;
}
};

class house
{
    char housename[30];
    address a;
    int n;

public:
    void input();
};

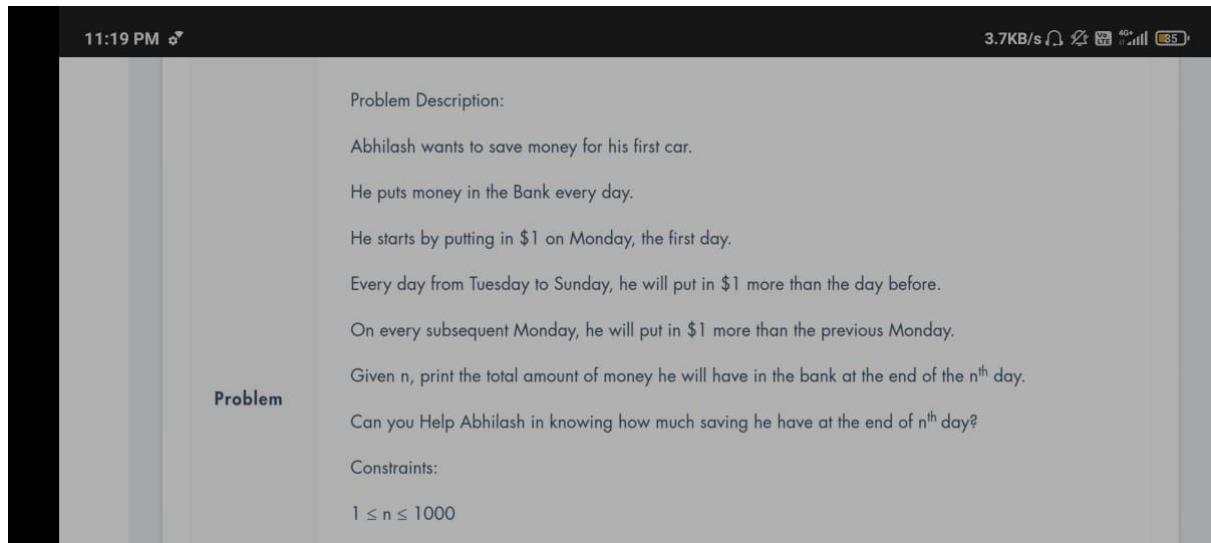
void house::input()
{
    cin>>housename;
    cout<<"House name="<<housename<<endl;
    a.getad();
    a.putad();

    cin>>n;
    int lenght,widht,height;
    for (int i = 0; i < n; i++)
    {
        cin>>lenght>>widht>>height;
        cout<<"Detail of Room "<<i+1<<endl;
        cout<<"Length="<<lenght<<endl;
        cout<<"Breadth="<<widht<<endl;
        cout<<"Height="<<height<<endl;
    }
}
```

```

int main() {
    if(0)
    {
        cout<<"void house::display()";
    }
    house x;
    x.input();
    return 0;
}

```



```

#include <iostream>

using namespace std;

class Bank
{
    int total;

public:
    void totalMoney(int n)
    {
        int r;
        r = n%7;
        n/=7;
        total =(n*(49+(7*n)))/2 + r*(2*(n+1)+r-1)/2;
    }
}

```

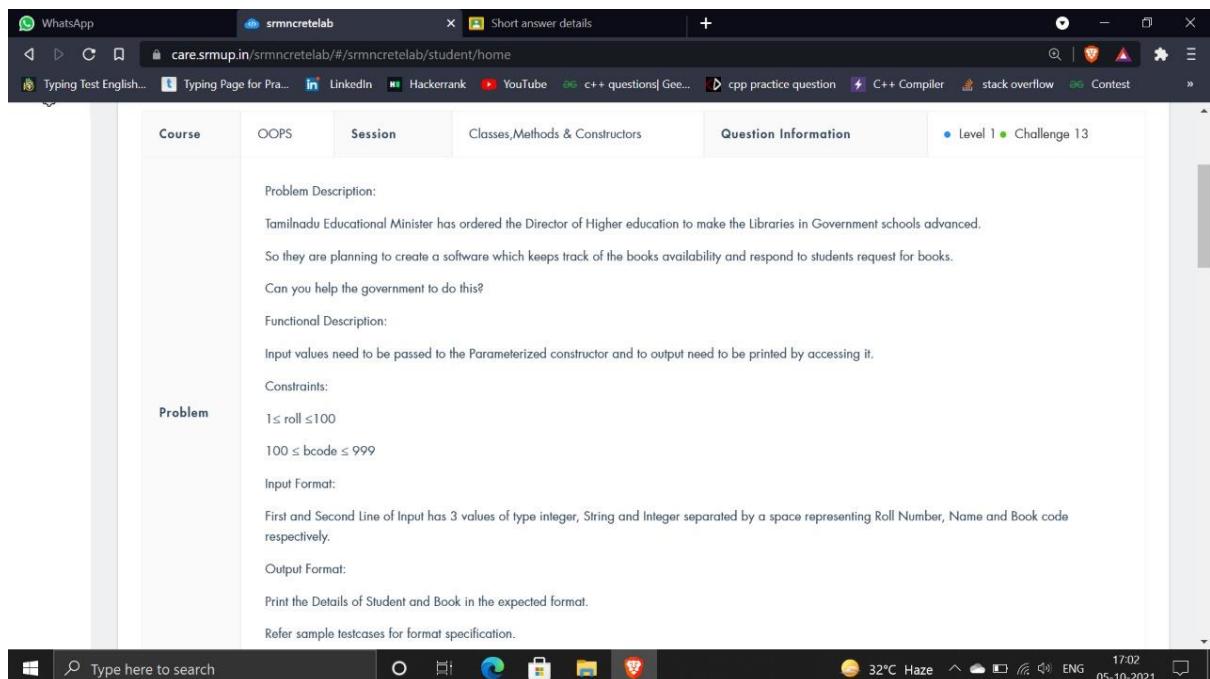
```

cout<<total;
}

};

int main(){
    int n;
    cin>>n;
    Bank CalculateMoney;
    CalculateMoney.totalMoney(n);
    return 0;
}

```



```

#include <iostream>

using namespace std;

class student
{
    string name;
    int roll;
    float height, weight;

public:
    student(){name="Bhagavan";roll=1593;height=172.5;weight=60.4;}

```

```

void set_data()
{
    cin>>name>>roll>>height>>weight;
}

void displaydata()
{
    cout<<name<<" "<<roll<<" "<<height<<" "<<weight<<endl;
}

};

int main()
{
    student s1,s2;

    s1.set_data();
    s1.displaydata();

    s2.displaydata();

    return 0;
}

```

Problem Description:

Hassan Works for the Popular Telecommunication Company.

Recently TRAI has ordered all the internet service providers to modify the STD codes from 0 to 91 for all the customers in India

So Hasan has been assigned the task of collecting the phone numbers of their customers across the country from the customer Database of his company.

After Collecting the phone numbers he has to do the following:

Replace 0 with +91 as 1st digit of STD code.

The exchange code need to be retained as it is.

And the 3rd part need to be displayed as it is.

Since the number of customers in the Database is huge hasan is finding it difficult to complete his task.

Can you help him with the programming logic to do so?

Constraints:

000≤stdcode≤099

450≤exchangecode≤800

50000≤num≤90000

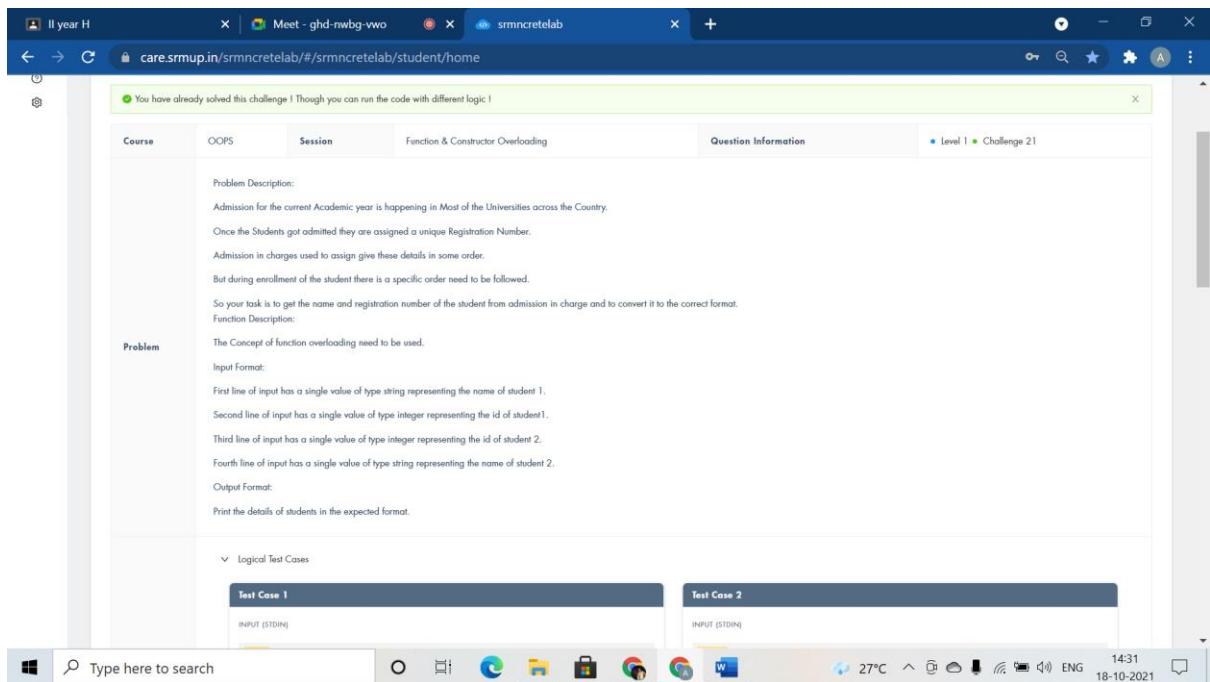
Input Format:

Only one line of input has a single value composed of STD code-Exchange Code-Number.

```
#include <iostream>
using namespace std;
```

```
class Phone
{
public:
    char n[14];
    void change()
    {
        cin>>n;
        n[0]='1';
        cout<<'9'<<n;
    }
};
```

```
int main()
{
    Phone obj;
    obj.change();
    return 0;
}
```



```
#include <iostream>

using namespace std;

class Student

{

public:

void Identity(string name,int id){

    cout<<name<<" "<<id<<endl;

}

void Identity(int id,string name){

    cout<<name<<" "<<id<<endl;

}

};

int main()

{

    Student Details;

    string name;

    int id;

    cin>>name>>id;
```

```

Details.Identity(name,id);

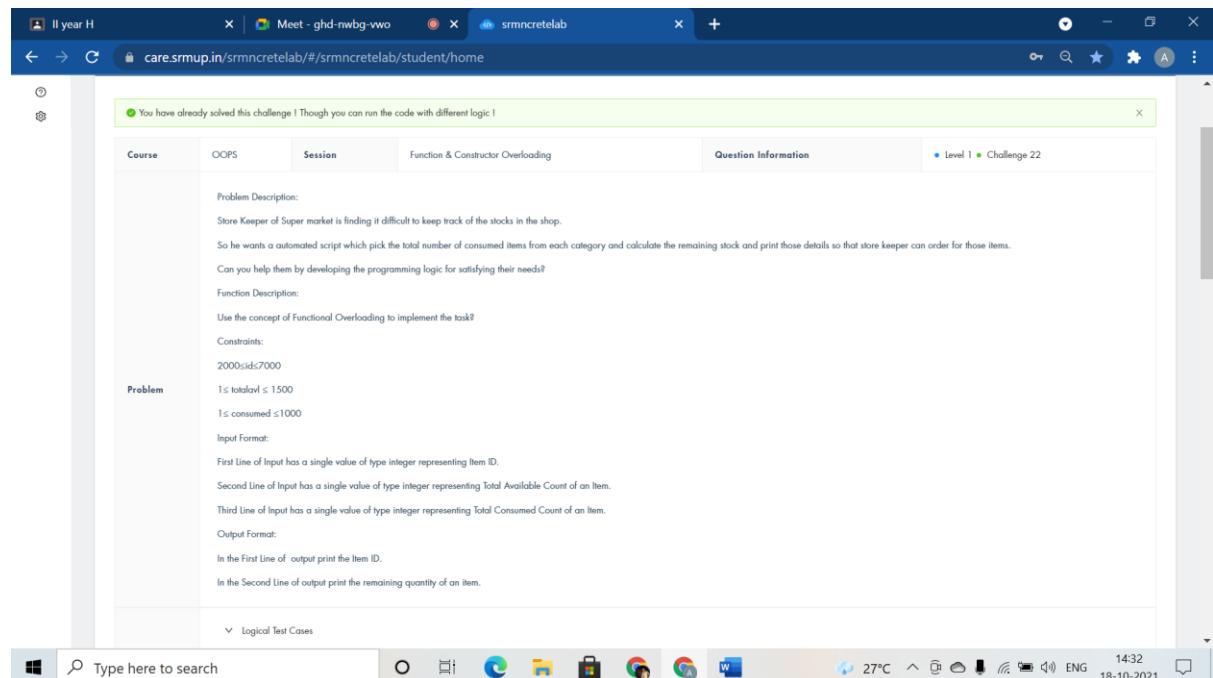
cin>>id>>name;

Details.Identity(id,name);

return 0;

}

```



```

#include <iostream>

using namespace std;

class Store{

public:

void itemcount(int id){

cout<<id<<endl;

}

void itemcount(int totalavl,int consumed){

cout<<totalavl - consumed<<endl;

}

};


```

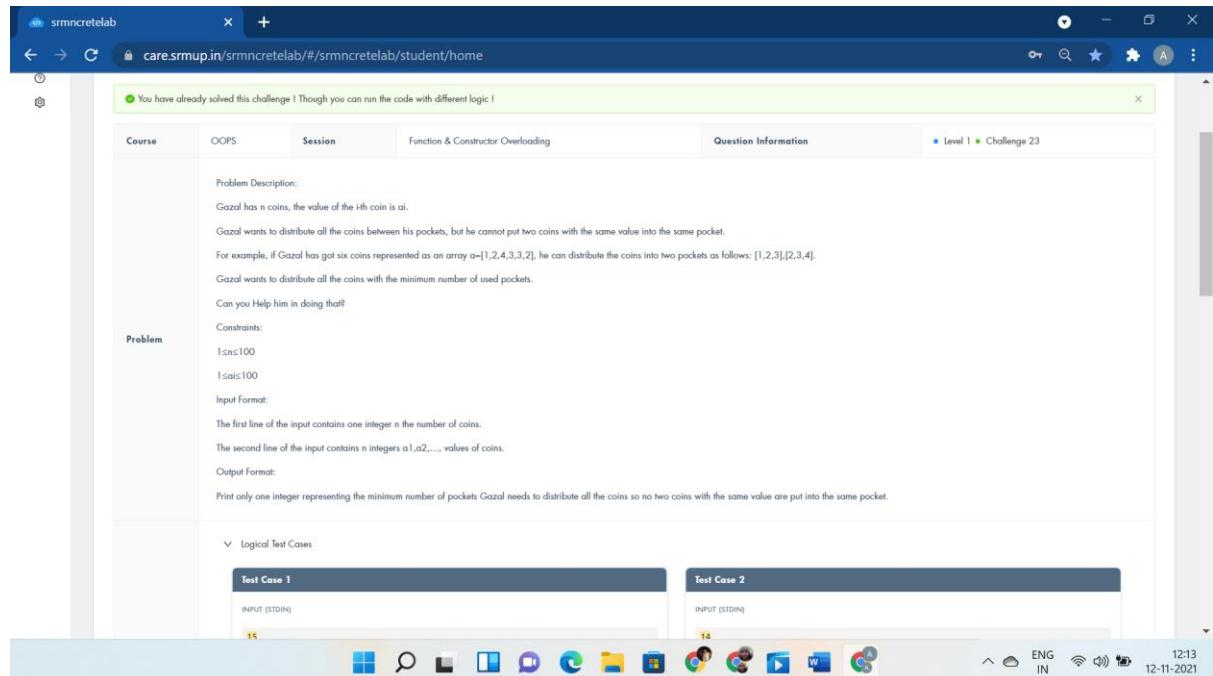
```

int main()
{
    Store purchase;

    int id,totalavl,consumed;
    cin>>id>>totalavl>>consumed;
    purchase.itemcount(id);
    purchase.itemcount(totalavl,consumed);

    return 0;
}

```



```

#include<bits/stdc++.h>

using namespace std;

int i,n,a,mx=INT_MIN,c[1000];

int res(int n);

int dis(int n,int mx);

int main(){

    cin>>n;

    mx=res(n);

    cout<<dis(n,mx);
}

```

```

    return 0;

cout<<"int* GazalCoin(int arr[],int n) int* GazalCoin(int arr[],int n,int i) GazalCoin(arr,n,0);";

}

int res(int n){

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

        cin>>a;
        c[a]++;
        mx=max(mx,c[a]);
    }

    return mx;
}

int dis(int n,int mx){

    if(n%mx==1 && n%11!=0)
        return mx+1;
    if(n%mx==1 && n%11 == 0)
        return mx;
    if(n%mx==2)
        return mx+1;
    return mx;
}

```

You have already solved this challenge! Though you can run the code with different logic!

**Course:** OOPS    **Session:** Function & Constructor Overloading    **Question Information:** Level 1 | Challenge 24

**Problem Description:**  
One of the famous politician was admitted in one of the famous hospital..  
Since the politician is one of the icon of the politics he has been given one of the best facilities available in the hospital.  
The politician was admitted in the hospital for more than a month so hospital wanted to calculate the Bill for Rooms and Medicines every week.

**Function Description:**  
Use Function Overloading to calculate the bills by taking into account the expenses and number of days

**Constraints:**  
5000 ≤ medicine ≤ 50000  
15000 ≤ room: 35000  
1 ≤ days ≤ 7

**Input Format:**  
First and Second Line of input has a single value of type integer representing amount for medicines and number of days respectively.  
Third and Fourth Line of input has a single value of type integer representing amount for room and number of days respectively.

**Output Format:**  
Print the Total Bill amount for Medicines and Room for n number of days in a separate line respectively.

**Logical Test Cases:**

Test Case 1	Test Case 2
INPUT [STDIN]	INPUT [STDIN]

27°C 14:32 18-10-2021

```
#include <iostream>

using namespace std;

class Hospital{

public:

void bill(long int mdeicinebill,int days){

    cout<<mdeicinebill*days<<endl;

}

void bill(int roomrent,int days){

    cout<<roomrent*days;

}

};

int main()

{

    Hospital ob;

    long int mdeicinebill,days;

    int roomrent;

    cin>>mdeicinebill>>days;

    ob.bill(mdeicinebill,days);
```

```

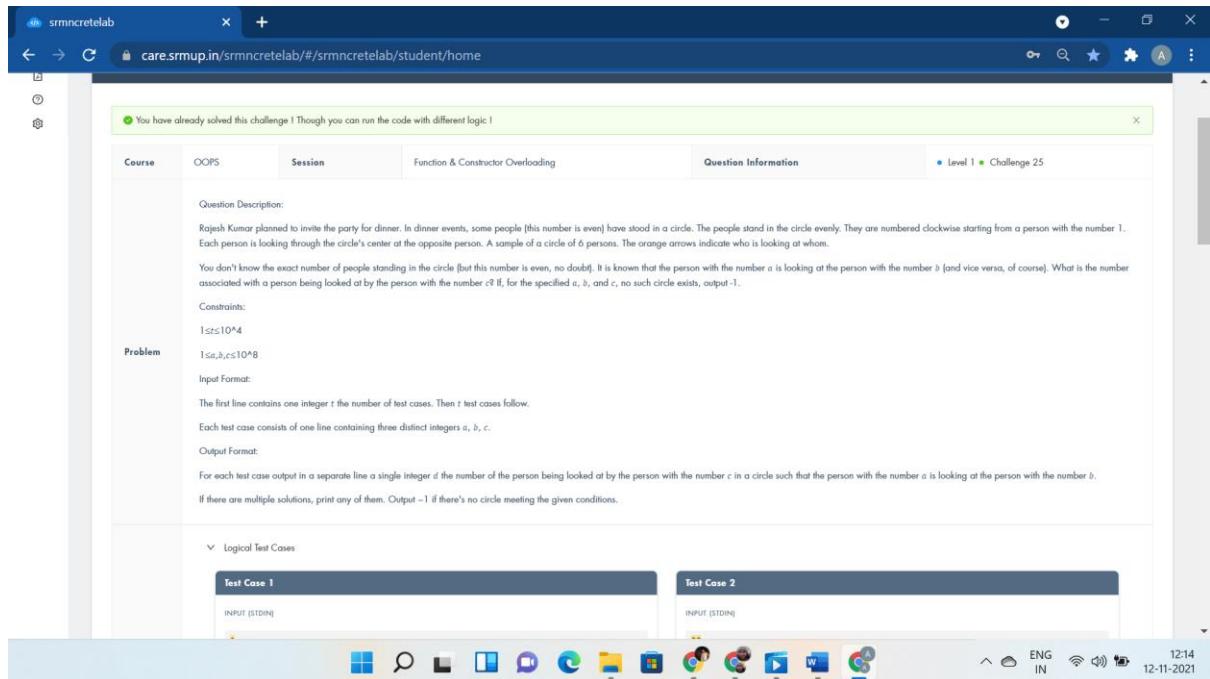
    cin>>roomrent>>days;

    ob.bill(roomrent,days);

    return 0;

}

```



```

#include<bits/stdc++.h>

using namespace std;

int i,T,a,b,c,n;

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

class solve{

public:

void get(){

    std::cin>>a>>b>>c;

    n=2*abs(a-b);

}

void get2(){

    if(c>n| | max(a,b)>n)

        cout<<"-1"<<endl;

    else if(c>n/2)

        cout<<c-n/2<<endl;
}

```

```

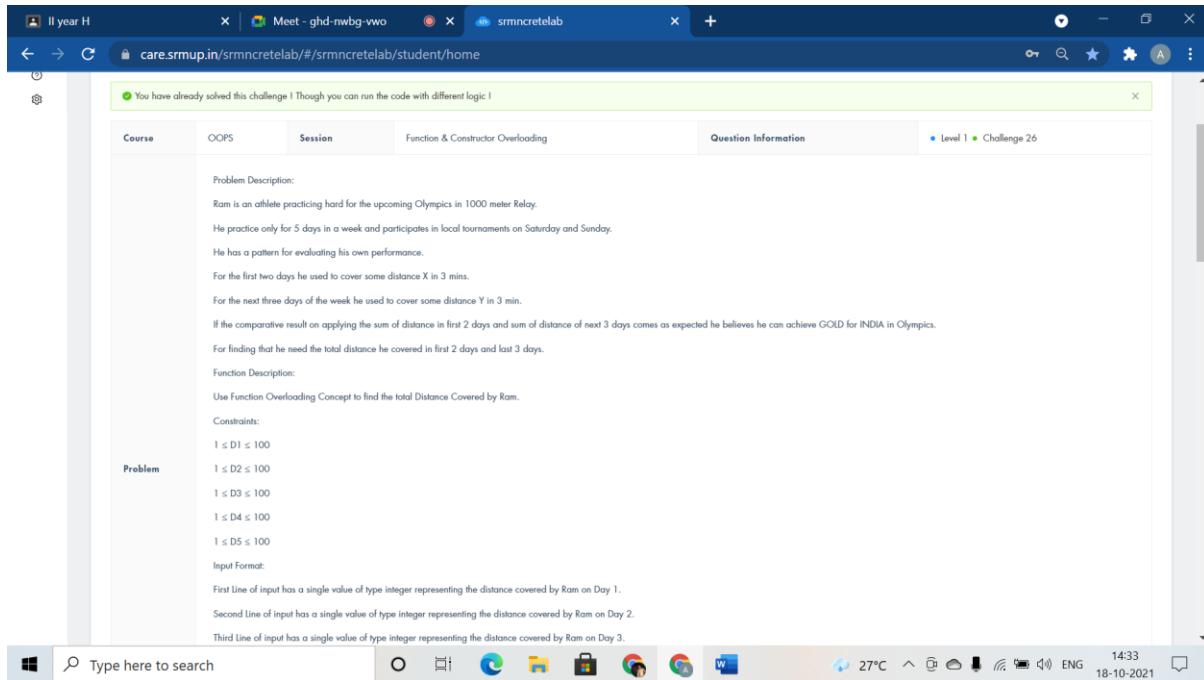
else
cout<<c+n/2<<endl;
}

};

int main(){
    cin>>T;
    solve p;
    f(i,0,T){
        p.get();
        p.get2();
    }
    return 0;
cout<<"void pline(int v[],int n) void pline(int v) else if(x>n | |x<=0)";

}

```



```
#include <iostream>
```

```
using namespace std;
```

```
class Olympic{
```

```

public:
void distance(int D1,int D2){
    cout<<D1+D2<<" meters"<<endl;
}

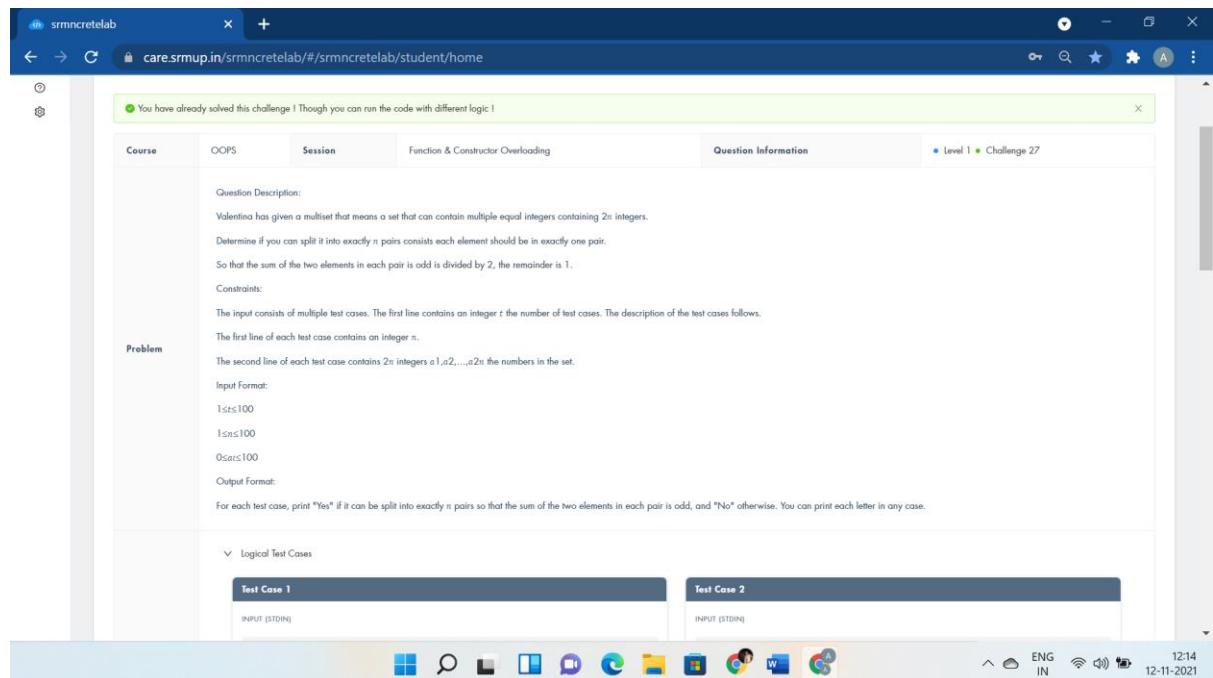
void distance(int D3, int D4, int D5){
    cout<<D3+D4+D5<<" meters"<<endl;
}

};

int main()
{
    Olympic Medal;
    int D1,D2,D3,D4,D5;
    cin>>D1>>D2>>D3>>D4>>D5;
    Medal.distance(D1,D2);
    Medal.distance(D3,D4,D5);

    return 0;
}

```



#include <iostream>

```
using namespace std;

int power(int x,int p);

int power(int x,int y,int p);

int main()

{

    int t;

    cin>>t;

    while(t--){

        int n,odd=0;

        cin>>n;

        int z=power(n,odd);

        //cout<<n<<z;

        power(n,z,1);

    }

    return 0;

}

int power(int x,int p){

    int a[2*x];

    for(int i=0;i<2*x;i++){

        cin>>a[i];

        if(a[i]%2==1)

            p++;

    }

    return p;

}

int power(int x,int y,int p){

    if(x==y)

        cout<<"Yes"<<endl;

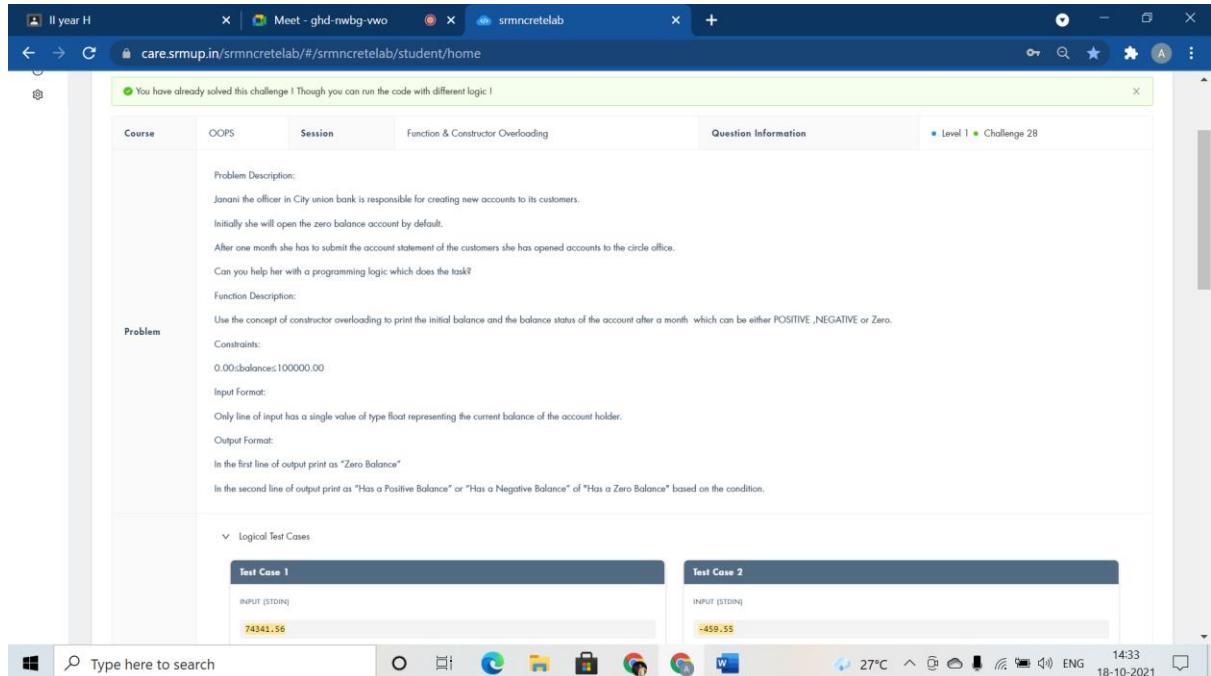
    else

        cout<<"No"<<endl;

    return 1;

}
```

}



```
#include <iostream>

using namespace std;

class AccBalance{

public:

    AccBalance(){cout<<"Zero Balance"<<endl;}

    AccBalance(int balance){

        if(balance<0)

            cout<<"Has a Negative Balance";

        else if(balance==0)

            cout<<"Has a Zero Balance";

        else

            cout<<"Has a Positive Balance";

    }

};

int main()
```

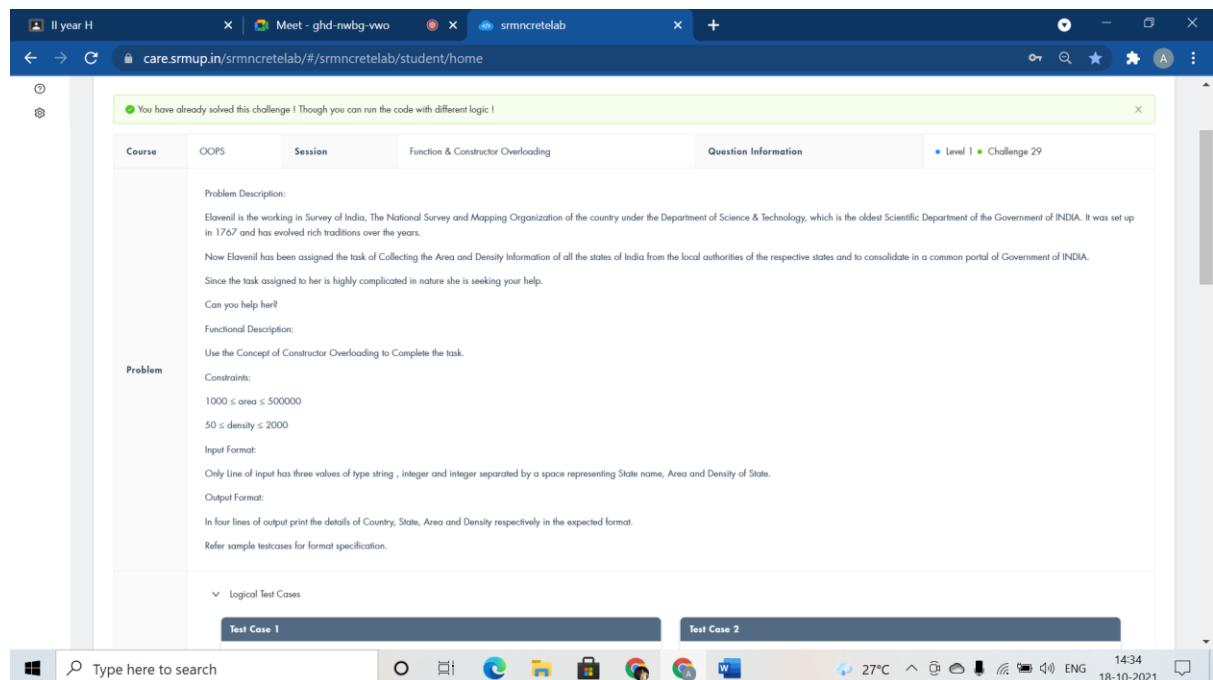
```
{
    AccBalance defltBal;

    int balance;

    cin>>balance;

    AccBalance currBal(balance);

    return 0;
}
```



```
#include <iostream>

using namespace std;

class Country{

public:

Country(){cout<<"Country:INDIA"<<endl;}

Country(char statename[100],int area,int density)

{

    cout<<"State:"<<statename<<endl<<"Area:"<<area<<endl<<"Density:"<<density<<endl;

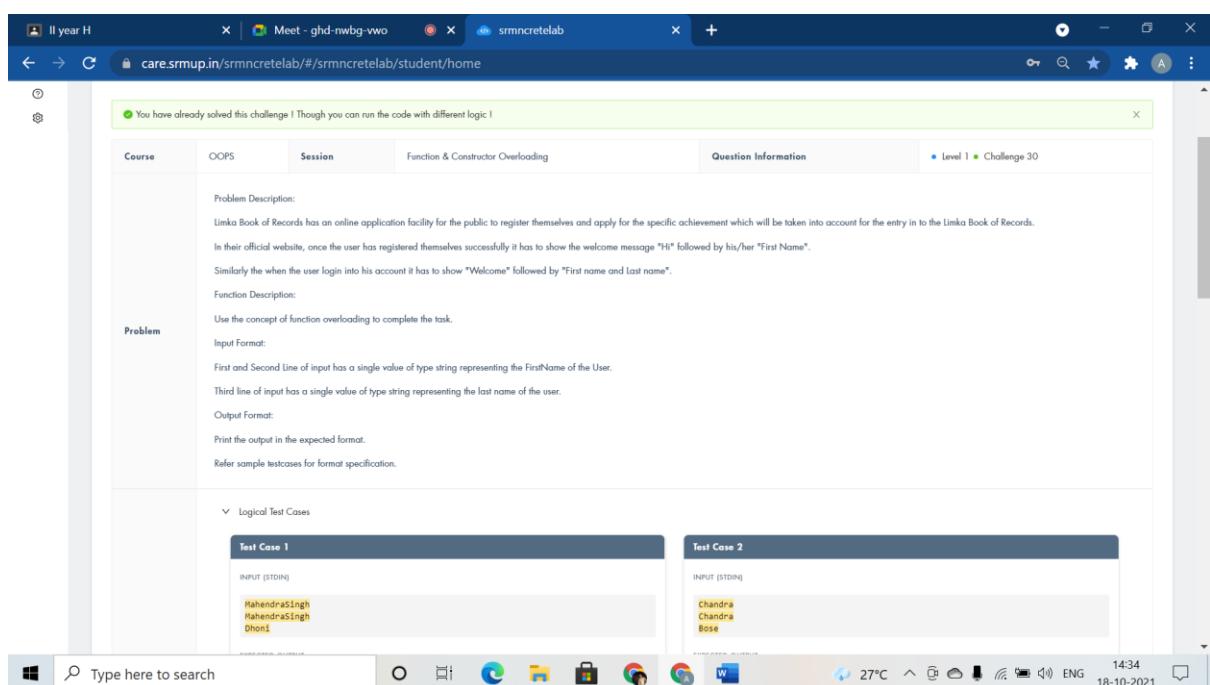
}
```

```

};

int main()
{
    Country country;
    char statename[100];
    int area,density;
    cin>>statename>>area>>density;
    Country statesofindia(statename,area,density);
    return 0;
}

```



```

#include <iostream>

using namespace std;

class Welcomemsg{
public:
    void msg(string fname){
        cout<<"Hi "<<fname<<endl;
    }
}

```

```
}

void msg(string fname,string lname){

    cout<<"Welcome "<<fname<<" "<<lname;

}

};

int main()

{

    Welcomemsg ob;

    string fname,lname;

    cin>>fname;

    ob.msg(fname);

    cin>>fname>>lname;

    ob.msg(fname,lname);

    return 0;

}
```

```
#include <iostream>

using namespace std;

class Fraction{

public:

int num,den;

Fraction(int n=0, int d=0)

{

    num=n;

    den=d;

}

Fraction operator /(Fraction const &obj){

    Fraction res;

    res.num=num * obj.den;

    res.den=den * obj.num;

    return res;

}

void display1(){

    cout<<num/den;

}

void display2(){

    cout<<num<<"+"<<den;

}
```

```

void display3(){
    cout<<"Error";
}

};

int main()
{
    int a,b,c,d;
    cin>>a>>b;
    cin>>c>>d;
    Fraction ob1(a,b), ob2(c,d);
    Fraction ob3 = ob1/ob2;
    if(ob1.den==0 || ob2.den==0){
        cout<<"Error";
        return 0;
    }
    if(ob3.den==1)
        ob3.display1();
    else{
        for(int i=2;i<50;i++)
        {
            if(ob3.num%i==0 && ob3.den%i==0)
            {
                ob3.num=ob3.num/i;
                ob3.den=ob3.den/i;
            }
        }
    }

    ob3.display2();
}

return 0;
}

```

You have already solved this challenge ! Though you can run the code with different logic !

**Course**      **OOPS**      **Session**      **Operator Overloading**      **Question Information**      **Level 1 • Challenge 33**

**Problem**

Question description  
Rahul and Ramesh are military officers. They are travelling to enjoy the vacation by train.  
They are planned to play a game during their travel that they are interested in how many ways there are in scrambling the letters.  
One fellow should said the length of the word added by 1 and other fellow should give the number of ways the letters to be scrambled.  
For example, If suppose Rahul gave the length of the word is 6. Then Ramesh should be subtracted that 1 and calculate for the word's length as 6-1. He have 5 choices for the first letter, once he have chosen the first letter there are 4 choices for the second letter, and then three choices for the third letter, two for the fourth letter, and only one choice for the last letter. Hence there are  $5[4][3][2][1] = 5 \times 120$  choices.

Can you help them to verify the answer?  
Constraints  
1≤n≤10  
Input Format  
The only line of input has one numbers n of type integer.  
Output Format  
Print the answer of the factorial of n-1.

Logical Test Cases

Test Case 1	Test Case 2
INPUT [STDIN] 6	INPUT [STDIN] 7
720	5040

```
#include <iostream>
```

```
using namespace std;
```

```
class Scrum{
```

```
public:
```

```
int n;
```

```
Scrum(int h)
```

```
{
```

```
n=h;
```

```
}
```

```
Scrum operator -- (int){
```

```
    Scrum T(int h);
```

```
    --n;
```

```
    return 1;
```

```
}
```

```
void display(){
```

```
    int res=1;
```

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

```
        res=res*i;
```

```
}
```

```
    cout<<res;
```

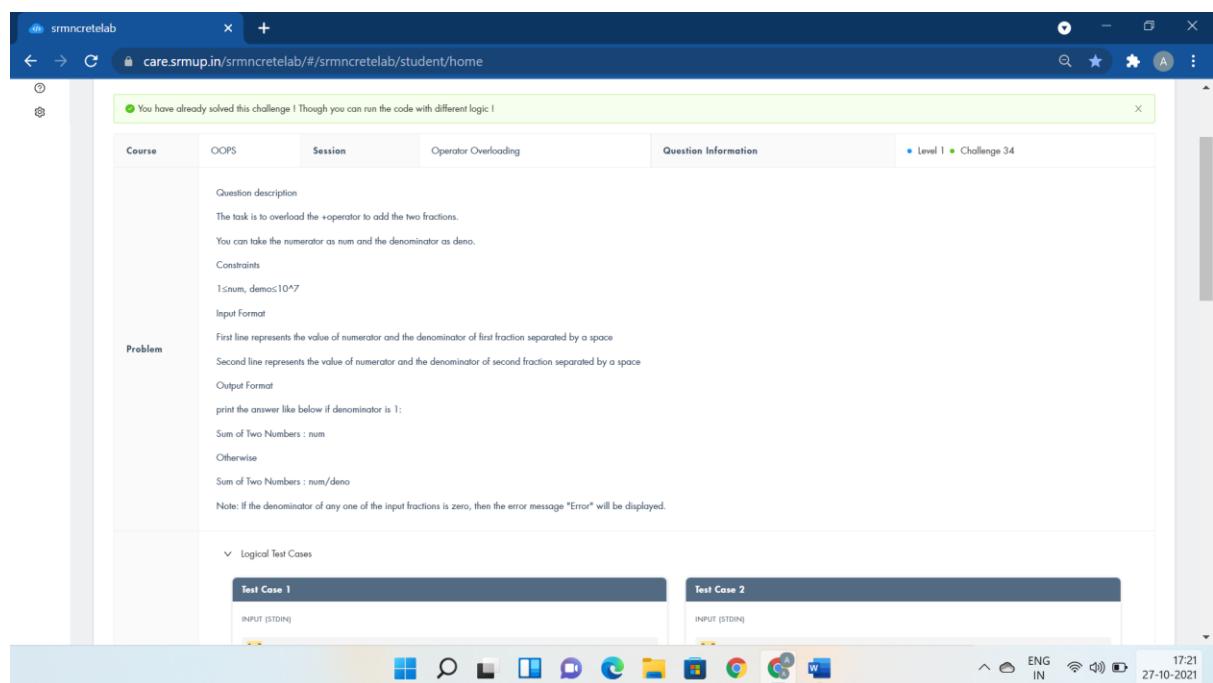
```
}
```

```
};
```

```

int main()
{
    int n;
    cin>>n;
    Scrum T(n);
    T--;
    T.display();
    return 0;
}

```



```

#include<iostream>

using namespace std;

class Fraction
{
public:
    int num,den;

    Fraction()
    {
        num=0;
        den=0;
    }
}

```

```
void getinput()
{
    cin>>num>>den;
}

Fraction operator +(Fraction obj)
{
    Fraction temp;
    temp.num=(num*obj.den)+(den*obj.num);
    temp.den=den*obj.den;
    return temp;
}

int main()
{
    Fraction f1,f2,add;
    f1.getinput();
    f2.getinput();
    add=f1+f2;
    if(add.den==0)
        cout<<"Error";
    else if(add.num%add.den == 0)
        cout<<add.num/add.den;
    else
        cout<<add.num<<"/"<<add.den;
    return 0;
}
```

You have already solved this challenge! Though you can run the code with different logic!

**Course:** OOPS    **Session:** Operator Overloading    **Question Information:** Level 1 | Challenge 35

**Problem:**

**Input Format:**

First line represent the first row elements of given matrix  
Second line represent the second row elements of given matrix

**Output Format:**

Print the determinant of a given matrix

**Logical Test Cases:**

Test Case 1	Test Case 2
INPUT (STDIN) 1 2 3 4	INPUT (STDIN) 2 0 1 2
EXPECTED OUTPUT	EXPECTED OUTPUT

```
#include <iostream>
```

```
using namespace std;
```

```
class matrix{
```

```
public:
```

```
int operator ~(){
```

```
    int a,b,c,d;
```

```
    cin>>a>>b>>c>>d;
```

```
    return a*d-b*c;
```

```
}
```

```
};
```

```
int main()
```

```
{
```

```
    matrix t;
```

```
    cout<<~t;
```

```
    return 0;
```

```
}
```

```
#include<iostream>

using namespace std;

class Complex {

private:
    int real, imag;

public:
    Complex(int r = 0, int i = 0) {real = r; imag = i;}

    Complex operator+(int a) {
        Complex res;
        res.real = real + a;
        res.imag = imag;
        return res;
    }

    Complex operator+(Complex obj) {
        Complex res;
        res.real = real + obj.real;
        res.imag = imag + obj.imag;
        return res;
    }

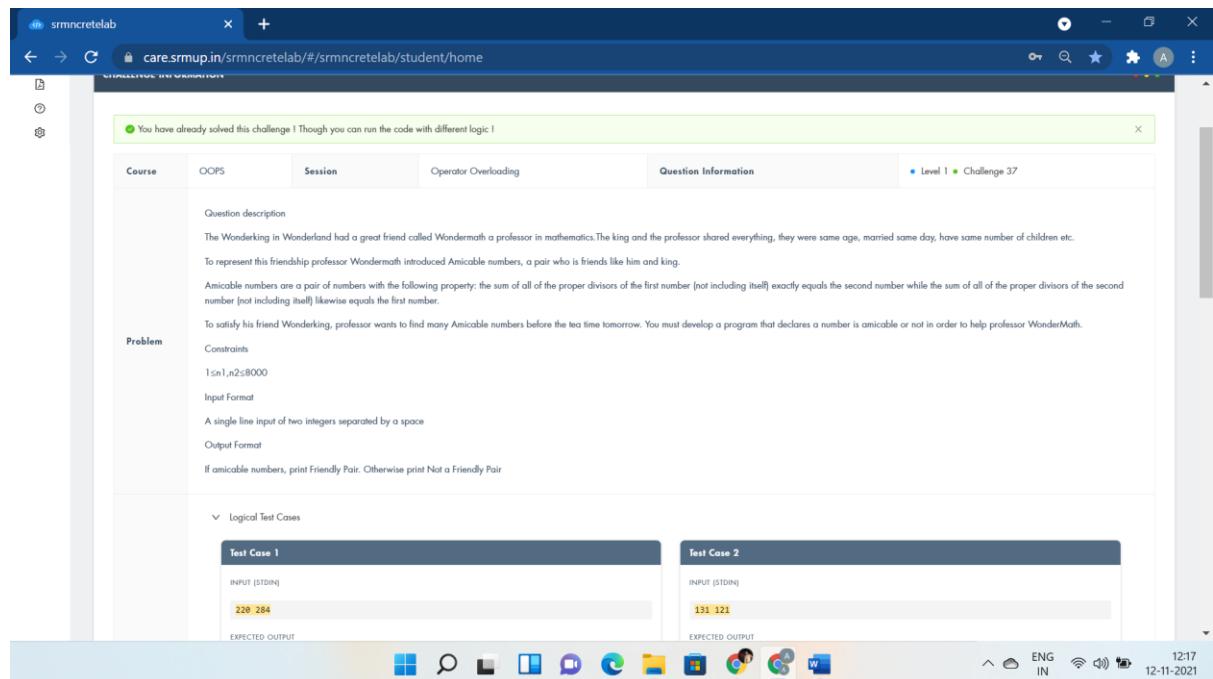
    void print() { cout << real << " + " << imag << "i" << endl; }

};
```

```

int main()
{
    int a,b,c;
    cin>>a>>b>>c;
    Complex i1(a, b);
    Complex i2 = i1 + c;
    i1.print();
    i2.print();
    (i1+i2).print();
}

```



```

#include<iostream>

using namespace std;

class compare{

public:

    int first,sum1=0;

    compare(int x){

        first=x;

    }

    void f(){

        //first1=first;
    }
}

```

```

for(int i=1; i<=first/2 ; i++)
{
    //finding and adding divisors of first number
    if(first%i==0)
        sum1=sum1+i;
}

void operator ==(compare t2){

    if(first==t2.sum1 && t2.first==sum1)
        cout<<"Friendly Pair";
    else
        cout<<"Not a Friendly Pair";
}

//main program

int main()
{
    int first,second;
    //user input
    cin>>first;
    //user input
    cin>>second;
    compare t1(first),t2(second);
    t1.f();
    t2.f();
    t1==t2;
    return 0;
}

```

You have already solved this challenge! Though you can run the code with different logic!

Course COPS Session Operator Overloading Question Information Level 1 Challenge 38

Question description  
The sum of the squares of the first ten natural numbers is,  
 $1^2 + 2^2 + 3^2 + \dots + 10^2 = 385$   
The square of the sum of the first ten natural numbers is,  
 $(1 + 2 + 3 + \dots + 10)^2 = 55^2 = 3025$   
Hence the difference between the sum of the squares of the first ten natural numbers and the square of the sum is  
 $3025 - 385 = 2640$   
Find the difference between the sum of the squares of the first  $n$  natural numbers and the square of the sum.  
Constraints  
 $1 \leq n \leq 100$   
Function Description  
Create a class Diff with a member functions sumofsquare and squareofsum with int datatype and use insertion overloading  
Constraints  
 $1 \leq n \leq 100$   
Input Format  
A single line input represent the first  $n$  natural numbers  
Output Format  
Print the difference of the sum of square and the square of sum of the series of first  $n$  natural numbers

```
#include <iostream>

using namespace std;

class Diff{

public:
    int n;

    void getdata(){
        cin>>n;
    }

    int sumofsquare();
    int sumofnumsq(){
        return n*(n+1)*(2*n+1)/6;
    }

};

int Diff :: sumofsquare(){
    return n*n*(n+1)*(n+1)/4;
}

int main()
{
    Diff n;
    if(0)
        cout<<"friend void operator >> (istream &in, Diff &obj )";
}
```

```

n.getdata();

//int sq=n*n*(n+1)*(n+1)/4;

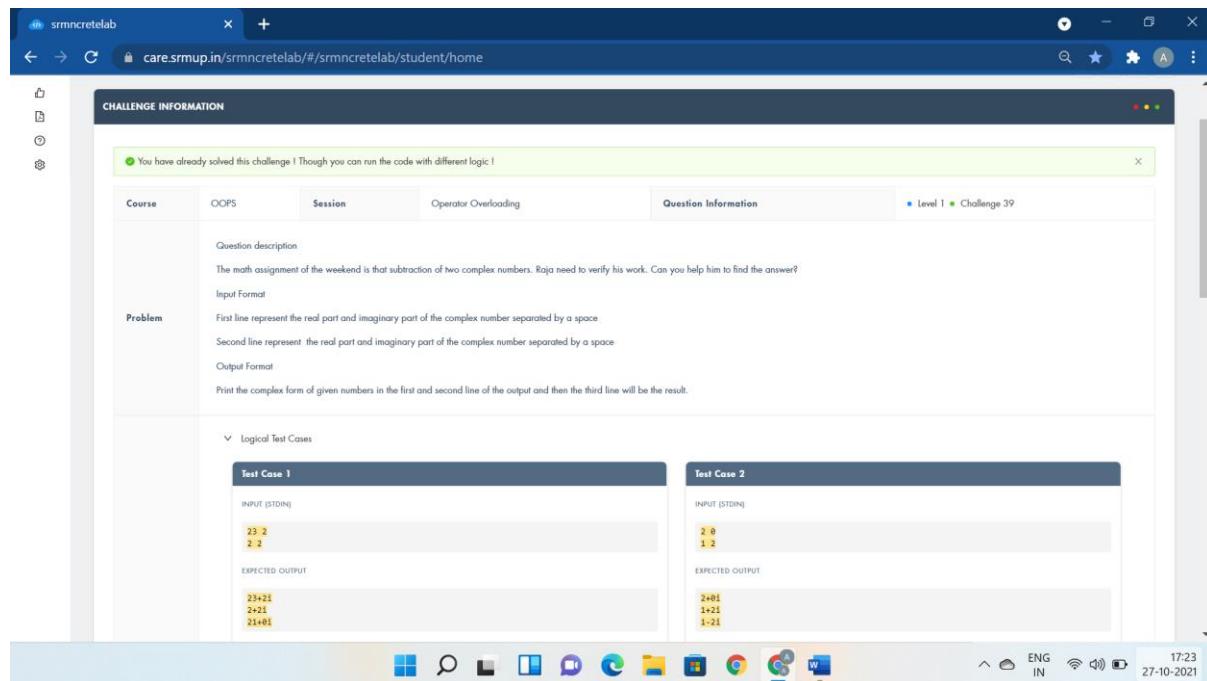
//int sq2=n*(n+1)*(2*n+1)/6;

cout<<n.sumofsquare()-n.sumofnumsq();

return 0;

}

```



```

#include <iostream>

using namespace std;

class complex

{
private:
    float real;
    float imag;

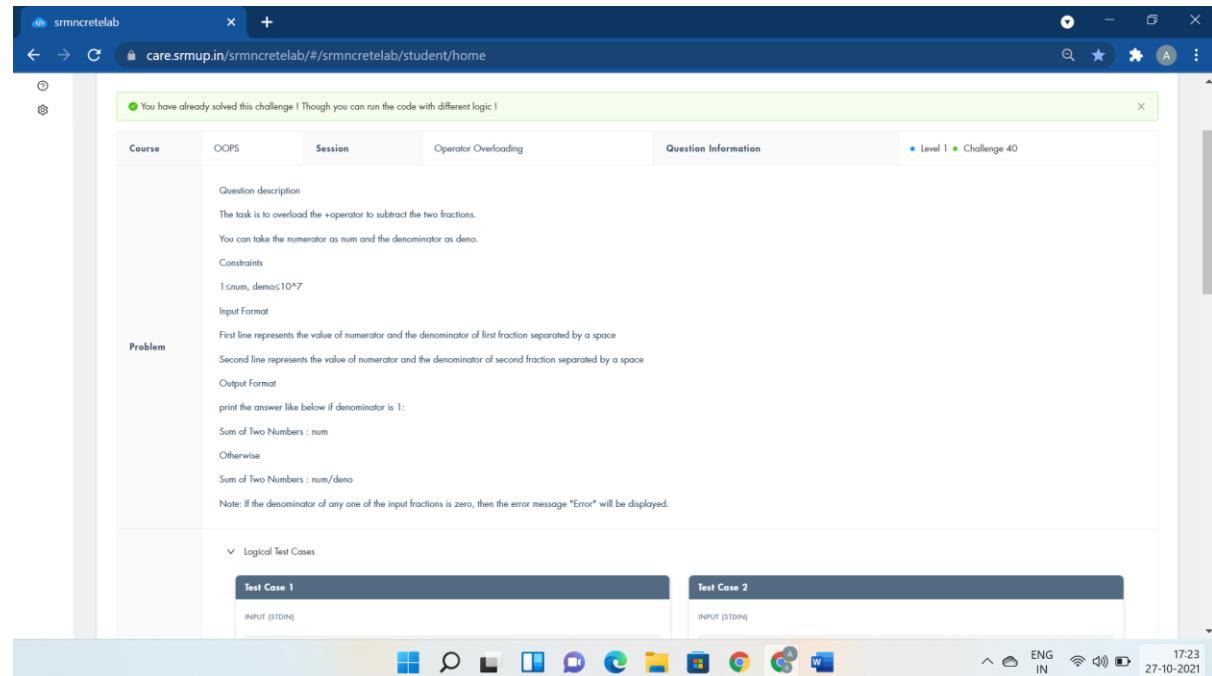
public:
    complex() {cin>>real>>imag;}
    complex operator-(complex ob)
    {
        complex t;
        t.real = real - ob.real;
        t.imag = imag - ob.imag;
        return t;
    }
}

```

```
}
```

```
void output()
{
    if(imag < 0)
        cout<< real << imag << "i" << endl;
    else
        cout<< real << "+" << imag << "i" << endl;
}
};

int main()
{
    complex c1, c2;
    c1.output();
    c2.output();
    (c1 - c2).output();
    return 0;
}
```



```
#include<iostream>

using namespace std;

class Fraction
```

```

{
public:
int num,den;

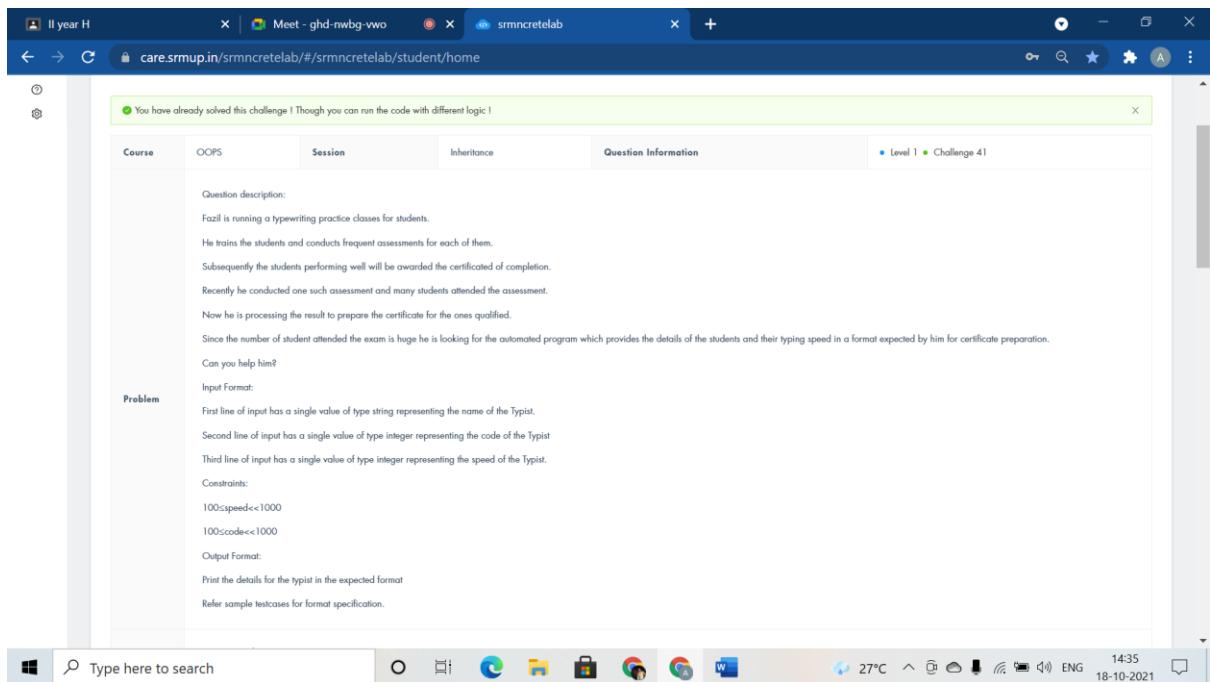
Fraction()
{
    num=0;
    den=0;
}

void getinput()
{
    cin>>num>>den;
}

Fraction operator -(Fraction obj)
{
    Fraction temp;
    temp.num=(num*obj.den)-(den*obj.num);
    temp.den=den*obj.den;
    return temp;
};

int main()
{
    Fraction f1,f2,add;
    f1.getinput();
    f2.getinput();
    add=f1-f2;
    if(add.den==0)
        cout<<"Error";
    else if(add.num%add.den == 0)
        cout<<add.num/add.den;
    else
        cout<<add.num<<"/"<<add.den;
    return 0;
}

```



```
#include <iostream>

using namespace std;

class staff{

public:

    int code,speed;

    string name;

    void getdata();

    void display();

};

void staff::getdata(){

    cin>>name>>code>>speed;

}

void staff::display(){

    cout<<"Name:"<<name<<endl<<"Code:"<<code<<endl<<"Speed"<<speed;

}

class typist: public staff{

public:

    void getdata();
```

```

void display();

};

void typist::getdata(){

    cin>>name>>code>>speed;

}

void typist::display(){

    cout<<"Name:"<<name<<endl<<"Code:"<<code<<endl<<"Speed:"<<speed;

}

int main()

{

typist t;

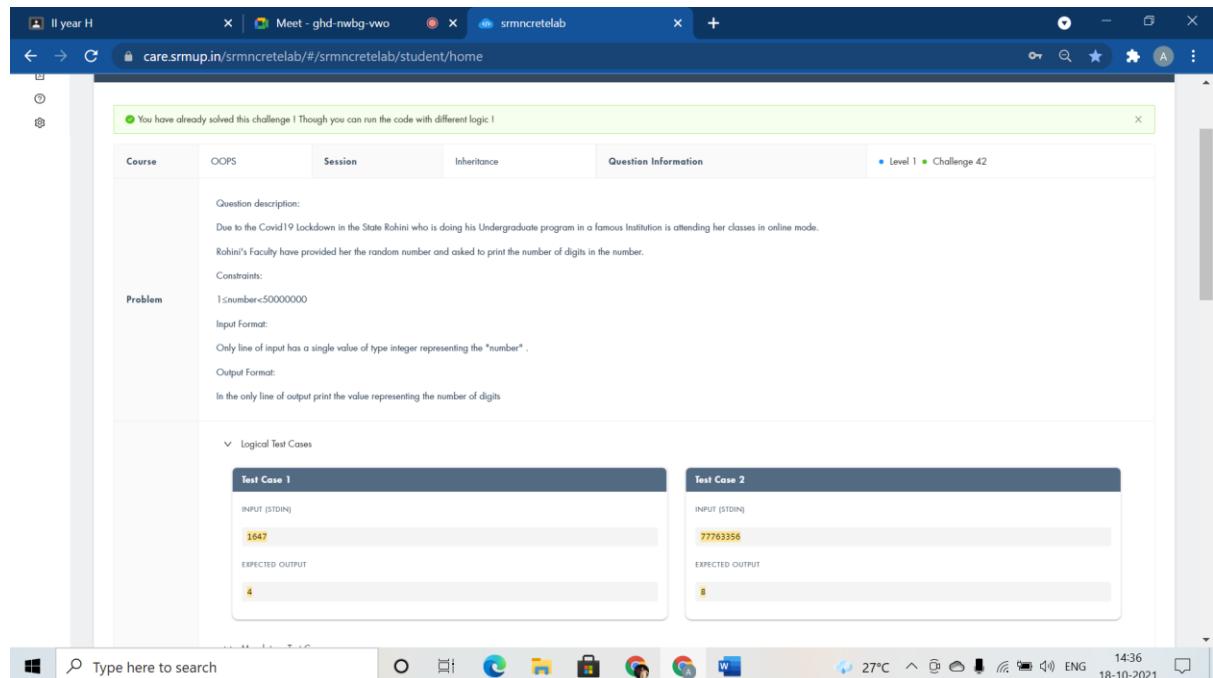
t.getdata();

t.display();

return 0;

}

```



```

#include <iostream>

using namespace std;

```

```
class Assignement{  
public:  
    int num;  
    void get(){  
        cin>>num;  
    }  
    void display(){  
        int count=0;  
        while(num!=0){  
            count++;  
            num/=10;  
        }  
        cout<<count;  
    }  
};  
class Student:public Assignement{  
};  
int main()  
{  
    Student obj;  
    obj.get();  
    obj.display();  
    return 0;  
}
```

You have already solved this challenge! Though you can run the code with different logic!

**Course:** OOPS    **Session:** Inheritance    **Question Information:** Level 1 • Challenge 43

**Question Description:**

Analia is developing an application to help customers who come to her supermarkets such as the price of the item that customers buy and display each item's price, the subtotal of the sale, the amount of sales tax, and the total.

Assume the sales tax is 6%. So you have help to Analia holds the prices of the five items in five variables.

**Constraints:**

`1 ≤ itemOne ≤ 100000`  
`1 ≤ itemTwo ≤ 100000`  
`1 ≤ Price of itemThree ≤ 100000`  
`1 ≤ Price of itemFour ≤ 100000`  
`1 ≤ Price of itemFive ≤ 100000`

**Problem:**

**Input Format:**

First line of input has a single value of type integer representing Price of itemOne.  
Second line of input has a single value of type integer representing Price of itemTwo.  
Third line of input has a single value of type integer representing Price of itemThree.  
Fourth line of input has a single value of type integer representing Price of itemFour.  
Fifth line of input has a single value of type integer representing Price of itemFive.

**Output Format:**

Print the result as per format.  
Refer sample testcases for format specification.

```
#include <iostream>

using namespace std;

class market{

public:

float i1,i2,i3,i4,i5;

float Subtotal,tax;

void items(){

    cin>>i1>>i2>>i3>>i4>>i5;

}

void buy(){

    Subtotal=(i1+i2+i3+i4+i5);

    cout<<"Subtotal="$<<Subtotal<<endl;

    tax=0.06*i1+0.06*i2+0.06*i3+0.06*i4+0.06*i5;

    cout<<"Tax="$<<tax<<endl;

    cout<<"Total="$<<Subtotal+tax;

}

};

class customer:public market{
```

```

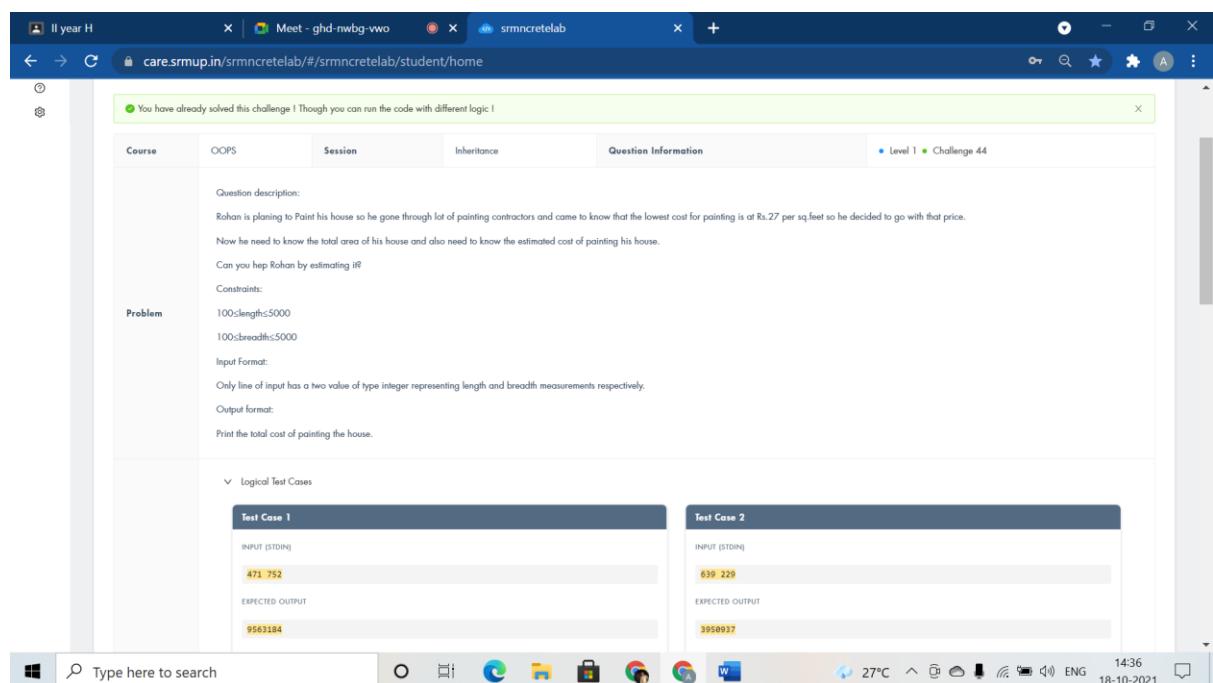
};

int main()
{
    customer c;

    c.items();

    c.buy();
}

```



```

#include <iostream>

using namespace std;

class ReceiveMesurement{

public:

    int l,b;

    void painingarea(){

        cin>>l>>b;

        cout<<l*b*27;
    }
}

```

```

};

class CalculateArea : public ReceiveMesurement{

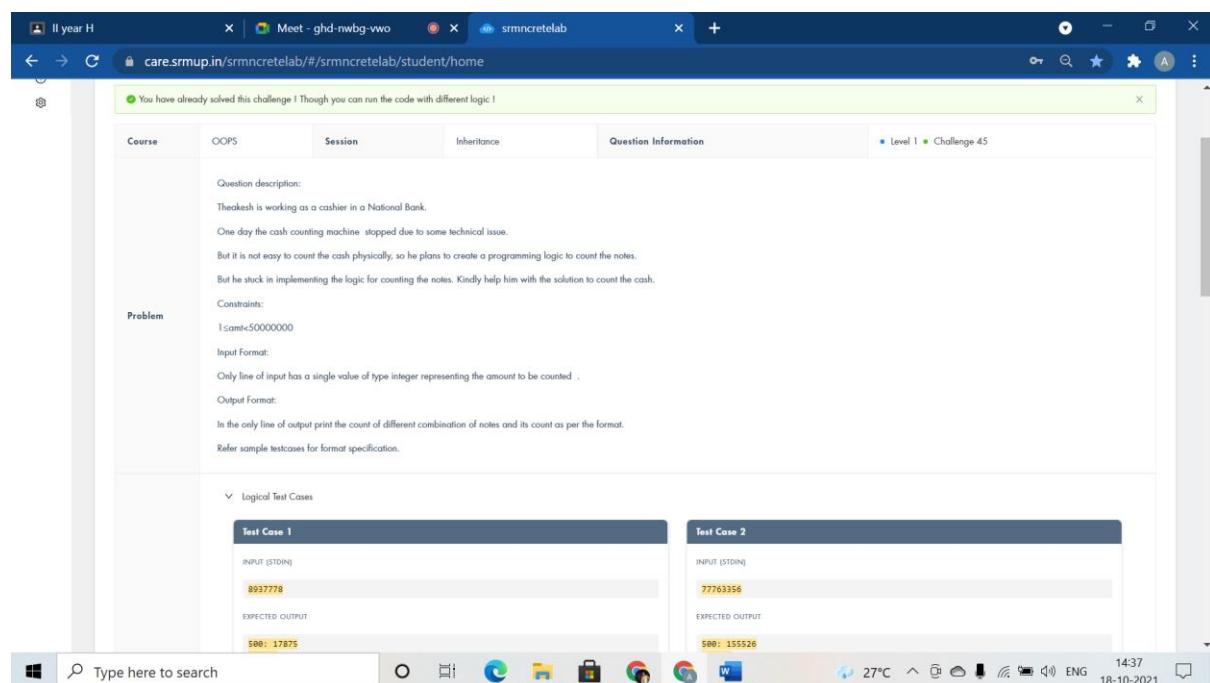
};

int main()
{
    CalculateArea mt;

    mt.painingarea();

    return 0;
}

```



```

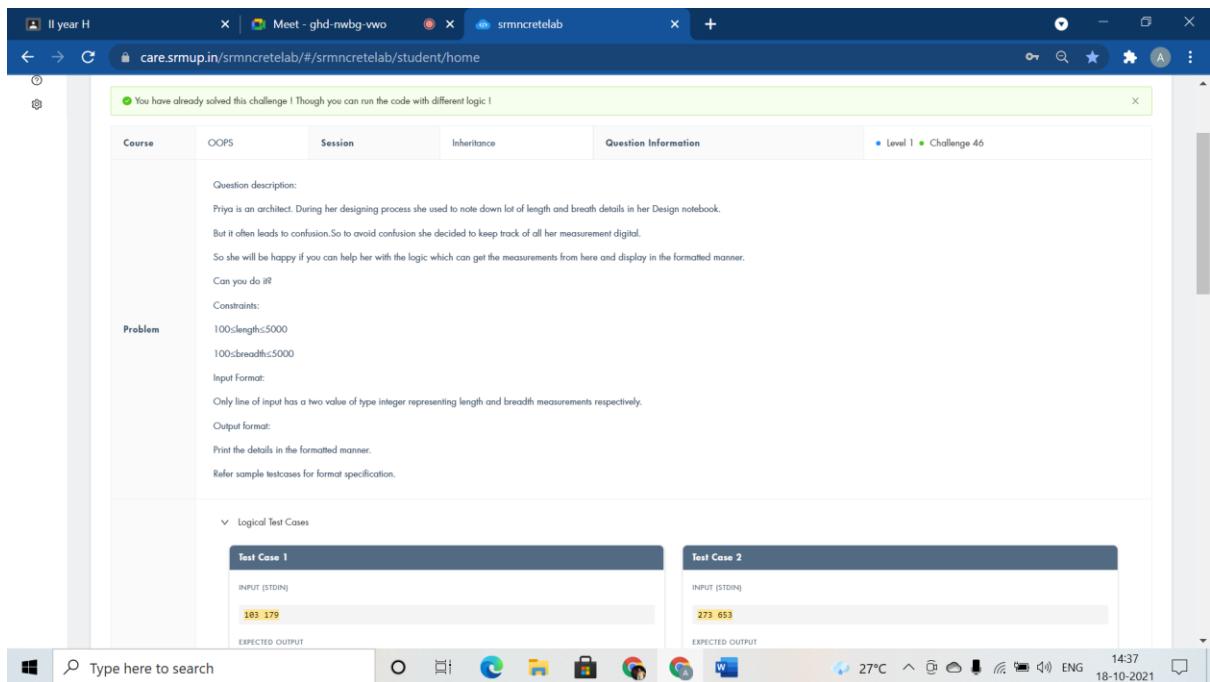
#include <iostream>

using namespace std;

class Bank{
public:
    int n;
    void get(){
        cin>>n;
    }
    void display(){
        cout<<"500: "<<n/500<<endl;
    }
}

```

```
n=n%500;  
cout<<"200: "<<n/200<<endl;  
n=n%200;  
cout<<"100: "<<n/100<<endl;  
n=n%100;  
cout<<"50: "<<n/50<<endl;  
n=n%50;  
cout<<"10: "<<n/10<<endl;  
n=n%10;  
cout<<"5: "<<n/5<<endl;  
n=n%5;  
cout<<"1: "<<n<<endl;  
}  
};  
class CashCounting:public Bank{  
};  
int main()  
{  
    CashCounting obj;  
    obj.get();  
    obj.display();  
    return 0;  
}
```



```
#include <iostream>

using namespace std;

class ReceiveMesurement{

public:

int l,b;

void display(){

cin>>l>>b;

cout<<"Length:"<<l<<" metres"<<endl;

cout<<"Breadth:"<<b<<" metres";

}

};

class FormatMesurement : public ReceiveMesurement{

};

int main()

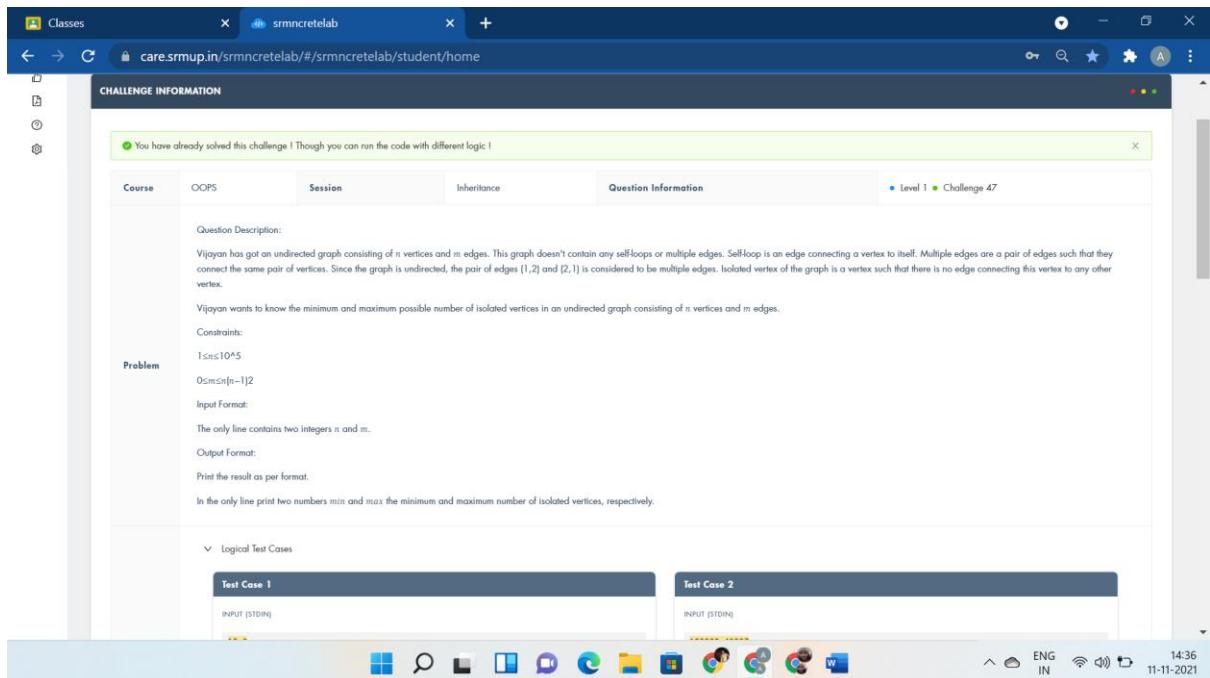
{

FormatMesurement mt;

mt.display();

return 0;
}
```

}



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

using namespace std;

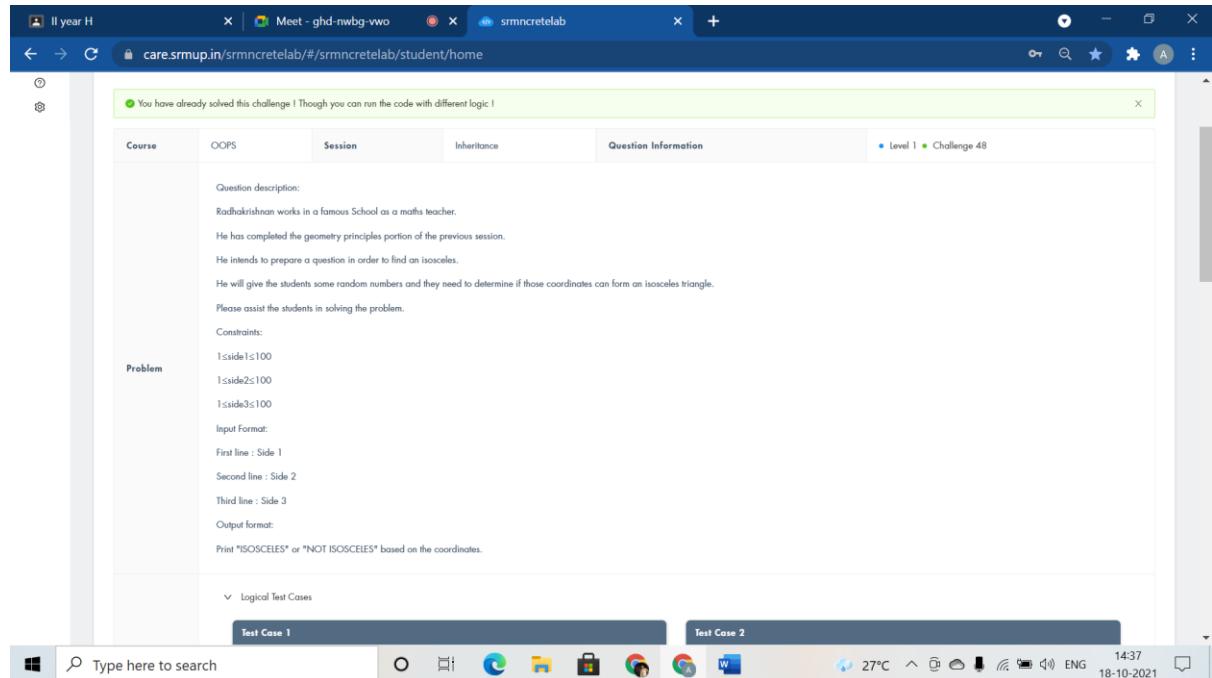
class graph{
public:
    void edge(){}
};

class pairs:public graph{
public:
    long long int n,m,k=0;
    void vertex(){}
    cin>>n>>m;
    cout<<max(0ll,n-2*m)<<" ";
    while(k*(k-1)/2<m) k++;
    cout<<n-k<<endl;
}
};
```

```

int main()
{
    pairs pa;
    pa.edge();
    pa.vertex();
    return 0;
}

```



```

#include <iostream>

using namespace std;

class triangle{
public:
    int a,b,c;
    void read(){
        cin>>a>>b>>c;
    }
    void check(){
        if(a==b || b==c || a==c)

```

```

cout<<"ISOSCELES";
else
cout<<"NOT ISOSCELES";
}

};

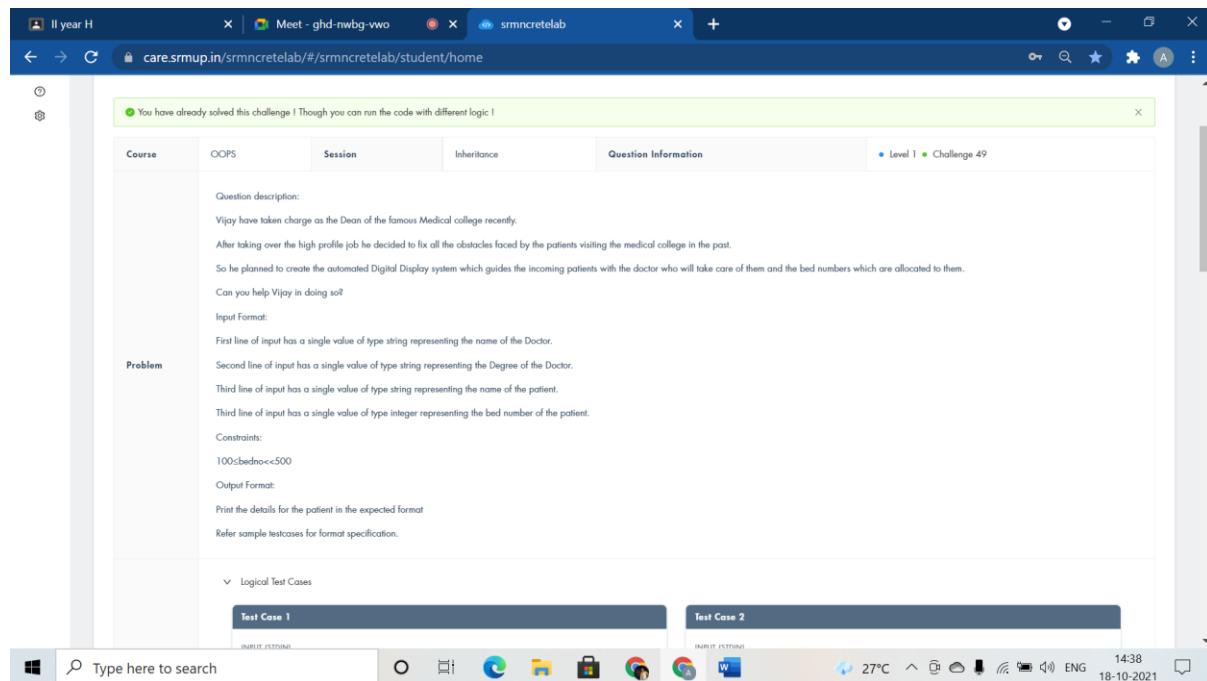
class isosceles : public triangle {

};

int main()
{
    isosceles obj;
    obj.read();
    obj.check();

    return 0;
}

```



```
#include <iostream>
```

```
using namespace std;
```

```
class doctor{
```

```
public:
```

```
string name,degree,pname;  
int no;  
  
void getedu(){  
    cin>>name>>degree>>pname;  
}  
  
void getdata(){  
    cin>>no;  
}  
  
void dispedu(){  
    cout<<"Doctor Name:"<<name<<endl<<"Doctorate Degree:"<<degree<<endl<<"Patient  
Name:"<<pname<<endl;  
}  
  
void dispdata(){  
    cout<<"Bed Number:"<<no;  
}  
};  
  
class patient:public doctor{  
};  
  
int main()  
{  
    patient p;  
    p.getedu();  
    p.getdata();  
    p.dispedu();  
    p.dispdata();  
    return 0;  
}
```

You have already solved this challenge! Though you can run the code with different logic!

**Course:** OOPS    **Session:** Inheritance    **Question Information:** Level 1 • Challenge 50

**Question Description:**  
Dayalan is a newly appointed lecturer of a government college in Sengipatti village near Thanjavur city. He is unhappy with the education system and is also worried about the pitiable condition of education of government colleges. After joining the college, he tries to change the college student environment. Dayalan's decision for the change does not go well with the other teachers and students. Slowly, Dayalan gets popular among the class students. One day Dayalan tells his students to use programming and multiplication table 10,3,8,7 based on the user choice concept.  
Option as follows 1 for 10 tables. 2 for three tables. 3 for eight tables. 4 for seven table

**Problem:**

**Constraints:**  
1 ≤ option ≤ 4

**Input Format:**  
The first line of input has a single value of type integer representing option.

**Output Format:**  
Print the result as per format.  
Refer sample test cases for format specification.

**Logical Test Cases:**

Test Case 1	Test Case 2
INPUT [STDIN] 1	INPUT [STDIN] 4
EXPECTED OUTPUT 10	EXPECTED OUTPUT 8

```
#include <iostream>
```

```
using namespace std;
```

```
class teacher{
```

```
public:
```

```
int num;
```

```
void setdata(int n)
```

```
{
```

```
if(n==1)
```

```
    num=10;
```

```
else
```

```
    num=7;
```

```
}
```

```
void setdata2(int n)
```

```
{
```

```
if(n==2)
```

```
    num=3;
```

```
else
```

```
    num=8;
```

```
}

void tentable(){

    for(int i=1;i<=10;i++)
        cout<<num<<"*"<<j<<"="<<num*i<<endl;
}

};

class ten:public teacher{

};

class three:public teacher{

};

class eight:public teacher{

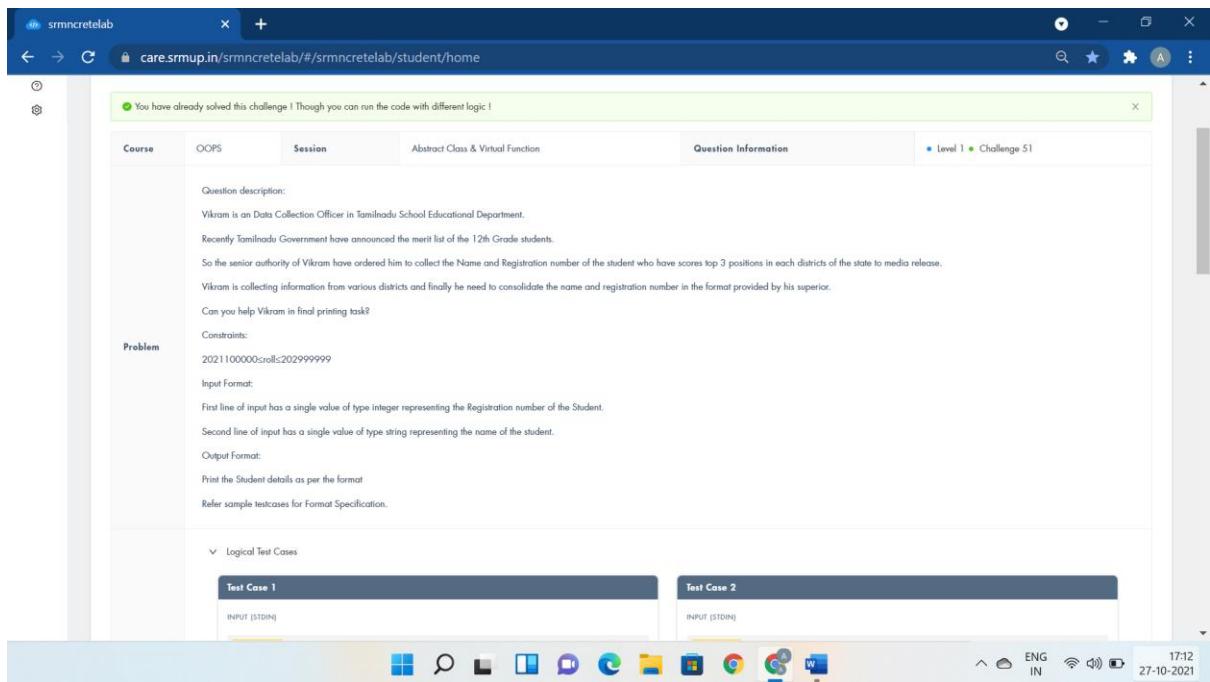
};

class seven:public teacher{

};

int main()

{
    int n;
    cin>>n;
    teacher t;
    if(n==1 || n==4)
        t.setdata(n);
    if(n==2 || n==3)
        t.setdata2(n);
    t.tentable();
    return 0;
}
```



```
#include <iostream>

using namespace std;

class School{

public:

    int roll;

    string name;

    virtual void getdata(){};

    virtual void display(){};

};

class District : public School{

    void getdata();

    void display();

};

void District :: getdata(){

    cin>>roll>>name;

}

void District :: display(){

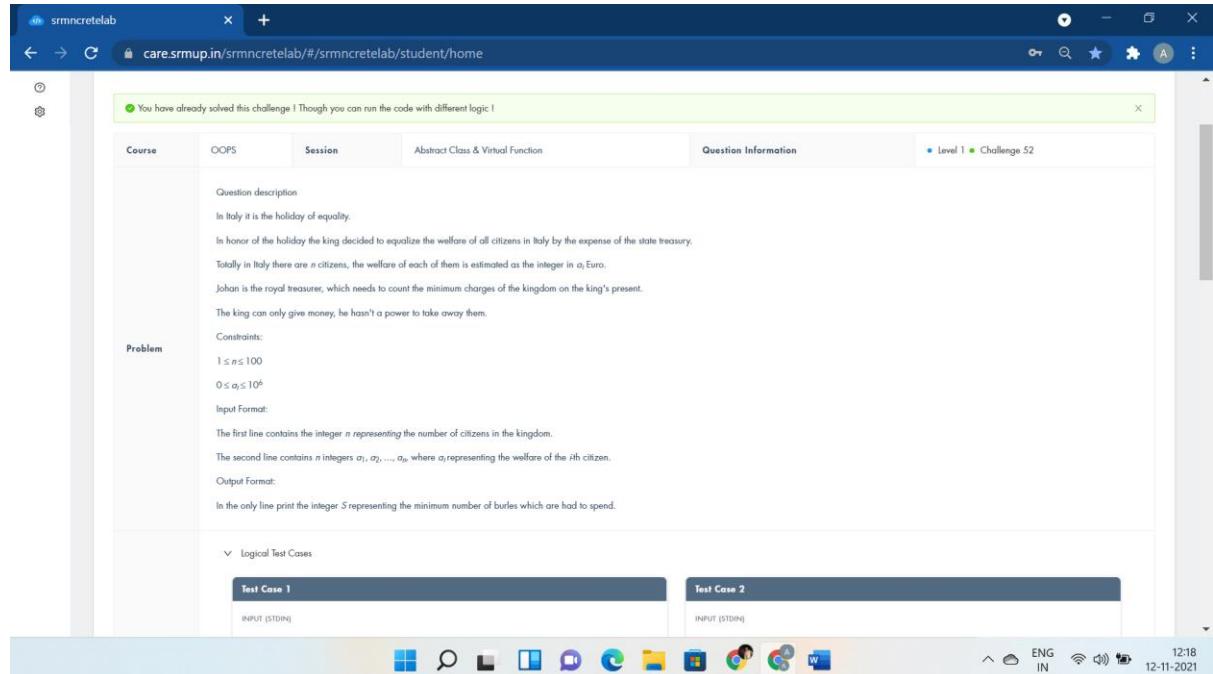
    cout<<"Student Name is: "<<name<<endl<<"Student Roll no is: "<<roll;

}
```

```

int main()
{
    District obj;
    School* ptr;
    ptr = &obj;
    ptr -> getdata();
    ptr -> display();
    return 0;
}

```



```

#include <bits/stdc++.h>

using namespace std;

int a,b,c,d,i;

class Holiday{

public:virtual void Expenses()=0;

};

class Citizen:public Holiday{

public:

void Expenses(){

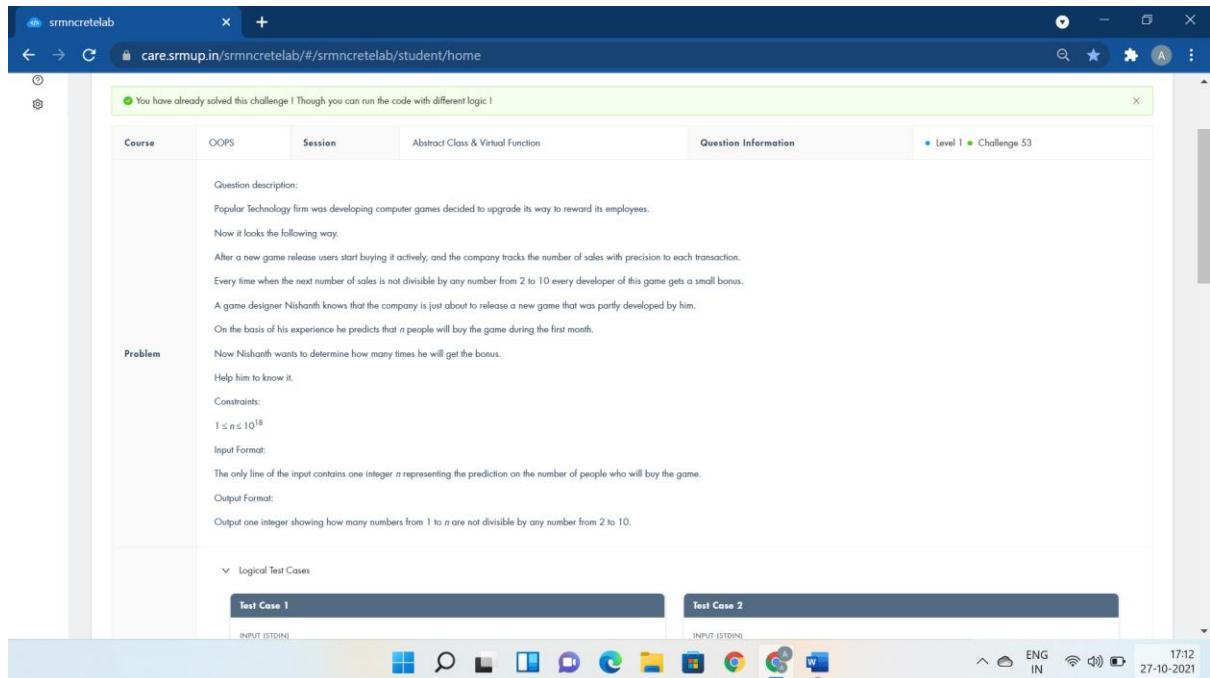

```

```

cin>>c;
for (i=0; i<c; i++){
    cin>>a;
    if (d<a) d=a;
    b=b+a;
}
cout<<d*c-b;
};

int main (){
Citizen obj;
obj.Expenses();
return 0;
}

```



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

```
using namespace std;

class Employees{

public:virtual void BuyingGame()=0;
};

class Reward:public Employees{

public:
    int n;

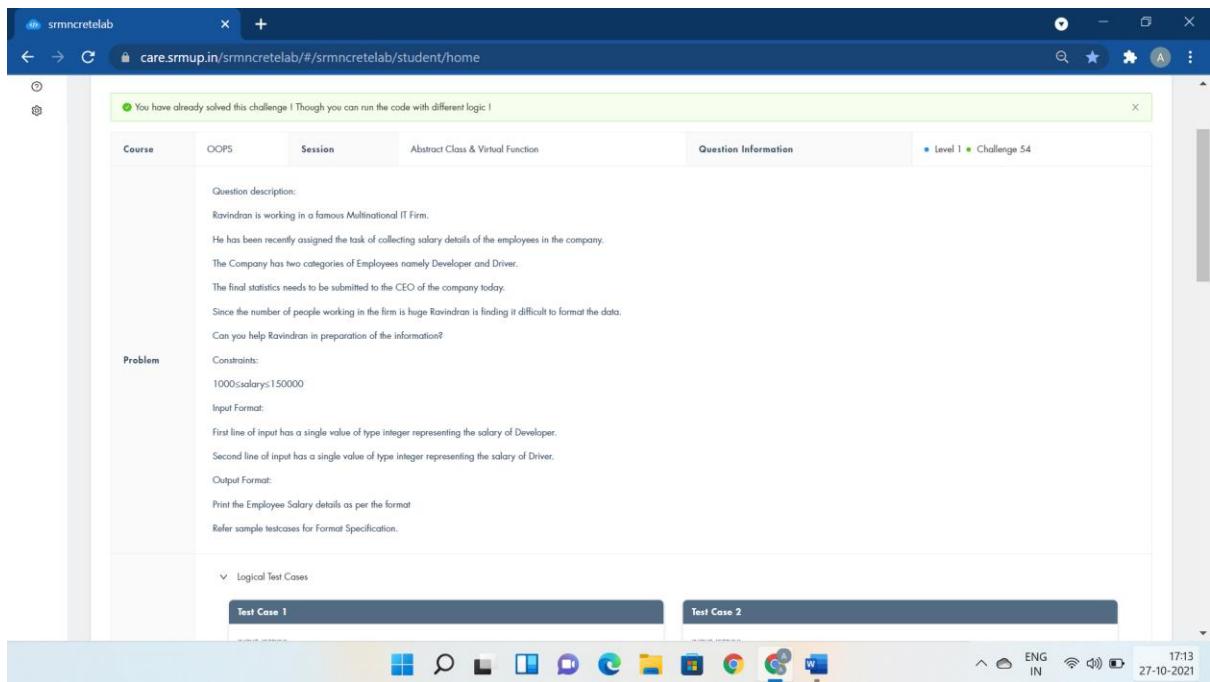
void BuyingGame(){

    cin>>n;
    cout<<n - n / 2 - n / 3 - n / 5 - n / 7
        + n / 6 + n / 10 + n / 14 + n / 15 + n / 21 + n / 35
        - n / 30 - n / 42 - n / 70 - n / 105 + n / 210;

}

};

int main()
{
    Reward obj;
    obj.BuyingGame();
    return 0;
}
```



```
#include <iostream>

using namespace std;

class Employee{
public:
    int s1,s2;
};

class Developer : public Employee{
public:
    void getSalary(){
        cin>>s1;
        cout<<"Salary of Developer:"<<s1<<endl;
    }
};

class Driver : public Employee{
public:
    void getSalary(){
        cin>>s2;
        cout<<"Salary of Driver:"<<s2<<endl;
    }
};
```

```

};

int main()
{
    Developer d1;

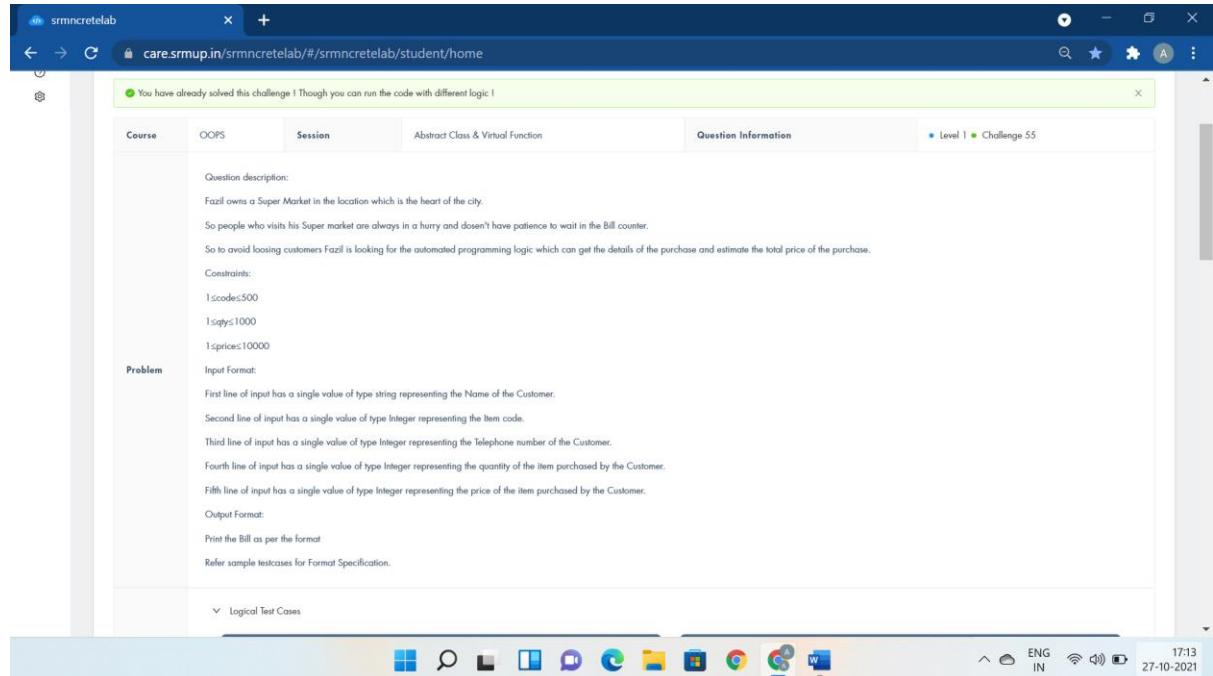
    Driver d2;

    d1.getSalary();

    d2.getSalary();

    return 0;
}

```



```

#include <iostream>

using namespace std;

class consumer{

public:

    string name;

    virtual void getdata()=0;

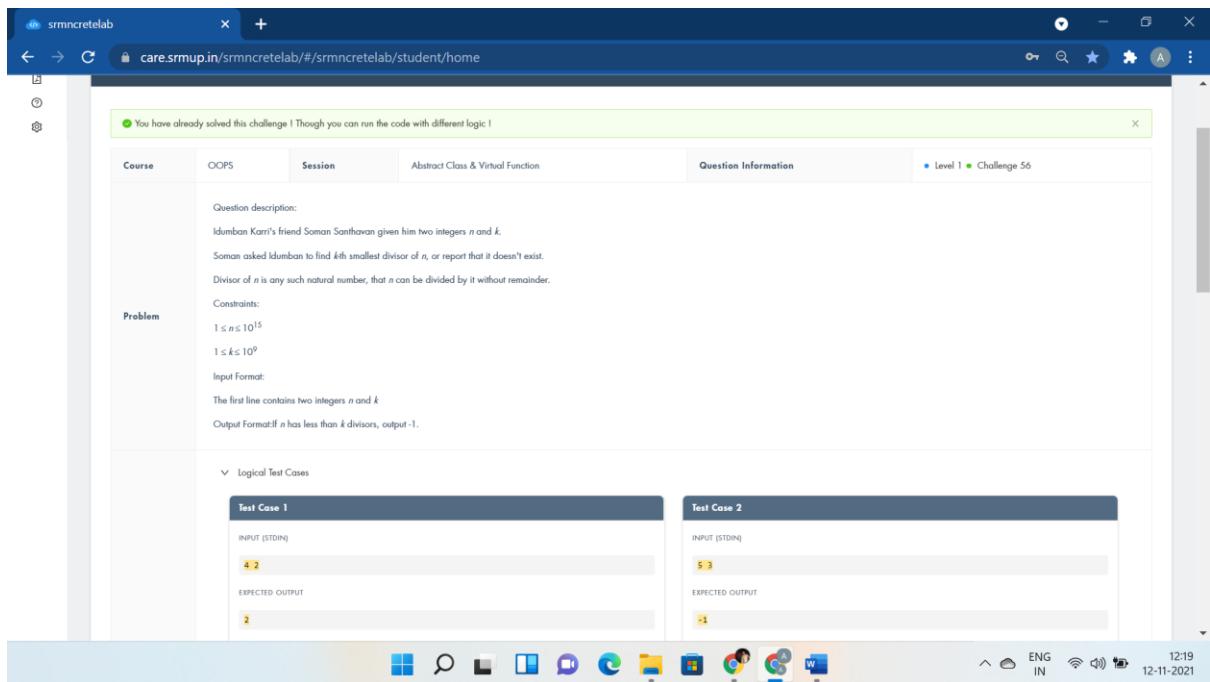
    virtual void display()=0;

};

class transaction: public consumer{

```

```
public:  
int code;  
long tel;  
int quan,price;  
void getdata(){  
    cin>>name>>code;  
    cin>>tel;  
    cin>>quan;  
    cin>>price;  
}  
void display(){  
    cout<<"Name : "<<name<<endl<<"Code : "<<code<<endl<<"Telephone : "<<tel<<endl;  
    cout<<"Quantity : "<<quan<<endl<<"Price : "<<price<<endl<<"Total Price : "  
    "<<quan*price<<endl;  
}  
};  
int main()  
{  
    consumer* o1;  
    transaction o2;  
    o1=&o2;  
    o1->getdata();  
    o1->display();  
    return 0;  
}
```



```
#include<iostream>

using namespace std;

class Problem {

public:virtual void Divisor()=0;
};

class Calculation:public Problem{

public:

    int n,k,i;

    void Divisor(){

        cin>>n>>k;
    }

    int Display()

    {

        int count;

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

        {

            if(n%i==0)

            {

                count++;
            }
        }
    }
}
```

```

        if(count==k){

            cout<<i;

            return 1;

        }

    }

    cout<<-1;

    return 1;

}

};

int main()

{

    Calculation obj;

    obj.Divisor();

    obj.Display();

    return 0;

}

```

You have already solved this challenge! Though you can run the code with different logic!

Course	OOPS	Session	Abstract Class & Virtual Function	Question Information	Level 1	Challenge 57

**Question description:**  
Young Varun has a birthday today! He got kit of  $n$  cubes as a birthday present from his parents. Every cube has a number  $a_i$  which is written on it.  
Varun put all the cubes in a row and went to unpack other presents.  
In this time, Varun's elder brother, Saran reordered the cubes using the following rule. Suppose the cubes are numbered from 1 to  $n$  in their order.  
Saran performs several steps, on step  $i$  he reverses the segment of cubes from  $i$ th to  $(n-i+1)$ th. He does this while  $i \leq n-i+1$ .  
After performing the operations Saran went away, being very proud of himself.  
When Varun returned to his cubes, he understood that their order was changed.

**Help Varun as fast as you can and save the holiday — restore the initial order of the cubes using information of their current location.**

**Problem**

**Constraints:**  
 $1 \leq n \leq 2 \cdot 10^5$   
 $-10^9 \leq a_i \leq 10^9$

**Input Format:**  
The first line contains single integer  $n$  representing the number of cubes.  
The second line contains  $n$  integers  $a_1, a_2, \dots, a_n$ , where  $a_i$  is the number written on the  $i$ th cube after Saran has changed their order.

**Output Format:**  
Print  $n$  integers, separated by spaces — the numbers written on the cubes in their initial order.

Logical Test Cases

Test Case 1    Test Case 2

17:14  
27-10-2021

#include <iostream>

```
using namespace std;

class Gift {

public:virtual void Cubes()=0;

};

class Birthday:public Gift{

public:

int a[10],n;

void Cubes(){

cin>>n;

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

cin>>a[i];

for(int i=0;i<n/2;i+=2)

/*int temp=a[i];

a[i]=a[n-i-1];

a[n-i-1]=temp;*/

swap(a[i],a[n-i-1]);

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

cout<<a[i]<<" ";

}

};

int main()

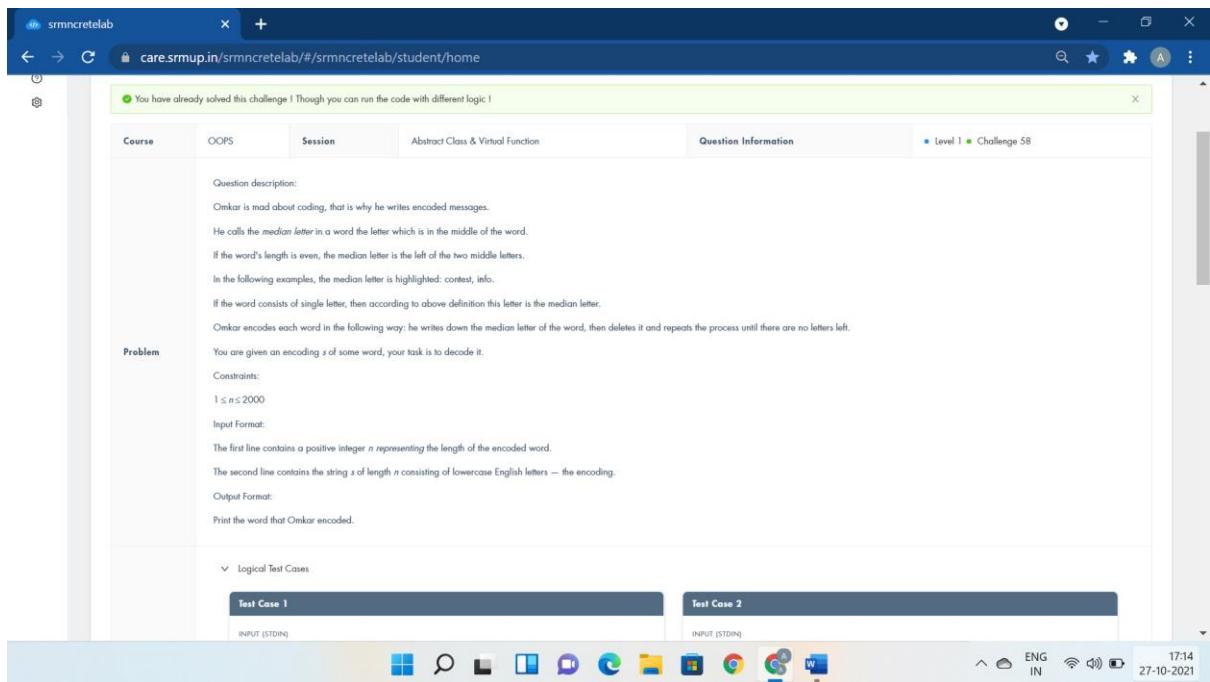
{

Birthday obj;

obj.Cubes();

return 0;

}
```



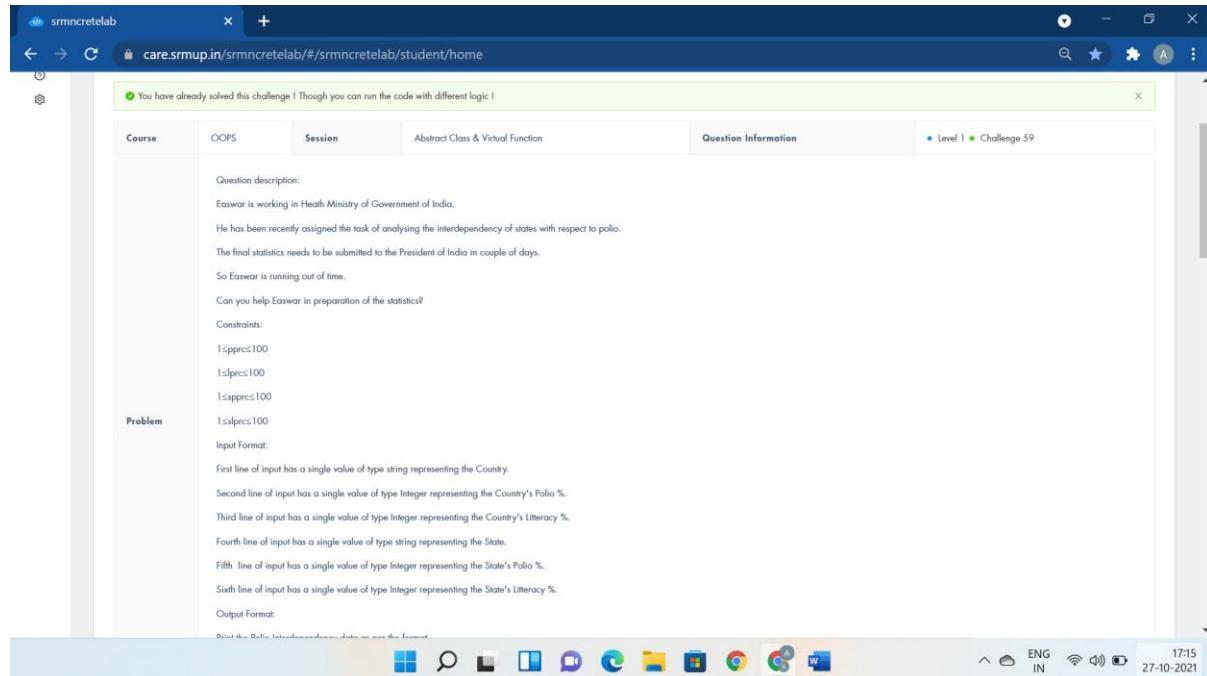
```
#include <iostream>
#include<string>
using namespace std;
class Decode{
public:virtual void Convert()=0;
};
class Word:public Decode{
public:
    string s1,s2;
    int n;
    void Convert(){
        cin>>n>>s1;
        for(int i=0;i<n;i++){
            if((n-i)%2==1)
                s2=s2+s1[i];
            else
                s2=s1[i]+s2;
        }
        cout<<s2;
    }
};
```

```

    }
};

int main()
{
    Word obj;
    obj.Convert();
}

```



```

#include <iostream>

using namespace std;

class country
{
public:
    virtual void getdata() = 0;
    virtual void display() = 0;
};

class state:public country
{

```

```

public:
char a[20];
int b,c;
char d[20];
int e,f;
void getdata(){
    cin>>a>>b>>c>>d>>e>>f;
}
void display()
{
    cout<<"Country:"<<a<<endl<<"Country's Polio %:"<<b<<endl;
    cout<<"Country Literacy %:"<<c<<endl<<"Interdependency Rate:"<<(float)b/c<<endl;
    cout<<"State Name:"<<d<<endl<<% of Polio of State:"<<e<<endl;
    cout<<% of Literacy of State:"<<f<<endl<<"Interdependency Rate:"<<(float)e/f;
}
};

int main() {
if(0)
    cout<<"country::getdata();";
    country *o1;
    state o2;
    o1=&o2;
    o1->getdata();
    o2.display();
return 0;
}

```

You have already solved this challenge! Though you can run the code with different logic!

**Course**   **OOPS**   **Session**   **Abstract Class & Virtual Function**   **Question Information**   **Level 1**   **Challenge 60**

**Problem**

Question description:  
Janani loves listening to music via her smartphone.  
But the smartphone doesn't have much memory, so Janani listens to her favorite songs in a well-known social network InTalk.  
Unfortunately, internet is not that fast in the city of Manali and the song takes a lot of time to download.  
But Janani is quite impatient. The song's duration is 7 seconds. Janani downloads the first  $S$  seconds of the song and plays it.  
When the playback reaches the point that has not yet been downloaded, Janani immediately plays the song from the start [the loaded part of the song stays in her phone, and the download is continued from the same place], and it happens until the song is downloaded completely and Janani listens to it to the end.  
For  $q$  seconds of real time the Internet allows you to download  $q-1$  seconds of the track.  
Tell Janani, for how many times he will start the song, including the very first start.

Constraints:  
 $2 \leq q \leq 10^4$   
 $1 \leq S \leq 10^5$

Input Format:  
The single line contains three integers  $T, S, q$ .

Output Format:  
Print a single integer representing the number of times the song will be restarted.

Logical Test Cases

Test Case 1   Test Case 2

ENG IN 12-11-2021 12:20

```
#include<iostream>

using namespace std;

class Smartphone{

public:virtual void Listening()=0;

};

class LoveForMusic:public Smartphone{

public:

int T,S,q,c=0;

void Listening(){

cin>>T>>S>>q;

while(S<T){

    c++;

    S*=q;

}

cout<<c;

}

};

int main()

{
```

```
LoveForMusic obj;  
obj.Listening();  
return 0;  
}
```

You have already solved this challenge! Though you can run the code with different logic!

Course OOPS Session Templates Question Information Level 1 Challenge 61

Question description:  
Veeran the who was described as Son of Forest by his people lives in the middle of the forest.  
He has two girlfriends: Elavenil and Koyal, who live at the different ends of the forest, each one is unaware of the other one's existence.  
When Veeran has some free time, he goes to one of his girlfriends. He descends into the forest at some time, waits the first parosal to come and rides on it to the end of the forest to the corresponding girl.  
However, the parosal run with different frequencies: a parosal goes to Elavenil's direction every  $a$  minutes, but a parosal goes to Koyal's direction every  $b$  minutes.  
If two parosal approach at the same time, Veeran goes toward the direction with the lower frequency of going parosal, that is, to the girl, to whose directions the parosal go less frequently.  
We know that the parosal begin to go simultaneously before Veeran appears.

Problem  
That is the parosal schedule is such that there exists a moment of time when the two parosal arrive simultaneously.  
Help Veeran count to which girlfriend he will go more often.

Constraints:  
 $1 \leq a, b \leq 10^6$   
 $a \neq b$

Input Format:  
The first line contains two integers  $a$  and  $b$

Output Format:  
Print "Elavenil" if Veeran will go to Elavenil more frequently, "Koyal" if he will go to Koyal more frequently, or "Equal" if he will go to both girlfriends with the same frequency.

Logical Test Cases  
Test Case 1 Test Case 2

17:16 27-10-2021

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

using namespace std;

template <class Forest>

Forest Visit(Forest a,Forest b){

    if(a>b)

        cout<<"Koyal\n";

    else

        cout<<"Elavenil\n";

    return 1;
}

int main()

{

    int a,b;

    cin>>a>>b;

    if(a%(a-b)==0 && b%(a-b)==0)

        cout<<"Equal\n";

    else

        Visit(a,b);

    return 0;
}
```

You have already solved this challenge ! Though you can run the code with different logic !

Question Information: Level 1 • Challenge 62

Question description:

A progress bar is an element of graphical interface that displays the progress of a process for this very moment before it is completed.

Let's take a look at the following form of such a bar.

A bar is represented as  $n$  squares, located in line.

To add clarity, let's number them with positive integers from 1 to  $n$  from the left to the right.

Each square has saturation ( $a_i$  for the  $i$ -th square), which is measured by an integer from 0 to  $k$ .

When the bar for some  $i$  ( $1 \leq i \leq n$ ) is displayed, squares 1, 2, ...,  $i-1$  has the saturation  $k$ , squares  $i+1, i+2, \dots, n$  has the saturation 0, and the saturation of the square  $i$  can have any value from 0 to  $k$ .

So some first squares of the progress bar always have the saturation  $k$ . Some last squares always have the saturation 0. And there is no more than one square that has the saturation different from 0 and  $k$ .

The degree of the process's completion is measured in percents.

Let the process be  $t\%$  completed. Then the following inequation is fulfilled:

$$\frac{\sum_{i=1}^n a_i}{nk} \leq \frac{t}{100} < \frac{(\sum_{i=1}^n a_i) + 1}{nk}$$

For the given  $n, k, t$  determine the measures of saturation for all the squares  $a_i$  of the progress bar.

Constraints:

$1 \leq n, k \leq 100$

$0 \leq t \leq 100$

Input Format:

Single line of input has 3 space-separated integers  $n, k, t$

Output Format:

12-11-2021

```
#include <iostream>

using namespace std;

template <class Interface>

Interface Bar(Interface n,Interface k,Interface t){

    t = t*k*n/100.0;

    while(n--){

        cout<<min(t,k)<<" ";

        t-=min(t,k);

    }

    return 1;
}

int main()

{

    int n,k,t;

    cin>>n>>k>>t;

    Bar(n,k,t);

    return 0;
}
```

You have already solved this challenge ! Though you can run the code with different logic !

Course   OOPS   Session   Templates   Question Information   Level 1   Challenge 63

Question Description:  
Hameed and Zaheer were involved in the discussion on cricket.  
Each of them says their favourite cricketers names respectively.  
Now both Hameed and Zaheer would like to interchange the names of their favourite cricketer.  
But both of them don't have any idea of how to interchange the names.  
Can you help them to complete the interchanging process?  
Input Format:  
Only line of input has two values of type string representing the name of their favourite players said by Hameed and Zaheer respectively.  
Output Format:  
Print the names of the players after interchanging.

Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN) Jadeja Siraj	INPUT (STDIN) Virat Rohit
EXPECTED OUTPUT Siraj Jadeja	EXPECTED OUTPUT Rohit Virat

```
#include <iostream>

using namespace std;

template <class T>

void InterchangeFavPlayers(T &player1,T &player2){

    cout<<player2<<" "<<player1;

}

int main()

{

    string player1,player2;

    cin>>player1>>player2;

    InterchangeFavPlayers(player1,player2);

    return 0;

}
```

You have already solved this challenge ! Though you can run the code with different logic !

Question description:  
As a result of the recent Taliban Attack on Algan Magical Clock the Central attraction of the city Kabal is damaged.  
The bullets of the gun made several holes in the clock, that's why the residents are concerned about the repair.  
The Magical clock can be represented as an infinite Cartesian plane, where the origin corresponds to the clock center. The clock was painted two colors black and white.  
This coloring naturally extends to infinity.  
The bullet can be taken to be points on the plane.  
Your task is to find the color of the area, damaged by the given ball.

Problem  
All the points located on the border of one of the areas have to be considered painted black.

Constraints:  
Each of the numbers  $x$  and  $y$  has an absolute value that does not exceed 1000.

Input Format:  
The first and single line contains two integers  $x$  and  $y$  representing the coordinates of the hole made in the clock by the ball.

Output Format:  
In a single line print the color.  
All the points between which and the origin of coordinates the distance is integral-value are painted black.

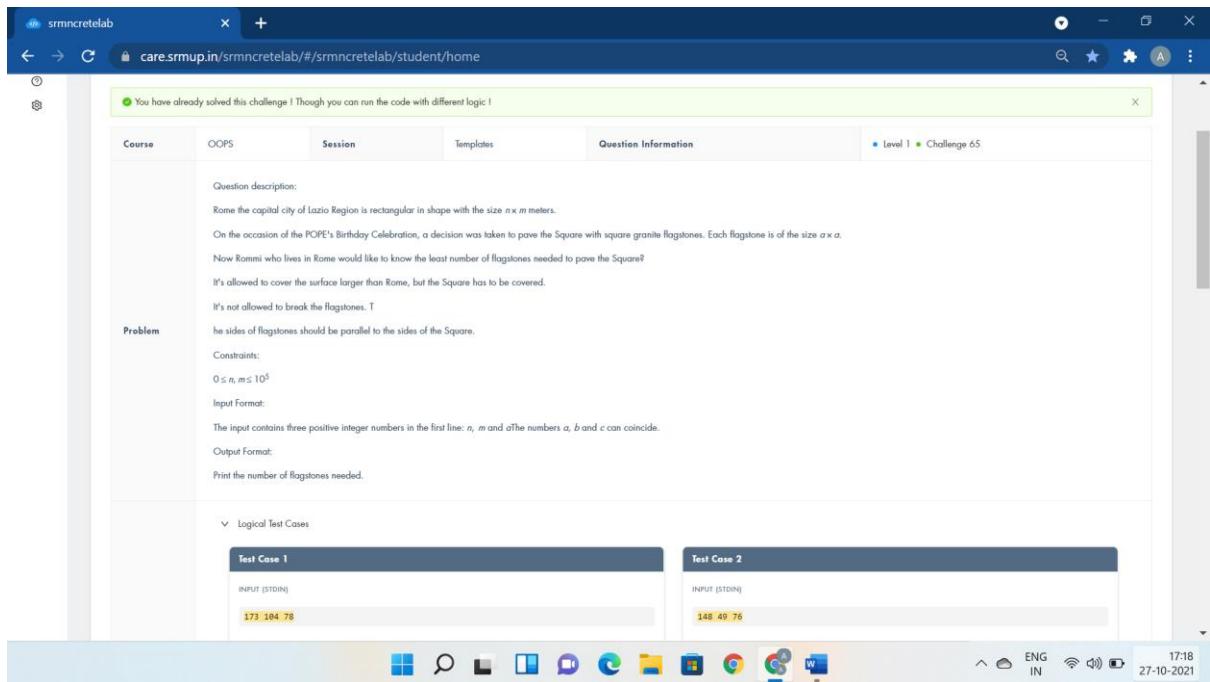
Logical Test Cases  
Test Case 1    Test Case 2

```
#include <iostream>
#include<cmath>
using namespace std;
template <class Hole>
Hole MagicClock(Hole x,Hole y){
    int c;
    c=sqrt(x*x+y*y);
    if(c*c==x*x+y*y){
        cout<<"black\n";
        return 0;
    }
    if(x*y<0)
        c++;
    if(c%2==0)
        cout<<"black";
    else cout<<"white";
    return 1;
}
using namespace std;
int main()
{
    int x,y;
```

```

    cin>>x>>y;
    MagicClock(x,y);
    return 0;
}

```



```

#include <iostream>

using namespace std;

template <class Celebration>

Celebration Rome(Celebration a,Celebration b,Celebration c){

cout<<((b+c-1)/c)*((a+c-1)/c);

return 1;

}

int main()

{

    int a,b,c;

    cin>>a>>b>>c;

    Rome(a,b,c);

    return 0;

}

```

You have already solved this challenge! Though you can run the code with different logic!

Question description:  
Jenni had a square painted on a piece of paper, the square's side equals  $n$  meters.  
Rohit draws crosses on the square's perimeter.  
Rohit paints the first cross in the lower left corner of the square.  
Then Rohit moves along the square's perimeter in the clockwise direction [first upwards, then to the right, then downwards, then to the left and so on].  
Every time he walks  $(n + 1)$  meters, he draws a cross [see picture for clarifications].

Constraints:  
 $1 \leq T \leq 50$   
 $1 \leq n_j \leq 10^9$

Input Format:  
The first line contains integer  $t$  representing the number of testcases.  
The second line contains  $t$  space-separated integers  $n_j$  representing the sides of the square for each test sample.

Output Format:  
For each test sample print on a single line the answer to it, that is, the number of crosses Rohit will draw as he will move along the square of the corresponding size.  
Print the answers to the samples in the order in which the samples are given in the input.

Logical Test Cases

Test Case 1  
INPUT (STDIN)  
Test Case 2  
INPUT (STDIN)

17:18  
27-10-2021

```
#include <iostream>

using namespace std;

template <class Paper>
Paper Square(Paper T){

    if(T%2==0)

        return 4*T+1;

    else if(T%4==1)

        return 2*T+1;

    else

        return T+1;

}

int main()

{

    int T,n;

    cin>>T;

    while(T--){

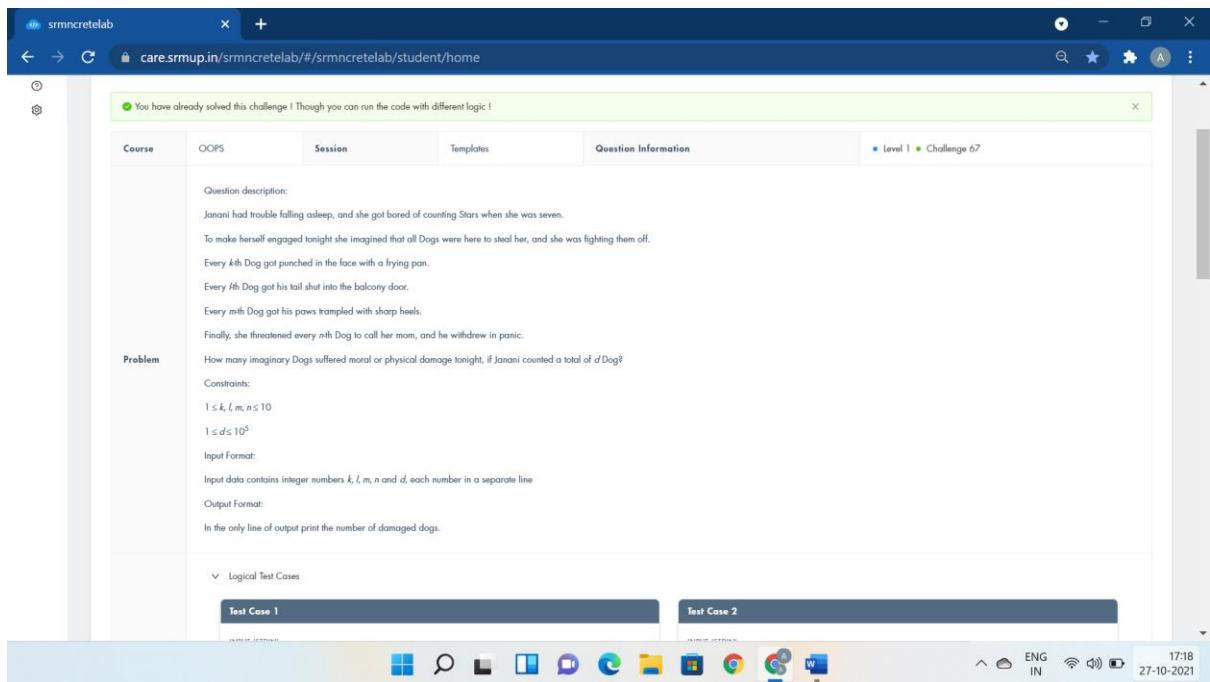
        cin>>n;

        cout<<Square(n)<<endl;

    }

    return 0;

}
```



```
#include <iostream>

using namespace std;

template <class LackofSleep>

LackofSleep Counting(LackofSleep k,LackofSleep l,LackofSleep m,LackofSleep n,LackofSleep d)

{

    int c=0;

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

        if(i%k==0 || i%l==0 || i%m==0 || i%n==0)

            c++;

    }

    return c-1;

}

int main()

{

    int k,l,m,n,d;

    cin>>k>>l>>m>>n>>d;

    cout<<Counting(k,l,m,n,d);

    return 0;

}
```

The screenshot shows a web browser window with the URL [care.srmup.in/srmncretelab/#/srmncretelab/student/home](http://care.srmup.in/srmncretelab/#/srmncretelab/student/home). The page title is "CHALLENGE INFORMATION". The navigation bar includes tabs for "Course", "OOOPS" (selected), "Session", "Templates", and "Question Information". A status bar at the bottom right shows "17:19" and "27-10-2021". A message bar at the top says "You have already solved this challenge! Though you can run the code with different logic!". The main content area contains a question description about two flies on a cube and their visibility, followed by input and output formats, and logical test cases for "Test Case 1" and "Test Case 2".

```
#include <iostream>

using namespace std;

template <class Universe>

Universe Planet (Universe x1,Universe y1,Universe z1,Universe x2,Universe y2,Universe z2){

    if(x1==x2 || y1 == y2 || z1==z2)
        cout<<"YES";
    else
        cout<<"NO";

    return 1;
}

int main()

{
    int x1,y1,z1,x2,y2,z2;
    cin>>x1>>y1>>z1>>x2>>y2>>z2;
    Planet(x1,y1,z1,x2,y2,z2);

    return 0;
}
```

You have already solved this challenge! Though you can run the code with different logic!

**Course**   **OOPS**   **Session**   **Templates**   **Question Information**   **Level 1**   **Challenge 69**

**Question description:**  
Walter has a ribbon, its length is  $n$ .  
He wants to cut the ribbon in a way that fulfills the following two conditions:

- After the cutting each ribbon piece should have length  $a$ ,  $b$  or  $c$ .
- After the cutting the number of ribbon pieces should be maximum.

Help Walter and find the number of ribbon pieces after the required cutting.

**Constraints:**  
 $1 \leq n, a, b, c \leq 2500$

**Input Format:**  
The first line contains four space-separated integers  $n$ ,  $a$ ,  $b$  and  $c$  representing the length of the original ribbon and the acceptable lengths of the ribbon pieces after the cutting, correspondingly.  
The numbers  $a$ ,  $b$  and  $c$  can coincide.

**Output Format:**  
In the only line of output print a single number representing the maximum possible number of ribbon pieces.

It is guaranteed that at least one correct ribbon cutting exists.

**Logical Test Cases**

Test Case 1	Test Case 2
INPUT (STDIN) 174 17 45 29	INPUT (STDIN) 378 28 13 79

12-11-2021 12:22

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

using namespace std;

template <class Ribbon>

Ribbon Pieces(Ribbon n,Ribbon a,Ribbon b,Ribbon c){

    int d=1,e,i,j;

    for(i=0;i<=4000;i++)

        for(j=0;j<=4000;j++) {

            e=n-a*i-b*j;

            if(e>=0&&e%c==0)

                d=max(d,i+j+e/c);

        }

    cout<<d;

    return 1;

}

int main(){

    int n,a,b,c;

    cin>>n>>a>>b>>c;

    Pieces(n,a,b,c);

}
```

You have already solved this challenge ! Though you can run the code with different logic !

**Question Description:**

Delhi was so hot nowadays and on one such hot day Priya and her friend Rohini decided to buy a Mangosteen. They chose the biggest and the ripest Mangosteen. After that the Mangosteen was weighed, and the scales showed  $w$  kilos. They reached home, dying of thirst, and decided to divide the Mangosteen, however they faced a hard problem.

Priya and Rohini are great fans of even numbers, that's why they want to divide the Mangosteen in such a way that each of the two parts weights even number of kilos, at the same time it is not obligatory that the parts are equal.

**Problem**

The girls are extremely tired and want to start their meal as soon as possible, that's why you should help them and find out, if they can divide the Mangosteen in the way they want.

For sure, each of them should get a part of positive weight.

**Constraints:**

$1 \leq w \leq 1000$

**Input Format:**

The only line of input line contains integer number  $w$  the weight of the Mangosteen bought by the girls.

**Logical Test Cases**

Test Case 1	Test Case 2
INPUT (STDIN) 461	INPUT (STDIN) 542
EXPECTED OUTPUT NO	EXPECTED OUTPUT YES

```
#include <iostream>

using namespace std;

template<class T>

T DivideMangosteen(T PurchasedWeight){

    if(PurchasedWeight%2==0)

        cout<<"YES";

    else

        cout<<"NO";

    return 1;

}

int main()

{

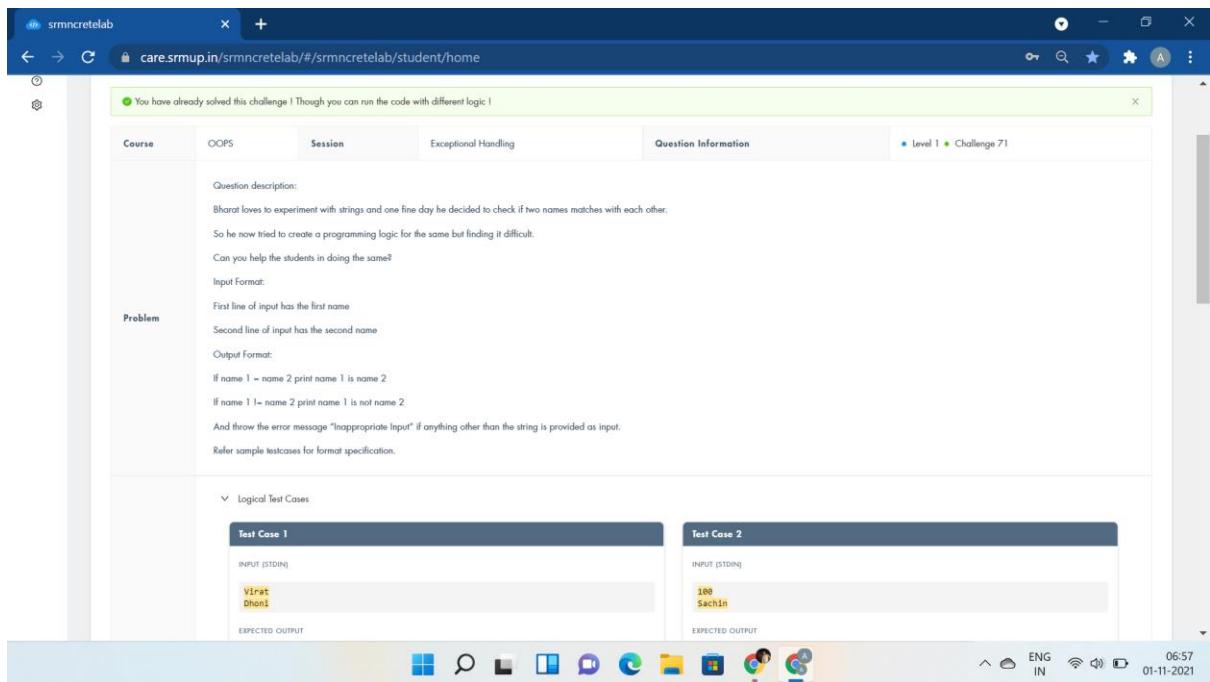
    int PurchasedWeight;

    cin>>PurchasedWeight;

    DivideMangosteen(PurchasedWeight);

    return 0;

}
```



```
#include <iostream>

using namespace std;

int main()

{

    string str1,str2;

    try{

        cin>>str1>>str2;

        int count, n=str1.size();

        if(cin){

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

                if((str1[i]>=48 && str1[i]<=57) || (str2[i]>=48&&str2[i]<=57) )

                    throw 0;

                if(str1[i]==str2[i])

                    count++;

            }

            if(count!=n)

                cout<<str1<<" is not "<<str2;

            else

                cout<<str1<<" is "<<str2;

        }

    }

    catch (int i){

        cout<<"Inappropriate Input";

    }

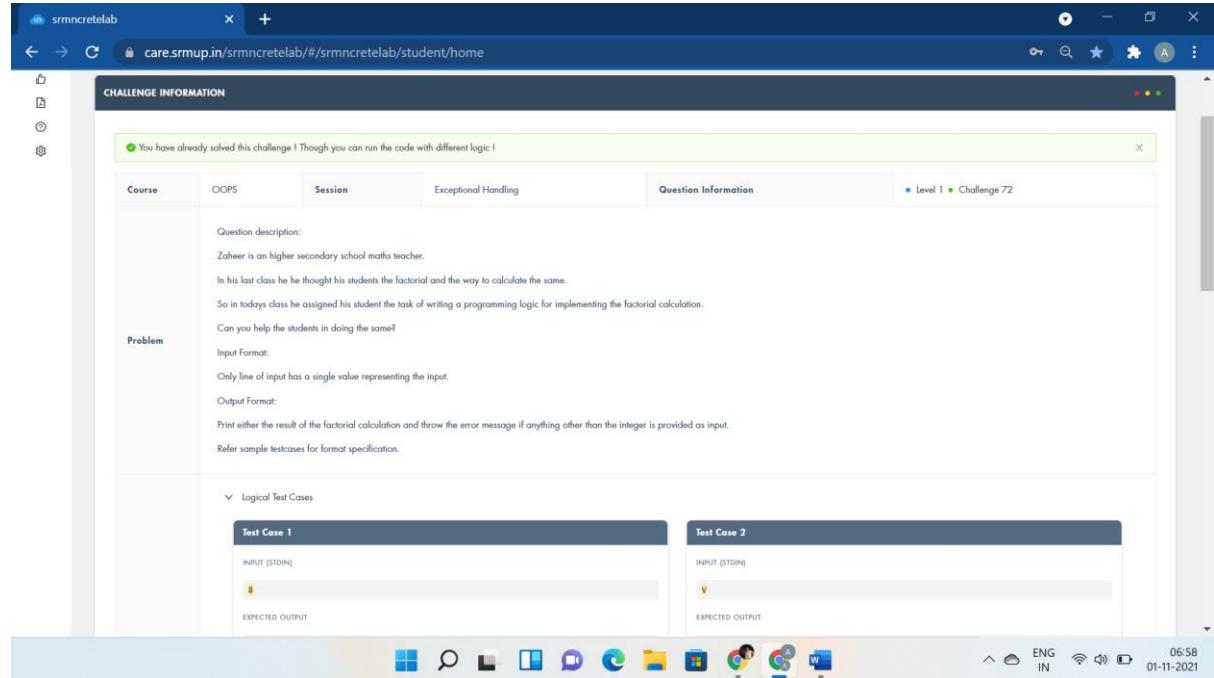
}
```

```

    }

    return 0;
}

```



```

#include <bits/stdc++.h>

#include <string.h>

using namespace std;

int main()

{

    int k;

    try{

        cin>>k;

        if(cin)

            cout<<fixed<<setprecision(0)<<tgamma(k+1);

        else

            throw "e";

    }

    catch (int i){

    }

    catch (const char *exp){

        cout<<"Input should be a Integer";
    }
}

```

```

    }

    return 0;
}

```

```

#include <iostream>

using namespace std;

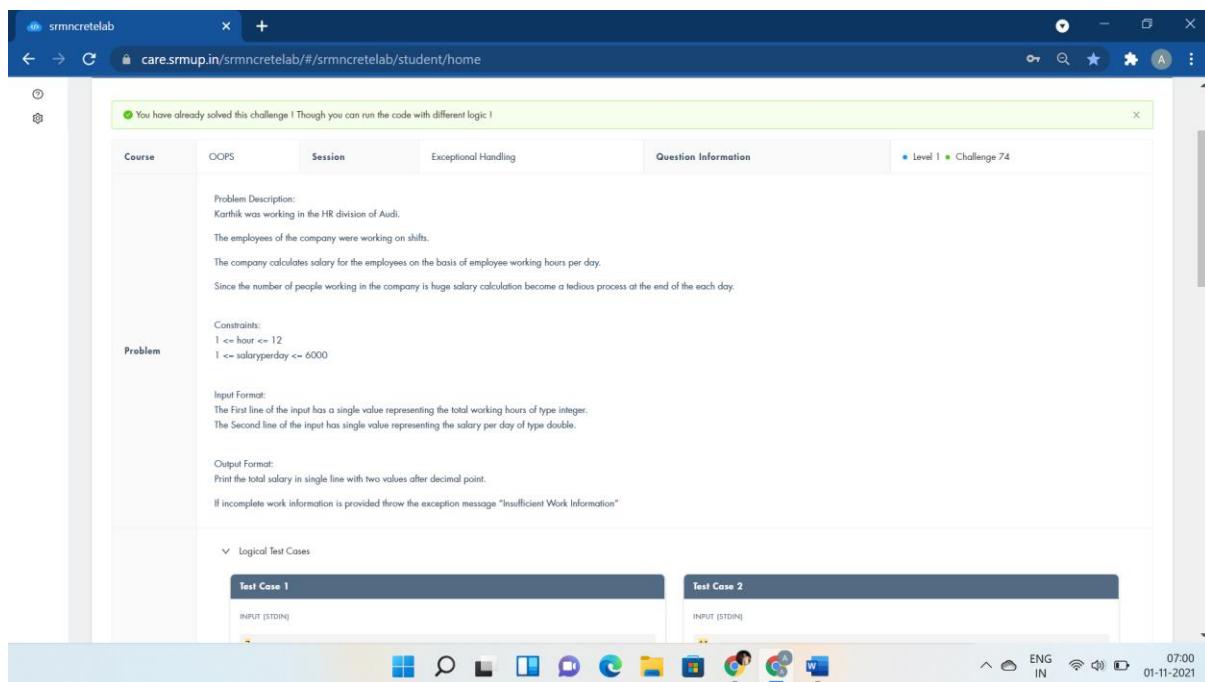
int main()
{
    int n,m;
    try{
        cin>>n;
        cin>>m;
        if(cin){
            cout<<n-1+(1+2*(n-1))*(m-1);
        }
        else
            throw 0;
    }
    catch(int griddimensions)
    {
        cout<<"Invalid Grid Dimensions";
    }
}

```

```

        return 0;
    }
}

```



```

#include<bits/stdc++.h>

using namespace std;

int main()
{
    float hour,salaryperday;

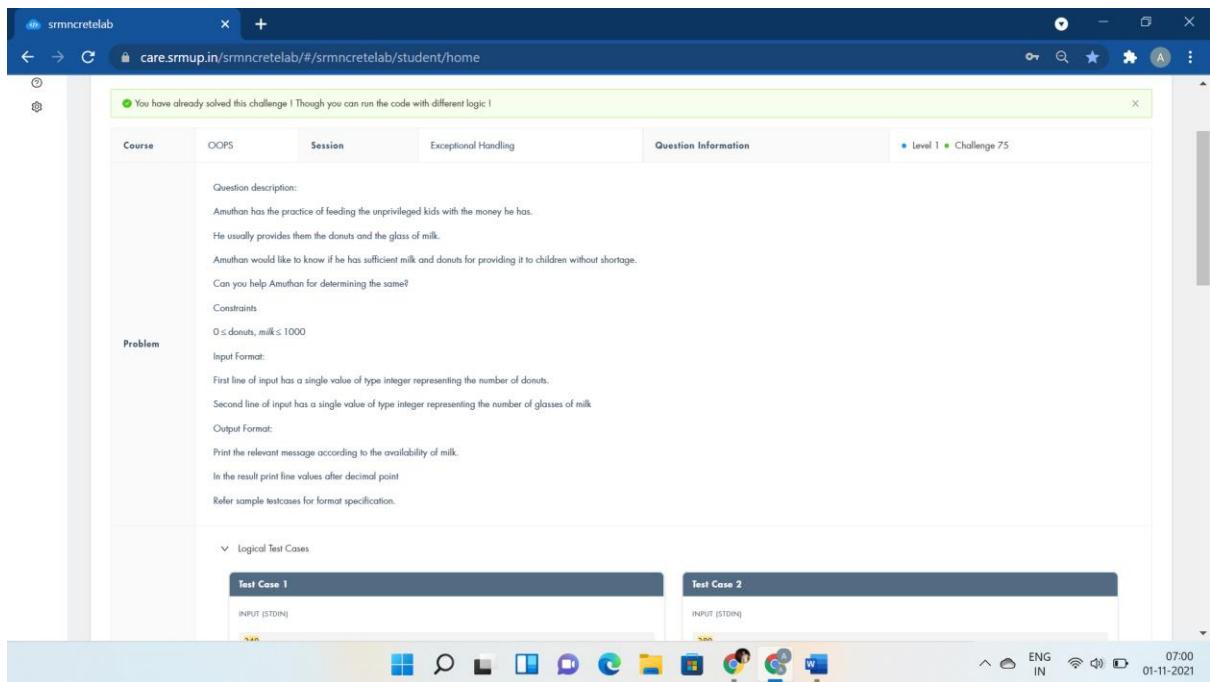
    try{
        cin>>hour;
        cin>>salaryperday;

        if(cin){
            cout<<fixed<<setprecision(2)<<hour*salaryperday;
        }
        else
            throw 0;
    }

    catch(int workstatus)
    {
        cout<<"Insufficient Work Information";
    }

    return 0;
}

```



```
#include <iostream>

using namespace std;

int main()

{

    int donuts,milk;

    try{

        cin>>donuts;

        cin>>milk;

        if(milk==0)

            throw donuts;

        else

            cout<<"You have "<<(float)donuts/milk<<" donuts for each glass of milk";

    }

    catch(int e){

        cout<<e<<" donuts and No Milk\nGo buy some milk";

    }

    return 0;

}
```

```
#include <iostream>
#include <math.h>
using namespace std;
int main()
{
    int a;
    try {
        cin>>a;
        if (a>0 && a<=100)
            cout<<"Valid Mark";
        else
            throw "e";
    }
    catch(const char* t){
        cout<<"Invalid Mark";
    }
}
```

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

using namespace std;

int main()

{

    int unitconsumed,costperunit;

    try{

        cin>>unitconsumed;

        cin>>costperunit;

        long int res;

        res=pow(unitconsumed,costperunit);

        if(cin){

            cout<<res;

        }

        else

        throw 0;

    }

    catch(int unit){

        cout<<"Incomplete Data";

    }

    return 0;

}
```

You have already solved this challenge! Though you can run the code with different logic!

**Course**   **OOPS**   **Session**   **Exceptional Handling**   **Question Information**   **Level 1 • Challenge 78**

**Problem Description:**  
Phoenix mall in the capital city of Washington and it is rectangular in shape when it is seen on the map with the size  $n \times m$  meters.  
On the occasion of the jubilee anniversary, a decision was taken to pave the Square with square marbles stones. Each stone is of the size  $a \times a$ .  
Can you find what is the least number of stones needed to pave the Square?  
It's allowed to cover the surface larger than the Mall Square, but the Square has to be covered.  
It's not allowed to break the stones.  
The sides of stones should be side by side[parallel] to the sides of the Square.

**Constraints:**  
**Problem**  
 $1 \leq n \leq 10^4$   
 $1 \leq m \leq 10^4$   
 $1 \leq a \leq 10^4$

**Input Format:**  
The only line of input contains three positive integer numbers  $n$ ,  $m$  and  $a$  separated by a space .

**Output Format:**  
Print the needed number of stones.  
If any of the input values  $n$  or  $m$  or  $a$  is missing in the input then raise the exception message as "Invalid Dimension"

```
#include <iostream>

using namespace std;

int main()
{
    int n,m,a;
    try{
        cin>>n>>m>>a;
        if(cin){
            cout<<((n+a-1)/a)*((m+a-1)/a);
        }
        else
            throw 0;
    }

    catch(int dimension){
        cout<<"Invalid Dimension";
    }

    return 0;
}
```

You have already solved this challenge! Though you can run the code with different logic!

Course   OOPS   Session   Exceptional Handling   Question Information   Level 1 • Challenge 79

Question description:  
Dino is an DTP operator in the Document formating firm.  
The document processor Dino uses accepts only characters which are alphabetic in nature.  
If the character is not alphabetic it is not accepted by the document processor.  
Can you help Dino in finding the nature of the characters in the document Dino is working with?

Problem  
Input Format:  
First line of input has a single value of type integer representing the number of testcases.  
Second line of input has the string to be checked in the document.  
Output Format:  
Print the relevant message for the input string.  
Refer sample testcases for format specification.

Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN) 1 v8 EXPECTED OUTPUT	INPUT (STDIN) 1 6e EXPECTED OUTPUT

07:02 01-11-2021

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

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

using namespace std;

int main(){

    int t,i,j;

    cin>>t;

    string str;

    f(j,0,t){

        f(i,0,2){

            try{

                cin>>str[i];

                if(isalpha(str[i])){

                    cout<<str[i]<<" is alphabetic"<<endl;

                }

                else

                    throw str[i];

            }

            catch (char f){

                cout<<f<<" is not alphabetic"<<endl;

            }

        }

    }

}
```

You have already solved this challenge! Though you can run the code with different logic!

**Course**   **OOPS**   **Session**   **Exceptional Handling**   **Question Information**   **Level 1**   **Challenge 80**

**Problem Description:**  
Selvan was playing with the object of random size for stress relief.  
Selvan knows that the Length, Width, and Height of the object.  
But he would like to know the surface area of the object he is playing with.  
Can you help him in finding it?

**Functional Description:**  
Surface area of the Object =  $2 \times (\text{width} \times \text{length} + \text{length} \times \text{height} + \text{height} \times \text{width})$

**Constraints:**  
 $1 \leq \text{length} \leq 20$   
 $1 \leq \text{width} \leq 20$   
 $1 \leq \text{height} \leq 20$

**Input Format:**  
First Line : Length of the object in Integer  
Second Line : Width of the object in Integer  
Third Line : Height of the object in Integer

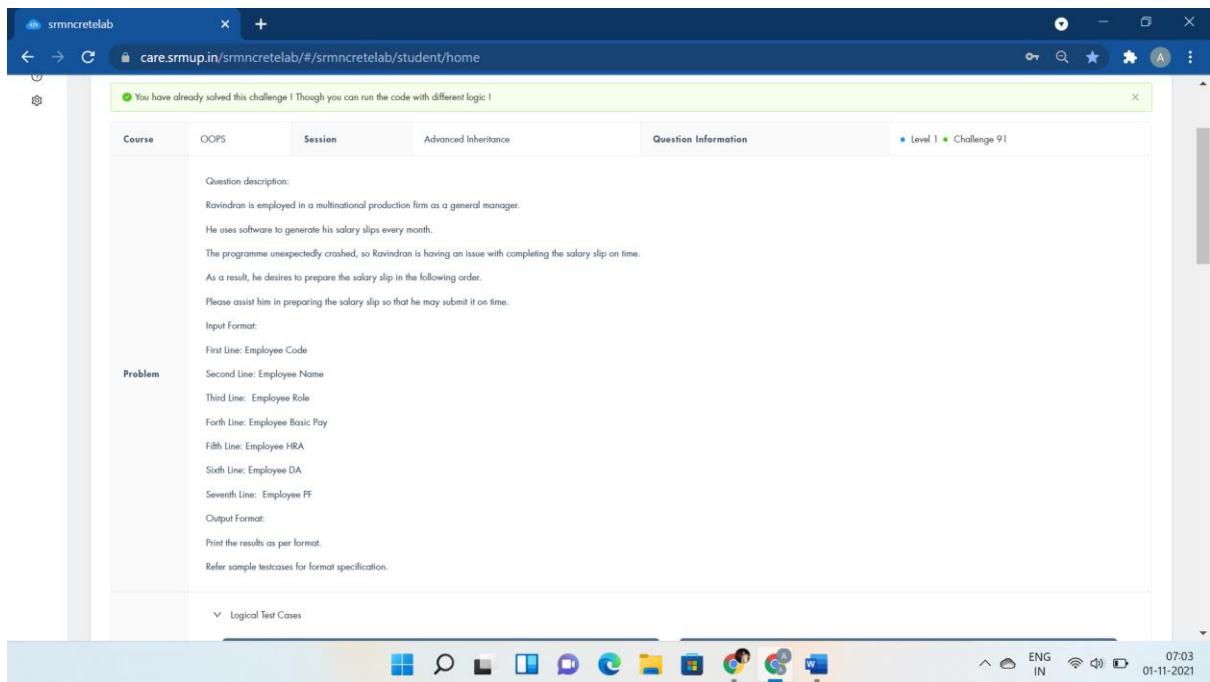
**Output Format:**  
Print a single integer value representing the surface area of the object selvan is playing with.  
If the information provided about the object is not sufficient for the calculation then throw an exception "Incomplete information about the object"

**Logical Test Cases**

```
#include <iostream>

using namespace std;

int main()
{
    int a,b,c;
    try{
        cin>>a>>b>>c;
        if(cin){
            cout<<2*(a*b+b*c+c*a);
        }
        else
            throw 0;
    }
    catch(int objectinfo){
        cout<<"Incomplete information about the object";
    }
    return 0;
}
```



```
#include <iostream>

using namespace std;

class Employee{

public:

};

class Salary : public Employee{

public:

    int code,basic,hra,da,pf,total;

    string name,position;

    void getEmpDetails(){

        cin>>code>>name>>position;

    }

    void getPayDetails(){

        cin>>basic>>hra>>da>>pf;

    }

    void calculate(){

        total=basic+hra+da-pf;

    }

    void display(){

        cout<<"Employee Number:"<<code<<endl;

        cout<<"Employee Name:"<<name<<endl;

        cout<<"Employee Role:"<<position<<endl;

        cout<<"Employee Net Pay:"<<total<<endl;

    }

}
```

```

    }

};

int main()
{
    Salary s;

    s.getEmpDetails();

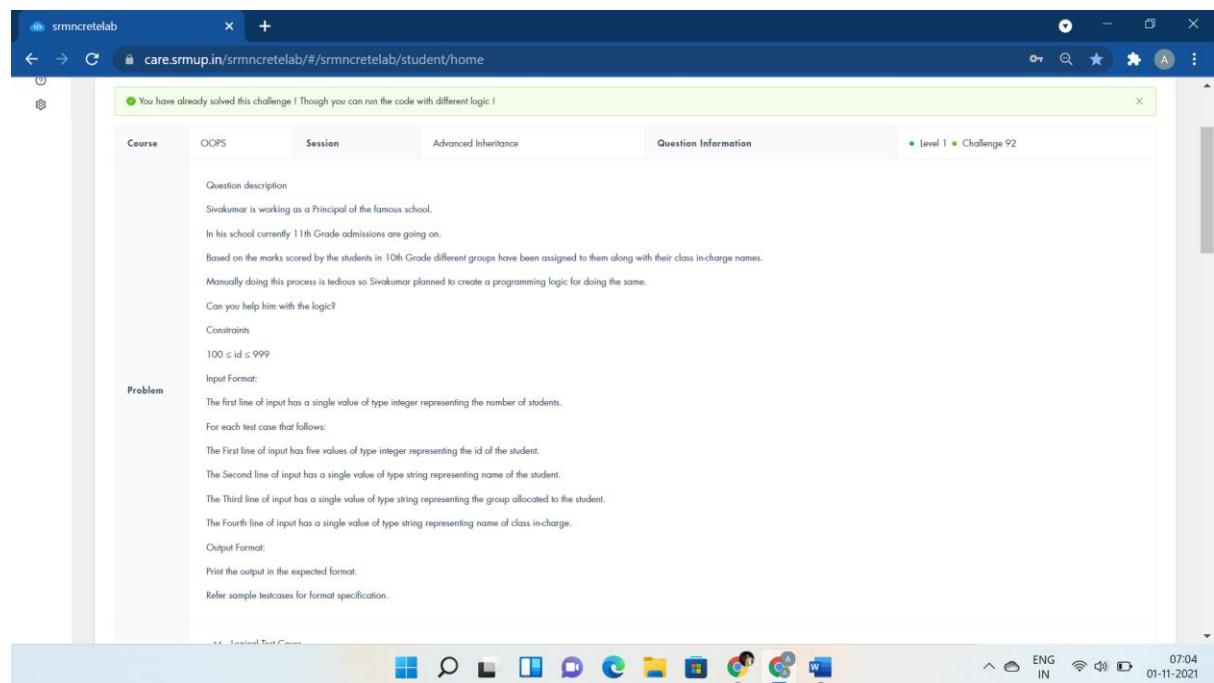
    s.getPayDetails();

    s.calculate();

    s.display();

    return 0;
}

```



```

#include <iostream>

using namespace std;

class Person{

};

class Teaching : public Person{

};

class Instructor : public Teaching{

public:

    int id;

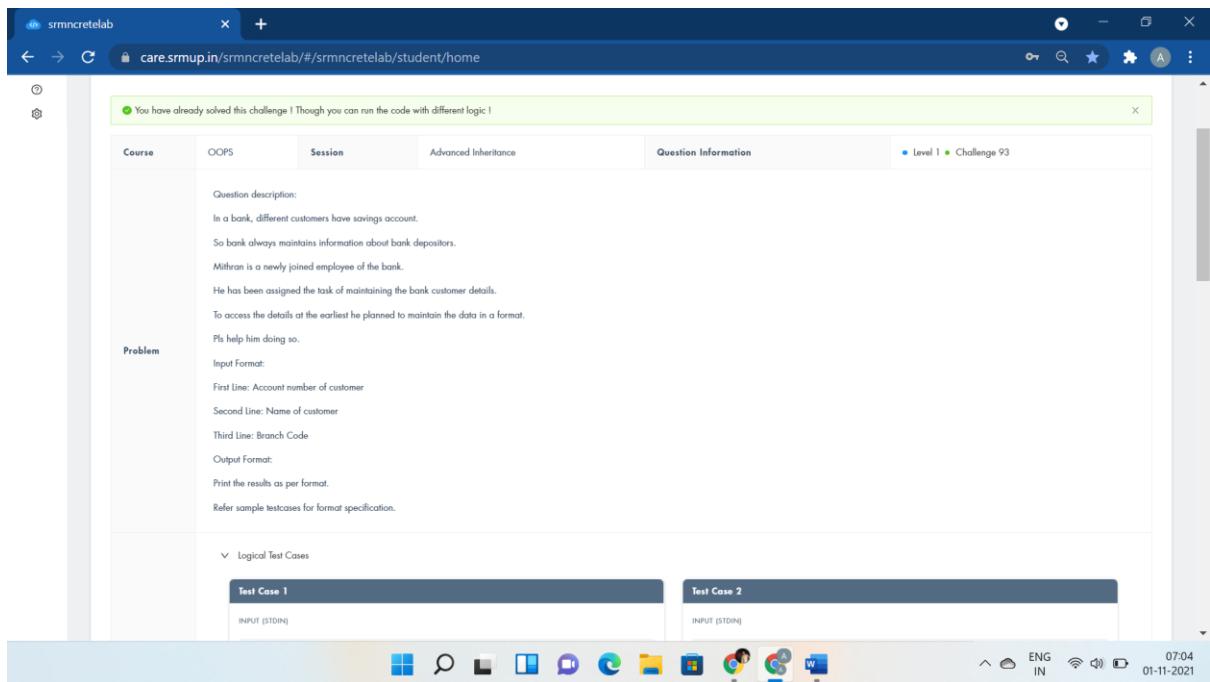
    string name,group,staff;

```

```
void accept_instructor_details(){
    cin>>id>>name>>group>>staff;
}

void display_instructor_details(){
    cout<<"Id:"<<id<<endl;
    cout<<"Name:"<<name<<endl;
    cout<<"Group:"<<group<<endl;
    cout<<"Staff:"<<staff<<endl;
}

int main()
{
    int n;
    cin>>n;
    Instructor inst[n];
    for(int i=0;i<n;i++){
        inst[i].accept_instructor_details();
        inst[i].display_instructor_details();
    }
    return 0;
    cout<<"Instructor *inst;";
}
```



```
#include <iostream>

using namespace std;

class acc{
public:
    int no;
    void getacc(){
        cin>>no;
    }
};

class branch:public acc{
public:
    string name;
    int code;
    void getbranch(){
        cin>>name>>code;
    }
    void display(){
        cout<<"Acc No:"<<no<<endl;
        cout<<"Name:"<<name<<endl;
        cout<<"Branch Code:"<<code<<endl;
    }
};

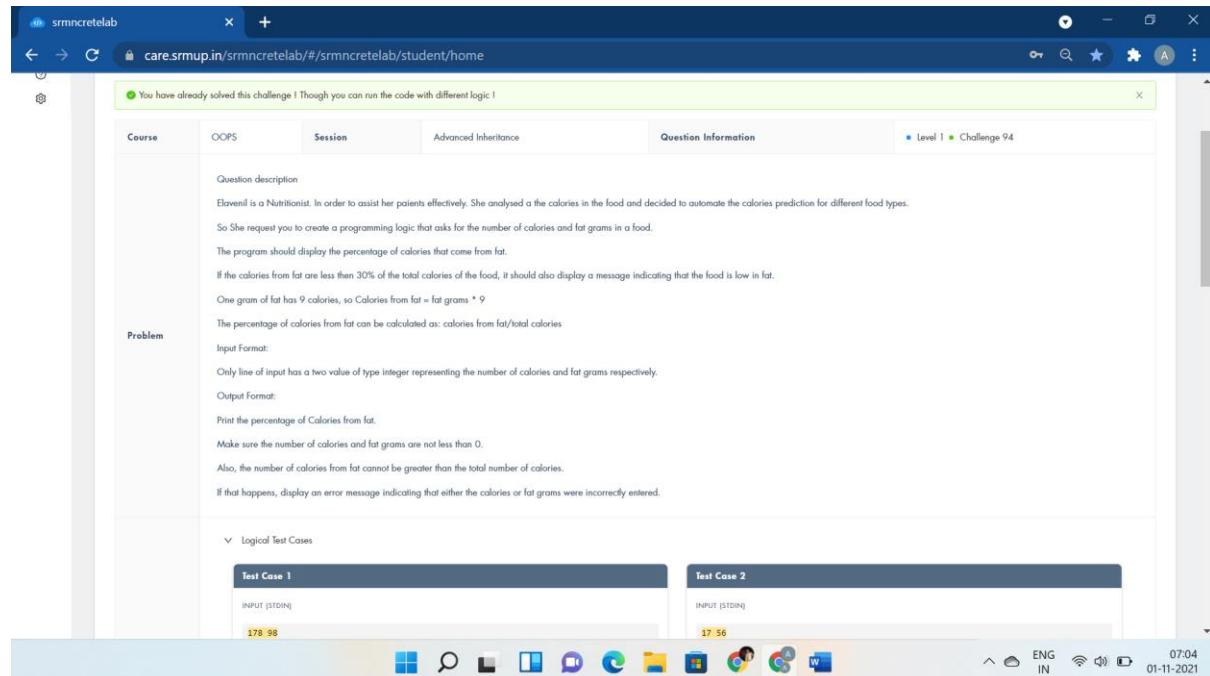
int main()
```

```

{
    branch b;
    b.getacc();
    b.getbranch();
    b.display();

    return 0;
}

```



```

#include <iostream>

using namespace std;

class Food{

};

class Nutritionist:public Food{

};

class Patient:public Nutritionist{

public:
    float cal,fat;

    void calorie(){
        cin>>cal>>fat;
    }

    void dplan(){
        if(cal<fat)
            cout<<"Fatgrams cannot be less than 0 or greater than calories"<<endl;
    }
}

```

```

cout<<"Calories from fat: "<<fat*9/cal*100<<"%";

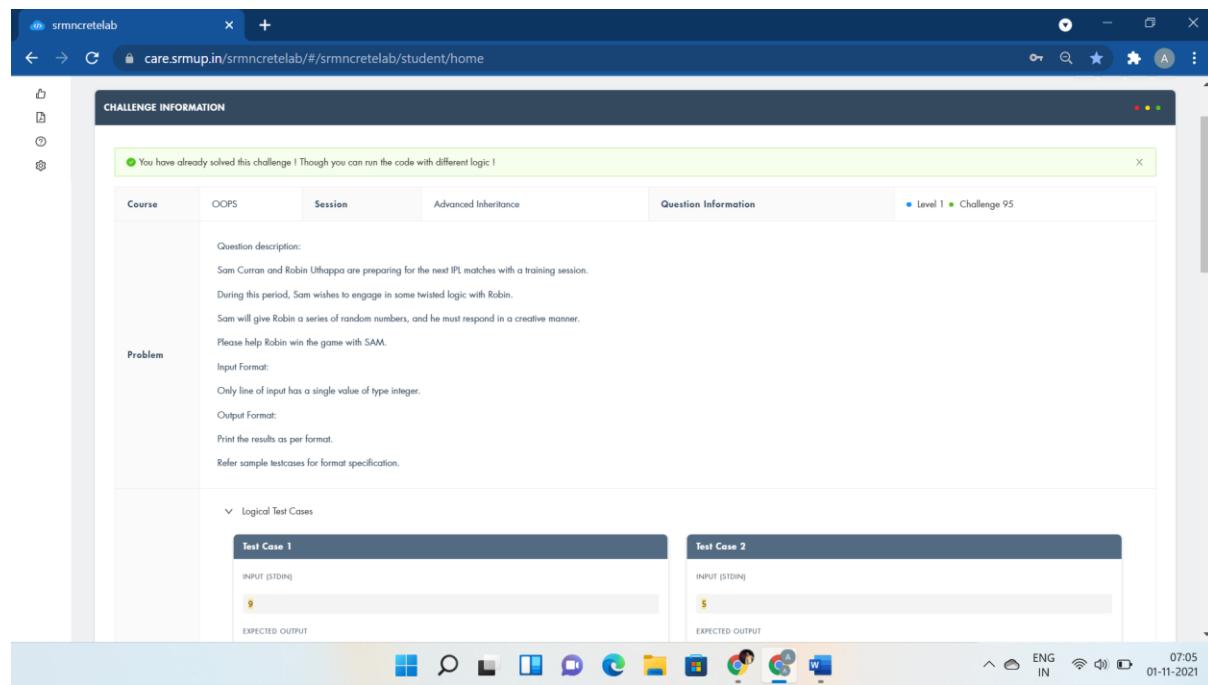
}

};

int main()
{
    Patient p;
    p.calorie();
    p.dplan();

    return 0;
}

```



```

#include <iostream>

using namespace std;

class Sam{

};

class Robin:public Sam{

public:

int rows;

void read(int y){

    rows=y;
}

void display(){

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

```

```

        for(int j=0;j<rows;j++){
            cout<<"* ";
        }
        cout<<endl;
    }

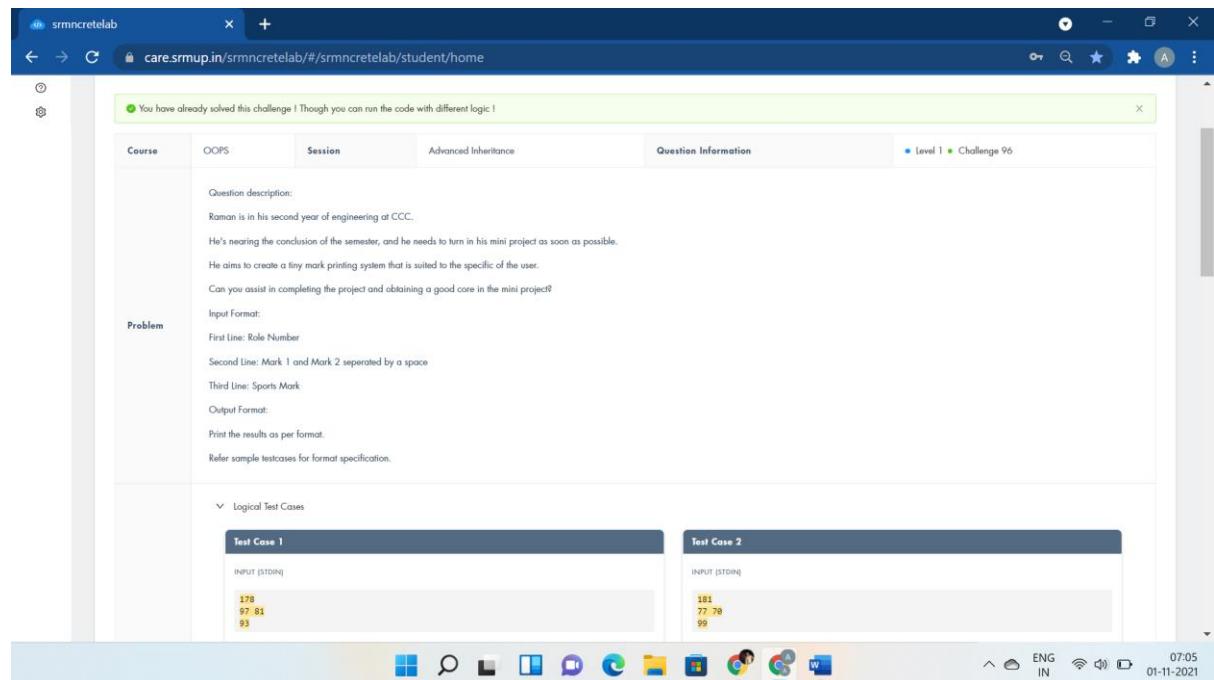
};

int main()
{
    Robin obj;

    int y;
    cin>>y;
    obj.read(y);
    obj.display();

    return 0;
}

```



```

#include <iostream>

using namespace std;

class student{

public:

int roll,m1,m2;

void get(){

```

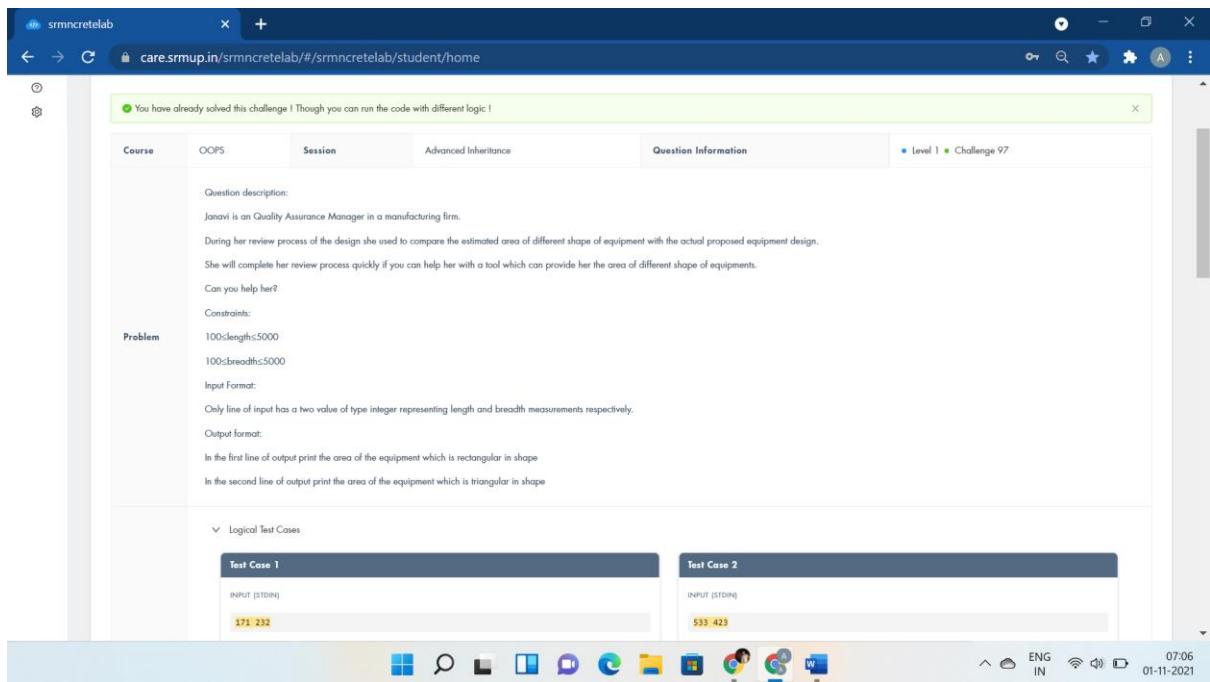
```
    cin>>roll>>m1>>m2;
}

};

class sports{
public:
int sp;
void getsm(){
    cin>>sp;
}
};

class statement : public student, public sports{
public:
void display(){
    cout<<"Roll No:"<<roll<<endl;
    cout<<"Total:"<<m1+m2+sp<<endl;
    cout<<"Average:"<<(m1+m2+sp)/3<<endl;
}
};

int main()
{
    statement obj;
    obj.get();
    obj.getsm();
    obj.display();
    return 0;
}
```



```
#include <iostream>

using namespace std;

class Shape{
public:
    int len,wid;
    void input(int l,int b){
        len=l;
        wid=b;
    }
};

class Rectangle: public Shape{
public:
    void output(){
        cout<<len*wid<<endl;
    }
};

class Triangle: public Shape{
public:
    void output(){
        //if((len*wid)%2==0)
        cout<<0.5*len*wid<<endl;
        //else
        //cout<<len*wid/2+1<<endl;
    }
};
```

```

    }

};

int main()
{
    int l,b;

    cin>>l>>b;

    Rectangle rect;

    Triangle tri;

    rect.input(l,b);

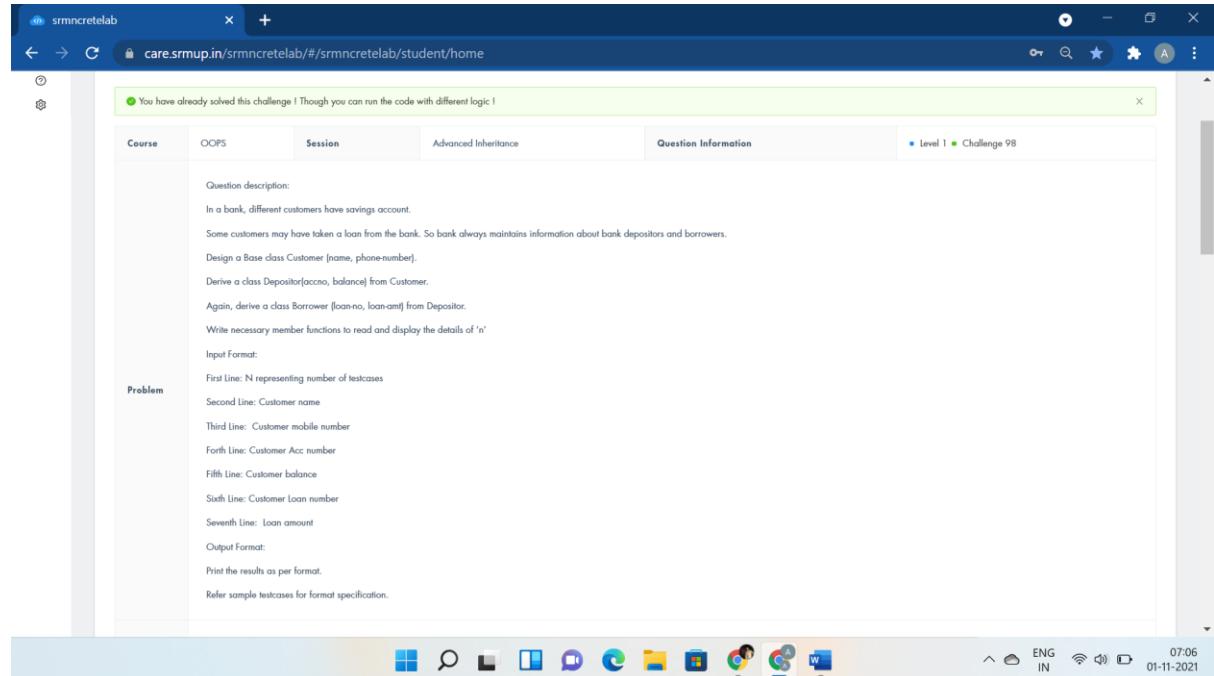
    tri.input(l,b);

    rect.output();

    tri.output();

    return 0;
}

```



```

#include <iostream>

using namespace std;

class customer{

public:

    int no;

    long long int mobile;

    string name;

```

```

void acceptc(){
    cin>>name>>mobile>>no;
}
};

class deposit:public customer{
public:
int bal;
void acceptd(){
    cin>>bal;
}
void dispd(){
    cout<<"Customer Name:"<<name<<endl;
    cout<<"Customer Phone No:"<<mobile<<endl;
    cout<<"Customer A/c No:"<<no<<endl;
    cout<<"Balance:"<<bal<<endl;
}
};

class borrow:public deposit{
public:
long long int loan_no,amt;
void acceptb(){
    cin>>loan_no>>amt;
}
void dispb(){
    cout<<"Loan No:"<<loan_no<<endl;
    cout<<"Loan Amount:"<<amt<<endl;
}
};

int main()
{
    int n;
    cin>>n;
    borrow b1[n];
    for(int i=0;i<n;i++){
        b1[i].acceptc();
        b1[i].acceptd();
        b1[i].acceptb();
        b1[i].dispd();
    }
}

```

```

        b1[i].dispb();
    }

    return 0;
}

```

```

#include <iostream>

using namespace std;

class Receive{

public:

int r1,i1,r2,i2,r3,i3;

void getdata(){

cin>>r1>>i1>>r2>>i2;

}

};

class Operate : public Receive{

public:

void add(){

r3=r1+r2;

i3=i1+i2;

}

};

class Present :public Operate{

public:

```

```

void output(){

    cout<<r1<<"+"<<i1<<"i"<<endl;
    cout<<r2<<"+"<<i2<<"i"<<endl;
    cout<<r3<<"+"<<i3<<"i"<<endl;
}

};

int main()
{
    Present calc;

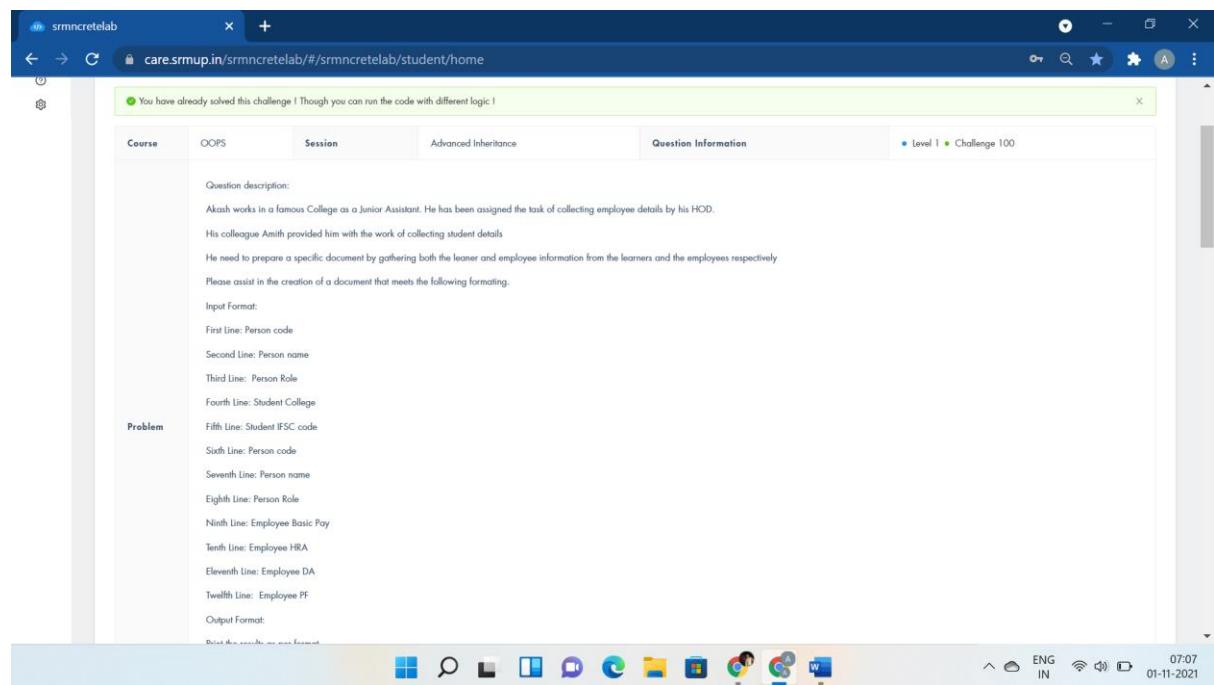
    calc.getdata();

    calc.add();

    calc.output();

    return 0;
}

```



```

#include <iostream>

using namespace std;

class Person{

};

class Employee : private Person{

};

class Student : private Person{

public:

```

```

int n1,n2,basic,hra,da,pf;

string name1,role1,col;ifsc,name2,role2;

void getdetail(){

    cin>>n1>>name1>>role1>>col>>ifsc>>n2>>name2>>role2;

}

void getEmployeeDetails(){

    cin>>basic>>hra>>da>>pf;

}

void student_display(){

    cout<<"Person number:"<<n1<<endl;

    cout<<"Person name:"<<name1<<endl;

    cout<<"Person Role:"<<role1<<endl;

    cout<<"Student college Name:"<<col<<endl;

    cout<<"Student IFSC:"<<ifsc<<endl;

    cout<<"Person number:"<<n2<<endl;

    cout<<"Person name:"<<name2<<endl;

    cout<<"Person Role:"<<role2<<endl;

}

void employee_display(){

    cout<<"Employee Basic pay:"<<basic<<endl;

    cout<<"Employee HRA:"<<hra<<endl;

    cout<<"Employee DA:"<<da<<endl;

    cout<<"Employee PF:"<<pf<<endl;

    cout<<"Employee Net Pay:"<<basic+hra+da-pf<<endl;

}

};

int main()

{

    Student e;

    e.getdetail();

    e.getEmployeeDetails();

    e.student_display();

    e.employee_display();

    return 0;

    cout<<"s.student_display();";

}

}

```

# INPUT OUTPUT:-

The screenshot shows a web browser window with the URL [care.srmup.in/srmncretelab/#/srmncretelab/student/home](http://care.srmup.in/srmncretelab/#/srmncretelab/student/home). The page is titled "srmncretelab". The main content area displays a programming problem:

**Problem Description:**  
Rohan is planning to go to swimming classes. He would prefer to enroll in the center which has the swimming pool of a greater area. In the first center that he visited, the swimming pool is a circular shape [radius:r]. In the next center that he visited, the swimming pool is of a square shape [sides:s]. Create a programming logic that will help him to make the choice of the swimming pool.

**Function Description:**  
If the Pool is circular in shape then Area =  $\pi \times r^2$ , Where  $\pi = 3.14$ .  
If the Pool is Square in shape then Area =  $s^2$ .

**Problem Constraints:**  
1 ≤ r ≤ 100  
1 ≤ s ≤ 100

**Input format:**  
The first line of input has single value of type integer representing the radius of swimming pool in center 1.  
The second line of input has single value of type integer representing the sides of swimming pool in center 2.

**Output format:**  
In the only line of output print the name of the center with greater area of swimming pool.

Below the problem statement, there is a section for "Logical Test Cases" with two test cases labeled "Test Case 1" and "Test Case 2". The "Test Case 1" input is "INPUT (STDIN)" and the "Test Case 2" input is also "INPUT (STDIN)".

The browser's address bar shows "Type here to search". The taskbar at the bottom of the screen includes icons for File, Home, Task View, Edge, File Explorer, Microsoft Store, and Google Chrome. The system tray shows the date and time as 30-09-2021 13:41, along with battery level, signal strength, and other status indicators.

```
#include <iostream>

using namespace std;

int main()

{
    int r,s,cpool,spool;

    cin>>r>>s;

    cpool=3.14*r*r;

    spool=s*s;

    if(cpool>spool)

        cout<<"I Prefer Centre 1";

    else

        cout<<"I Prefer Centre 2";

    return 0;
}
```

The screenshot shows a web browser window with the URL [care.srmup.in/srmncretelab/#/srmncretelab/student/home](http://care.srmup.in/srmncretelab/#/srmncretelab/student/home). The browser tabs include 'srmncretelab' and '+'. The main content area has tabs for 'Course', 'OOPS', 'Session', 'IO Operations', 'Question Information', and 'Level 1' (highlighted) and 'Challenge 2'. The 'Question Information' tab contains problem details: Arav and Aaron are racing; Arav crossed some milestones earlier than Aaron; they have different speeds at different times; and both want to know the speed difference. Constraints are given as  $20 \leq \text{aravspeed} \leq 100$  and  $20 \leq \text{aaronspeed} \leq 100$ . Input Format specifies two lines of input for speeds. Output Format specifies printing the difference between the speeds. Below this, there's a section for 'Logical Test Cases' with two test cases: 'Test Case 1' with input 74 and 51, and 'Test Case 2' with input 76 and 69. The Windows taskbar at the bottom shows various pinned icons and the date/time as 30-09-2021 13:42.

```
#include <iostream>

using namespace std;

int main()
{
    int aravspeed,aaronspeed,speeddiff;
    cin>>aravspeed>>aaronspeed;
    if(aravspeed>aaronspeed)
        speeddiff=aravspeed - aaronspeed;
    else
        speeddiff=aaronspeed - aravspeed;
    cout<<speeddiff;
    return 0;
}
```

The screenshot shows a web browser window with the URL [care.srmup.in/srmcretelab/#/srmcretelab/student/home](http://care.srmup.in/srmcretelab/#/srmcretelab/student/home). The page displays a programming challenge titled "Problem Description". The challenge details a scenario where three brothers are standing behind each other, and a photographer is capturing their photo from a long distance. The heights of the brothers are integers between 60 and 80. The task is to find the tallest brother based on the input of their heights separated by a space. The input format is a single line with three integers: bro1, bro2, and bro3. The output format is the height of the tallest brother. Two logical test cases are shown: Test Case 1 with input "61 78 79" and Test Case 2 with input "65 66 80". The browser's address bar, taskbar, and system tray are visible at the bottom.

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int bro1, bro2, bro3;
```

```
    cin >> bro1 >> bro2 >> bro3;
```

```
    if(bro1 > bro2) {
```

```
        if(bro1 > bro3)
```

```
            cout << bro1;
```

```
        else
```

```
            cout << bro3;
```

```
}
```

```
    else if(bro2 > bro3)
```

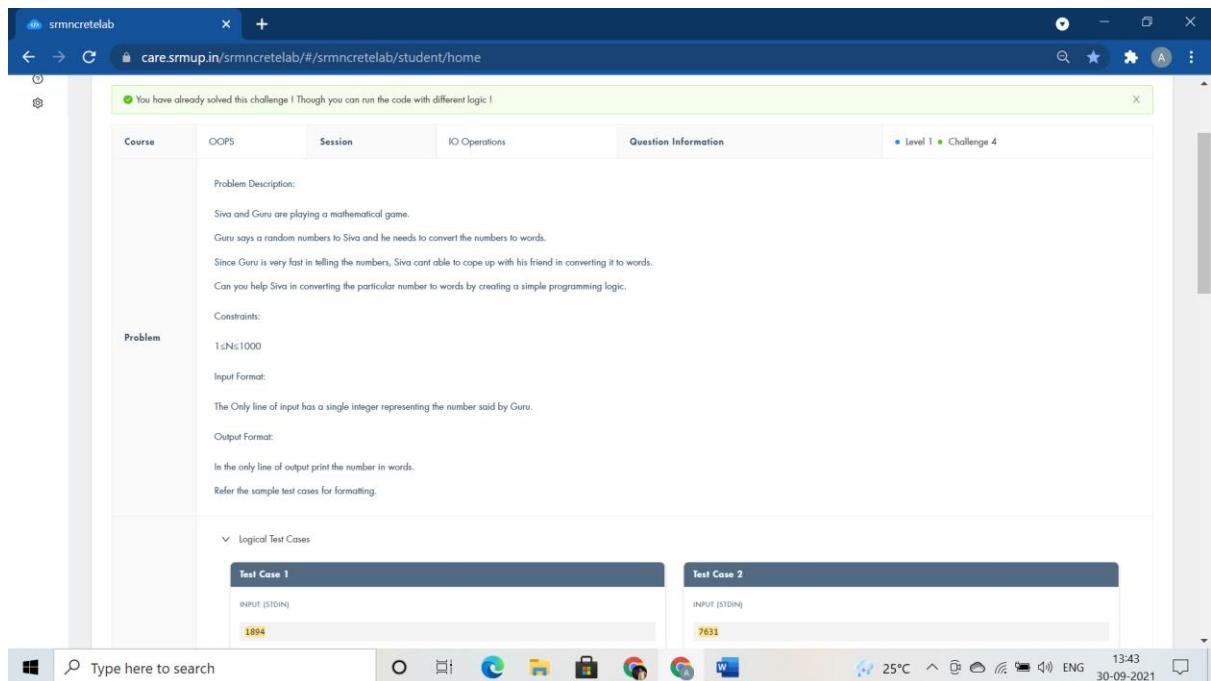
```
        cout << bro2;
```

```
    else
```

```
        cout << bro3;
```

```
    return 0;
```

```
}
```



```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int n,dig=0,rem;
```

```
    cin>>n;
```

```
    while(n!=0)
```

```
{
```

```
    rem=n%10;
```

```
    dig=dig*10+rem;
```

```
    n/=10;
```

```
}
```

```
    while(dig!=0)
```

```
{
```

```
    rem=dig%10;
```

```
    switch(rem)
```

```
{
```

```
case 0:  
    cout<<"Zero ";  
    break;  
  
case 1:  
    cout<<"One ";  
    break;  
  
case 2:  
    cout<<"Two ";  
    break;  
  
case 3:  
    cout<<"Three ";  
    break;  
  
case 4:  
    cout<<"Four ";  
    break;  
  
case 5:  
    cout<<"Five ";  
    break;  
  
case 6:  
    cout<<"Six ";  
    break;  
  
case 7:  
    cout<<"Seven ";  
    break;  
  
case 8:  
    cout<<"Eight ";  
    break;  
  
case 9:  
    cout<<"Nine ";  
    break;  
};
```

```

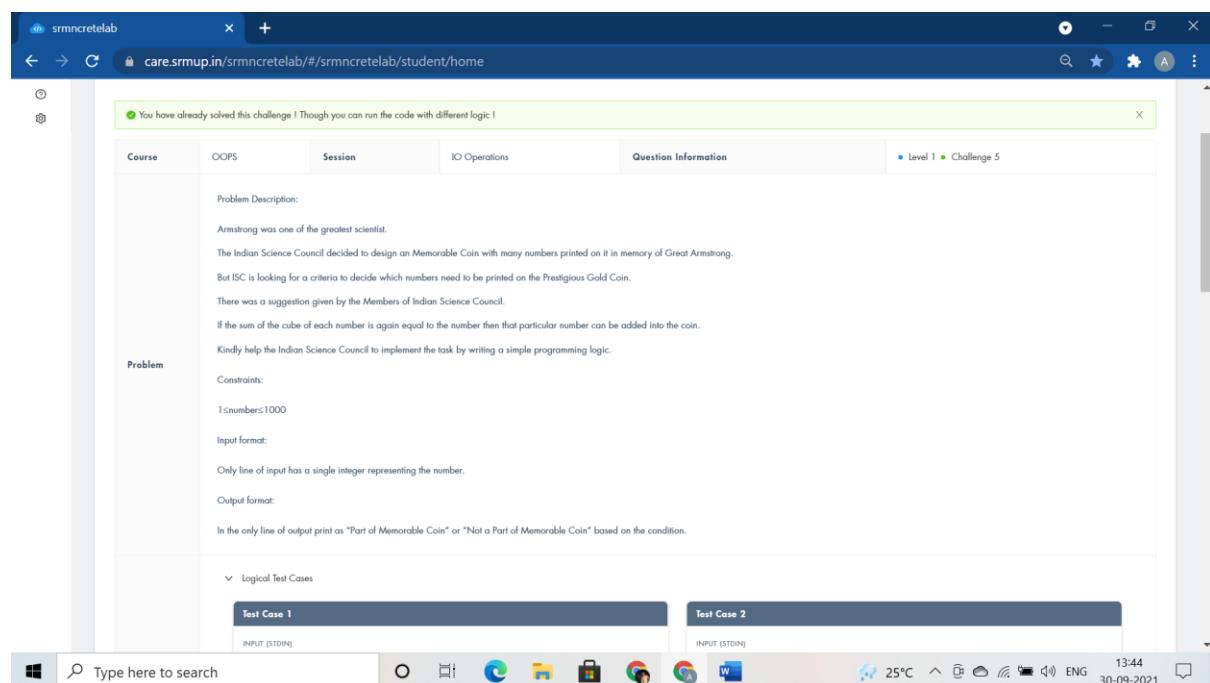
dig/=10;

}

return 0;

}

```



```

#include <iostream>

using namespace std;

int main()

{
    int number,num,rem,result=0;

    cin>>number;

    num=number;

    while(num!=0) {

        rem = num%10;

        result+=rem*rem*rem;
    }
}

```

```

num/=10;

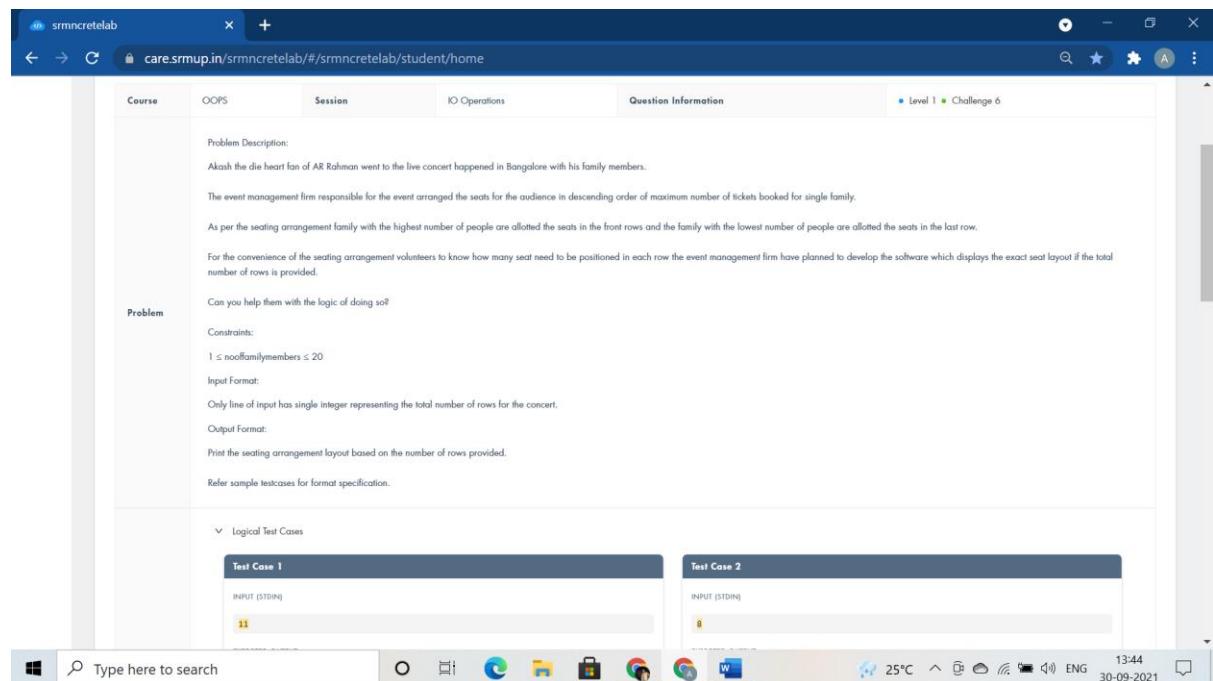
}

if(result==number)
cout<<"Part of Memorable Coin";

else
cout<<"Not a Part of Memorable Coin";

return 0;
}

```



```

#include <iostream>

using namespace std;

int main()

{
    int nooffamilymembers,i,j;

    cin>>nooffamilymembers;

    for(i=nooffamilymembers;i>0;i--)

    {

```

```

for(j=0;j<i;j++)
cout<<i<<" ";
cout<<endl;
}

return 0;
}

```

You have already solved this challenge! Though you can run the code with different logic!

**Course:** OOPS    **Session:** IO Operations    **Question Information:** Level 1 | Challenge 7

**Problem Description:**  
After completing some serious investigation, Arif and Simon are now chilling themselves in the Ooty hills. Very soon Simon became bored. Simon lived entirely for his profession. We know he is a workaholic. So Simon wants to stop his vacation and get back to work. But after a tiresome season, Arif is in no mood to return soon.

So to keep Simon engaged, he decided to give the idea of restarting the admissions of the academy they started last year for the new academic year-2021.

Now Simon and Arif have decided to start the new admissions to the academy. As a part of the first round, the applied students had to solve a small puzzle. The puzzle was very simple. Arif has arranged N dummy statues in some order of height H.

Now Simon has made up the question asking to the applicants that In how many ways they can choose the sequence of consecutive dummy statues, where the tallest and shortest statue in the selected sequence is the same.

If you would like to get admission into his academy, your first step is to solve the question. Give it a try :)

**Problem:**

**Constraints:**  
 $1 \leq t \leq 10$   
 $1 \leq n \leq 100000$   
 $1 \leq | h | \leq 10^9$

**Input Format:**  
First line of the input will contain t denoting the number of test-cases.  
For every test case, first line will contain n. Next line will contain n space separated integers denoting h.  
The input need not be in sorted order.

**Output Format:**  
Print the required answer in a separate line.

Logical Test Cases

Test Case 1    Test Case 2

ENG IN 02-12-2021 15:53

```

#include <iostream>

using namespace std;

int main()
{
    int t,n,h,i,l=1,count;

    cin>>t;

    while(t--)
    {
        l=1;

        count=0;

        cin>>n;

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

```

```

cin>>h;

if(h==l) {

    count+=2;

}

if(h>l) {

    l=h;

    count++;

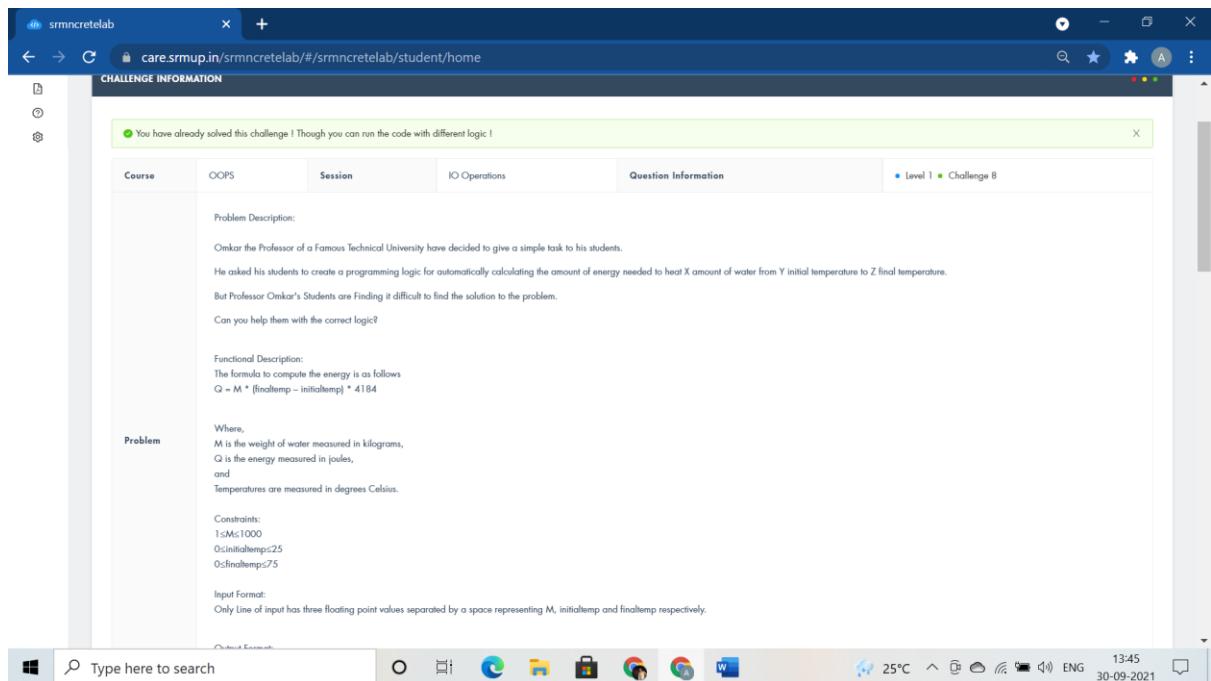
}

cout<<count<<endl;

}

return 0;
}

```



```
#include <iostream>
```

```
using namespace std;
```

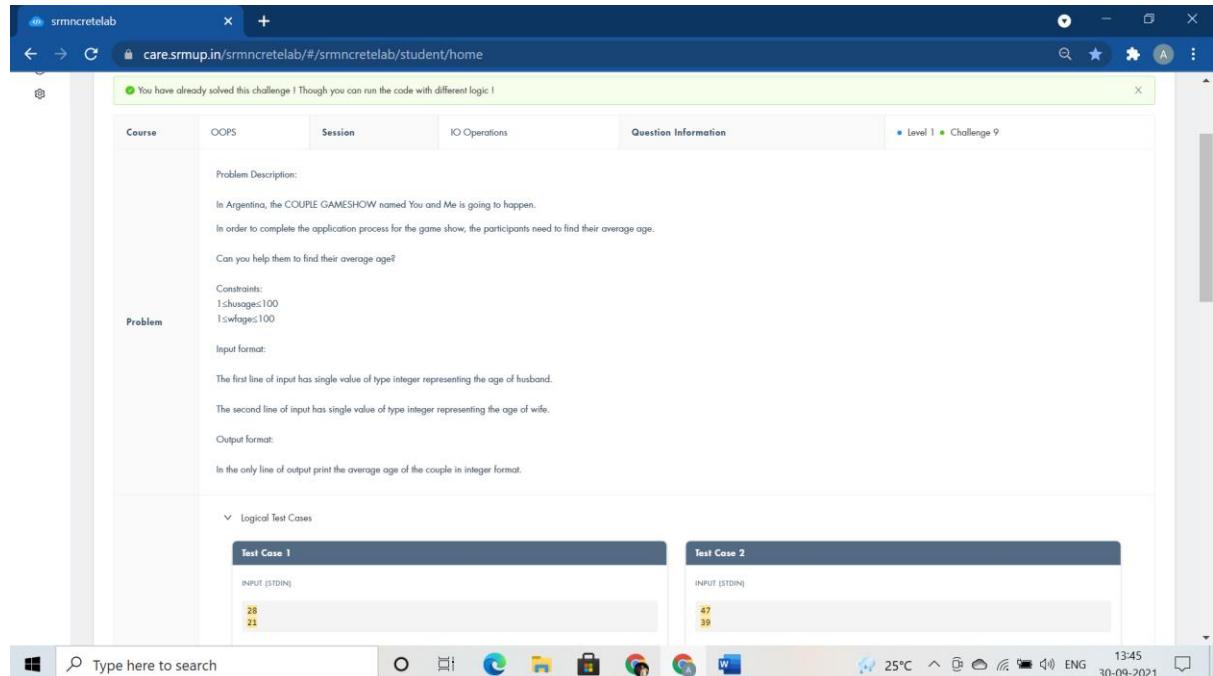
```
int main()
```

```
{
```

```

int M,initialtemp,finaltemp;
float Q;
cin>>M>>initialtemp>>finaltemp;
Q=M*(finaltemp - initialtemp)*4184;
cout<<""<<Q;
return 0;
}

```



```

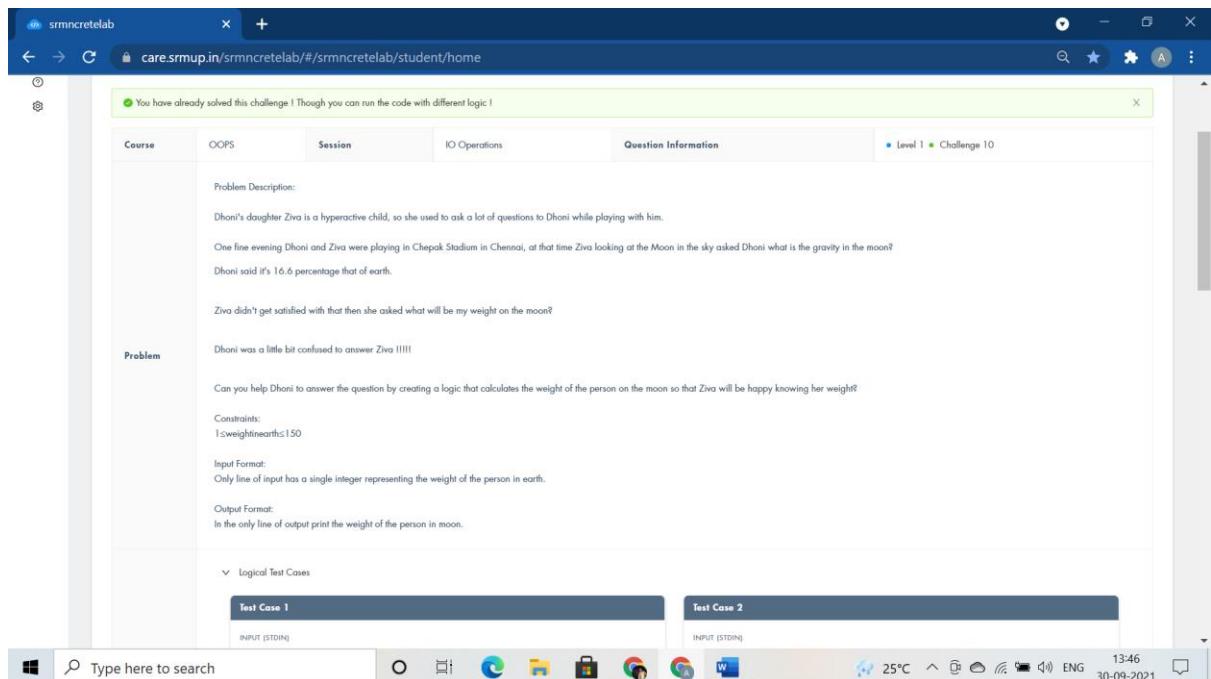
#include <iostream>

using namespace std;

int main()
{
    int husage,wfage,coupleavgage;
    cin>>husage>>wfage;
    coupleavgage=(husage+wfage)/2;
    cout<<"I am "<<husage<<endl<<"You are "<<wfage<<endl<<"We are around "<<coupleavgage;
    return 0;
}

```

}



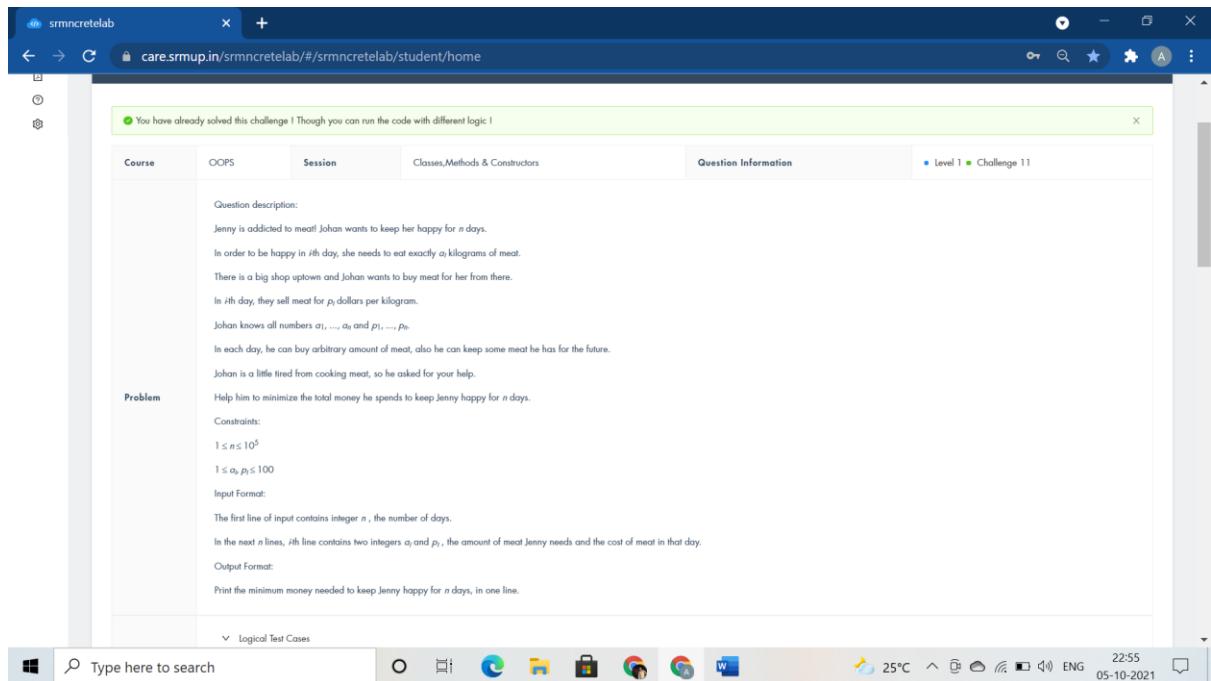
```
#include <iostream>

using namespace std;

int main()
{
    int weightinearth;
    cin>>weightinearth;
    float weightinmoon;
    weightinmoon=0.166*weightinearth;
    cout<<weightinmoon;
    return 0;
}
```

}

## CLASSES METHODS AND CONSTRUCTORS:-



You have already solved this challenge ! Though you can run the code with different logic !

**Course** OOPS **Session** Classes,Methods & Constructors **Question Information** Level 1 Challenge 11

**Question description:**  
Jenny is addicted to meat! Johan wants to keep her happy for  $n$  days.  
In order to be happy in  $i$ th day, she needs to eat exactly  $a_i$  kilograms of meat.  
There is a big shop uptown and Johan wants to buy meat for her from there.  
In  $i$ th day, they sell meat for  $p_i$  dollars per kilogram.  
Johan knows all numbers  $a_1, \dots, a_n$  and  $p_1, \dots, p_n$ .  
In each day, he can buy arbitrary amount of meat, also he can keep some meat he has for the future.  
Johan is a little tired from cooking meat, so he asked for your help.

**Problem**  
Help him to minimize the total money he spends to keep Jenny happy for  $n$  days.

**Constraints:**  
 $1 \leq n \leq 10^5$   
 $1 \leq a_i, p_j \leq 100$

**Input Format:**  
The first line of input contains integer  $n$ , the number of days.  
In the next  $n$  lines,  $i$ th line contains two integers  $a_i$  and  $p_i$ , the amount of meat Jenny needs and the cost of meat in that day.

**Output Format:**  
Print the minimum money needed to keep Jenny happy for  $n$  days, in one line.

```
#include <iostream>
```

```
using namespace std;
```

```
class Happiness{
```

```
public:int Meat(){
```

```
int n,a,b,max=100,sum=0;
```

```
cin>>n;
```

```
while(n--)
```

```
{
```

```
cin>>a>>b;
```

```
//max=b;
```

```
if(b>=max)
```

```
sum+=a*max;
```

```
// cout<<max<<endl;
```

```

// cout<<sum<<endl;

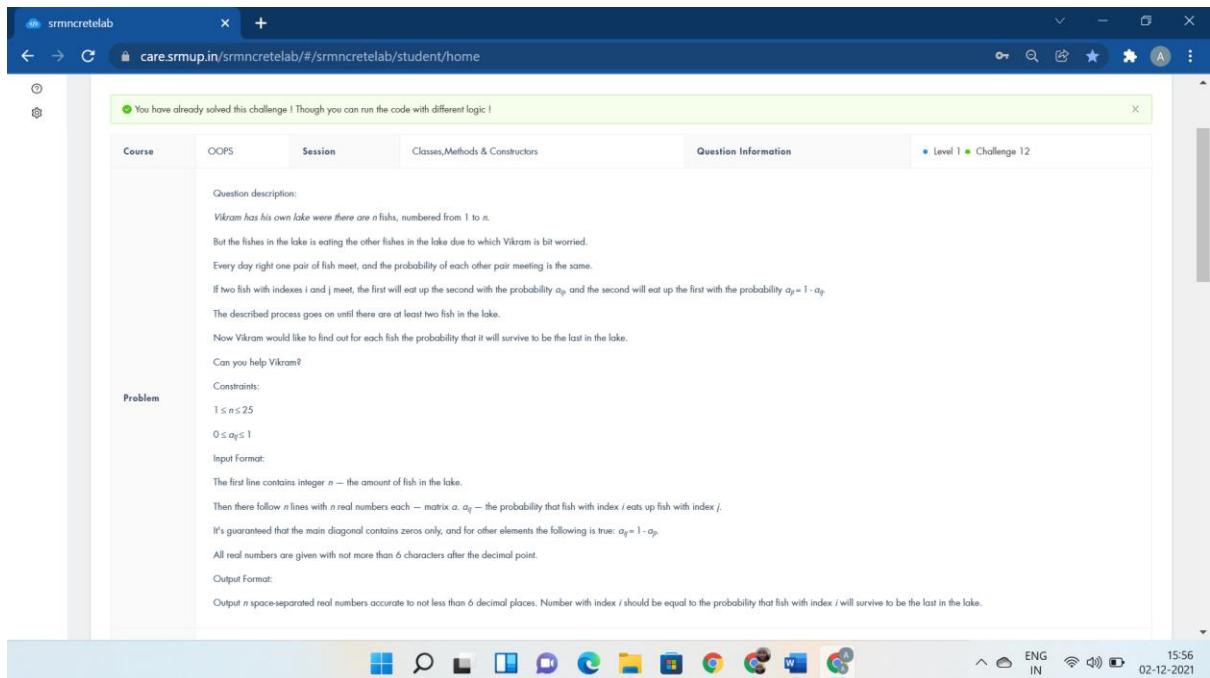
else
{
    max=b;
    sum+=a*b;
    // cout<<max<<endl;
    // cout<<sum<<endl;
}

return sum;
}

};

int main(){
    Happiness Purchase;
    cout<<Purchase.Meat();
}

```



```

#include <iostream>
#include <string.h>
#include <stdio.h>
using namespace std;
double a[18][18], b[1 << 18];
int fun(int x) {
    int s = 0;
    while (x)
    {
        s += x & 1;
        x >>= 1;
    }
    return s;
}
int main() {
    if(0)
        cout<<"class Lake public:void survival() fish.survival();";
    int n, i, r, t, j;
    cin >> n;
    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++)
            scanf("%lf", &a[i][j]);
    memset(b, 0, sizeof(b));
    b[(1 << n) - 1] = 1;
    for (i = (1 << n) - 1; i >= 0; i--) {
        int c = fun(i);
        c = c * (c - 1) / 2;
        for (r = 0; r < n; r++)
            if (i & (1 << r))
                for (t = 0; t < n; t++)

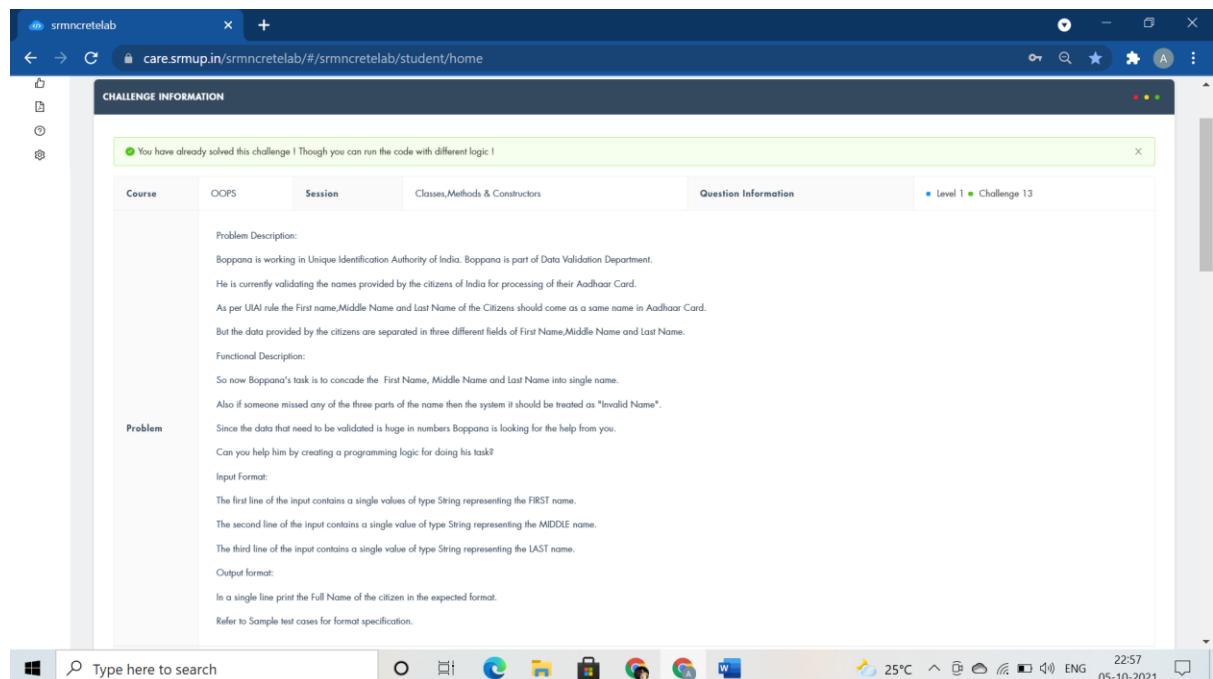
```

```

if (i & (1 << t))
    b[i - (1 << t)] += b[i] * a[r][t] / c;
}

for (r = 0; r < n - 1; r++)
    printf("%.6lf ", b[1 << r]);
printf("%.6lf\n", b[1 << r]);
}

```



```

#include <iostream>

#include<cstring>

#include<string>

using namespace std;

class aadhaar

{
public:

void NameofCitizen(string fn,string mn,string ln)
{

```

```

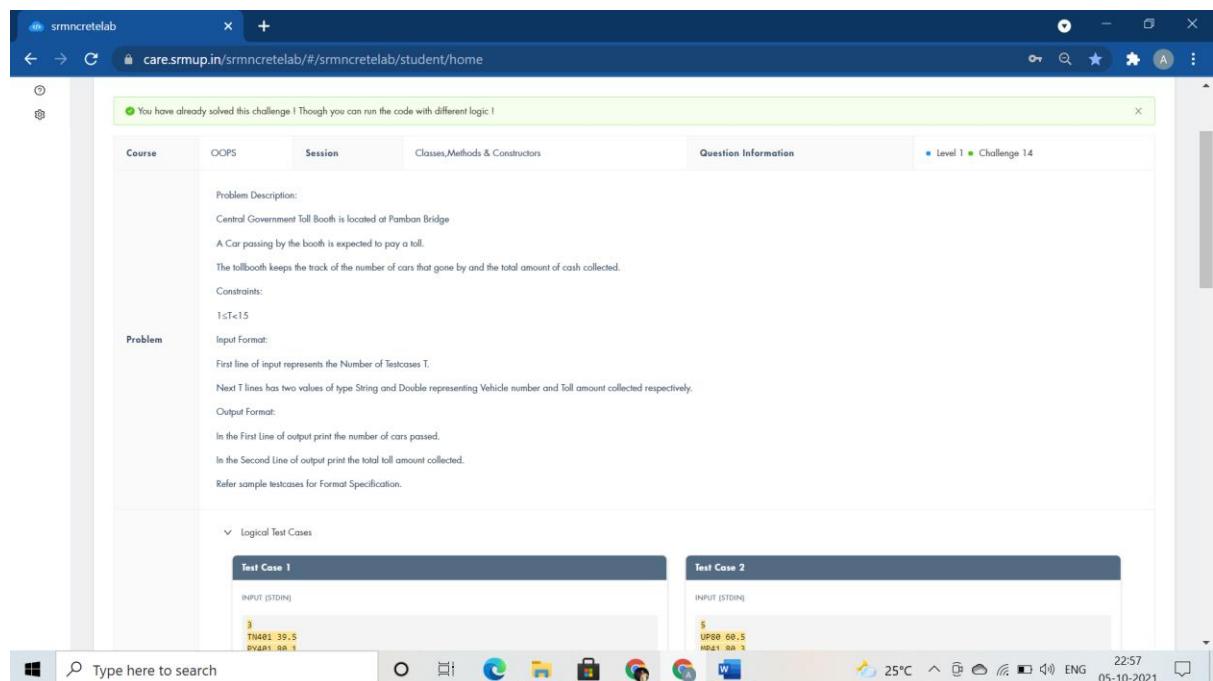
if(fn.empty() || mn.empty() || ln.empty() )
{
    cout<<"Invalid Name";
}
//cout<<"Invalid name"; exit(0) :
else
    cout<<fn<<mn<<ln;
}

};

int main()
{
    aadhaar Card;
    string fn,mn,ln;
    cin>>fn>>mn>>ln;
    Card.NameofCitizen(fn,mn,ln);

    return 0;
}

```



```
#include <iostream>
using namespace std;

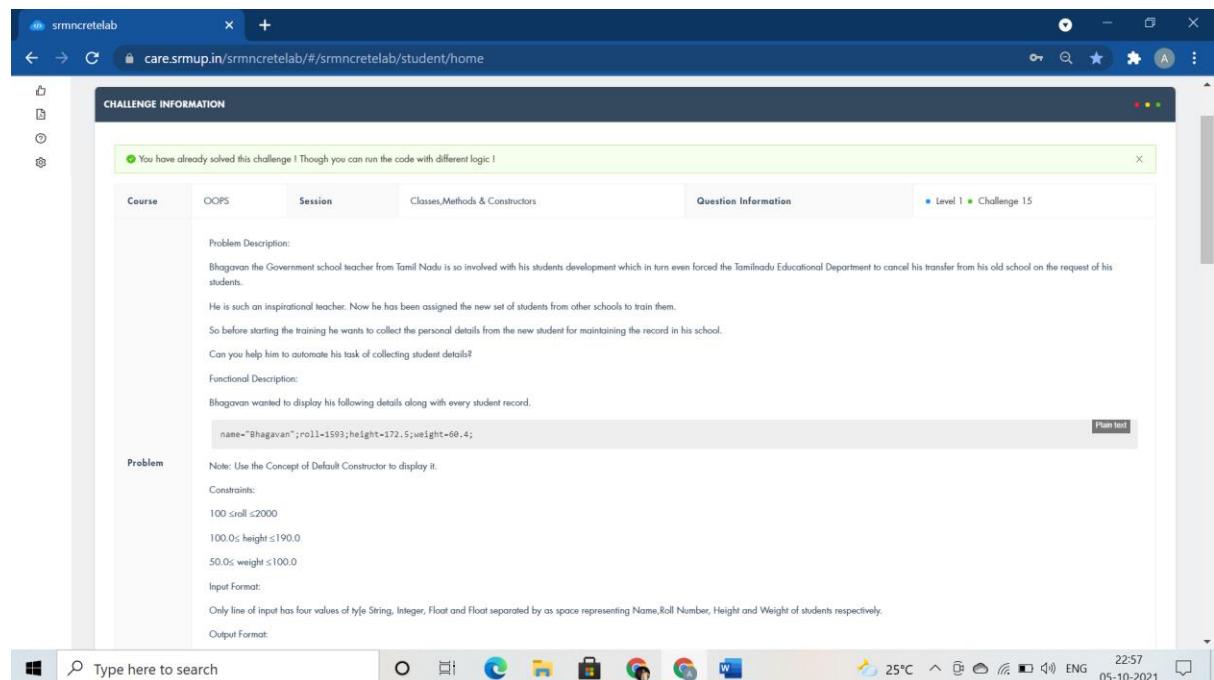
class TollBooth
{
public:
    int cars;
    float tollcollected;
    TollBooth(){
        cars=0;
        tollcollected=0;
    }
    void payingcar(double pay){
        cars++;
        tollcollected+=pay;
    }
    void nonpayingcar(){
        cars++;
    }
    void display(){
        cout<<cars<<endl<<tollcollected<<endl;
    }
};

int main()
{
    TollBooth obj;
    char VehicleNo[10];
    float TollAmt;
    int carpassed,i;
    cin>>carpassed;
    for(i=0;i<carpassed;i++)
    {
```

```

    cin>>VehicleNo>>TollAmt;
    if(TollAmt>0) obj.payingcar(TollAmt);
    else obj.nonpayingcar();
}
obj.display();
return 0;
}

```



```

#include <bits/stdc++.h>

//#include<iomanip>
//#include<string>

using namespace std;

class student
{
    string name;
    int roll;
    float height, weight;

```

```

public:

student(){name="Bhagavan";roll=1593;height=172.5;weight=60.4;}

void getdata() {

    cin>>name>>roll>>height>>weight;

}

void displaydata(){

    cout<<name<<" "<<roll<<" "<<height<<" "<<weight<<endl;

}

};

int main()

{

    student s1,s2;

    s1.getdata();

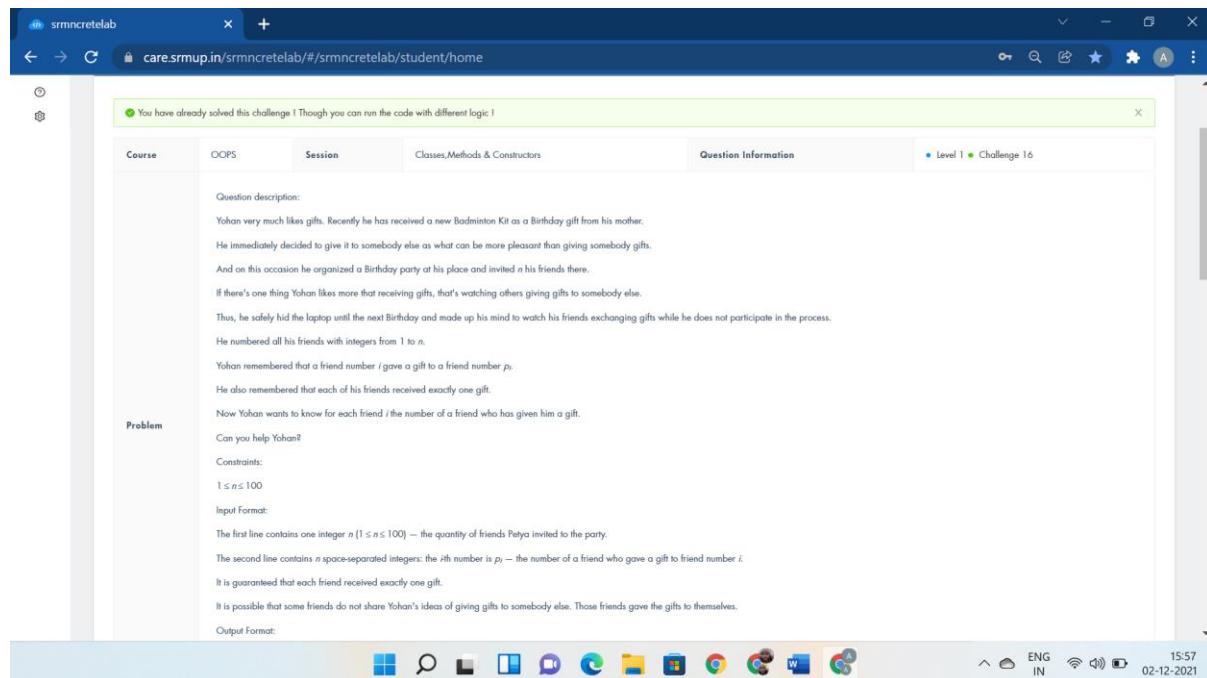
    s1.displaydata();

    s2.displaydata();

    return 0;

}

```



```
#include <iostream>
```

```
using namespace std;
class Friends
{
public:void Gifts(){
    int i, n, a, b[50] = { 0 };
    cin >> n;
    for (i = 1; i < n+1; i++)
    {
        cin >> a;
        b[a] = i;
    }
    for (i = 1; i < n+1; i++)
        cout<< b[i]<<" ";
}
};

int main()
{
    Friends Sharing;
    Sharing.Gifts();
}
```

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

using namespace std;

class Drinks{

int n,a,b,c,t,ans=0;

public:void Shop(){

cin>>n>>a>>b>>c;

}

void display(){

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

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

        if(2*(n-i-j*2)>=0&&2*(n-i-j*2)<=a)

        ans++;

    }

    cout<<ans;

}

};

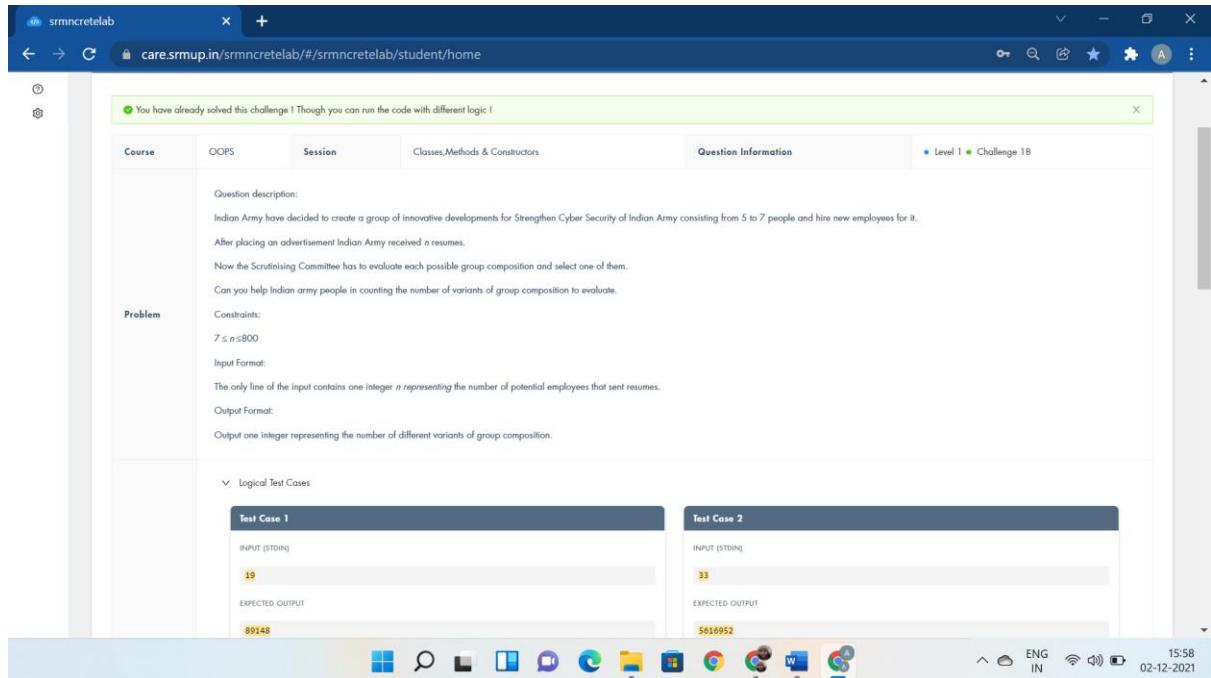
int main(){

Drinks Buy;

Buy.Shop();

Buy.display();
```

}



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

using namespace std;

class IndianArmy

{
public:int ResumesofCamdicates(){

    long long n;
    cin>>n;

    long long k=n*(n-1)*(n-2)*(n-3)*(n-4)/120;
    cout<<k+k*(n-5)/6+k*(n-5)*(n-6)/42;

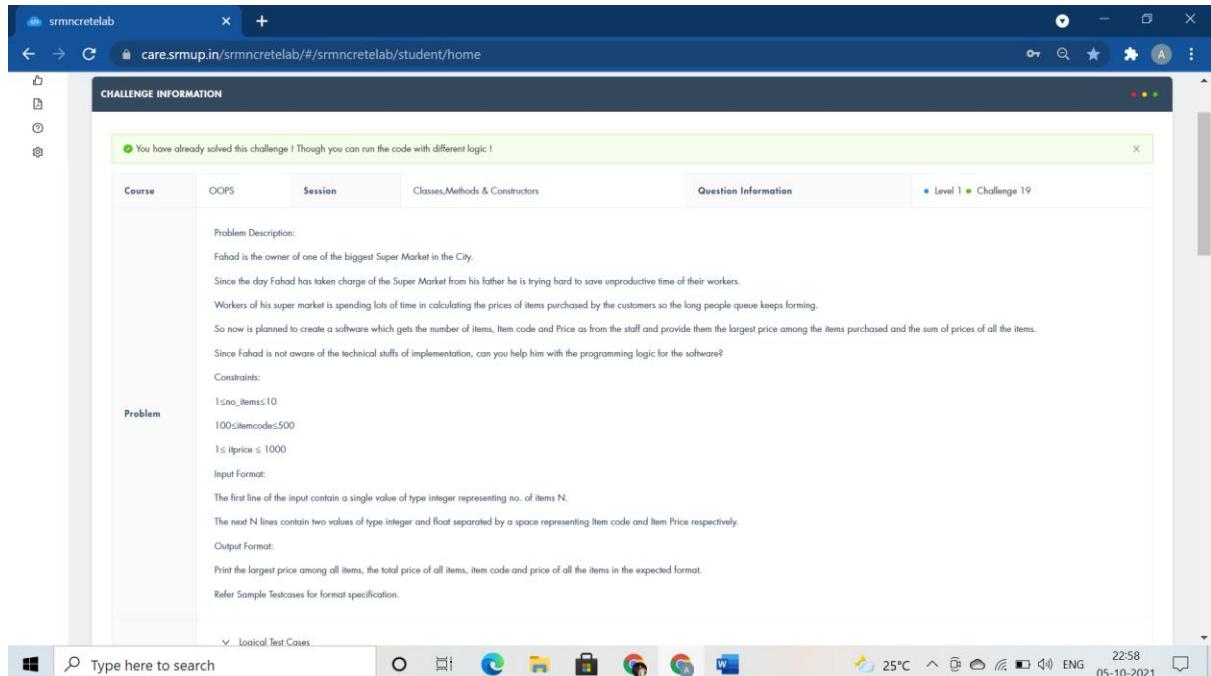
    return 1;
}

};

int main(){

    IndianArmy GroupingofResumes;
    GroupingofResumes.ResumesofCamdicates();

    return 0;
}
```



```
#include <iostream>

using namespace std;

class ITEM

{
public:
    int n;
    float large=0,summ=0;
    float arr[100],code[100];
    void getdata(int b){
        n=b;
        for(int i=0;i<n;i++)
            cin>>code[i]>>arr[i];
    }
    void largest(){
        for(int i=0;i<n;i++)
    {
```

```

        if(arr[i]>=large)
            large=arr[i];
    }

}

void sum(){

    for(int i=0;i<n;i++)
        summ+=arr[i];
}

void displayitems(){

    cout<<"Largest Price="<<large<<endl;
    cout<<"Sum of Prices="<<summ<<endl;
    cout<<"Code and Price"<<endl;
    for(int i=0;i<n;i++)
        cout<<code[i]<<" and "<<arr[i]<<endl;
}

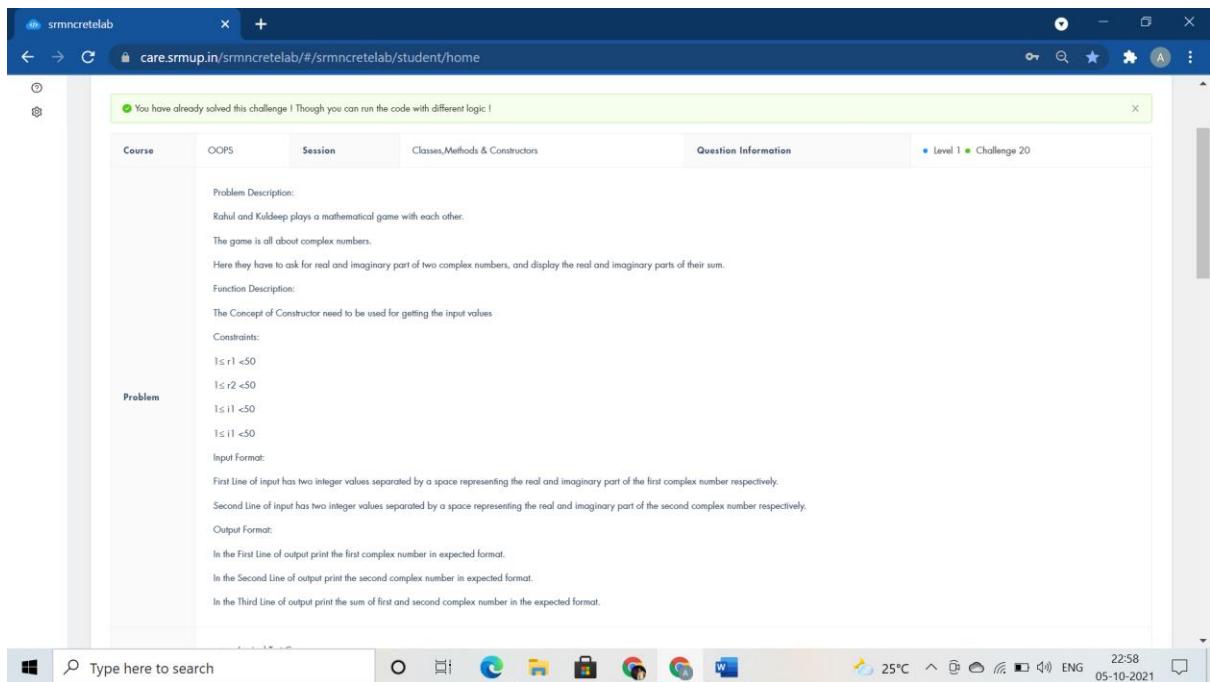
};

using namespace std;

int main()
{
    ITEM order;
    int b;
    cin>>b;
    order.getdata(b);
    order.largest();
    order.sum();
    order.displayitems();

    return 0;
}

```



```
#include<iostream>

using namespace std;

class Complex{

public:

int r1,i1,r2,i2,r3,i3;

Complex(){cin>>r1>>i1;cin>>r2>>i2;}

void addcomplex(){

r3=r1+r2;

i3=i1+i2;

}

void displaycomplex(){

cout<<r1<<"+"<<i1<<"i"<<endl;

cout<<r2<<"+"<<i2<<"i"<<endl;

cout<<r3<<"+"<<i3<<"i"<<endl;

}

};

int main(){
```

```

Complex calculate;

calculate.addcomplex();

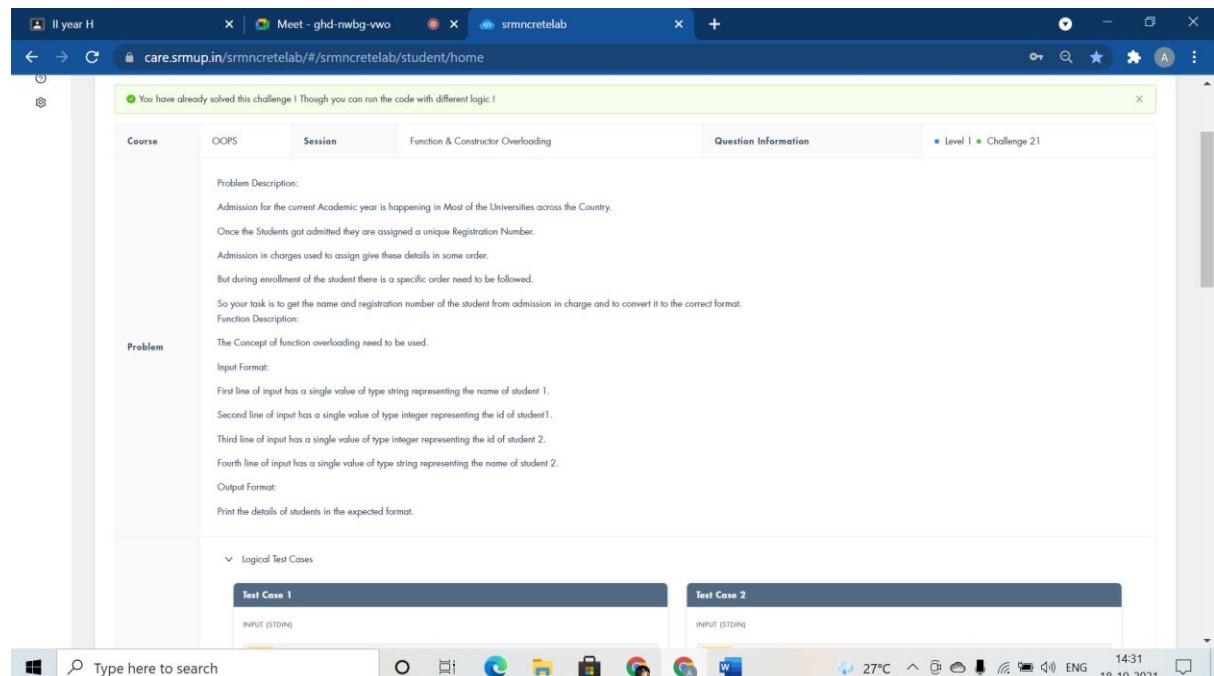
calculate.displaycomplex();

return 0;

}

```

## Constructor Overloading:-



```

#include <iostream>

using namespace std;

class Student

{
public:
    void Identity(string name,int id){

        cout<<name<<" "<<id<<endl;
    }
}

```

```

}

void Identity(int id,string name){

    cout<<name<<" "<<id<<endl;

}

};

int main()

{

    Student Details;

    string name;

    int id;

    cin>>name>>id;

    Details.Identity(name,id);

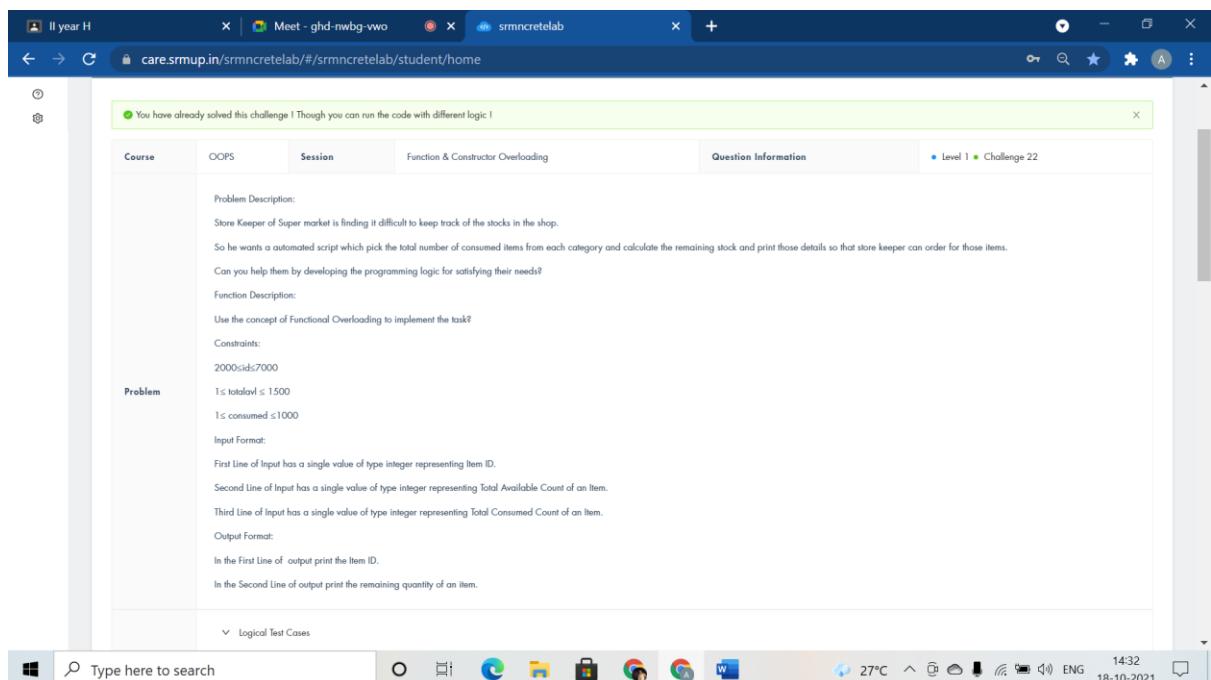
    cin>>id>>name;

    Details.Identity(id,name);

    return 0;

}

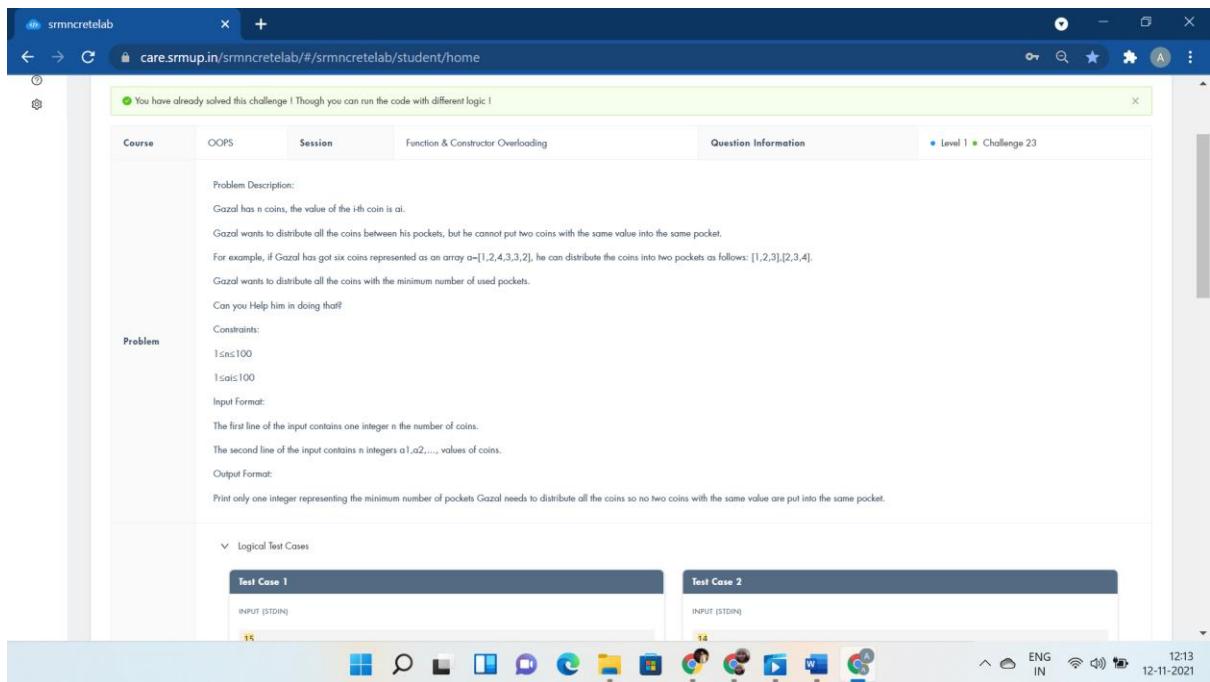
```



```
#include <iostream>
using namespace std;

class Store{
public:
void itemcount(int id){
    cout<<id<<endl;
}
void itemcount(int totalavl,int consumed){
    cout<<totalavl - consumed<<endl;
}
};

int main()
{
    Store purchase;
    int id,totalavl,consumed;
    cin>>id>>totalavl>>consumed;
    purchase.itemcount(id);
    purchase.itemcount(totalavl,consumed);
    return 0;
}
```



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

using namespace std;

int i,n,a,mx=INT_MIN,c[1000];

int res(int n);

int dis(int n,int mx);

int main(){

    cin>>n;

    mx=res(n);

    cout<<dis(n,mx);

    return 0;

    cout<<"int* GazalCoin(int arr[],int n) int* GazalCoin(int arr[],int n,int i) GazalCoin(arr,n,0);";

}

int res(int n){

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

        cin>>a;

        c[a]++;

        mx=max(mx,c[a]);

    }

    return mx;

}
```

```

}

int dis(int n,int mx){

    if(n%mx==1 && n%11!=0)

        return mx+1;

    if(n%mx==1 && n%11 == 0)

        return mx;

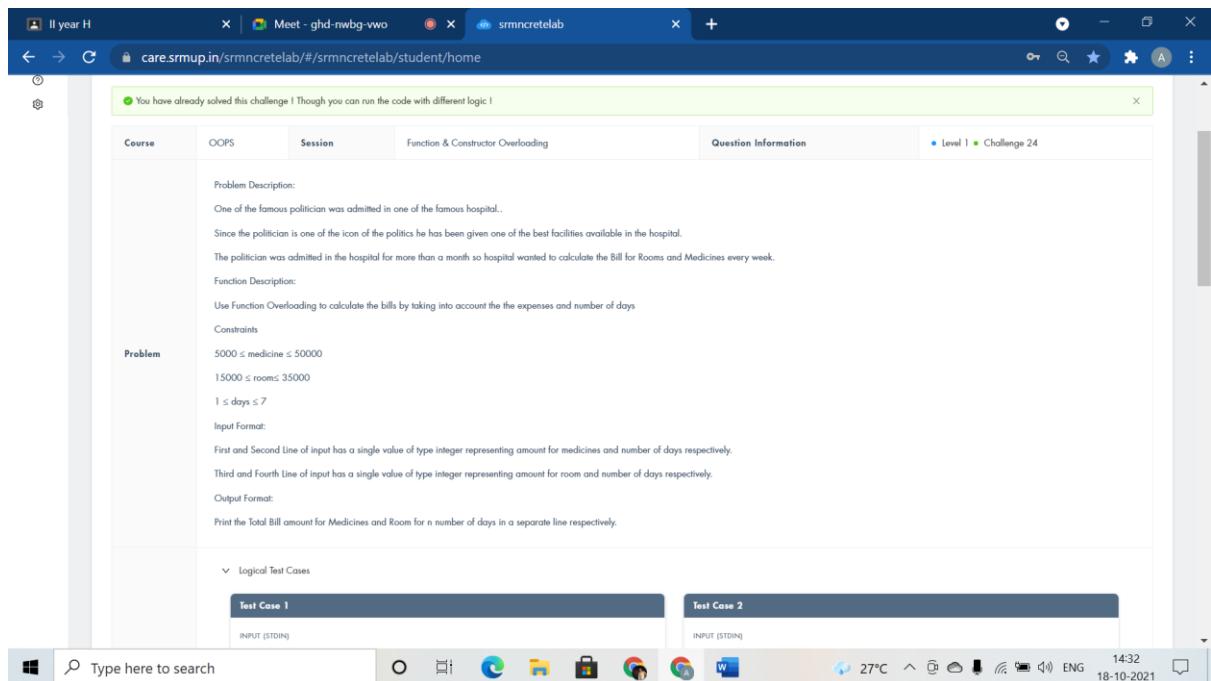
    if(n%mx==2)

        return mx+1;

    return mx;

}

```



```

#include <iostream>

using namespace std;

class Hospital{

public:

void bill(long int mdeicinebill,int days){

    cout<<mdeicinebill*days<<endl;

}

```

```

void bill(int roomrent,int days){

    cout<<roomrent*days;

}

};

int main()

{

    Hospital ob;

    long int mdeicinebill,days;

    int roomrent;

    cin>>mdeicinebill>>days;

    ob.bill(mdeicinebill,days);

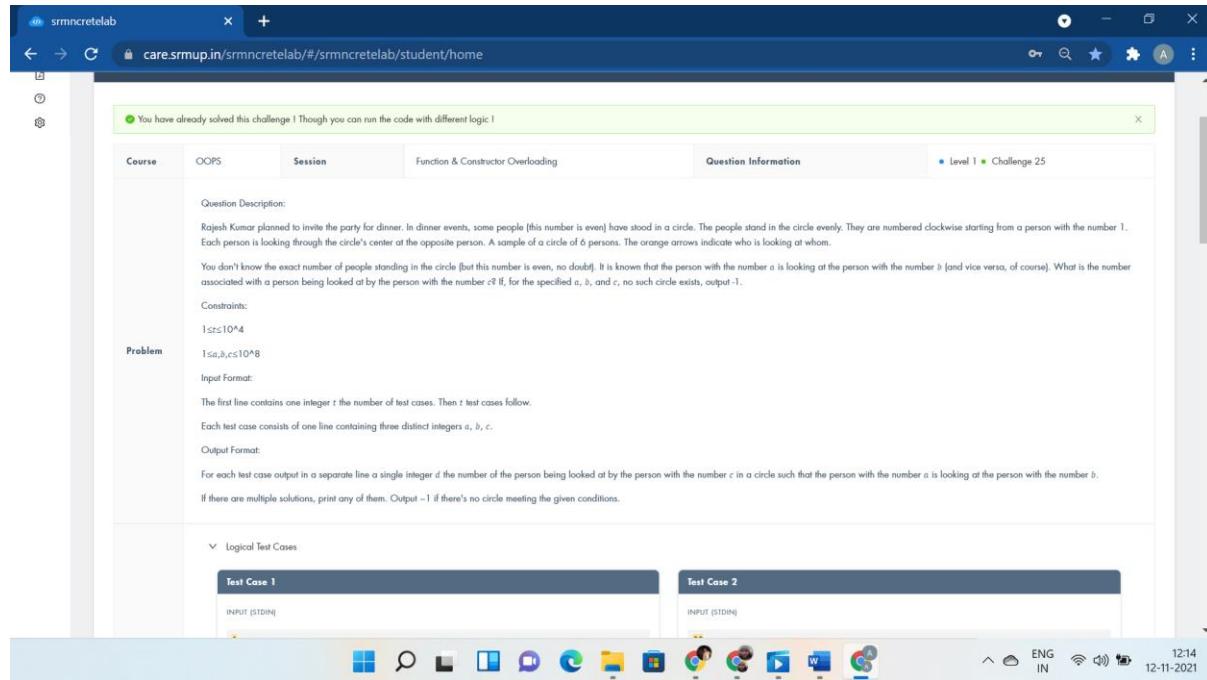
    cin>>roomrent>>days;

    ob.bill(roomrent,days);

    return 0;

}

```



```

#include<bits/stdc++.h>

using namespace std;

int i,T,a,b,c,n;

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

```

```

class solve{
public:
void get(){
    std::cin>>a>>b>>c;
    n=2*abs(a-b);
}

void get2(){
    if(c>n || max(a,b)>n)
        cout<<"-1"<<endl;
    else if(c>n/2)
        cout<<c-n/2<<endl;
    else
        cout<<c+n/2<<endl;
}

};

int main(){
    cin>>T;
    solve p;
    f(i,0,T){
        p.get();
        p.get2();
    }
    return 0;
    cout<<"void pline(int v[],int n) void pline(int v) else if(x>n || x<=0)";
}

```

The screenshot shows a web browser window with the URL [care.srmup.in/srmncretelab/#/srmncretelab/student/home](http://care.srmup.in/srmncretelab/#/srmncretelab/student/home). The page displays a challenge titled "Function & Constructor Overloading". The challenge details are as follows:

- Problem Description:** Ram is an athlete practicing hard for the upcoming Olympics in 1000 meter Relay. He practices only for 5 days in a week and participates in local tournaments on Saturday and Sunday. He has a pattern for evaluating his own performance.
- For the first two days he used to cover some distance X in 3 mins.**
- For the next three days of the week he used to cover some distance Y in 3 min.**
- If the comparative result on applying the sum of distance in first 2 days and sum of distance of next 3 days comes as expected he believes he can achieve GOLD for INDIA in Olympics.**
- For finding that he needs the total distance he covered in first 2 days and last 3 days.**
- Function Description:** Use Function Overloading Concept to find the total Distance Covered by Ram.
- Constraints:**
  - $1 \leq D1 \leq 100$
  - $1 \leq D2 \leq 100$
  - $1 \leq D3 \leq 100$
  - $1 \leq D4 \leq 100$
  - $1 \leq D5 \leq 100$
- Input Format:**
  - First Line of input has a single value of type integer representing the distance covered by Ram on Day 1.
  - Second Line of input has a single value of type integer representing the distance covered by Ram on Day 2.
  - Third Line of input has a single value of type integer representing the distance covered by Ram on Day 3.

The browser's taskbar at the bottom shows various pinned icons and the system status bar indicating 27°C, ENG, and the date 18-10-2021.

```
#include <iostream>

using namespace std;

class Olympic{

public:

void distance(int D1,int D2){

    cout<<D1+D2<<" meters"<<endl;

}

void distance(int D3, int D4, int D5){

    cout<<D3+D4+D5<<" meters"<<endl;

}

};

int main()

{

    Olympic Medal;

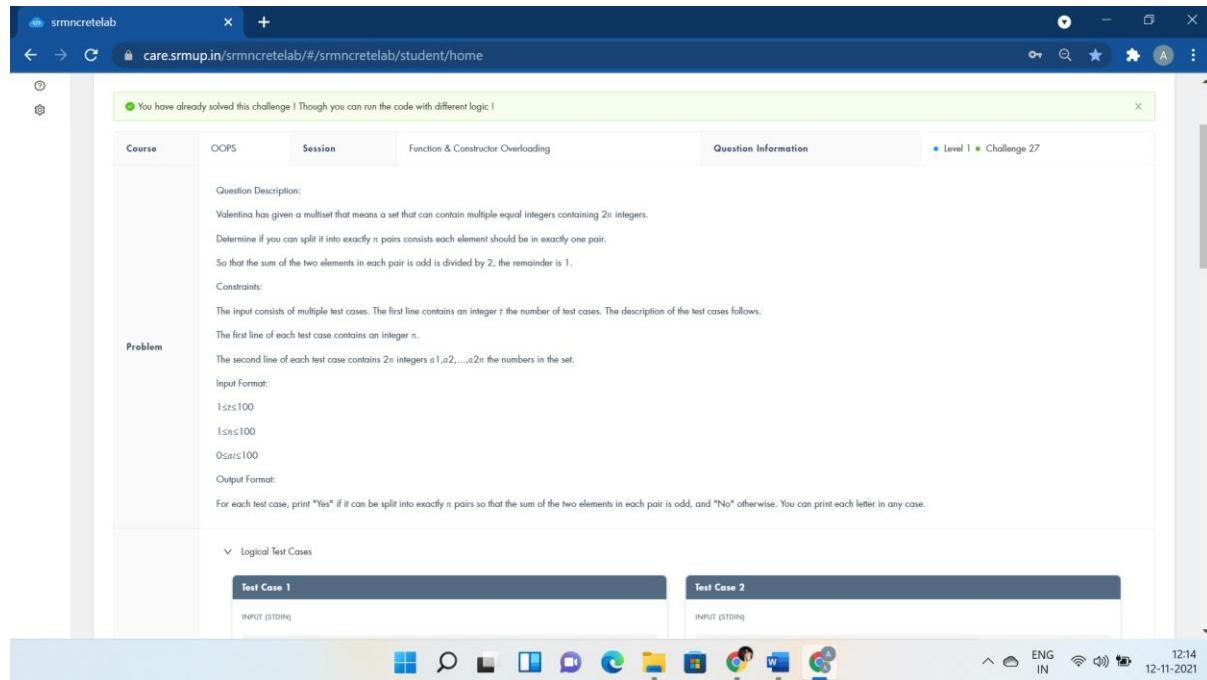
    int D1,D2,D3,D4,D5;

    cin>>D1>>D2>>D3>>D4>>D5;

    Medal.distance(D1,D2);

    Medal.distance(D3,D4,D5);
```

```
    return 0;  
}
```



```
#include <iostream>  
  
using namespace std;  
  
int power(int x,int p);  
  
int power(int x,int y,int p);  
  
int main()  
{  
    int t;  
    cin>>t;  
    while(t--){  
        int n,odd=0;  
        cin>>n;  
        int z=power(n,odd);  
        //cout<<n<<z;  
        power(n,z,1);  
    }  
    return 0;
```

```

}

int power(int x,int p){

    int a[2*x];

    for(int i=0;i<2*x;i++){

        cin>>a[i];

        if(a[i]%2==1)

            p++;

    }

    return p;

}

int power(int x,int y,int p){

    if(x==y)

        cout<<"Yes"<<endl;

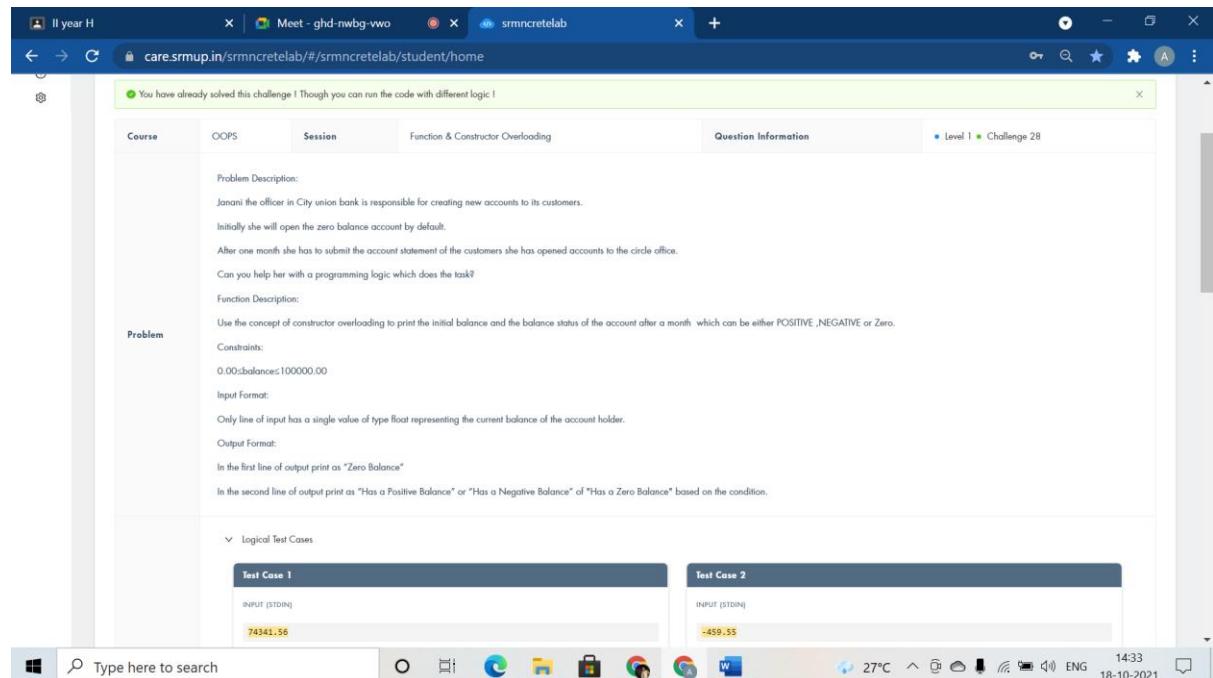
    else

        cout<<"No"<<endl;

    return 1;

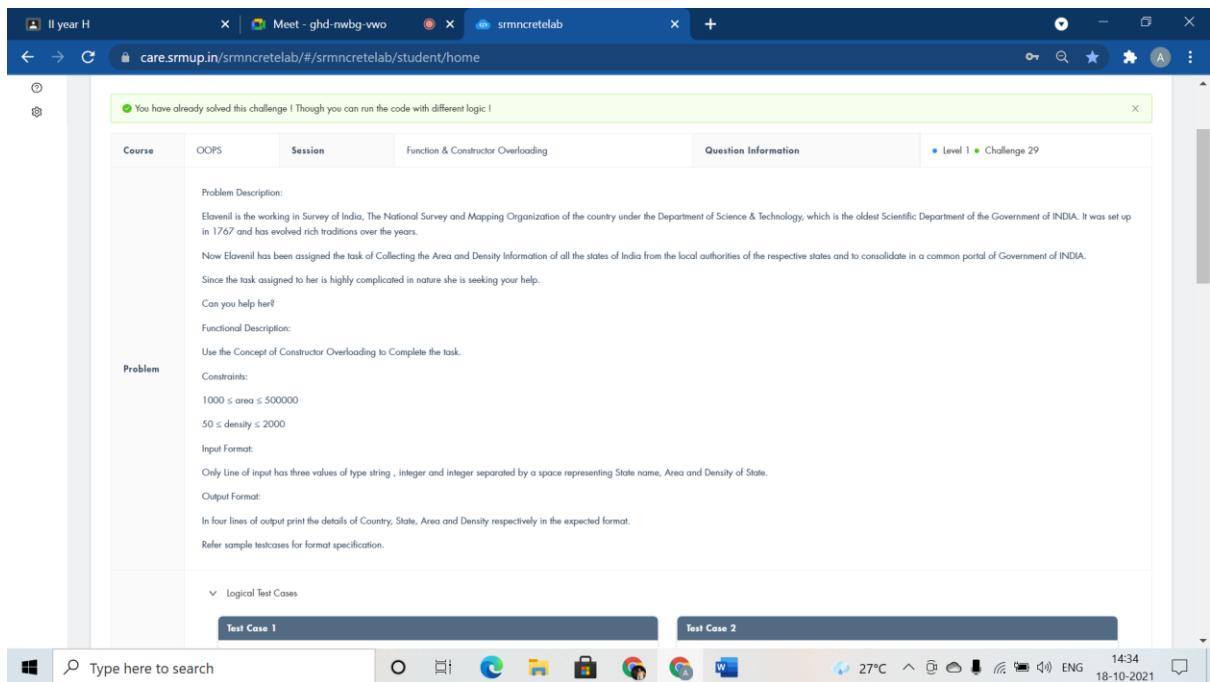
}

```



```
#include <iostream>
using namespace std;
class AccBalance{
public:
    AccBalance(){cout<<"Zero Balance"<<endl;}
    AccBalance(int balance){
        if(balance<0)
            cout<<"Has a Negative Balance";
        else if(balance==0)
            cout<<"Has a Zero Balance";
        else
            cout<<"Has a Positive Balance";
    }
};

int main()
{
    AccBalance defltBal;
    int balance;
    cin>>balance;
    AccBalance currBal(balance);
    return 0;
}
```



```
#include <iostream>

using namespace std;

class Country{

public:

Country(){cout<<"Country:INDIA"<<endl;}

Country(char statename[100],int area,int density)

{

    cout<<"State:"<<statename<<endl<<"Area:"<<area<<endl<<"Density:"<<density<<endl;

}

};

int main()

{

    Country country;

    char statename[100];

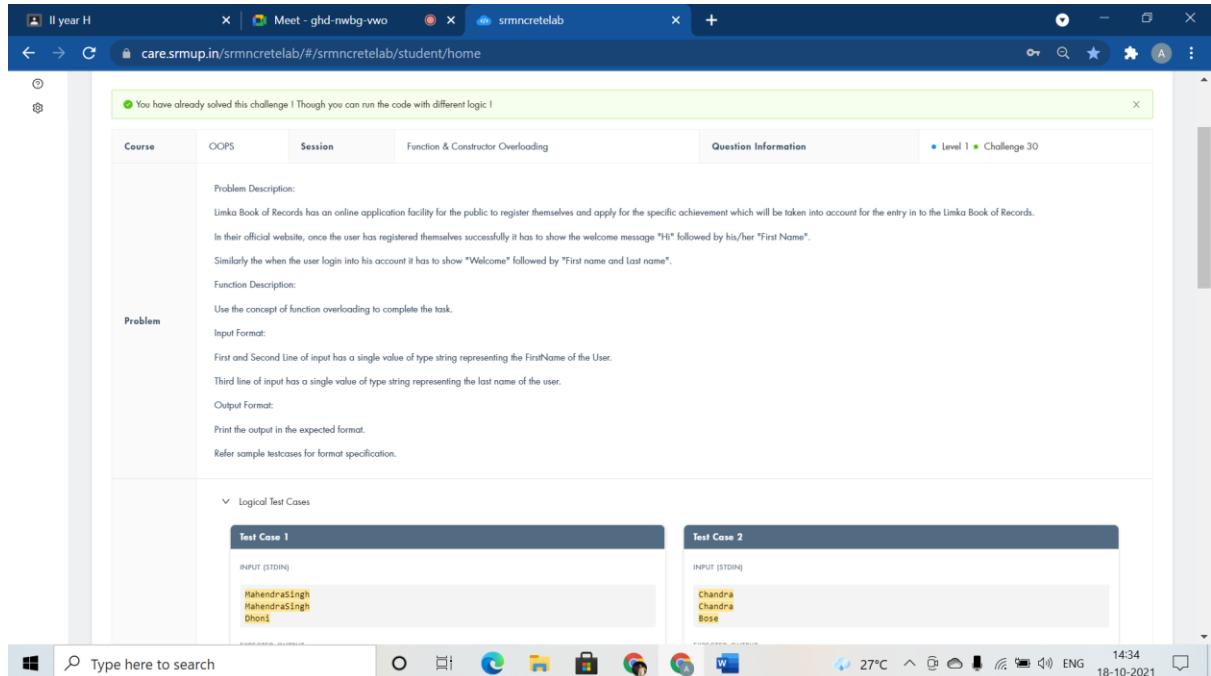
    int area,density;

    cin>>statename>>area>>density;

    Country statesofindia(statename,area,density);

    return 0;
}
```

```
}
```



```
#include <iostream>

using namespace std;

class Welcomemsg{

public:

void msg(string fname){

    cout<<"Hi "<<fname<<endl;

}

void msg(string fname,string lname){

    cout<<"Welcome "<<fname<<" "<<lname;

}

};

int main()

{



Welcomemsg ob;

string fname,lname;
```

```

    cin>>fname;
    ob.msg(fname);
    cin>>fname>>lname;
    ob.msg(fname,lname);
    return 0;
}

```

## Operator Overloading:-

You have already solved this challenge! Though you can run the code with different logic!

**Course**   **OOPS**   **Session**   **Operator Overloading**   **Question Information**   **Level 1**   **Challenge 31**

**Problem**

Question description  
The task is to overload the /operator to divide the fraction with other fraction.  
You can take the numerator as num and the denominator as deno.

Constraints  
 $1 \leq \text{num}, \text{deno} \leq 10^4$

Input Format  
First line represents the value of numerator and the denominator of first fraction separated by a space  
Second line represents the value of numerator and the denominator of second fraction separated by a space

Output Format  
print the answer like below if denominator is 1:  
Sum of Two Numbers : num  
Otherwise  
Sum of Two Numbers : num/deno

Note: If the denominator of any one of the input fractions is zero, then the error message "Error" will be displayed.

Logical Test Cases

**Test Case 1**   **Test Case 2**

INPUT (STDIN)

ENG IN   17:20   27-10-2021

```

#include <iostream>

using namespace std;

class Fraction{

public:

    int num,den;

    Fraction(int n=0, int d=0)

    {

        num=n;

        den=d;

    }

    Fraction operator /(Fraction const &obj){

```

```

Fraction res;

res.num=num * obj.den;
res.den=den * obj.num;
return res;
}

void display1(){
cout<<num/den;
}

void display2(){
cout<<num<<"/"<<den;
}

void display3(){
cout<<"Error";
}

int main()
{
int a,b,c,d;
cin>>a>>b;
cin>>c>>d;
Fraction ob1(a,b), ob2(c,d);
Fraction ob3 = ob1/ob2;
if(ob1.den==0 || ob2.den==0){
cout<<"Error";
return 0;
}
if(ob3.den==1)
ob3.display1();
else{
for(int i=2;i<50;i++)
{
if(ob3.num%i==0 && ob3.den%i==0)
{
ob3.num=ob3.num/i;
ob3.den=ob3.den/i;
}
}
}
}

```

```

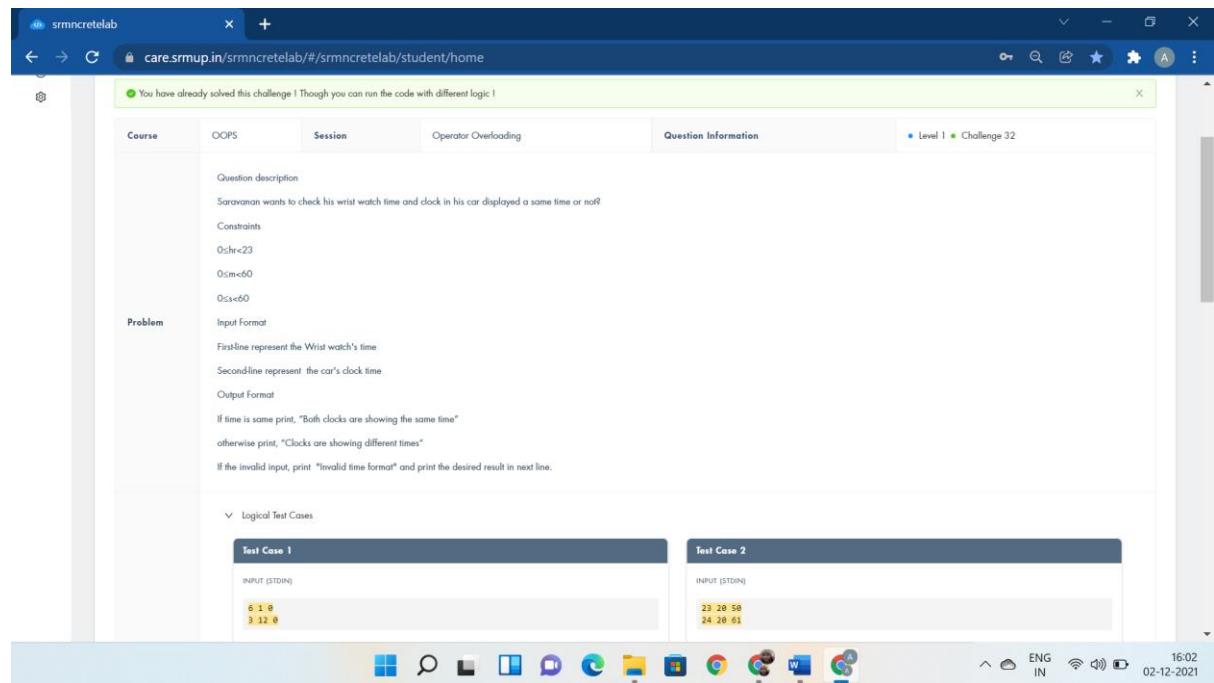
        }
    }

    ob3.display2();

}

return 0;
}

```



```

#include <iostream>

using namespace std;

class Time
{
    int h,m,s;

public:
    Time()
    {
        cin>>h>>m>>s;
    }

    void check()
    {
        if(h>23 || m>59 || s>59 )
    }
}

```

```
cout<<"Invalid time format\n";
}

bool operator==(Time t2);
};

bool Time::operator==(Time t2)
{
    if(h==t2.h && m==t2.m && s==t2.s)
        return true;
    else
        return false;
}

int main()
{
    Time t1,t2;
    t1.check();
    t2.check();
    if(t1==t2)
        cout<<"Both clocks are showing the same time";
    else
        cout<<"Clocks are showing different times";
    return 0;
}
```

You have already solved this challenge ! Though you can run the code with different logic !

**Course**      **OOPS**      **Session**      **Operator Overloading**      **Question Information**      **Level 1 • Challenge 33**

**Problem**

Question description  
Rahul and Ramesh are military officers. They are travelling to enjoy the vacation by train.  
They are planned to play a game during their travel that they are interested in how many ways there are in scrambling the letters.  
One fellow should said the length of the word added by 1 and other fellow should give the number of ways the letters to be scrambled.  
For example, If suppose Rahul gave the length of the word is 6. Then Ramesh should be subtracted that 1 and calculate for the word's length as 6-1. He have 5 choices for the first letter, once he have chosen the first letter there are 4 choices for the second letter, and then three choices for the third letter, two for the fourth letter, and only one choice for the last letter. Hence there are  $5[4][3][2][1] = 5 \times 120$  choices.

Can you help them to verify the answer?  
Constraints  
1≤n≤10  
Input Format  
The only line of input has one numbers n of type integer.  
Output Format  
Print the answer of the factorial of n-1.

Logical Test Cases

Test Case 1	Test Case 2
INPUT [STDIN] 6	INPUT [STDIN] 7
720	5040

```
#include <iostream>
```

```
using namespace std;
```

```
class Scrum{
```

```
public:
```

```
int n;
```

```
Scrum(int h)
```

```
{
```

```
n=h;
```

```
}
```

```
Scrum operator -- (int){
```

```
    Scrum T(int h);
```

```
    --n;
```

```
    return 1;
```

```
}
```

```
void display(){
```

```
    int res=1;
```

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

```
        res=res*i;
```

```
}
```

```
    cout<<res;
```

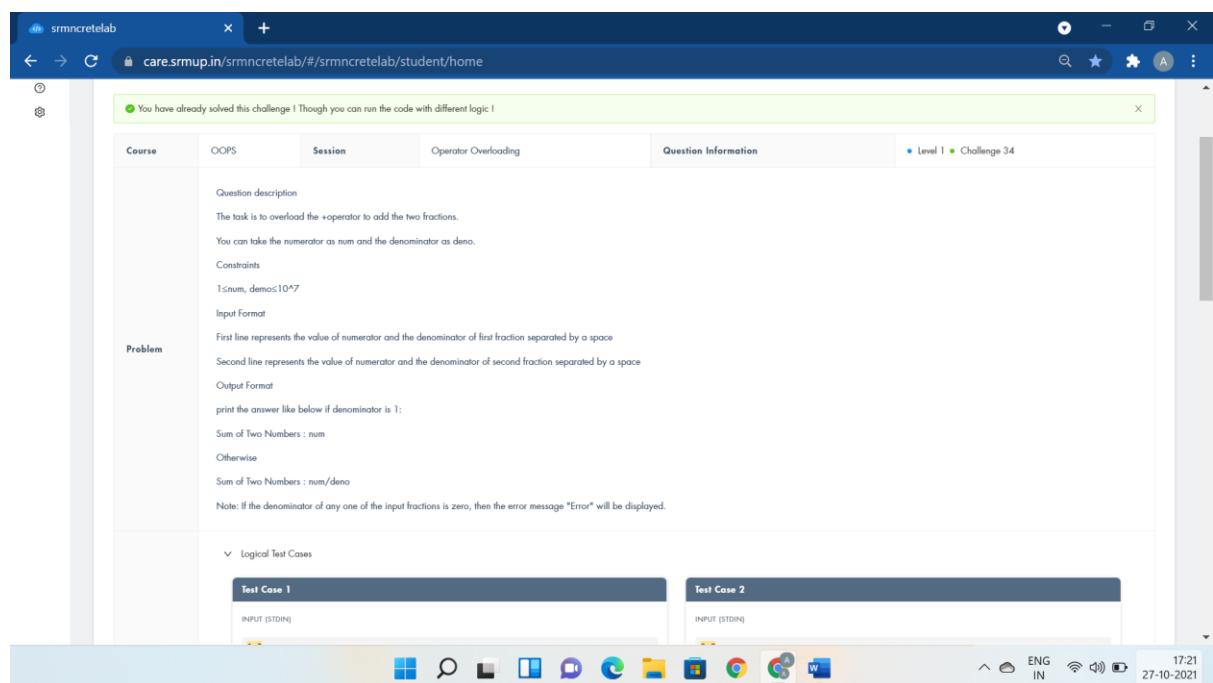
```
}
```

```
};
```

```

int main()
{
    int n;
    cin>>n;
    Scrum T(n);
    T--;
    T.display();
    return 0;
}

```



```

#include<iostream>

using namespace std;

class Fraction

{
public:
    int num,den;

    Fraction()
    {
        num=0;
        den=0;
    }
}

```

```
void getinput()
{
    cin>>num>>den;
}

Fraction operator +(Fraction obj)
{
    Fraction temp;
    temp.num=(num*obj.den)+(den*obj.num);
    temp.den=den*obj.den;
    return temp;
}

int main()
{
    Fraction f1,f2,add;
    f1.getinput();
    f2.getinput();
    add=f1+f2;
    if(add.den==0)
        cout<<"Error";
    else if(add.num%add.den == 0)
        cout<<add.num/add.den;
    else
        cout<<add.num<<"/"<<add.den;
    return 0;
}
```

You have already solved this challenge! Though you can run the code with different logic!

**Course:** OOPS    **Session:** Operator Overloading    **Question Information:** Level 1 | Challenge 35

**Problem:**

**Input Format:**

First line represent the first row elements of given matrix  
Second line represent the second row elements of given matrix

**Output Format:**

Print the determinant of a given matrix

**Logical Test Cases:**

Test Case 1	Test Case 2
INPUT (STDIN) 1 2 3 4	INPUT (STDIN) 2 0 1 2
EXPECTED OUTPUT	EXPECTED OUTPUT

```
#include <iostream>
```

```
using namespace std;
```

```
class matrix{
```

```
public:
```

```
int operator ~(){
```

```
    int a,b,c,d;
```

```
    cin>>a>>b>>c>>d;
```

```
    return a*d-b*c;
```

```
}
```

```
};
```

```
int main()
```

```
{
```

```
    matrix t;
```

```
    cout<<~t;
```

```
    return 0;
```

```
}
```

```
#include<iostream>

using namespace std;

class Complex {

private:
    int real, imag;

public:
    Complex(int r = 0, int i = 0) {real = r; imag = i;}

    Complex operator+(int a) {
        Complex res;
        res.real = real + a;
        res.imag = imag;
        return res;
    }

    Complex operator+(Complex obj) {
        Complex res;
        res.real = real + obj.real;
        res.imag = imag + obj.imag;
        return res;
    }

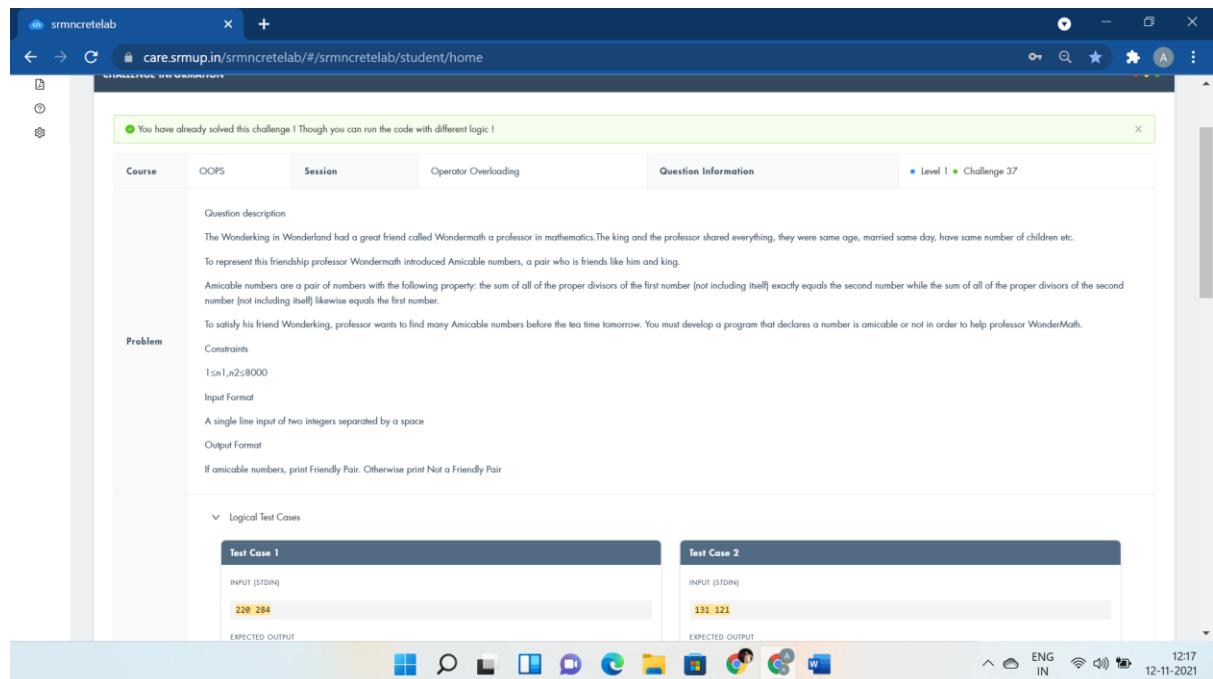
    void print() { cout << real << " + " << imag << "i" << endl; }

};
```

```

int main()
{
    int a,b,c;
    cin>>a>>b>>c;
    Complex i1(a, b);
    Complex i2 = i1 + c;
    i1.print();
    i2.print();
    (i1+i2).print();
}

```



```

#include<iostream>

using namespace std;

class compare{

public:

    int first,sum1=0;

    compare(int x){

        first=x;

    }

    void f(){

        //first1=first;
    }
}

```

```

for(int i=1; i<=first/2 ; i++)
{
    //finding and adding divisors of first number
    if(first%i==0)
        sum1=sum1+i;
}

void operator ==(compare t2){

    if(first==t2.sum1 && t2.first==sum1)
        cout<<"Friendly Pair";
    else
        cout<<"Not a Friendly Pair";
}

//main program

int main()
{
    int first,second;
    //user input
    cin>>first;
    //user input
    cin>>second;
    compare t1(first),t2(second);
    t1.f();
    t2.f();
    t1==t2;
    return 0;
}

```

```
#include <iostream>

using namespace std;

class Diff{

public:

int n;

void getdata(){

    cin>>n;

}

int sumofsquare();

int sumofnumsq(){

    return n*(n+1)*(2*n+1)/6;

}

};

int Diff :: sumofsquare(){

    return n*n*(n+1)*(n+1)/4;

}

int main()

{

    Diff n;

    if(0)

        cout<<"friend void operator >> (istream &in, Diff &obj )";
```

```

n.getdata();

//int sq=n*n*(n+1)*(n+1)/4;

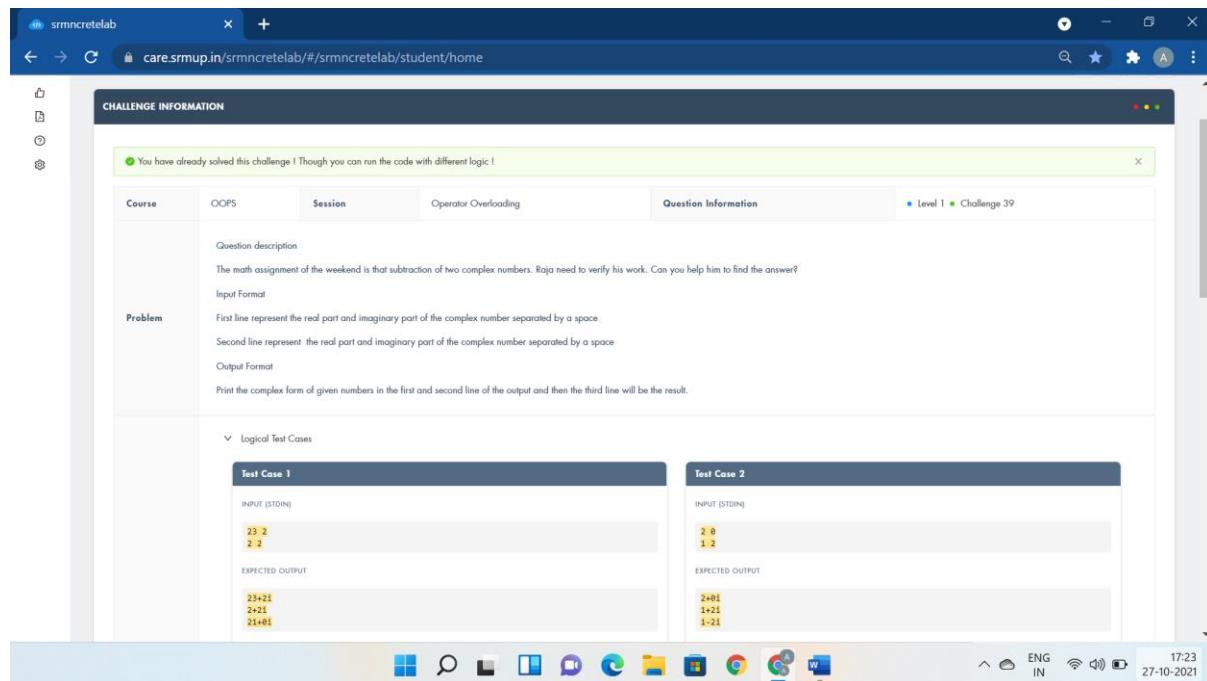
//int sq2=n*(n+1)*(2*n+1)/6;

cout<<n.sumofsquare()-n.sumofnumsq();

return 0;

}

```



```

#include <iostream>

using namespace std;

class complex

{
private:
    float real;
    float imag;

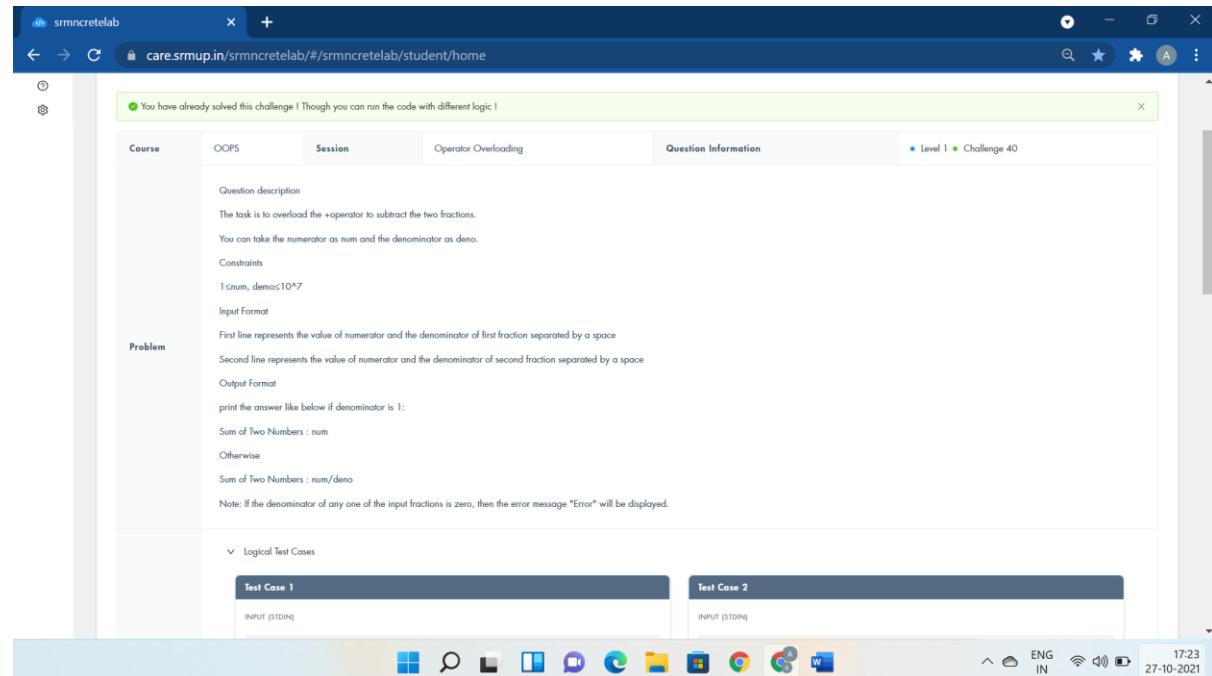
public:
    complex() {cin>>real>>imag;}
    complex operator-(complex ob)
    {
        complex t;
        t.real = real - ob.real;
        t.imag = imag - ob.imag;
        return t;
    }
}

```

```
}
```

```
void output()
{
    if(imag < 0)
        cout<< real << imag << "i" << endl;
    else
        cout<< real << "+" << imag << "i" << endl;
}
};

int main()
{
    complex c1, c2;
    c1.output();
    c2.output();
    (c1 - c2).output();
    return 0;
}
```



```
#include<iostream>

using namespace std;

class Fraction
```

```

{
public:
int num,den;

Fraction()
{
    num=0;
    den=0;
}

void getinput()
{
    cin>>num>>den;
}

Fraction operator -(Fraction obj)
{
    Fraction temp;
    temp.num=(num*obj.den)-(den*obj.num);
    temp.den=den*obj.den;
    return temp;
};

int main()
{
    Fraction f1,f2,add;
    f1.getinput();
    f2.getinput();
    add=f1-f2;
    if(add.den==0)
        cout<<"Error";
    else if(add.num%add.den == 0)
        cout<<add.num/add.den;
    else
        cout<<add.num<<"/"<<add.den;
    return 0;
}

```

# Inheritance:-

The screenshot shows a web browser window with a yellow header bar containing the text "Inheritance:-". The main content area is a challenge page from "care.srmup.in/srmncretelab/#/srmncretelab/student/home". The page has tabs for "Course", "OOOPS", "Session", "Inheritance" (which is selected), and "Question Information". The "Question Information" tab shows "Level 1" and "Challenge 41". The challenge description is as follows:

You have already solved this challenge ! Though you can run the code with different logic !

**Question description:**  
Fazil is running a typewriting practice classes for students.  
He trains the students and conducts frequent assessments for each of them.  
Subsequently the students performing well will be awarded the certificate of completion.  
Recently he conducted one such assessment and many students attended the assessment.  
Now he is processing the result to prepare the certificate for the ones qualified.  
Since the number of student attended the exam is huge he is looking for the automated program which provides the details of the students and their typing speed in a format expected by him for certificate preparation.

**Can you help him?**

**Input Format:**  
First line of input has a single value of type string representing the name of the Typist.  
Second line of input has a single value of type integer representing the code of the Typist  
Third line of input has a single value of type integer representing the speed of the Typist.

**Constraints:**  
100≤speed<1000  
100≤code<1000

**Output Format:**  
Print the details for the typist in the expected format  
Refer sample testcases for format specification.

The browser taskbar at the bottom shows the Windows Start button, a search bar with "Type here to search", and various pinned application icons. The system tray indicates it's 27°C, 14:35, and the date is 18-10-2021.

```
#include <iostream>

using namespace std;

class staff{
public:
    int code,speed;
    string name;
    void getdata();
    void display();
};

void staff::getdata(){
    cin>>name>>code>>speed;
}

void staff::display(){
    cout<<"Name:"<<name<<endl<<"Code:"<<code<<endl<<"Speed"<<speed;
}

class typist: public staff{
public:
```

```

void getdata();

void display();

};

void typist::getdata(){

    cin>>name>>code>>speed;

}

void typist::display(){

    cout<<"Name:"<<name<<endl<<"Code:"<<code<<endl<<"Speed:"<<speed;

}

int main()

{

typist t;

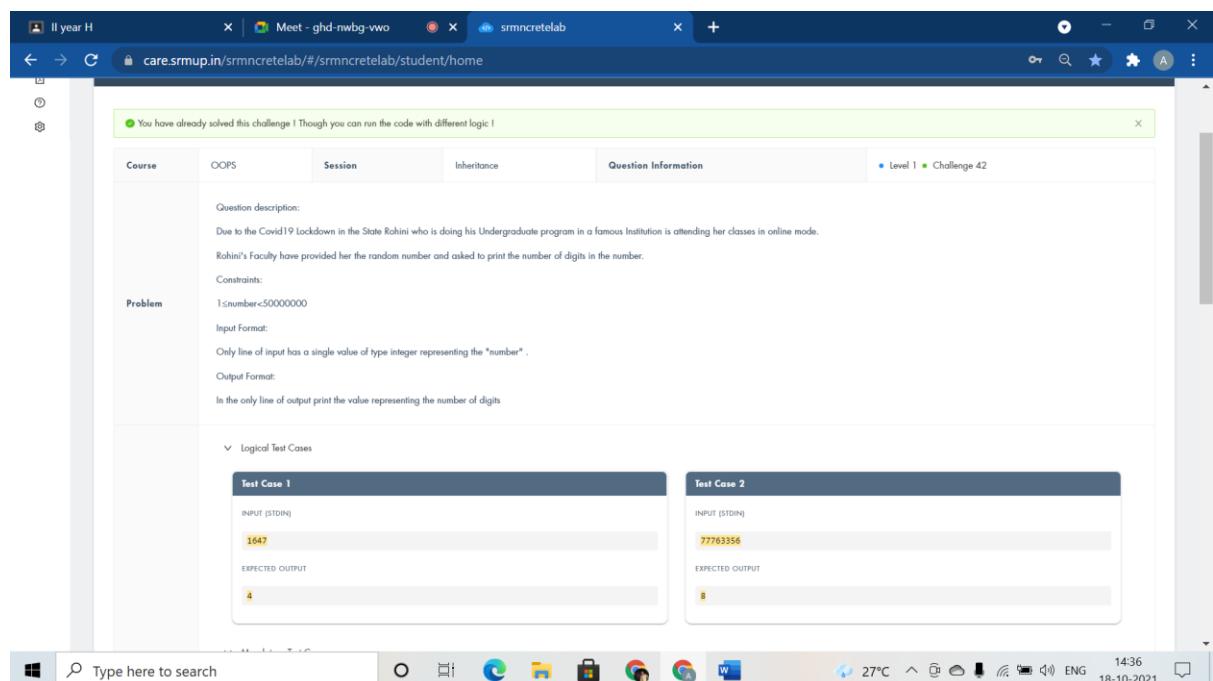
t.getdata();

t.display();

return 0;

}

```



```
#include <iostream>
```

```
using namespace std;

class Assignement{

public:
    int num;

    void get(){
        cin>>num;
    }

    void display(){
        int count=0;

        while(num!=0){
            count++;
            num/=10;
        }

        cout<<count;
    }
};

class Student:public Assignement{
};

int main()
{
    Student obj;

    obj.get();
    obj.display();

    return 0;
}
```

You have already solved this challenge ! Though you can run the code with different logic !

**Course**    **OOPS**    **Session**    **Inheritance**    **Question Information**    **Level 1**    **Challenge 43**

**Question Description:**  
Analia is developing an application to help customers who come to her supermarkets such as the price of the item that customers buy and display each item's price, the subtotal of the sale, the amount of sales tax, and the total.  
Assume the sales tax is 6%. So you have help to Analia holds the prices of the five items in five variables.

**Constraints:**  
1≤itemOne≤100000  
1≤itemTwo≤100000  
1≤Price of itemThree ≤100000  
1≤Price of item Four ≤100000  
1≤Price of itemFive≤100000

**Problem**  
**Input Format:**  
First line of input has a single value of type integer representing Price of itemOne.  
Second line of input has a single value of type integer representing Price of itemTwo.  
Third line of input has a single value of type integer representing Price of itemThree.  
Fourth line of input has a single value of type integer representing Price of item Four.  
Fifth line of input has a single value of type integer representing Price of itemFive.  
**Output Format:**  
Print the result as per format.  
Refer sample testcases for format specification.

```
#include <iostream>

using namespace std;

class market{

public:

float i1,i2,i3,i4,i5;

float Subtotal,tax;

void items(){

    cin>>i1>>i2>>i3>>i4>>i5;

}

void buy(){

    Subtotal=(i1+i2+i3+i4+i5);

    cout<<"Subtotal="$<<Subtotal<<endl;

    tax=0.06*i1+0.06*i2+0.06*i3+0.06*i4+0.06*i5;

    cout<<"Tax="$<<tax<<endl;

    cout<<"Total="$<<Subtotal+tax;

}

};

class customer:public market{
```

```

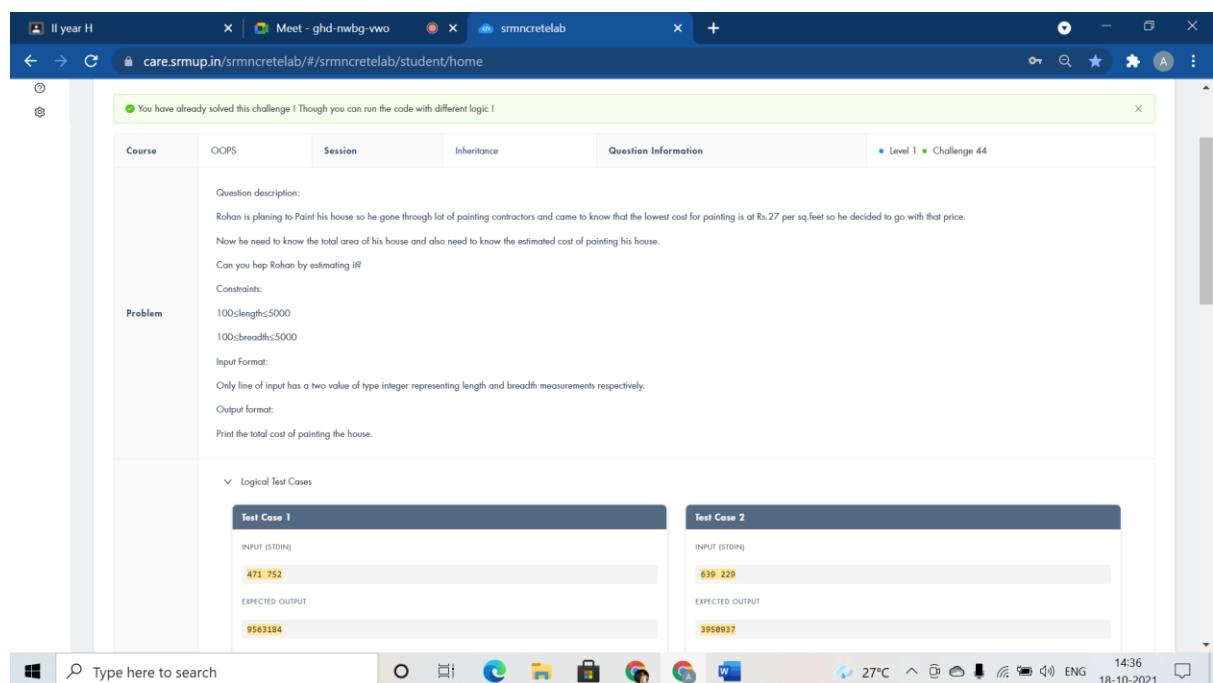
};

int main()
{
    customer c;

    c.items();

    c.buy();
}

```



```

#include <iostream>

using namespace std;

class ReceiveMesurement{

public:

    int l,b;

    void painingarea(){

        cin>>l>>b;

        cout<<l*b*27;
    }
}

```

```

};

class CalculateArea : public ReceiveMesurement{

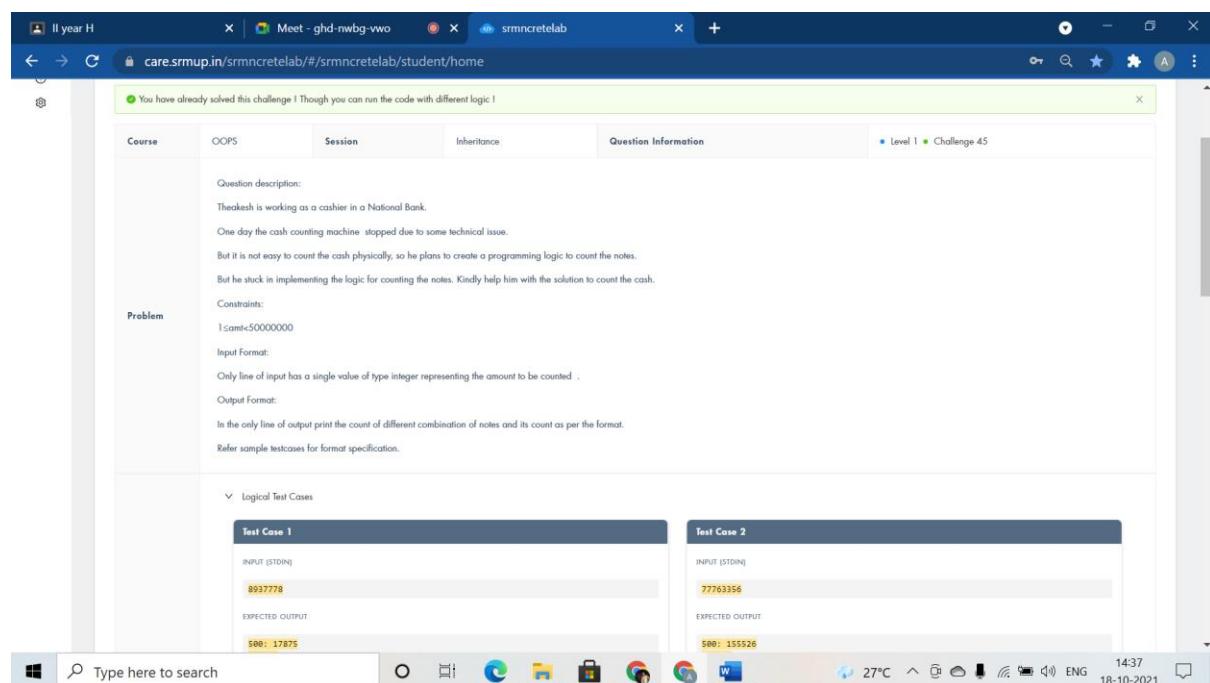
};

int main()
{
    CalculateArea mt;

    mt.painingarea();

    return 0;
}

```



```

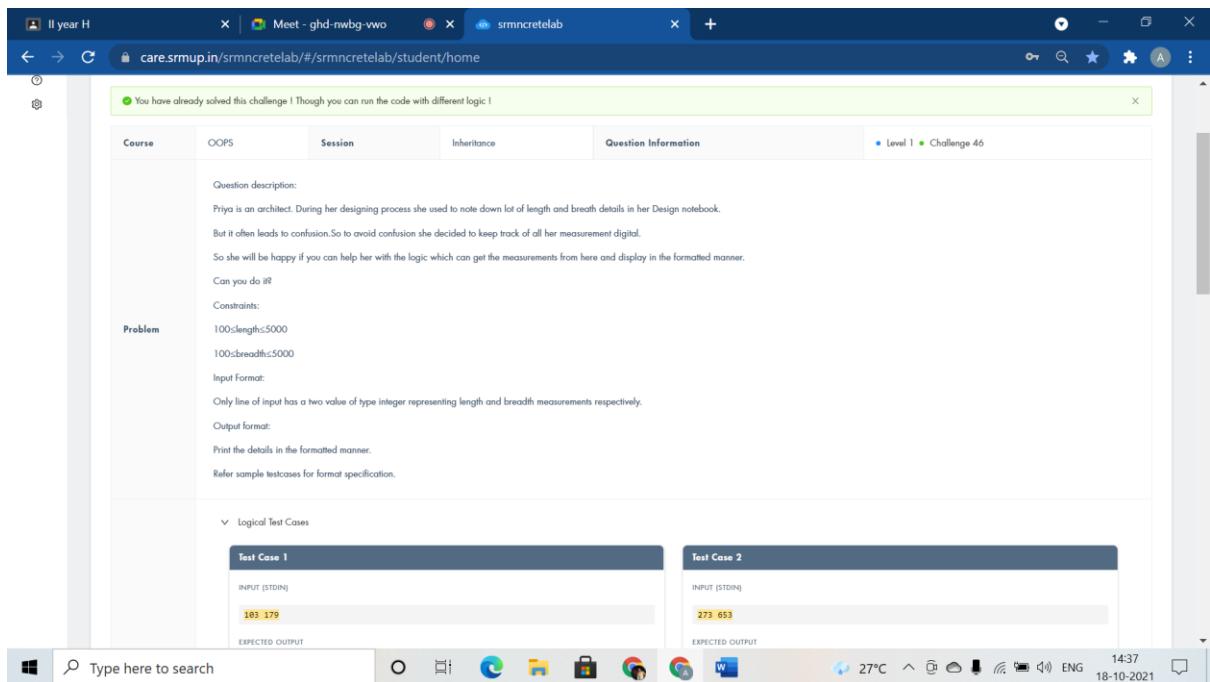
#include <iostream>

using namespace std;

class Bank{
public:
    int n;
    void get(){
        cin>>n;
    }
    void display(){
        cout<<"500: "<<n/500<<endl;
    }
}

```

```
n=n%500;  
cout<<"200: "<<n/200<<endl;  
n=n%200;  
cout<<"100: "<<n/100<<endl;  
n=n%100;  
cout<<"50: "<<n/50<<endl;  
n=n%50;  
cout<<"10: "<<n/10<<endl;  
n=n%10;  
cout<<"5: "<<n/5<<endl;  
n=n%5;  
cout<<"1: "<<n<<endl;  
}  
};  
class CashCounting:public Bank{  
};  
int main()  
{  
    CashCounting obj;  
    obj.get();  
    obj.display();  
    return 0;  
}
```



```
#include <iostream>

using namespace std;

class ReceiveMesurement{

public:

int l,b;

void display(){

cin>>l>>b;

cout<<"Length:"<<l<<" metres"<<endl;

cout<<"Breadth:"<<b<<" metres";

}

};

class FormatMesurement : public ReceiveMesurement{

};

int main()

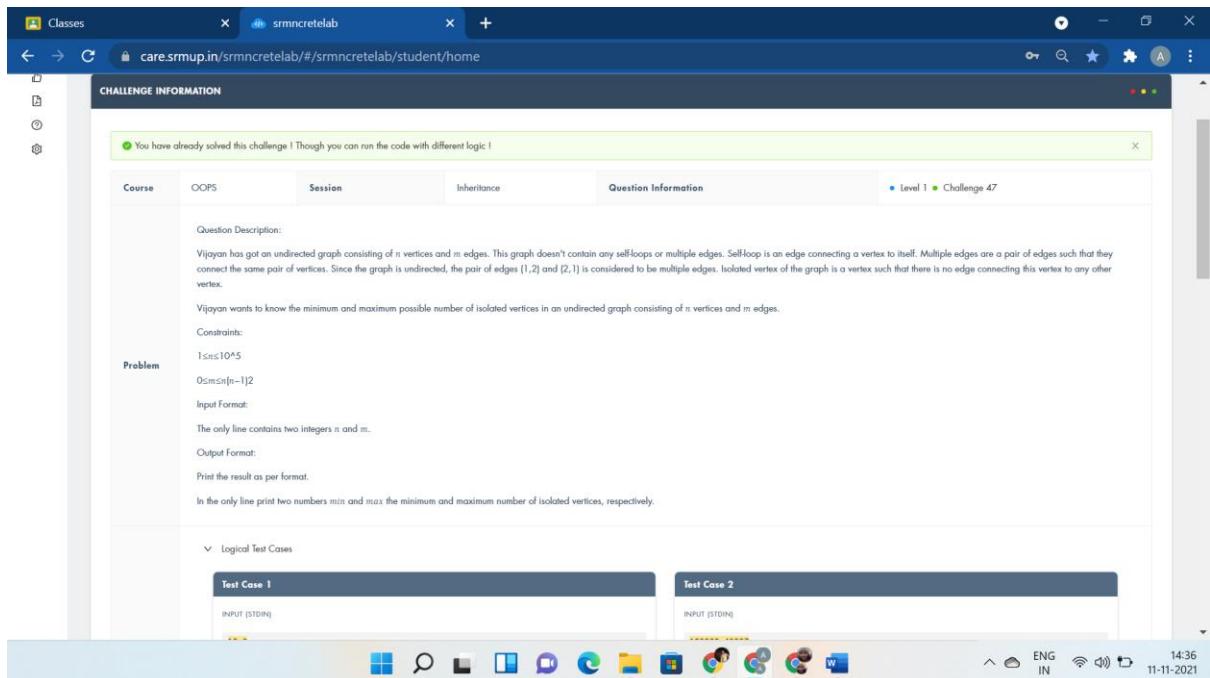
{

FormatMesurement mt;

mt.display();

return 0;
}
```

}



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

using namespace std;

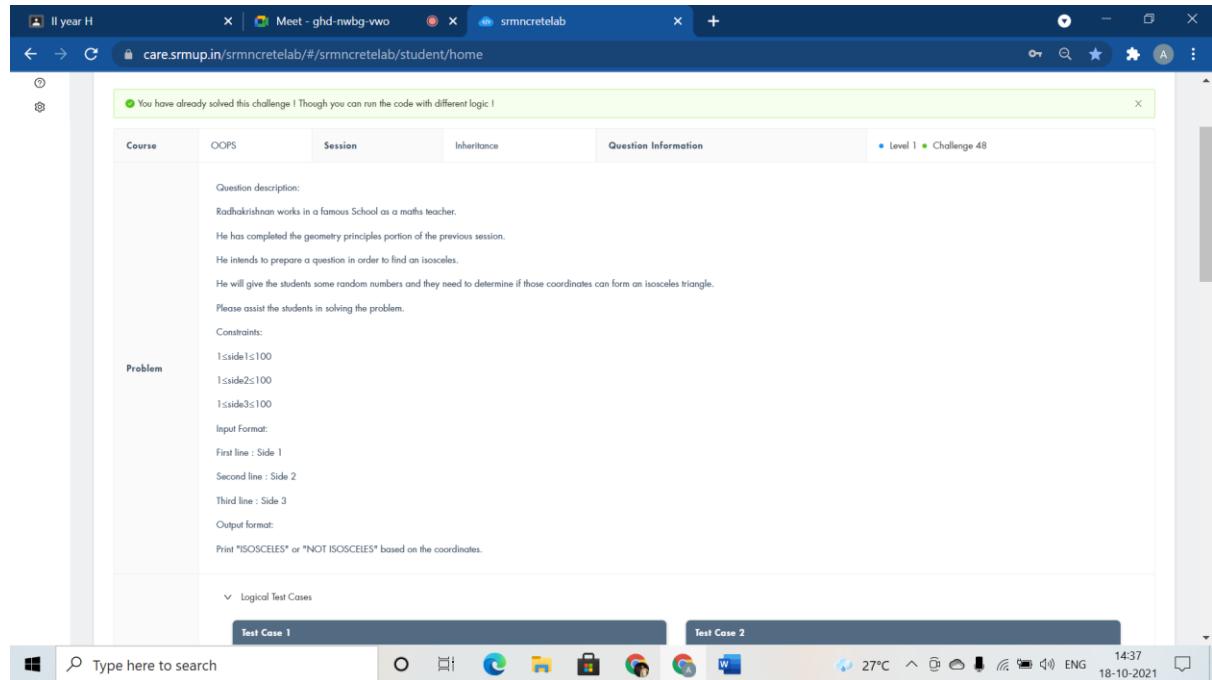
class graph{
public:
    void edge(){}
};

class pairs:public graph{
public:
    long long int n,m,k=0;
    void vertex(){}
    cin>>n>>m;
    cout<<max(0ll,n-2*m)<<" ";
    while(k*(k-1)/2<m) k++;
    cout<<n-k<<endl;
}
};
```

```

int main()
{
    pairs pa;
    pa.edge();
    pa.vertex();
    return 0;
}

```



```

#include <iostream>

using namespace std;

class triangle{
public:
    int a,b,c;
    void read(){
        cin>>a>>b>>c;
    }
    void check(){
        if(a==b || b==c || a==c)

```

```

cout<<"ISOSCELES";
else
cout<<"NOT ISOSCELES";
}

};

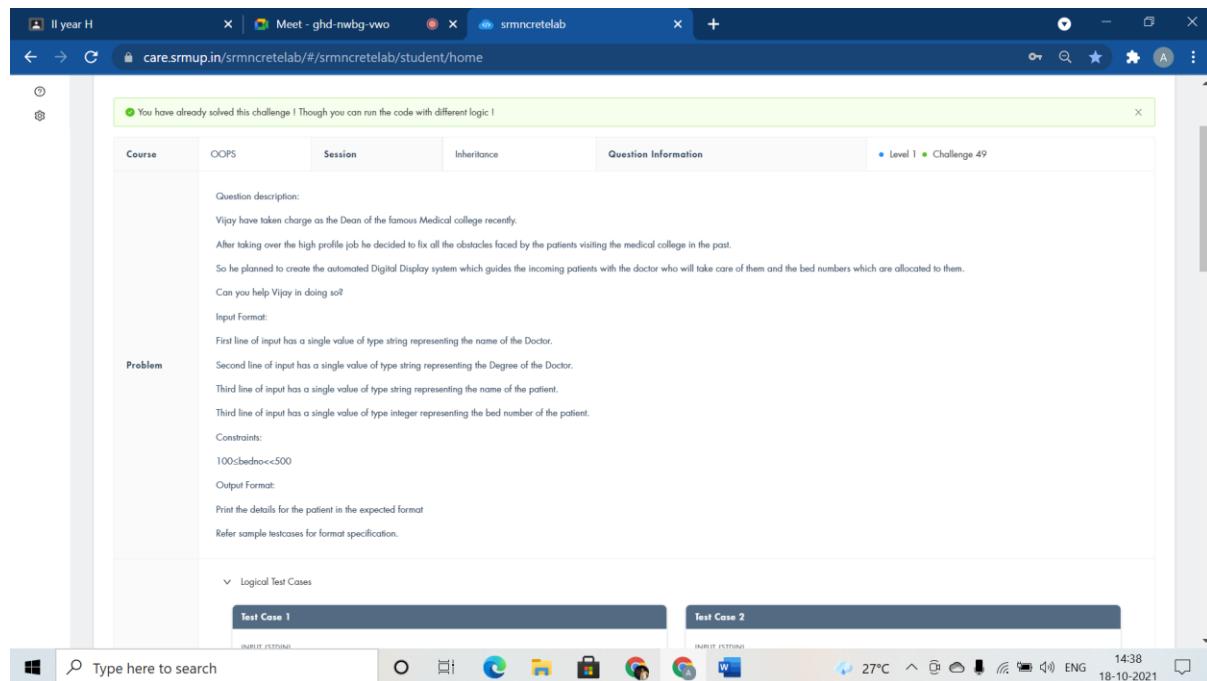
class isosceles : public triangle {

};

int main()
{
    isosceles obj;
    obj.read();
    obj.check();

    return 0;
}

```



```
#include <iostream>
```

```
using namespace std;
```

```
class doctor{
```

```
public:
```

```

string name,degree,pname;
int no;

void getedu(){
    cin>>name>>degree>>pname;
}

void getdata(){
    cin>>no;
}

void dispedu(){
    cout<<"Doctor Name:"<<name<<endl<<"Doctorate Degree:"<<degree<<endl<<"Patient
Name:"<<pname<<endl;
}

void dispdata(){
    cout<<"Bed Number:"<<no;
}

};

class patient:public doctor{
};

int main()
{
    patient p;
    p.getedu();
    p.getdata();
    p.dispedu();
    p.dispdata();

    return 0;
}

```

You have already solved this challenge! Though you can run the code with different logic!

**Course:** OOPS    **Session:** Inheritance    **Question Information:** Level 1 • Challenge 50

**Question Description:**  
Dayalan is a newly appointed lecturer of a government college in Sengipatti village near Thanjavur city. He is unhappy with the education system and is also worried about the pitiable condition of education of government colleges. After joining the college, he tries to change the college student environment. Dayalan's decision for the change does not go well with the other teachers and students. Slowly, Dayalan gets popular among the class students. One day Dayalan tells his students to use programming and multiplication table 10,3,8,7 based on the user choice concept.  
Option as follows 1 for 10 tables. 2 for three tables. 3 for eight tables. 4 for seven table

**Constraints:**  
1 Options: 4

**Input Format:**  
The first line of input has a single value of type integer representing option.

**Output Format:**  
Print the result as per format.  
Refer sample test cases for format specification.

**Logical Test Cases:**

Test Case 1	Test Case 2
INPUT [STDIN] 1	INPUT [STDIN] 4
EXPECTED OUTPUT 10	EXPECTED OUTPUT 8

```
#include <iostream>
```

```
using namespace std;
```

```
class teacher{
```

```
public:
```

```
int num;
```

```
void setdata(int n)
```

```
{
```

```
if(n==1)
```

```
num=10;
```

```
else
```

```
num=7;
```

```
}
```

```
void setdata2(int n)
```

```
{
```

```
if(n==2)
```

```
num=3;
```

```
else
```

```
num=8;
```

```

}

void tentable(){

    for(int i=1;i<=10;i++)
        cout<<num<<"*"<<i<<"="<<num*i<<endl;
}

};

class ten:public teacher{

};

class three:public teacher{

};

class eight:public teacher{

};

class seven:public teacher{

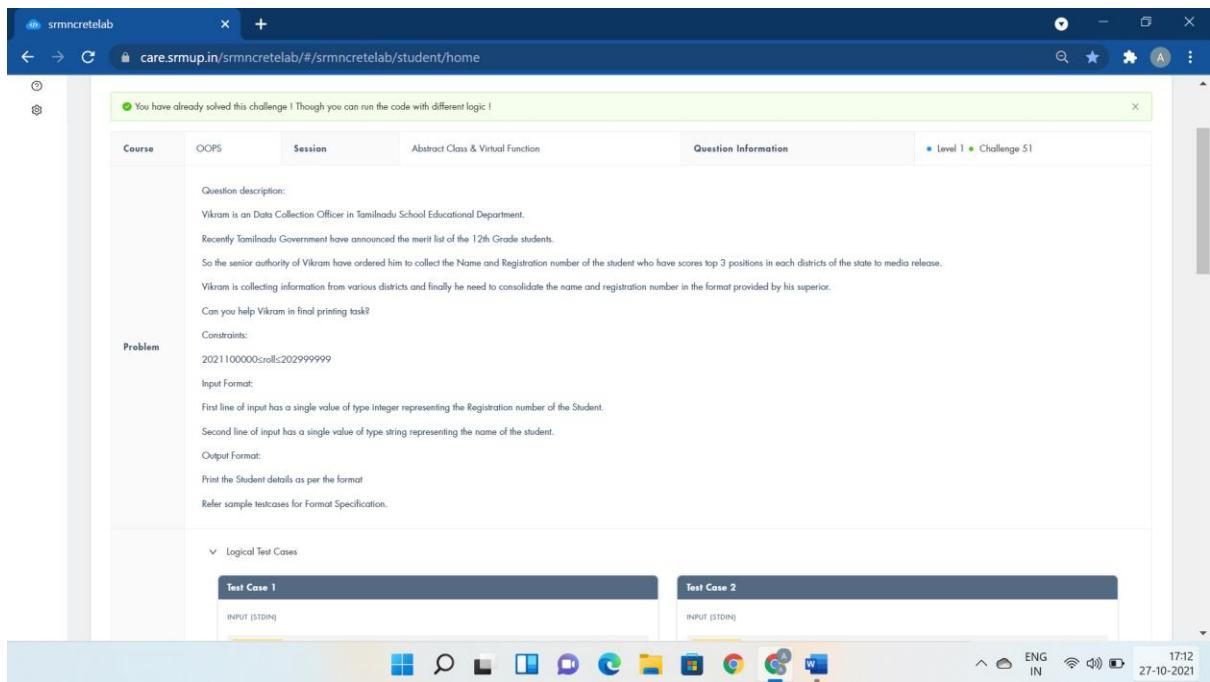
};

int main()

{
    int n;
    cin>>n;
    teacher t;
    if(n==1 || n==4)
        t.setdata(n);
    if(n==2 || n==3)
        t.setdata2(n);
    t.tentable();
    return 0;
}

```

## Abstract Class and Virtual Functions:-



```
#include <iostream>

using namespace std;

class School{

public:

    int roll;

    string name;

    virtual void getdata(){};

    virtual void display(){};

};

class District : public School{

    void getdata();

    void display();

};

void District :: getdata(){

    cin>>roll>>name;

}

void District :: display(){

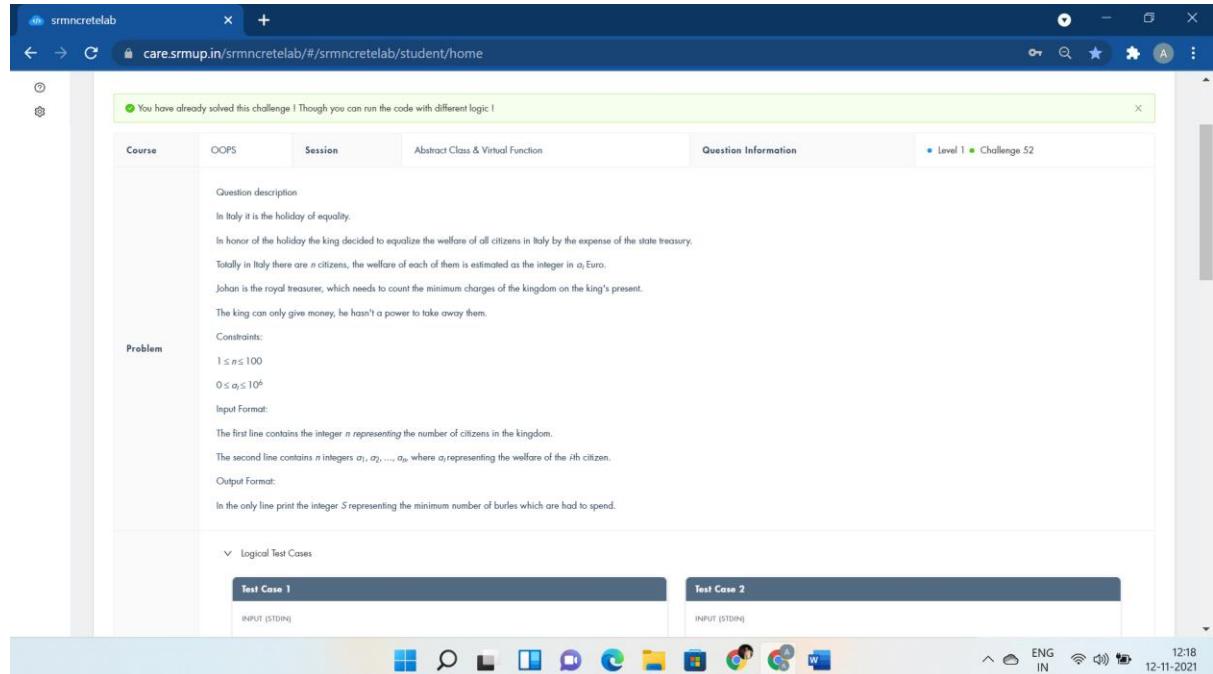
    cout<<"Student Name is: "<<name<<endl<<"Student Roll no is: "<<roll;

}
```

```

int main()
{
    District obj;
    School* ptr;
    ptr = &obj;
    ptr -> getdata();
    ptr -> display();
    return 0;
}

```



```

#include <bits/stdc++.h>

using namespace std;

int a,b,c,d,i;

class Holiday{

public:virtual void Expenses()=0;

};

class Citizen:public Holiday{

public:

void Expenses(){

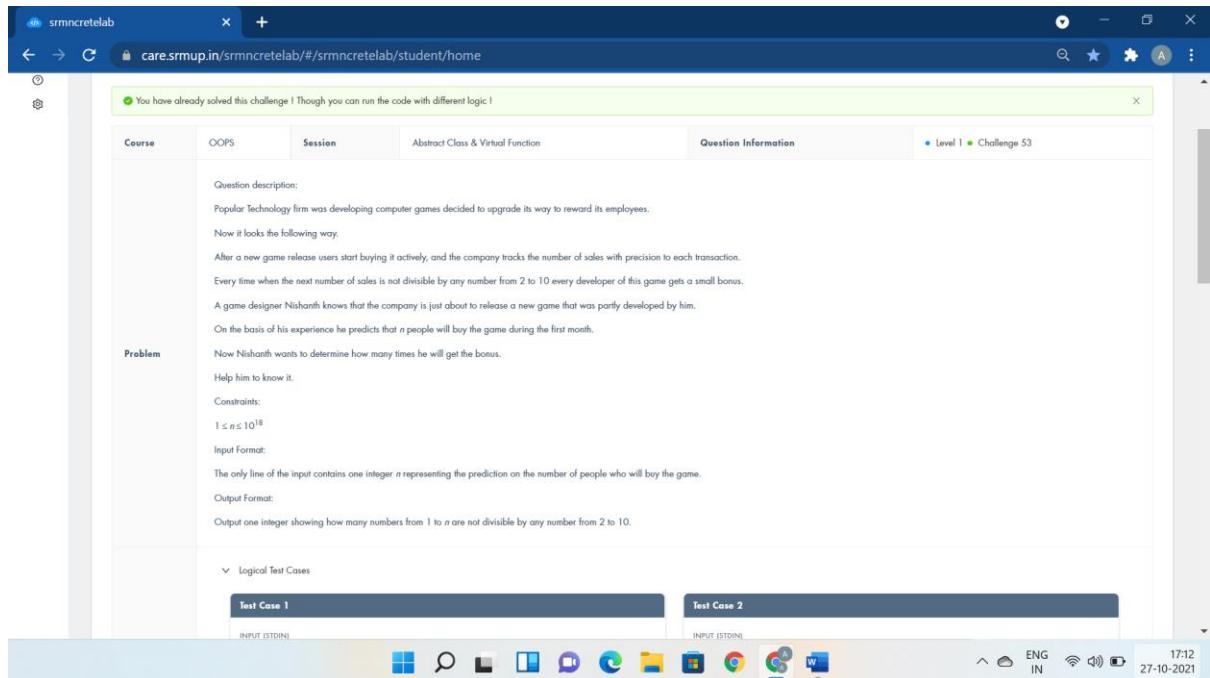

```

```

cin>>c;
for (i=0; i<c; i++){
    cin>>a;
    if (d<a) d=a;
    b=b+a;
}
cout<<d*c-b;
};

int main (){
Citizen obj;
obj.Expenses();
return 0;
}

```



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

```
using namespace std;

class Employees{

public:virtual void BuyingGame()=0;
};

class Reward:public Employees{

public:
    int n;

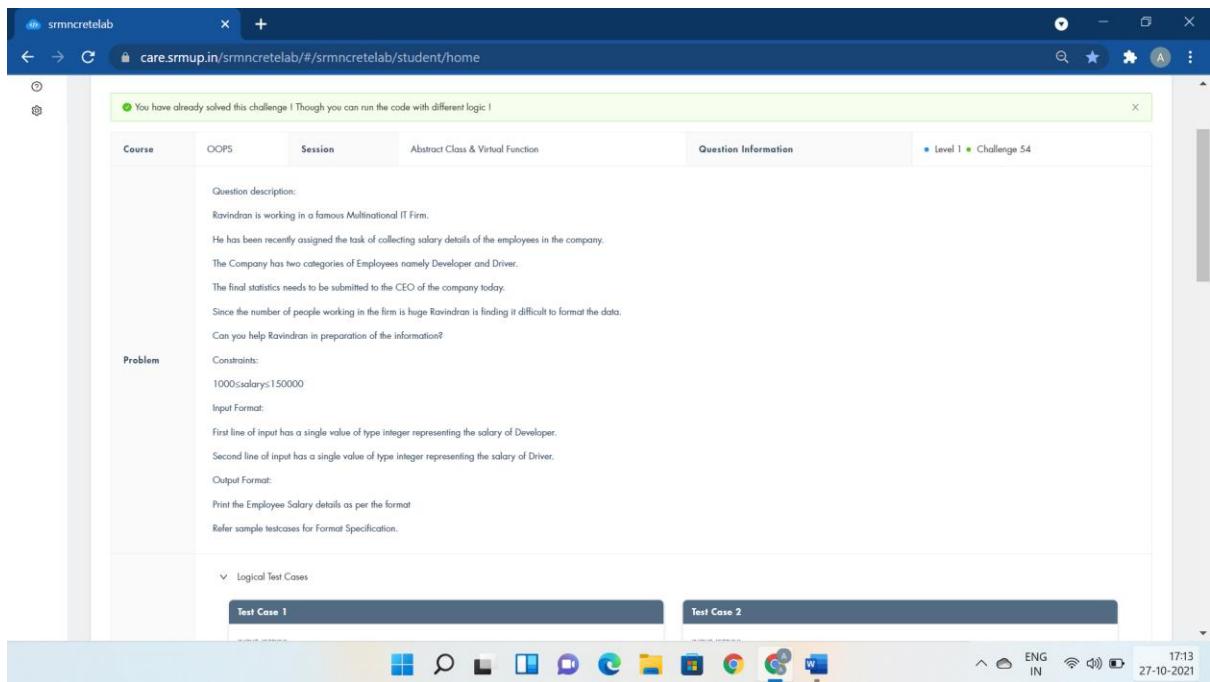
void BuyingGame(){

    cin>>n;
    cout<<n - n / 2 - n / 3 - n / 5 - n / 7
        + n / 6 + n / 10 + n / 14 + n / 15 + n / 21 + n / 35
        - n / 30 - n / 42 - n / 70 - n / 105 + n / 210;

}

};

int main()
{
    Reward obj;
    obj.BuyingGame();
    return 0;
}
```



```
#include <iostream>

using namespace std;

class Employee{
public:
    int s1,s2;
};

class Developer : public Employee{
public:
    void getSalary(){
        cin>>s1;
        cout<<"Salary of Developer:"<<s1<<endl;
    }
};

class Driver : public Employee{
public:
    void getSalary(){
        cin>>s2;
        cout<<"Salary of Driver:"<<s2<<endl;
    }
};
```

```

};

int main()
{
    Developer d1;

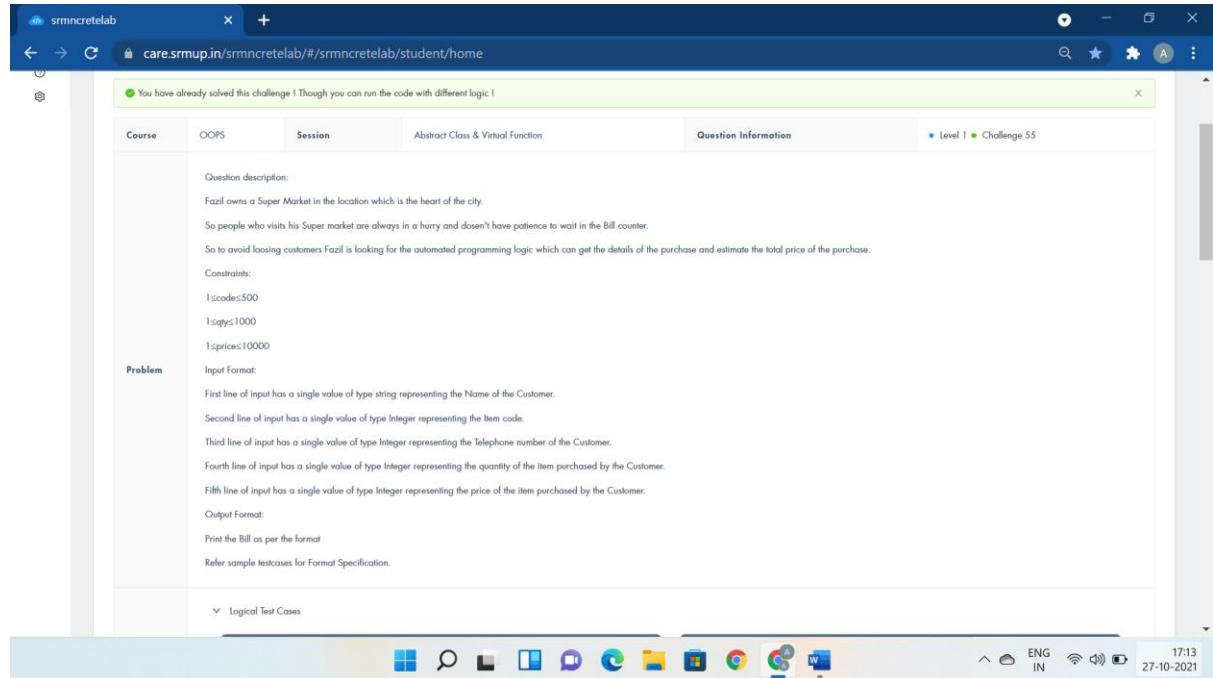
    Driver d2;

    d1.getSalary();

    d2.getSalary();

    return 0;
}

```



```

#include <iostream>

using namespace std;

class consumer{

public:

    string name;

    virtual void getdata()=0;

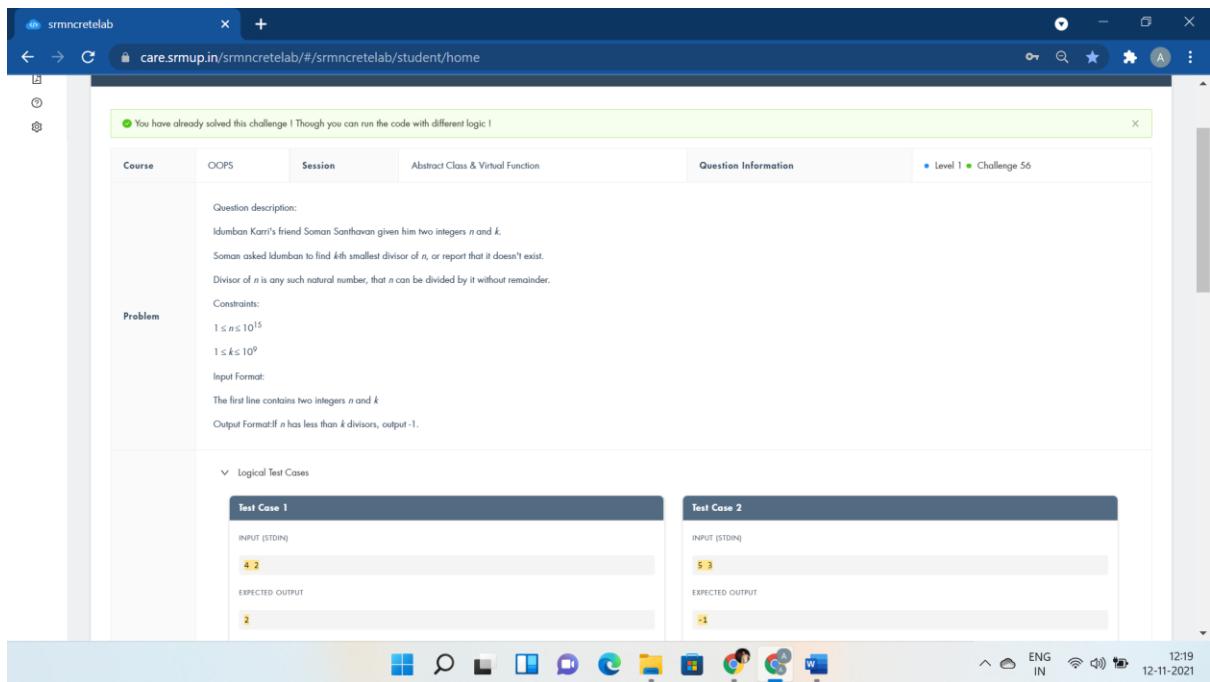
    virtual void display()=0;

};

class transaction: public consumer{

```

```
public:  
int code;  
long tel;  
int quan,price;  
void getdata(){  
    cin>>name>>code;  
    cin>>tel;  
    cin>>quan;  
    cin>>price;  
}  
void display(){  
    cout<<"Name : "<<name<<endl<<"Code : "<<code<<endl<<"Telephone : "<<tel<<endl;  
    cout<<"Quantity : "<<quan<<endl<<"Price : "<<price<<endl<<"Total Price : "  
    "<<quan*price<<endl;  
}  
};  
int main()  
{  
    consumer* o1;  
    transaction o2;  
    o1=&o2;  
    o1->getdata();  
    o1->display();  
    return 0;  
}
```



```
#include<iostream>

using namespace std;

class Problem {

public:virtual void Divisor()=0;
};

class Calculation:public Problem{

public:

    int n,k,i;

    void Divisor(){

        cin>>n>>k;

    }

    int Display()

    {

        int count;

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

        {

            if(n%i==0)

            {

                count++;

            }

        }

    }

}
```

```

        if(count==k){

            cout<<i;

            return 1;

        }

    }

    cout<<-1;

    return 1;

}

};

int main()

{

    Calculation obj;

    obj.Divisor();

    obj.Display();

    return 0;

}

```

You have already solved this challenge! Though you can run the code with different logic!

Course	OOPS	Session	Abstract Class & Virtual Function	Question Information	Level 1	Challenge 57

**Question description:**  
Young Varun has a birthday today! He got kit of  $n$  cubes as a birthday present from his parents. Every cube has a number  $a_i$  which is written on it.  
Varun put all the cubes in a row and went to unpack other presents.  
In this time, Varun's elder brother, Saran reordered the cubes using the following rule. Suppose the cubes are numbered from 1 to  $n$  in their order.  
Saran performs several steps, on step  $i$  he reverses the segment of cubes from  $i$ th to  $(n-i+1)$ th. He does this while  $i \leq n-i+1$ .  
After performing the operations Saran went away, being very proud of himself.  
When Varun returned to his cubes, he understood that their order was changed.

**Help Varun as fast as you can and save the holiday — restore the initial order of the cubes using information of their current location.**

**Problem**

**Constraints:**  
 $1 \leq n \leq 2 \cdot 10^5$   
 $-10^9 \leq a_i \leq 10^9$

**Input Format:**  
The first line contains single integer  $n$  representing the number of cubes.  
The second line contains  $n$  integers  $a_1, a_2, \dots, a_n$ , where  $a_i$  is the number written on the  $i$ th cube after Saran has changed their order.

**Output Format:**  
Print  $n$  integers, separated by spaces — the numbers written on the cubes in their initial order.

Logical Test Cases

Test Case 1    Test Case 2

17:14  
27-10-2021

#include <iostream>

```
using namespace std;

class Gift {

public:virtual void Cubes()=0;

};

class Birthday:public Gift{

public:

int a[10],n;

void Cubes(){

cin>>n;

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

cin>>a[i];

for(int i=0;i<n/2;i+=2)

/*int temp=a[i];

a[i]=a[n-i-1];

a[n-i-1]=temp;*/

swap(a[i],a[n-i-1]);

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

cout<<a[i]<<" ";

}

};

int main()

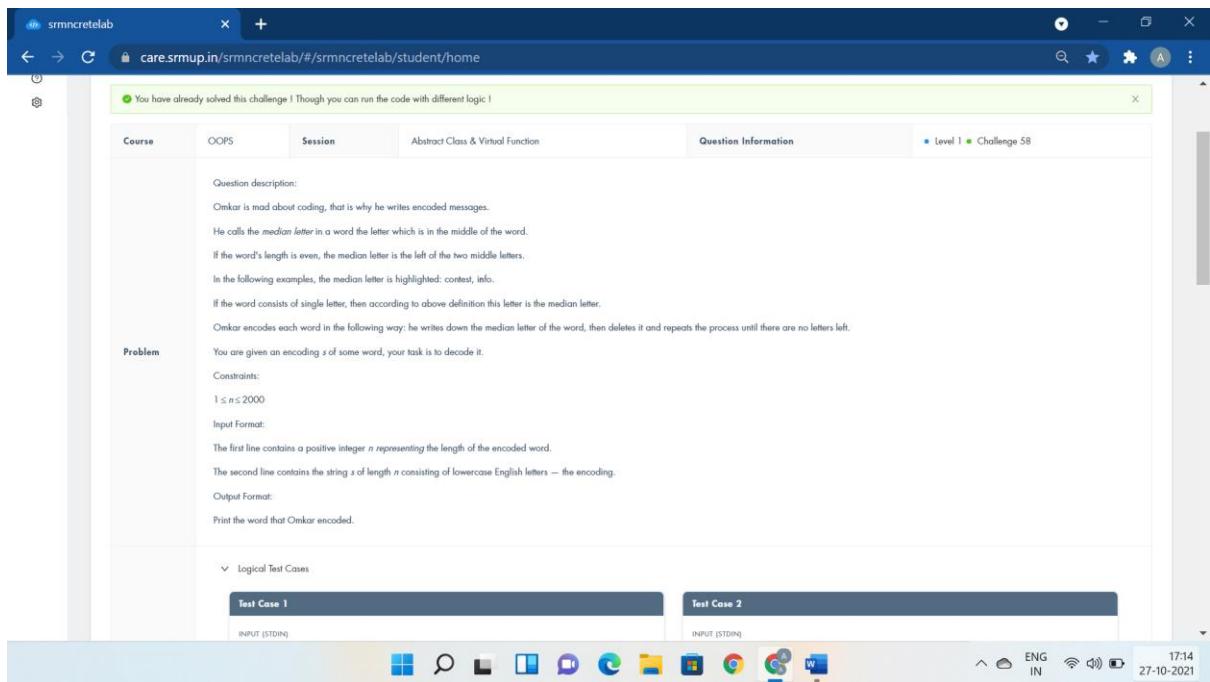
{

Birthday obj;

obj.Cubes();

return 0;

}
```



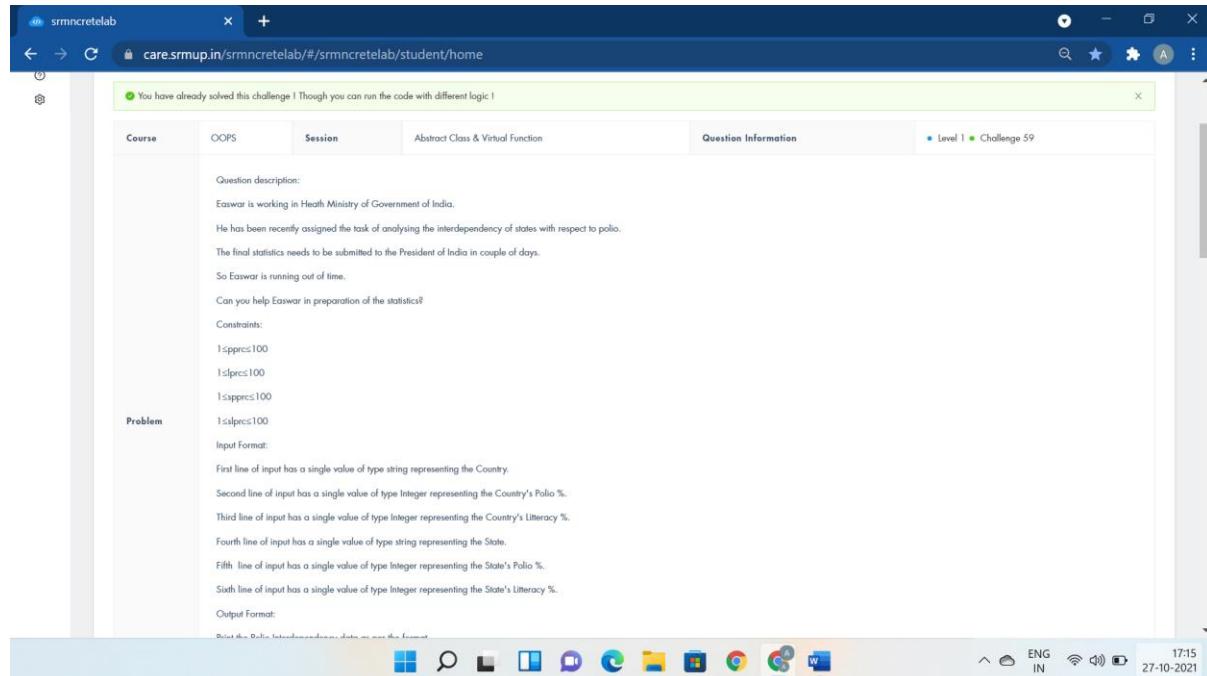
```
#include <iostream>
#include<string>
using namespace std;
class Decode{
public:virtual void Convert()=0;
};
class Word:public Decode{
public:
    string s1,s2;
    int n;
    void Convert(){
        cin>>n>>s1;
        for(int i=0;i<n;i++){
            if((n-i)%2==1)
                s2=s2+s1[i];
            else
                s2=s1[i]+s2;
        }
        cout<<s2;
    }
};
```

```

    }
};

int main()
{
    Word obj;
    obj.Convert();
}

```



```

#include <iostream>

using namespace std;

class country
{
public:
    virtual void getdata() = 0;
    virtual void display() = 0;
};

class state:public country
{

```

```

public:
char a[20];
int b,c;
char d[20];
int e,f;
void getdata(){
    cin>>a>>b>>c>>d>>e>>f;
}
void display()
{
    cout<<"Country:"<<a<<endl<<"Country's Polio %:"<<b<<endl;
    cout<<"Country Literacy %:"<<c<<endl<<"Interdependency Rate:"<<(float)b/c<<endl;
    cout<<"State Name:"<<d<<endl<<% of Polio of State:"<<e<<endl;
    cout<<% of Literacy of State:"<<f<<endl<<"Interdependency Rate:"<<(float)e/f;
}
};

int main() {
if(0)
    cout<<"country::getdata();";
    country *o1;
    state o2;
    o1=&o2;
    o1->getdata();
    o2.display();
return 0;
}

```

You have already solved this challenge! Though you can run the code with different logic!

Course    OOPS    Session    Abstract Class & Virtual Function    Question Information    Level 1    Challenge 60

Question description:  
Janani loves listening to music via her smartphone.  
But the smartphone doesn't have much memory, so Janani listens to her favorite songs in a well-known social network InTalk.  
Unfortunately, internet is not that fast in the city of Manali and the song takes a lot of time to download.  
But Janani is quite impatient. The song's duration is 7 seconds. Janani downloads the first  $S$  seconds of the song and plays it.  
When the playback reaches the point that has not yet been downloaded, Janani immediately plays the song from the start [the loaded part of the song stays in her phone, and the download is continued from the same place], and it happens until the song is downloaded completely and Janani listens to it to the end.  
For  $q$  seconds of real time the Internet allows you to download  $q-1$  seconds of the track.  
Tell Janani, for how many times he will start the song, including the very first start.

Constraints:  
 $2 \leq q \leq 10^4$   
 $1 \leq S, T \leq 10^5$

Input Format:  
The single line contains three integers  $T, S, q$ .

Output Format:  
Print a single integer representing the number of times the song will be restarted.

Logical Test Cases

Test Case 1    Test Case 2

```
#include<iostream>

using namespace std;

class Smartphone{

public:virtual void Listening()=0;

};

class LoveForMusic:public Smartphone{

public:

int T,S,q,c=0;

void Listening(){

cin>>T>>S>>q;

while(S<T){

    c++;

    S*=q;

}

cout<<c;

}

};

int main()

{
```

```

LoveForMusic obj;
obj.Listening();
return 0;
}

```

## Templates:-

You have already solved this challenge! Though you can run the code with different logic!

**Course:** OOPS    **Session:** Templates    **Question Information:** level 1 | Challenge 61

**Question description:**

Veeran the who was described as Son of Forest by his people lives in the middle of the forest.

He has two girlfriends: Elavenil and Koyal, who live at the different ends of the forest, each one is unaware of the other one's existence.

When Veeran has some free time, he goes to one of his girlfriends. He descends into the forest at some time, waits the first parasal to come and rides on it to the end of the forest to the corresponding girl.

However, the parasal run with different frequencies: a parasal goes to Elavenil's direction every  $a$  minutes, but a parasal goes to Koyal's direction every  $b$  minutes.

If two parasal approach at the same time, Veeran goes toward the direction with the lower frequency of going parasal, that is, to the girl, to whose directions the parasal go less frequently.

We know that the parasal begin to go simultaneously before Veeran appears.

**Problem:**

That is the parasal schedule is such that there exists a moment of time when the two parasal arrive simultaneously.

Help Veeran count to which girlfriend he will go more often.

**Constraints:**

$1 \leq a, b \leq 10^6$

$a \neq b$

**Input Format:**

The first line contains two integers  $a$  and  $b$ .

**Output Format:**

Print "Elavenil" if Veeran will go to Elavenil more frequently, "Koyal" if he will go to Koyal more frequently, or "Equal" if he will go to both girlfriends with the same frequency.

**Logical Test Cases:**

Test Case 1    Test Case 2

17:16  
27-10-2021

```

#include <bits/stdc++.h>

using namespace std;

template <class Forest>

Forest Visit(Forest a,Forest b){

    if(a>b)

        cout<<"Koyal\n";

    else

        cout<<"Elavenil\n";

    return 1;

}

int main()

{

    int a,b;

    cin>>a>>b;
}

```

```

if(a%(a-b)==0 && b%(a-b)==0)

cout<<"Equal\n";

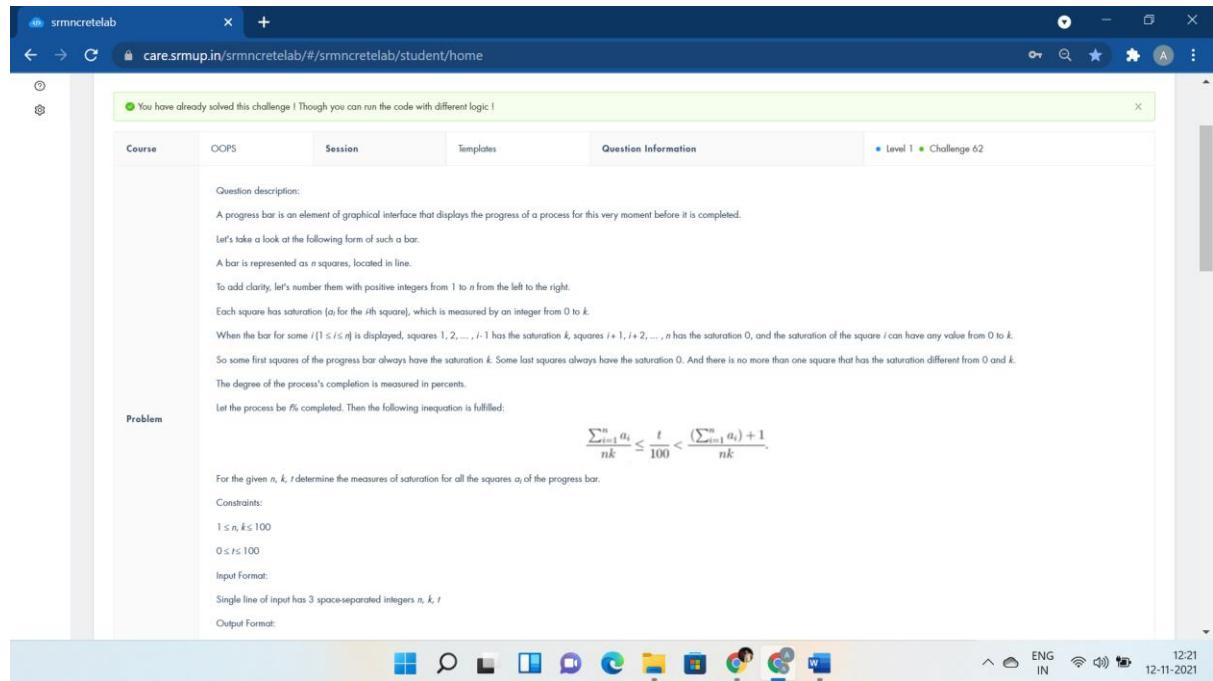
else

Visit(a,b);

return 0;

}

```



```

#include <iostream>

using namespace std;

template <class Interface>

Interface Bar(Interface n,Interface k,Interface t){

    t = t*k*n/100.0;

    while(n--){

        cout<<min(t,k)<<" ";

        t-=min(t,k);

    }

    return 1;

}

int main()

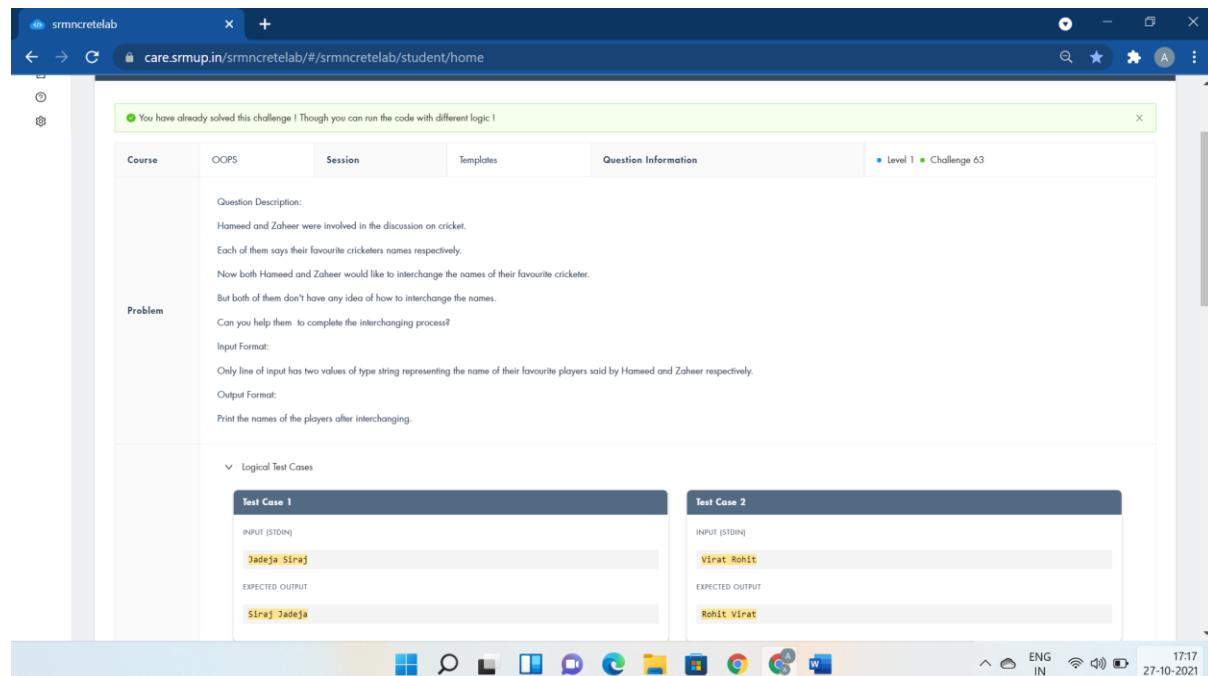
{

```

```

int n,k,t;
cin>>n>>k>>t;
Bar(n,k,t);
return 0;
}

```

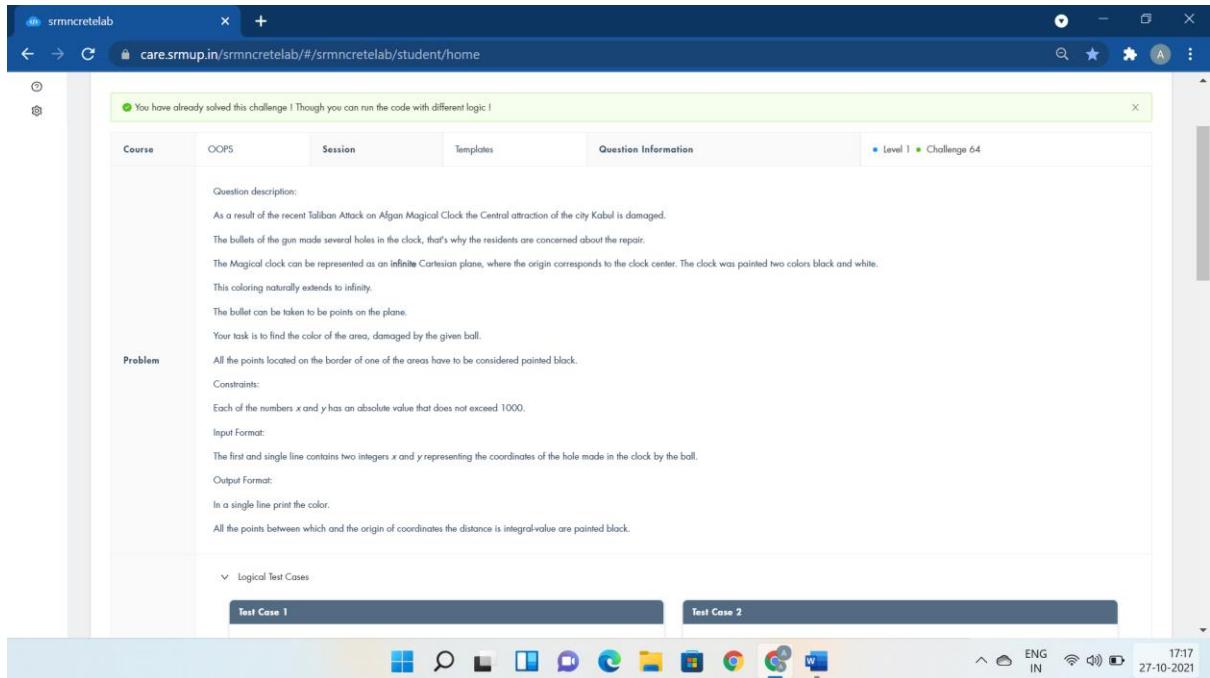


```

#include <iostream>
using namespace std;
template <class T>
void InterchangeFavPlayers(T &player1,T &player2){
    cout<<player2<<" "<<player1;
}
int main()
{
    string player1,player2;
    cin>>player1>>player2;
    InterchangeFavPlayers(player1,player2);
    return 0;
}

```

}



```
#include <iostream>
#include<cmath>

using namespace std;

template <class Hole>

Hole MagicClock(Hole x,Hole y){

    int c;

    c=sqrt(x*x+y*y);

    if(c*c==x*x+y*y){

        cout<<"black\n";

        return 0;

    }

    if(x*y<0)

        c++;

    if(c%2==0)

        cout<<"black";

    else cout<<"white";

    return 1;

}

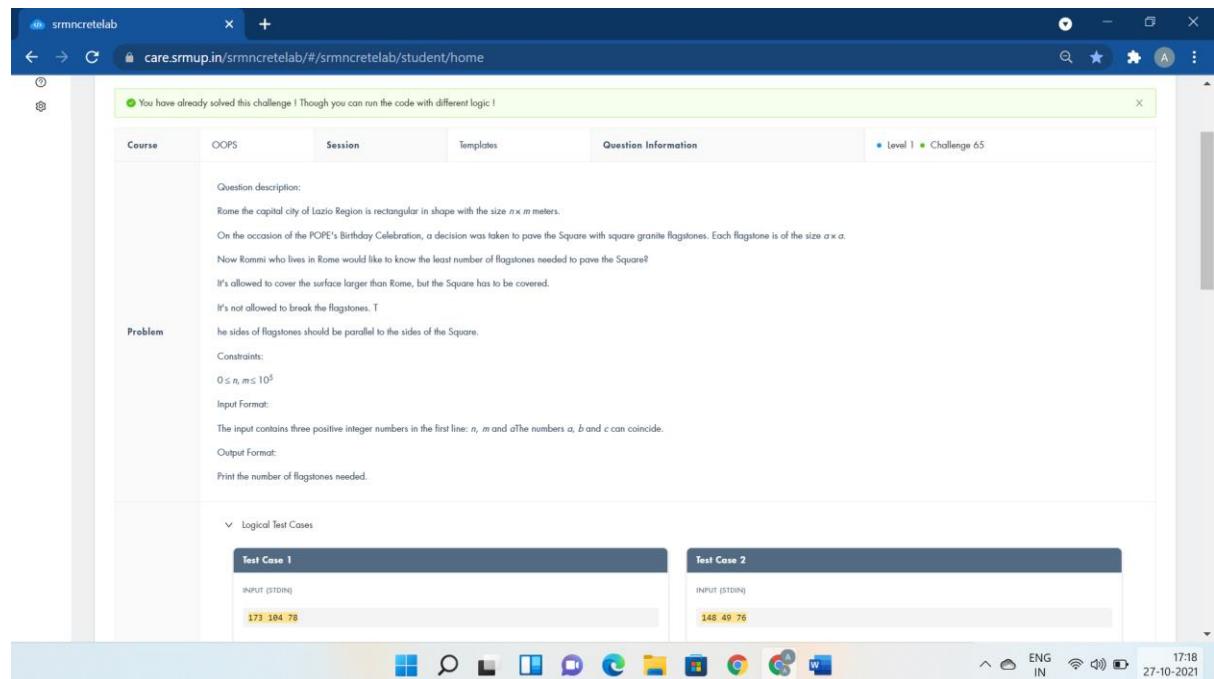
using namespace std;

int main()
```

```

{
    int x,y;
    cin>>x>>y;
    MagicClock(x,y);
    return 0;
}

```



```

#include <iostream>

using namespace std;

template <class Celebration>

Celebration Rome(Celebration a,Celebration b,Celebration c){

    cout<<((b+c-1)/c)*((a+c-1)/c);

    return 1;

}

int main()

{

    int a,b,c;

    cin>>a>>b>>c;

    Rome(a,b,c);

    return 0;

}

```

```
#include <iostream>

using namespace std;

template <class Paper>
Paper Square(Paper T){

    if(T%2==0)

        return 4*T+1;

    else if(T%4==1)

        return 2*T+1;

    else

        return T+1;

}

int main()

{

    int T,n;

    cin>>T;

    while(T--){

        cin>>n;

        cout<<Square(n)<<endl;

    }

    return 0;

}
```

You have already solved this challenge! Though you can run the code with different logic!

Course      OOPS      Session      Templates      Question Information      Level 1      Challenge 67

Question description:  
Janani had trouble falling asleep, and she got bored of counting Stars when she was seven.  
To make herself engrossed tonight she imagined that all Dogs were here to steal her, and she was fighting them off.  
Every  $k$ th Dog got punched in the face with a frying pan.  
Every  $l$ th Dog got his tail shut into a balcony door.  
Every  $m$ th Dog got his paws trampled with sharp heels.  
Finally, she threatened every  $n$ th Dog to call her mom, and he withdrew in panic.

Problem  
How many imaginary Dogs suffered moral or physical damage tonight, if Janani counted a total of  $d$  Dogs?

Constraints:  
 $1 \leq k, l, m, n \leq 10$   
 $1 \leq d \leq 10^5$

Input Format:  
Input data contains integer numbers  $k, l, m, n$  and  $d$ , each number in a separate line

Output Format:  
In the only line of output print the number of damaged dogs.

Logical Test Cases

Test Case 1      Test Case 2

```
#include <iostream>

using namespace std;

template <class LackofSleep>

LackofSleep Counting(LackofSleep k,LackofSleep l,LackofSleep m,LackofSleep n,LackofSleep d)

{

    int c=0;

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

        if(i%k==0 || i%l==0 || i%m==0 || i%n==0)

            c++;

    }

    return c-1;

}

int main()

{

    int k,l,m,n,d;

    cin>>k>>l>>m>>n>>d;

    cout<<Counting(k,l,m,n,d);

    return 0;

}
```

The screenshot shows a web browser window with the URL [care.srmup.in/srmncretelab/#/srmncretelab/student/home](http://care.srmup.in/srmncretelab/#/srmncretelab/student/home). The page title is "CHALLENGE INFORMATION". A message bar at the top says "You have already solved this challenge! Though you can run the code with different logic!". Below this, there are tabs for "Course", "OOOPS" (which is selected), "Session", "Templates", and "Question Information". The "Question Information" tab shows "Level 1" and "Challenge 68". The main content area contains a "Question description:" section with text about Rohan's interest in space research and a cubic planet. It also includes sections for "Problem", "Input Format", "Output Format", and "Logical Test Cases" with two test cases labeled "Test Case 1" and "Test Case 2". The browser interface includes a toolbar with various icons and a status bar at the bottom right showing "17:19" and "27-10-2021".

```
#include <iostream>

using namespace std;

template <class Universe>

Universe Planet (Universe x1,Universe y1,Universe z1,Universe x2,Universe y2,Universe z2){

    if(x1==x2 || y1 == y2 || z1==z2)
        cout<<"YES";
    else
        cout<<"NO";

    return 1;
}

int main()

{
    int x1,y1,z1,x2,y2,z2;
    cin>>x1>>y1>>z1>>x2>>y2>>z2;
    Planet(x1,y1,z1,x2,y2,z2);

    return 0;
}
```

You have already solved this challenge! Though you can run the code with different logic!

Course   OOPS   Session   Templates   Question Information   Problem

Question description:  
Walter has a ribbon, its length is  $n$ .  
He wants to cut the ribbon in a way that fulfills the following two conditions:

- After the cutting each ribbon piece should have length  $a$ ,  $b$  or  $c$ .
- After the cutting the number of ribbon pieces should be maximum.

Help Walter find the number of ribbon pieces after the required cutting.

Constraints:  
 $1 \leq n, a, b, c \leq 2500$

Input Format:  
The first line contains four space-separated integers  $n$ ,  $a$ ,  $b$  and  $c$  representing the length of the original ribbon and the acceptable lengths of the ribbon pieces after the cutting, correspondingly.  
The numbers  $a$ ,  $b$  and  $c$  can coincide.

Output Format:  
In the only line of output print a single number representing the maximum possible number of ribbon pieces.  
It is guaranteed that at least one correct ribbon cutting exists.

Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN) 174 17 45 29	INPUT (STDIN) 378 28 13 79
1	3

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

using namespace std;

template <class Ribbon>

Ribbon Pieces(Ribbon n,Ribbon a,Ribbon b,Ribbon c){

    int d=1,e,i,j;

    for(i=0;i<=4000;i++)

        for(j=0;j<=4000;j++) {

            e=n-a*i-b*j;

            if(e>=0&&e%c==0)

                d=max(d,i+j+e/c);

        }

    cout<<d;

    return 1;

}

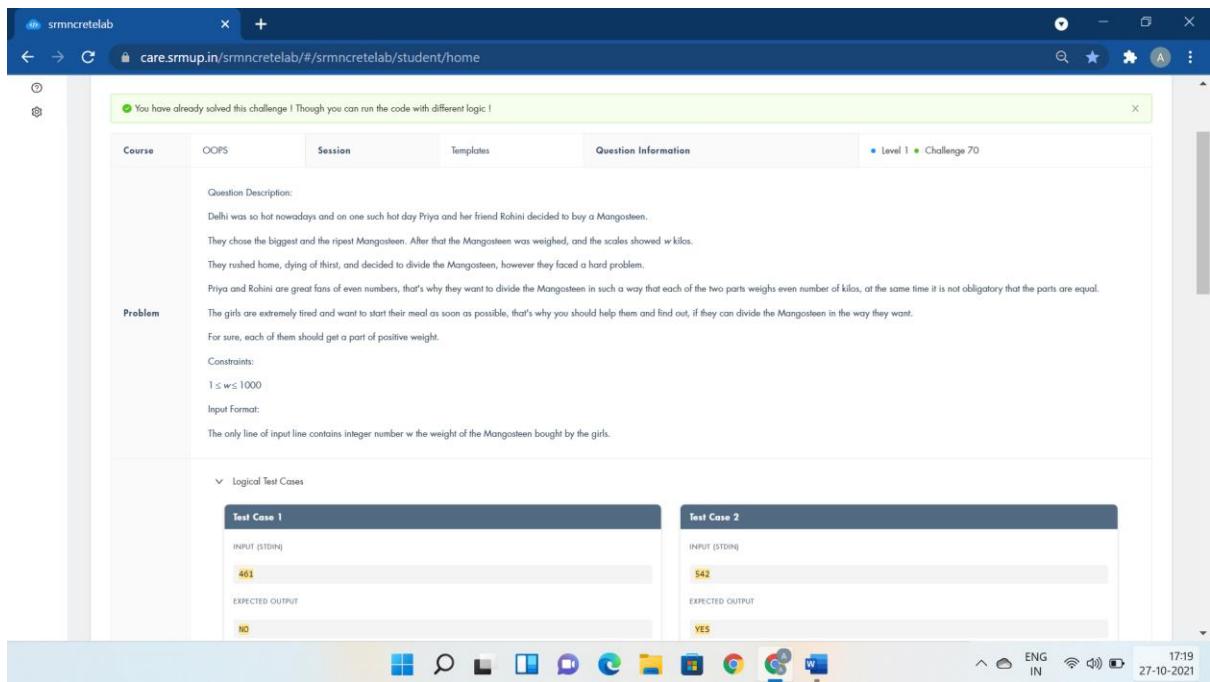
int main(){

    int n,a,b,c;

    cin>>n>>a>>b>>c;

    Pieces(n,a,b,c);

}
```



```
#include <iostream>

using namespace std;

template<class T>

T DivideMangosteen(T PurchasedWeight){

    if(PurchasedWeight%2==0)

        cout<<"YES";

    else

        cout<<"NO";

    return 1;
}

int main()

{

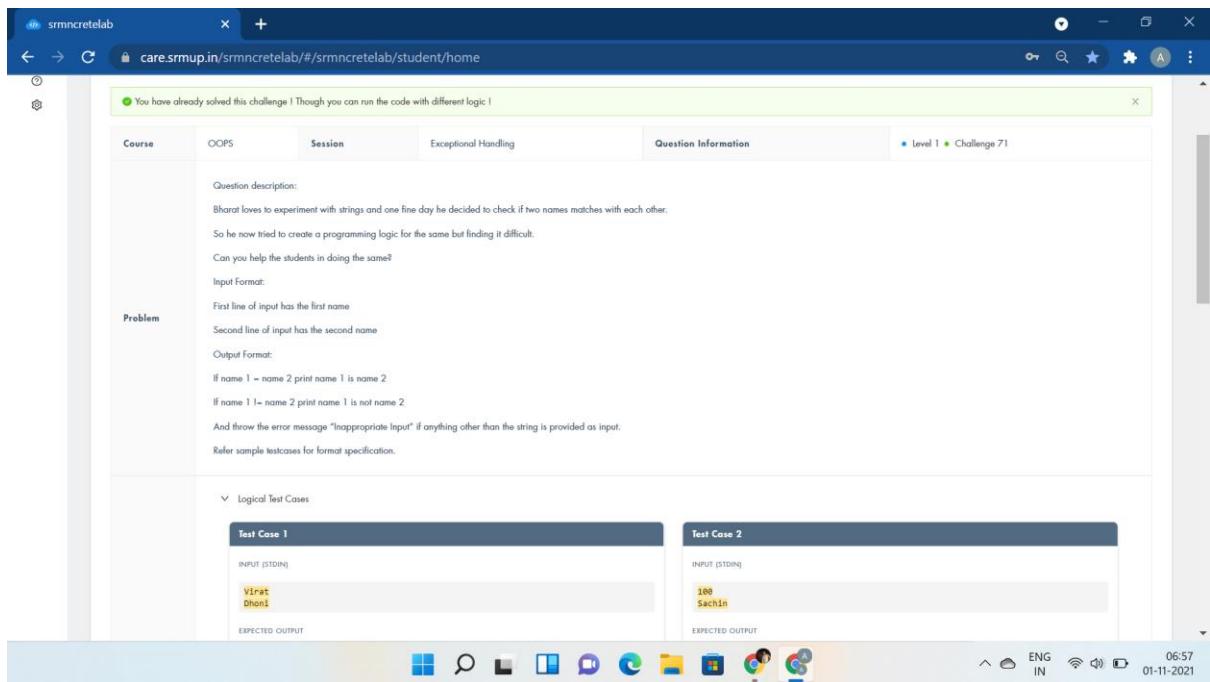
    int PurchasedWeight;

    cin>>PurchasedWeight;

    DivideMangosteen(PurchasedWeight);

    return 0;
}
```

## Exceptional Handling:-



```
#include <iostream>

using namespace std;

int main()

{

    string str1,str2;

    try{

        cin>>str1>>str2;

        int count, n=str1.size();

        if(cin){

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

                if((str1[i]>=48 && str1[i]<=57) || (str2[i]>=48&&str2[i]<=57) )

                    throw 0;

                if(str1[i]==str2[i])

                    count++;

            }

            if(count!=n)

                cout<<str1<<" is not "<<str2;

            else

                cout<<str1<<" is "<<str2;

        }

    }

    catch (int i){

        cout<<"Inappropriate Input";

    }

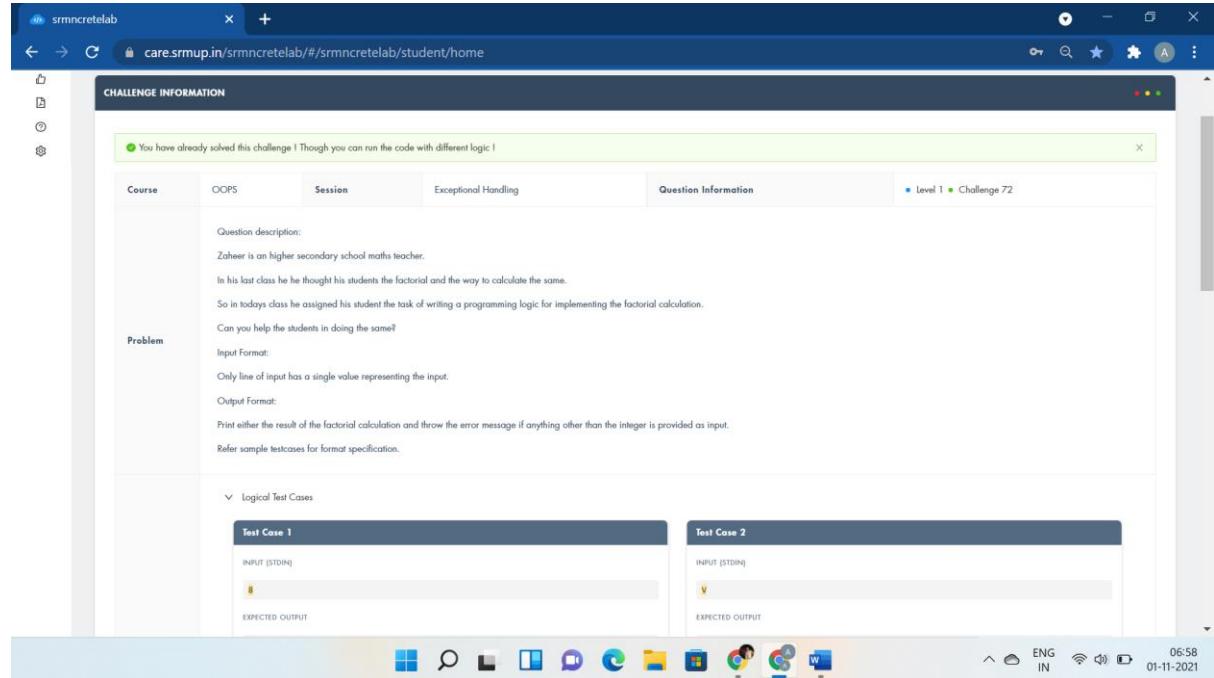
}
```

```

    }

    return 0;
}

```



```

#include <bits/stdc++.h>

#include <string.h>

using namespace std;

int main()

{

    int k;

    try{

        cin>>k;

        if(cin)

            cout<<fixed<<setprecision(0)<<tgamma(k+1);

        else

            throw "e";

    }

    catch (int i){

    }

    catch (const char *exp){

        cout<<"Input should be a Integer";
    }
}

```

```

    }

    return 0;
}

```

```

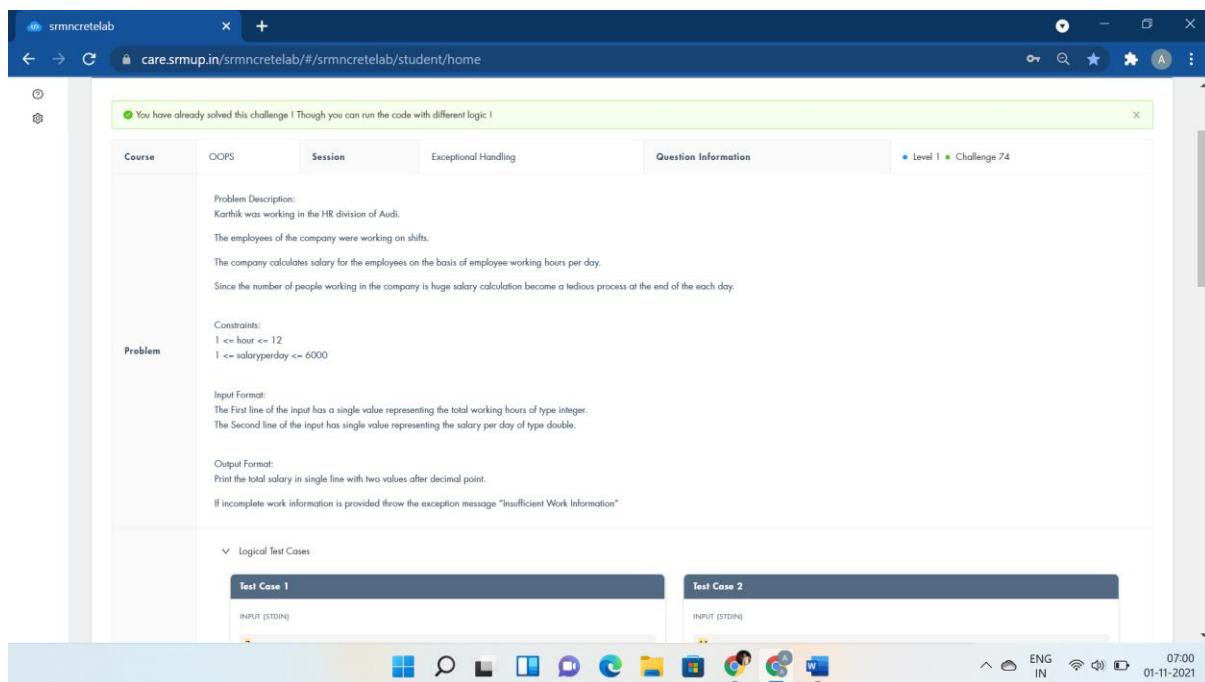
#include <iostream>

using namespace std;

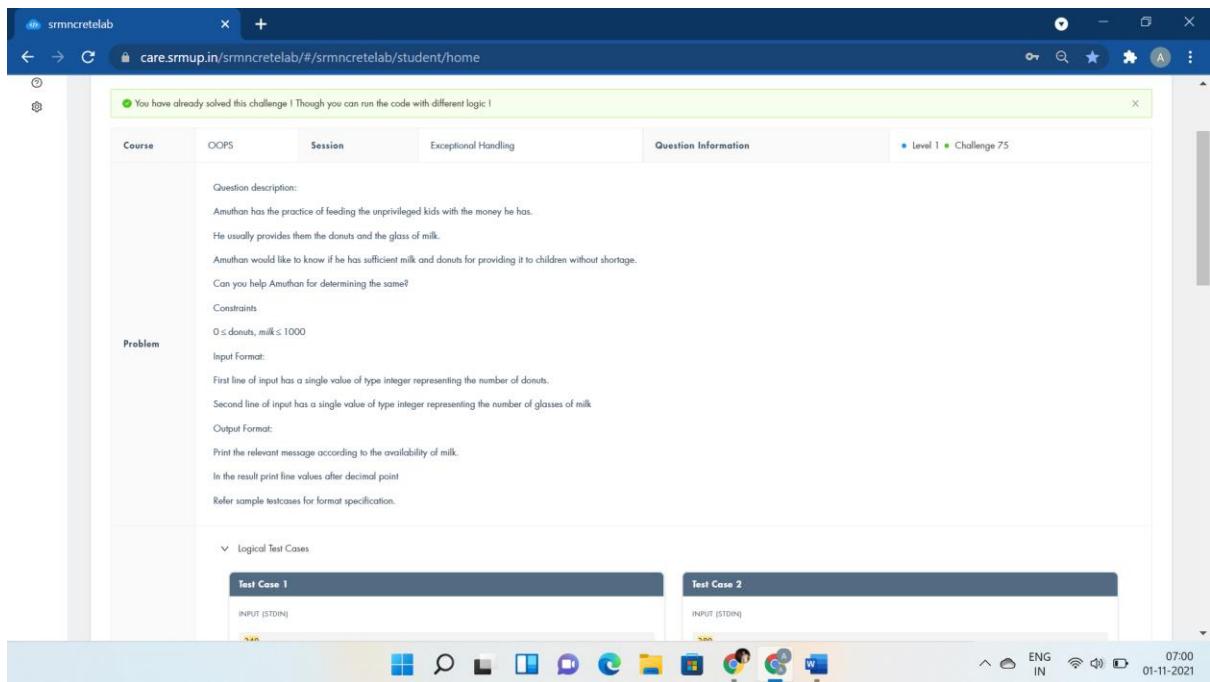
int main()
{
    int n,m;
    try{
        cin>>n;
        cin>>m;
        if(cin){
            cout<<n-1+(1+2*(n-1))*(m-1);
        }
        else
            throw 0;
    }
    catch(int griddimensions)
    {
        cout<<"Invalid Grid Dimensions";
    }
}

```

```
    return 0;  
}
```



```
#include<bits/stdc++.h>  
  
using namespace std;  
  
int main()  
{  
    float hour,salaryperday;  
  
    try{  
        cin>>hour;  
        cin>>salaryperday;  
  
        if(cin){  
            cout<<fixed<<setprecision(2)<<hour*salaryperday;  
        }  
        else  
            throw 0;  
    }  
  
    catch(int workstatus)  
{  
        cout<<"Insufficient Work Information";  
    }  
  
    return 0;  
}
```



```
#include <iostream>

using namespace std;

int main()
{
    int donuts,milk;
    try{
        cin>>donuts;
        cin>>milk;
        if(milk==0)
            throw donuts;
        else
            cout<<"You have "<<(float)donuts/milk<<" donuts for each glass of milk";
    }
    catch(int e){
        cout<<e<<" donuts and No Milk\nGo buy some milk";
    }
    return 0;
}
```

You have already solved this challenge! Though you can run the code with different logic.

Question description:

Bogar was given a task to check whether the entered mark is valid or not.

Bogar framed three rules for checking the validity of the mark

Rule 1: The mark should be greater than 0 and less than or equal to 100 [ 0 < m <=100 ]

Rule 2: The mark should not exceed 100.

Rule 3: No negative Marks

Rule 4: It should be a valid integer number

Kindly help Bogar the Tamil SIDDHAR to perform the operations.

Problem

Constraints:

1≤n≤1000

Input Format:

Only one line of input has a single value representing the input.

Output Format:

If the input value satisfies the above mentioned rules of Bogar print "Valid Mark"

And throw the error message "Invalid Mark" if the input value doesn't satisfy the rules of Bogar.

Refer sample testcases for format specification.

Logical Test Cases

Test Case 1      Test Case 2

```
#include <iostream>
#include <math.h>
using namespace std;
int main()
{
    int a;
    try {
        cin>>a;
        if (a>0 && a<=100)
            cout<<"Valid Mark";
        else
            throw "e";
    }
    catch(const char* t){
        cout<<"Invalid Mark";
    }
}
```

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

using namespace std;

int main()

{

    int unitconsumed,costperunit;

    try{

        cin>>unitconsumed;

        cin>>costperunit;

        long int res;

        res=pow(unitconsumed,costperunit);

        if(cin){

            cout<<res;

        }

        else

        throw 0;

    }

    catch(int unit){

        cout<<"Incomplete Data";

    }

    return 0;

}
```

You have already solved this challenge! Though you can run the code with different logic!

Course      OOPS      Session      Exceptional Handling      Question Information      Level 1      Challenge 78

Problem Description:  
Phoenix mall in the capital city of Washington and it is rectangular in shape when it is seen on the map with the size  $n \times m$  meters.  
On the occasion of the jubilee anniversary, a decision was taken to pave the Square with square marbles stones. Each stone is of the size  $a \times a$ .  
Can you find what is the least number of stones needed to pave the Square?  
It's allowed to cover the surface larger than the Mall Square, but the Square has to be covered.  
It's not allowed to break the stones.  
The sides of stones should be side by side[parallel] to the sides of the Square.

Constraints:  
Problem       $1 \leq n \leq 10^4$   
                 $1 \leq m \leq 10^4$   
                 $1 \leq a \leq 10^4$

Input Format:  
The only line of input contains three positive integer numbers  $n$ ,  $m$  and  $a$  separated by a space .

Output Format:  
Print the needed number of stones.  
If any of the input values  $n$  or  $m$  or  $a$  is missing in the input then raise the exception message as "Invalid Dimension"

```
#include <iostream>

using namespace std;

int main()
{
    int n,m,a;
    try{
        cin>>n>>m>>a;
        if(cin){
            cout<<((n+a-1)/a)*((m+a-1)/a);
        }
        else
            throw 0;
    }

    catch(int dimension){
        cout<<"Invalid Dimension";
    }

    return 0;
}
```

You have already solved this challenge! Though you can run the code with different logic!

Course   OOPS   Session   Exceptional Handling   Question Information   Level 1 • Challenge 79

Question description:  
Dino is an DTP operator in the Document formating firm.  
The document processor Dino uses accepts only characters which are alphabetic in nature.  
If the character is not alphabetic it is not accepted by the document processor.  
Can you help Dino in finding the nature of the characters in the document Dino is working with?

Problem  
Input Format:  
First line of input has a single value of type integer representing the number of testcases.  
Second line of input has the string to be checked in the document.  
Output Format:  
Print the relevant message for the input string.  
Refer sample testcases for format specification.

Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN) 1 v8 EXPECTED OUTPUT	INPUT (STDIN) 1 6e EXPECTED OUTPUT

07:02 01-11-2021

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

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

using namespace std;

int main(){

    int t,i,j;

    cin>>t;

    string str;

    f(j,0,t){

        f(i,0,2){

            try{

                cin>>str[i];

                if(isalpha(str[i])){

                    cout<<str[i]<<" is alphabetic"<<endl;

                }

                else

                    throw str[i];

            }

            catch (char f){

                cout<<f<<" is not alphabetic"<<endl;

            }

        }

    }

}
```

You have already solved this challenge! Though you can run the code with different logic!

**Course**   **OOPS**   **Session**   **Exceptional Handling**   **Question Information**   **Level 1 Challenge 80**

**Problem Description:**  
Selvan was playing with the object of random size for stress relief.  
Selvan knows that the Length, Width, and Height of the object.  
But he would like to know the surface area of the object he is playing with.  
Can you help him in finding it?

**Functional Description:**  
Surface area of the Object =  $2 \times (\text{width} \times \text{length} + \text{length} \times \text{height} + \text{height} \times \text{width})$

**Constraints:**  
 $1 \leq \text{length} \leq 20$   
 $1 \leq \text{width} \leq 20$   
 $1 \leq \text{height} \leq 20$

**Input Format:**  
First Line : Length of the object in Integer  
Second Line : Width of the object in Integer  
Third Line : Height of the object in Integer

**Output Format:**  
Print a single integer value representing the surface area of the object selvan is playing with.  
If the information provided about the object is not sufficient for the calculation then throw an exception "Incomplete information about the object"

**Logical Test Cases**

```
#include <iostream>

using namespace std;

int main()
{
    int a,b,c;
    try{
        cin>>a>>b>>c;
        if(cin){
            cout<<2*(a*b+b*c+c*a);
        }
        else
            throw 0;
    }
    catch(int objectinfo){
        cout<<"Incomplete information about the object";
    }
    return 0;
}
```

# STL:-

You have already solved this challenge! Though you can run the code with different logic!

**Course**   **OOPS**   **Session**   **STL**   **Question Information**   **Level 1**   **Challenge 81**

**Question description**

Tina administers a large cluster of computers with hard drives that use various file system types to store data. Tina recently decided to unify the file systems to the same type.

That is quite a challenge since all the drives are currently in use, all of them are filled with important data to the limits of their capacities, and you cannot afford to lose any of the data.

Moreover, reformatting a drive to use a new file system may significantly change the drive's capacity. To make the reformat possible, Tina will have to buy an extra hard drive. Obviously, you want to save money by minimizing the size of such extra storage.

Tina can reformat the drives in any order. Prior to reformatting a drive, you must move all data from that drive to one or more other drives, splitting the data if necessary.

After a drive is reformatted, you can immediately start using it to store data from other drives. It is not necessary to put all the data on the same drives they originally started on – in fact, this might even be impossible if some of the drives have smaller capacity with the new file system.

It is also allowed for some data to end up on the extra drive.

Can you help Tina with this complicated task?

**Constraints:**

$1 \leq n \leq 10^6$

$1 \leq a, b \leq 10^9$

**Input Format:**

The input begins with a line containing one integer  $n$ , which is the number of drives in your cluster.

Following are  $n$  lines, each describing a drive as two integers  $a$  and  $b$ , where  $a$  is the capacity with the old file system and  $b$  is the capacity with the new file system.

All capacities are given in gigabytes and satisfy: (One thousand petabytes should be enough for everyone, right?)

**Output Format:**

Print the total extra capacity in gigabytes you must buy to reformat the drives.

16:48 25-11-2021

```
#include <algorithm>
#include <iostream>
#include <vector>

using namespace std;

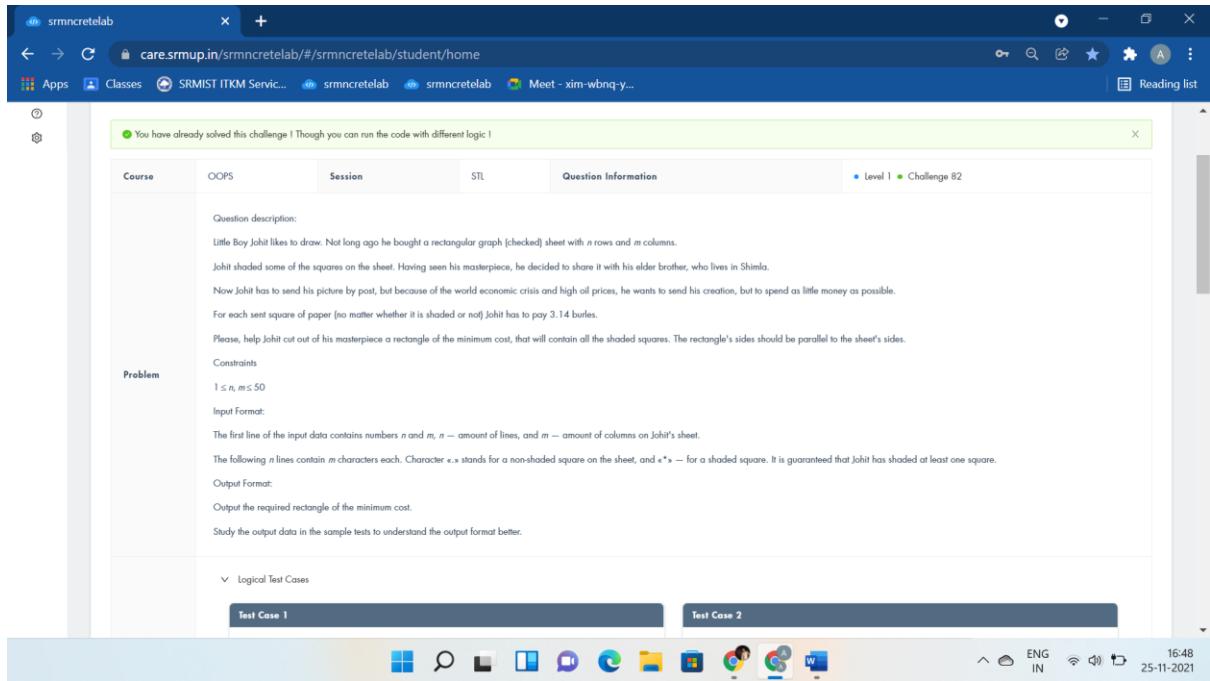
int main() {
    int N, a, b;
    while (cin>>N) {
        vector<pair<int,pair<int,int>>>StorageDrives;
        for (int i = 0; i < N; i++) {
            cin>>a>>b;
            StorageDrives.push_back(make_pair((b>a) ? a : 2000000001-b, make_pair(a, b)));
        }
        long long ret = 0, cap = 0;
        sort(StorageDrives.begin(),StorageDrives.end());
        int z=StorageDrives.size();
        for (int i = 0; i < z; i++) {
            if (cap < StorageDrives[i].second.first) {
```

```

    ret += StorageDrives[i].second.first - cap;
    cap = StorageDrives[i].second.first;
}
cap += StorageDrives[i].second.second - StorageDrives[i].second.first;
}

cout << ret << endl;
}
}

```



```

#include<bits/stdc++.h>

using namespace std;

int n,m,sx=99999,sy=99999,x,y;

char a[55][55];

int main(){

    cin>>n>>m;

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

        for(int j=1;j<=m;j++){

            cin>>a[i][j];

            if(a[i][j]=='*'){

                x=max(x,i),y=max(y,j),sx=min(sx,i),sy=min(sy,j);

            }

        }

    }

}

```

```

    }

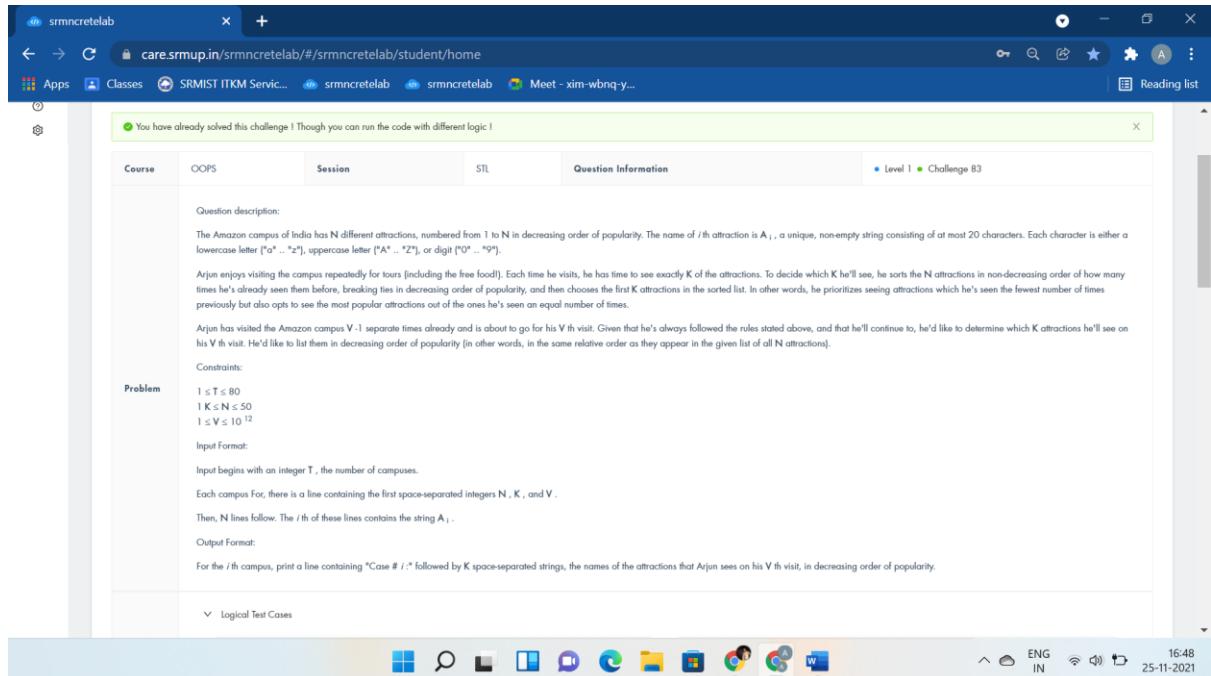
    for(int i=sx;i<=x;i++){
        for(int j=sy;j<=y;j++){
            cout<<a[i][j];
        }
        cout<<endl;
    }

    return 0;
}

cout<<"vector<vector<char>>drawing(n,vector<char>(m,'0')); drawing[row][col]";

}

```



```

#include <bits/stdc++.h>

using namespace std;

typedef long long LL;

const int N=55;

LL n, k, v, idx;

string name[N];

int main(){

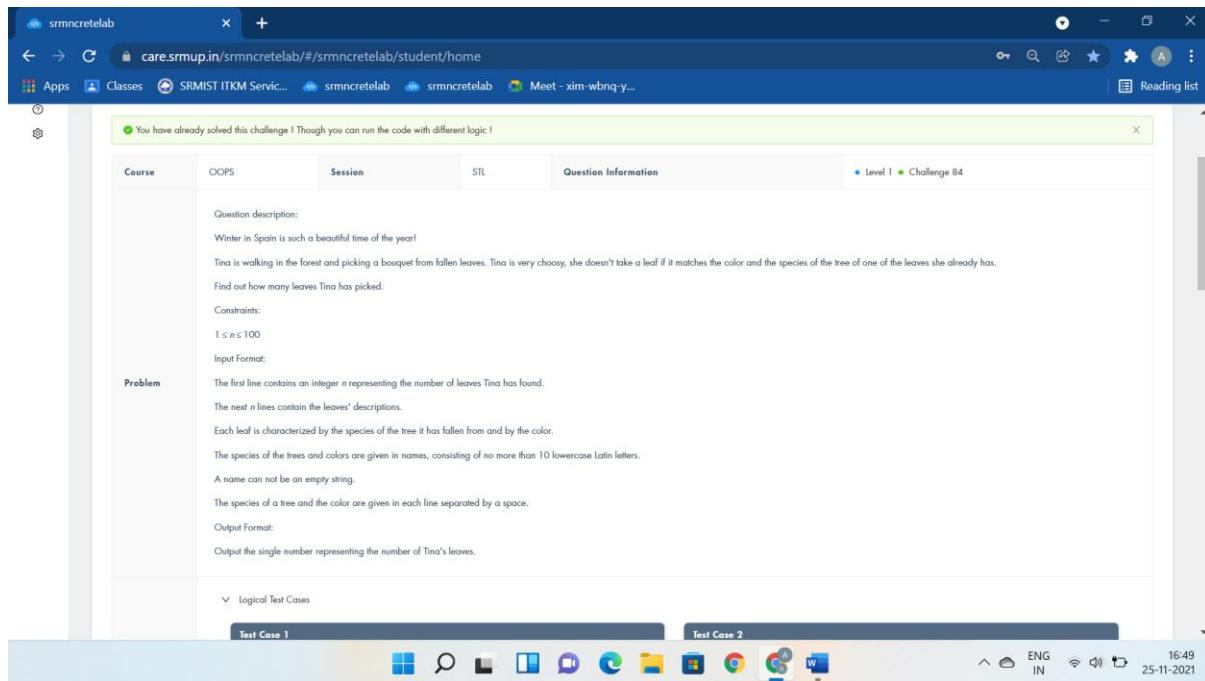
    LL t; cin>>t; while(t--){
        cin>>n>>k>>v;
        for(int i=0; i<n; i++)
            cin>>name[i];
        LL st=((v-1)*k)%n;

```

```

//cout<<"Case #"<<(++idx)<<"";
vector<int> ans;
for(int i=0; i<k; i++)
    ans.push_back((st+i)%n);
sort(ans.begin(), ans.end());
for(int id: ans)
    cout<<name[id]<<" ";
cout<<"\n";
}
return 0;
cout<<"vector<string>visit(n); vector<pair<int,string>>seenattraction; sort(seenattraction.begin(),seenattraction.end());";
}

```



```

#include <bits/stdc++.h>

using namespace std;

int main()
{
    int n;
    cin>>n;
    set<pair<string,string>>Descriptionofleaves;
    string species,color;
    while(n--){

```

```

    cin>>species>>color;

    Descriptionofleaves.insert(make_pair(species,color));

}

cout<<Descriptionofleaves.size();

return 0;

}

```

You have already solved this challenge! Though you can run the code with different logic!

Course	OOPS	Session	STL	Question Information
				Level 1 • Challenge 85

**Question description**

Bank of Spain have been opened again after the major heist by Professor Sergio and Gang. The news of Bank of Spain have spread all around Spain and some of the students have been petrified due to seeing the basilisk. The Former Director of Bank of Spain got fired and now Tokyo is trying to enter the Bank of Spain chamber. These aren't good news for Joji Mathew the new Director of Bank of Spain. The problem is, he doesn't want anybody to be able to enter the Bank's chamber.

The Bank of Spain Chamber is an  $n \times m$  rectangular grid in which some of the cells are columns. A light ray passes through the columns without changing its direction.

But with some spell we can make a column magic to reflect the light ray in all four directions when it receives the ray. This is shown in the figure below.

**Problem**

The basilisk is located at the right side of the lower right cell of the grid and is looking to the left [in the direction of the lower left cell].

According to the legend, anyone who meets a basilisk's gaze directly dies immediately. But if someone meets a basilisk's gaze through a column, this person will get petrified.

We know that the door to the Bank of Spain chamber is located on the left side of the upper left corner of the grid and anyone who wants to enter will look in the direction of its movement [in the direction of the upper right cell] from that position.

Given the dimensions of the Bank's chamber and the location of regular columns, Joji Mathew has asked you to find the minimum number of columns that we need to make magic so that anyone who wants to enter the chamber would be

```

#include <bits/stdc++.h>

using namespace std;

void sum(){}
int n,m;
vector <int> use[2020];
int cost[2020];
string g[1010];
int main()
{
    cin>>n>>m;
    for(int i=0;i<n;i++)
    {
        cin>>g[i];
        for(int j=0;j<m;j++)
        {

```

```

        if(g[i][j]=='#')
    {
        use[i].push_back(j+n);
        use[j+n].push_back(i);
    }
}

queue<int>BankChamber;
BankChamber.push(n-1);
cost[n-1]=1;
while(!BankChamber.empty())
{
    int t=BankChamber.front();
    BankChamber.pop();
    int z=use[t].size();
    for(int i=0;i<z;i++)
    {
        if(cost[use[t][i]]==0)
        {
            cost[use[t][i]]=cost[t]+1;
            BankChamber.push(use[t][i]);
        }
    }
}
cout<<cost[0]-1<<endl;
sum();
return 0;
cout<<"BankChamber.push(n);";
}

```

You have already solved this challenge! Though you can run the code with different logic!

Course	OOPS	Session	STL	Question Information
				Level 1 Challenge 86

**Question description:**

In Spain, there is the national holiday coming. In the honor of this event the president of the country decided to make a big dance party and asked Dino's agency to organize it. He has several conditions:

- overall, there must be  $m$  dances;
- exactly three people must take part in each dance;
- each dance must have one dancer in white clothes, one dancer in red clothes and one dancer in blue clothes [these are the colors of the national flag of Spain].

The agency has  $n$  dancers, and their number can be less than  $3m$ . That is, some dancers will probably have to dance in more than one dance. All of Dino's dancers must dance on the party.

However, if some dance has two or more dancers from a previous dance, then the current dance stops being spectacular. Dino agency cannot allow that to happen, so each dance has at most one dancer who has danced in some previous dance.

Dino considered all the criteria and made the plan for the  $m$  dances: each dance had three dancers participating in it. Dino task is to determine the clothes color for each of the  $n$  dancers so that the President's third condition fulfilled: each dance must have a dancer in white, a dancer in red and a dancer in blue.

The dancers cannot change clothes between the dances.

**Constraints:**

$3 \leq n \leq 10^5$   
 $1 \leq m \leq 10^8$

**Input Format:**

The first line contains two space-separated integers  $n$  and  $m$  representing the number of dancers and the number of dances, correspondingly.

Then  $m$  lines follow, describing the dances in the order of dancing them. The  $i$ -th line contains three distinct integers — the numbers of the dancers that take part in the  $i$ -th dance.

The dancers are numbered from 1 to  $n$ .

Each dancer takes part in at least one dance.

**Output Format:**

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

```
using namespace std;
```

```
typedef long long int ll;
```

```
ll a[100006],c[3];
```

```
int main()
```

```
{
```

```
    ll n,m,i,j,k,l,sum=0;
```

```
    cin>>n>>m;
```

```
    for(i=0;i<m;i++)
```

```
{
```

```
    sum=0;
```

```
    for(j=0;j<3;j++)
```

```
{
```

```
        cin>>c[j];
```

```
        sum=sum+a[c[j]];
```

```
}
```

```
    l=1;
```

```
    for(k=0;k<3;k++)
```

```
{
```

```
        if(l==sum)
```

```
            l++;
```

```
        if(a[c[k]]==0)
```

```

{
    a[c[k]]=l++;
}

}

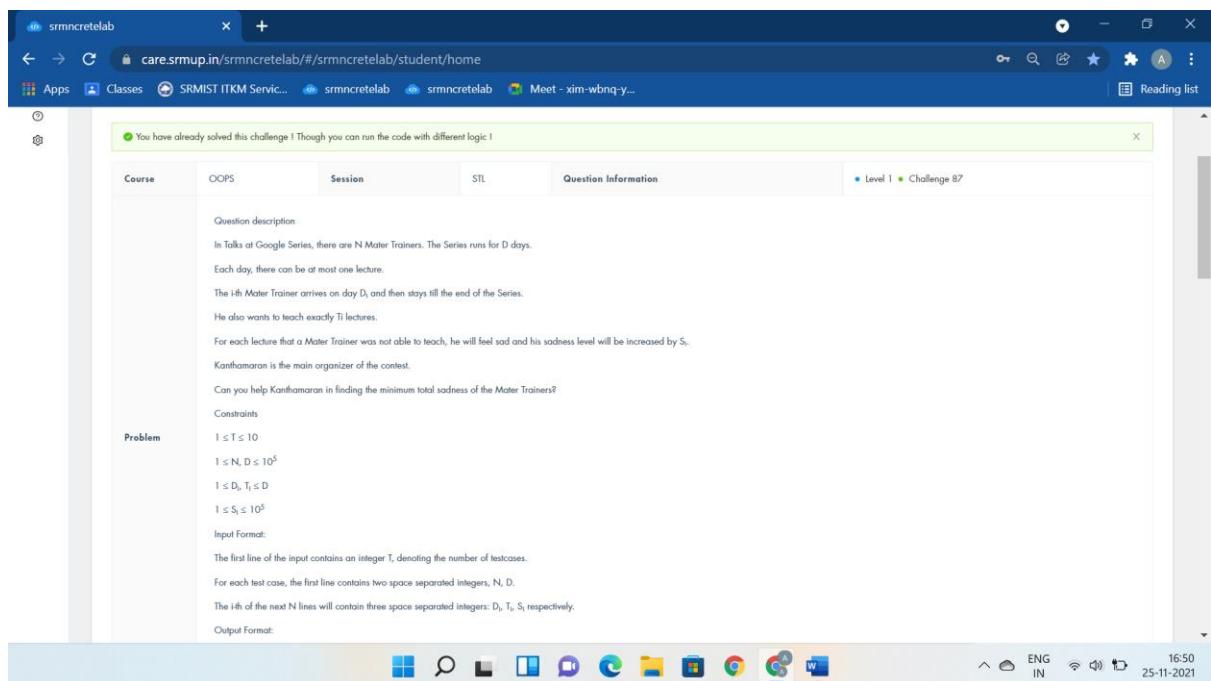
}

for(i=1;i<=n;i++)
    cout<<a[i]<<" ";

return 0;

cout<<"map<int,int>dance; set<int>dancer;";}

```



```

#include <bits/stdc++.h>

#define ll long long

using namespace std;

int main(){

    int t;
    cin >> t;

    while (t--) {

        int n, d;
        cin >> n >> d;

        map<ll, vector<pair<long,long>>>TGS;

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

            ll day, lec, sad;

```

```

cin >> day >> lec >> sad;
TGS[day].push_back({sad, lec}); }

priority_queue<pair<long,long>>PQ;

for (int i = 1; i <= d; i++) {
    for (auto x : TGS[i])
        PQ.push(x);
    if (!PQ.empty())
    {
        pair<ll, ll> p = PQ.top();
        PQ.pop();
        p.second--;
        if (p.second == 0) {}
        else
            PQ.push({p.first, p.second});
    }
}

ll cnt = 0;
while (!PQ.empty()) {
    pair<ll, ll> p = PQ.top();
    cnt += (p.first * p.second);
    PQ.pop();
}
cout << cnt << endl;
}

return 0;
cout<<"vector<pair<long,long>>TGS PQ.top().first;PQ.top().second ";}

```

You have already solved this challenge! Though you can run the code with different logic!

**Course**   **OOPS**   **Session**   **STL**   **Question Information**   **Level I** | **Challenge 88**

**Question description:**  
Virat in his recent examination got very bad marks in algebra again. To avoid such unpleasant events in future he decided to train his arithmetic skills. He wrote four integer numbers  $a, b, c, d$  on the blackboard.  
During each of the next three minutes he took two numbers from the blackboard (not necessarily adjacent) and replaced them with their sum or their product. In the end he got one number.  
Unfortunately, due to the awful memory he forgot that number, but he remembers four original numbers, sequence of the operations and his surprise because of the very small result.  
Help Virat remember the forgotten number to find the smallest number that can be obtained from the original numbers by the given sequence of operations.

**Constraints:**  
 $0 \leq a, b, c, d \leq 1000$

**Input Format:**  
First line contains four integers separated by space representing the the original numbers.  
Second line contains three signs ('+' or '\*' each) separated by space representing the sequence of the operations in the order of performing. ('+' stands for addition, '\*' — multiplication)

**Output Format:**  
Output one integer number representing the minimal result which can be obtained.

**Logical Test Cases**

Test Case 1	Test Case 2
INPUT [STDIN] 228 127 597 394 * + *	INPUT [STDIN] 8 1 7 14 + + *

16:50 25-11-2021

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

using namespace std;

long long ans=1e15;

deque<char>Operations(20);

void solve(vector<long long> a,int id){

    if((int)a.size()==1){

        ans=min(ans,a[0]);

        return;

    }

    for(int i=0;i<(int)a.size();i++){

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

            vector<long long> b;

            if(Operations[id]=='+') b.push_back(a[i]+a[j]);

            else b.push_back(a[i]*a[j]);

            for(int k=0;k<(int)a.size();k++){

                if(k!=i && k!=j) b.push_back(a[k]);

            }

            solve(b,id+1);

        }

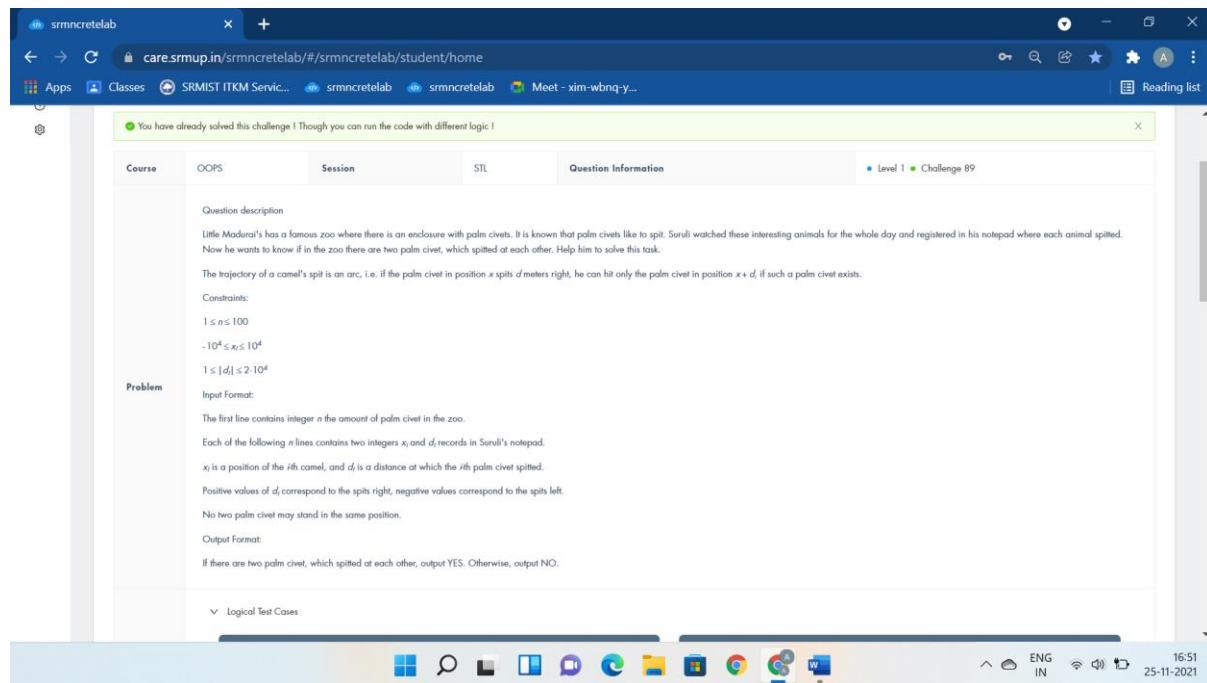
    }

}
```

```

int main() {
    vector<long long>numbers(4);
    for(int i=0;i<4;i++) cin>>numbers[i];
    for(int i=0;i<3;i++) cin>>Operations[i];
    solve(numbers,0);
    cout<<ans;
    return 0;
}

```



```

#include <bits/stdc++.h>

using namespace std;

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

int i,j,n,x[110],d[110];

int main(){

    cin>>n;

    f(i,1,n+1) cin>>x[i]>>d[i];

    f(i,1,n+1){

        f(j,i+1,n+1){

            if(x[i]+d[i]==x[j] && x[j]+d[j]==x[i]){

                cout << "YES\n";

            }

        }

    }

    return 0;
}

```

```

    }

}

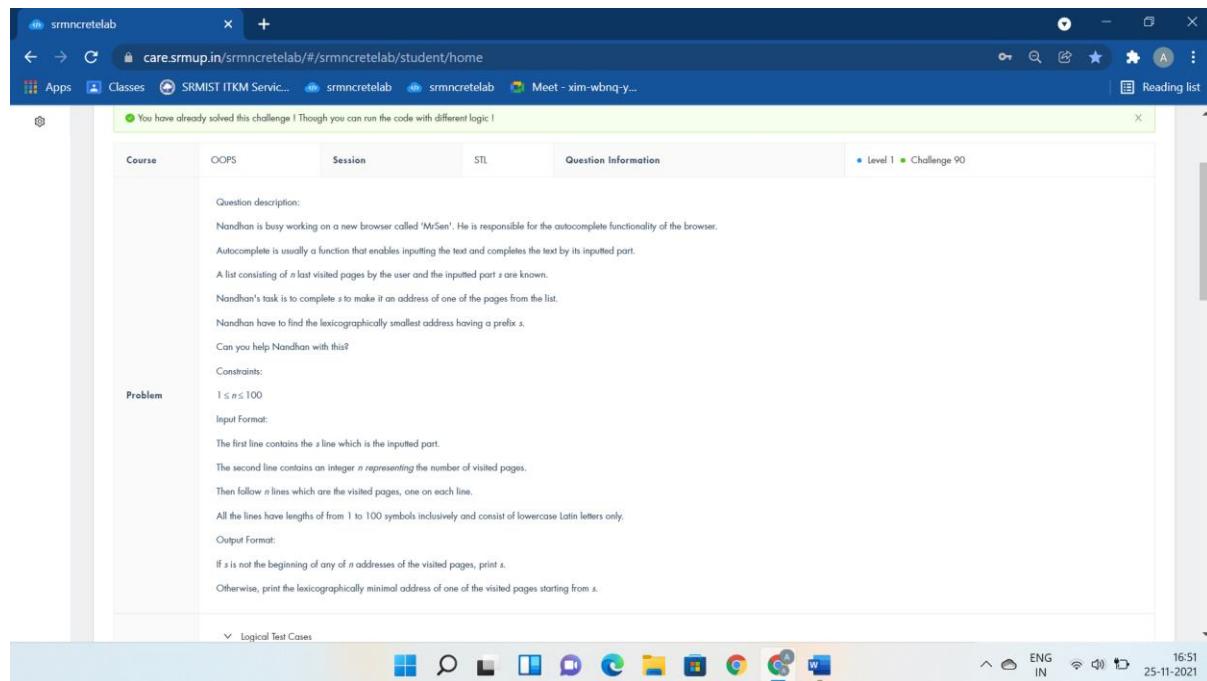
cout << "NO";

return 0;

cout<<"map<long long,long long>palm; ";

}

```



```

#include<bits/stdc++.h>

using namespace std;

int i,n;

string s,t,u;

int D()

{

    for(i=0;s[i];i++)if(s[i]^t[i])return 0;

    return 1;

}

int main()

{

    for(cin>>s>>n;n--)

    {

        cin>>t;

        if(D()&&(u.empty()||t<u))u=t;
    }
}

```

```

    }

    if(u.empty())cout<<s;

    else cout<<u;

    return 0;

    cout<<"unordered_map<string,string>website; map<string,bool>searchlist; cin>>n;" ;

}

```

## Advanced Inheritance:-

You have already solved this challenge! Though you can run the code with different logic!

Course	OOPS	Session	Advanced Inheritance	Question Information	Level 1 • Challenge 91

**Question description:**  
Ravindran is employed in a multinational production firm as a general manager.  
He uses software to generate his salary slips every month.  
The programme unexpectedly crashed, so Ravindran is having an issue with completing the salary slip on time.  
As a result, he desires to prepare the salary slip in the following order.  
Please assist him in preparing the salary slip so that he may submit it on time.

**Input Format:**  
First Line: Employee Code  
Second Line: Employee Name  
Third Line: Employee Role  
Fourth Line: Employee Basic Pay  
Fifth Line: Employee HRA  
Sixth Line: Employee DA  
Seventh Line: Employee PF

**Output Format:**  
Print the results as per format.  
Refer sample testcases for format specification.

**Logical Test Cases**

```

#include <iostream>

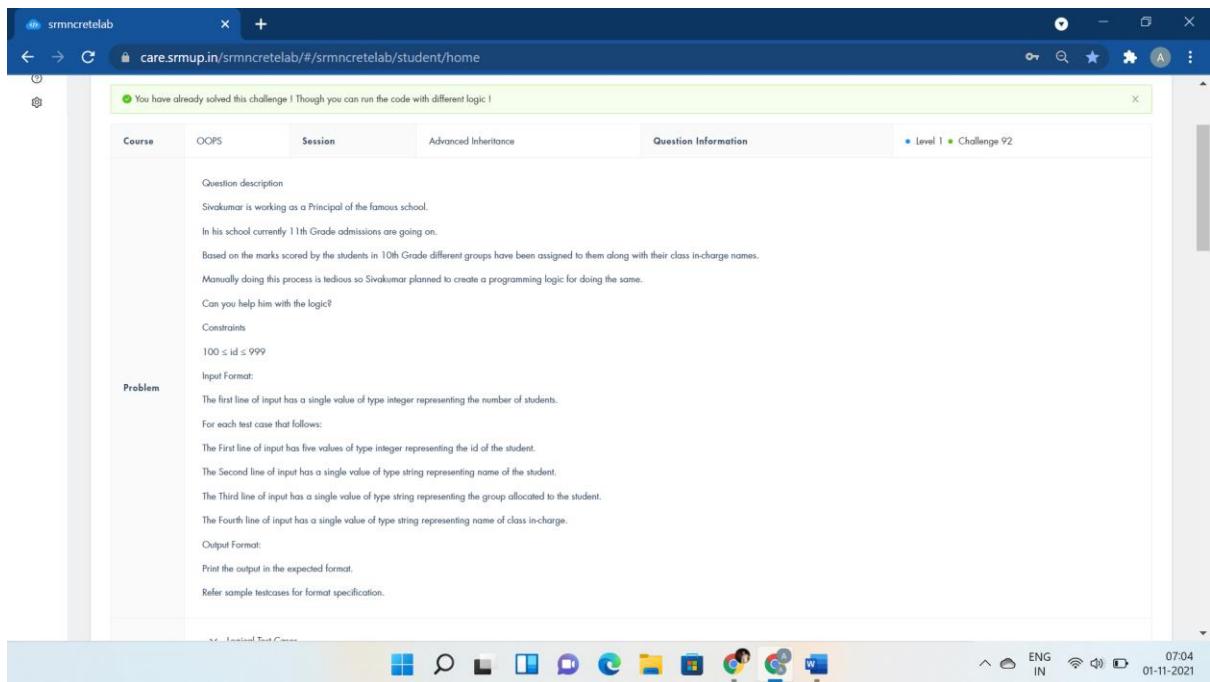
using namespace std;

class Employee{
public:
};

class Salary : public Employee{
public:
    int code,basic,hra,da,pf,total;
    string name,position;
    void getEmpDetails(){

```

```
cin>>code>>name>>position;  
}  
  
void getPayDetails(){  
    cin>>basic>>hra>>da>>pf;  
}  
  
void calculate(){  
    total=basic+hra+da-pf;  
}  
  
void display(){  
    cout<<"Employee Number:"<<code<<endl;  
    cout<<"Employee Name:"<<name<<endl;  
    cout<<"Employee Role:"<<position<<endl;  
    cout<<"Employee Net Pay:"<<total<<endl;  
}  
};  
  
int main()  
{  
    Salary s;  
    s.getEmpDetails();  
    s.getPayDetails();  
    s.calculate();  
    s.display();  
    return 0;  
}
```



```
#include <iostream>

using namespace std;

class Person{

};

class Teaching : public Person{

};

class Instructor : public Teaching{

public:

    int id;

    string name,group,staff;

    void accept_instructor_details(){

        cin>>id>>name>>group>>staff;

    }

    void display_instructor_details(){

        cout<<"Id:"<<id<<endl;

        cout<<"Name:"<<name<<endl;

        cout<<"Group:"<<group<<endl;

        cout<<"Staff:"<<staff<<endl;

    }

};

int main()

{



    int n;
```

```

cin>>n;

Instructor inst[n];

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

    inst[i].accept_instructor_details();

    inst[i].display_instructor_details();

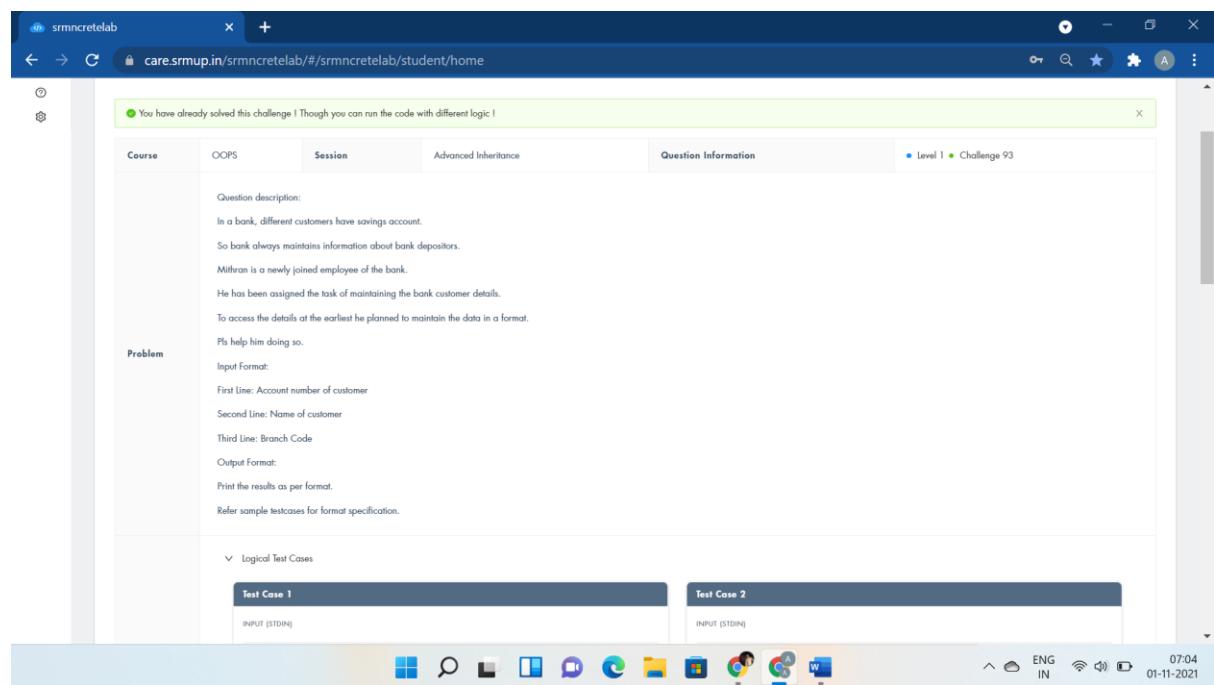
}

return 0;

cout<<"Instructor *inst;";

}

```



```

#include <iostream>

using namespace std;

class acc{
public:
    int no;
    void getacc(){
        cin>>no;
    }
};

class branch:public acc{
public:
    string name;
}

```

```

int code;

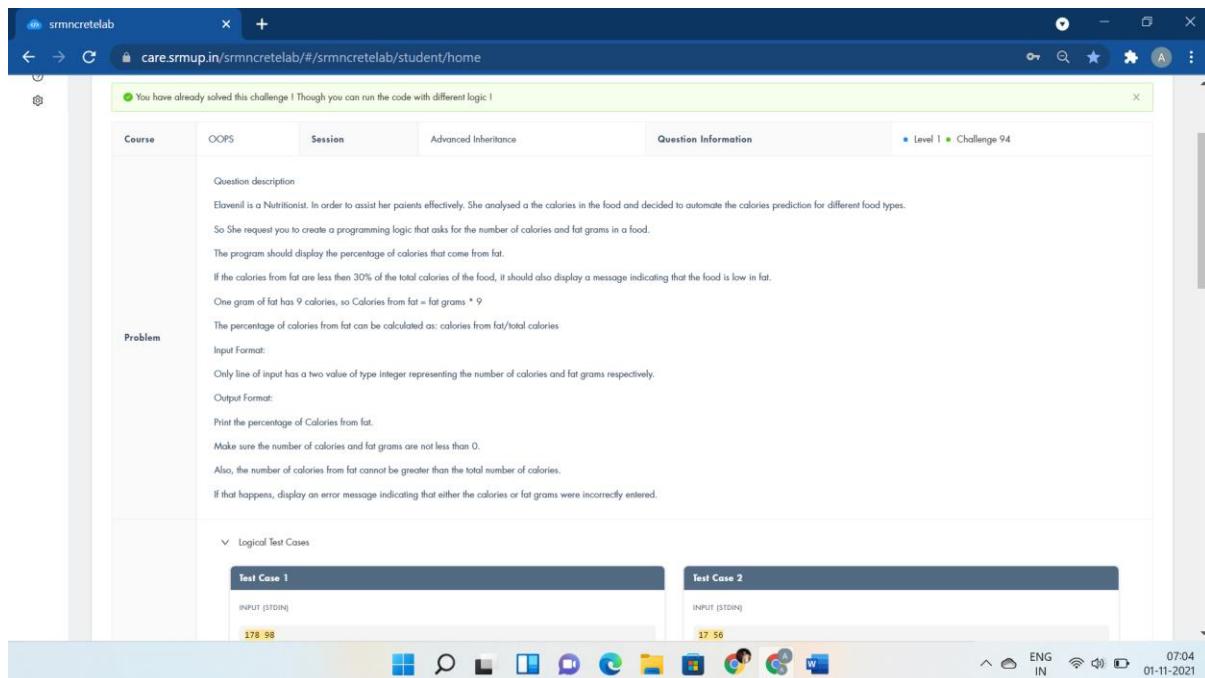
void getbranch(){
    cin>>name>>code;
}

void display(){
    cout<<"Acc No:"<<no<<endl;
    cout<<"Name:"<<name<<endl;
    cout<<"Branch Code:"<<code<<endl;
}

int main()
{
    branch b;
    b.getacc();
    b.getbranch();
    b.display();

    return 0;
}

```



```

#include <iostream>

using namespace std;

class Food{
};

```

```

class Nutritionist:public Food{
};

class Patient:public Nutritionist{

public:

float cal,fat;

void calorie(){

    cin>>cal>>fat;

}

void dplan(){

if(cal<fat)

cout<<"Fatgrams cannot be less than 0 or greater than calories"<<endl;

cout<<"Calories from fat: "<<fat*9/cal*100<<"%";

}

};

int main()

{

Patient p;

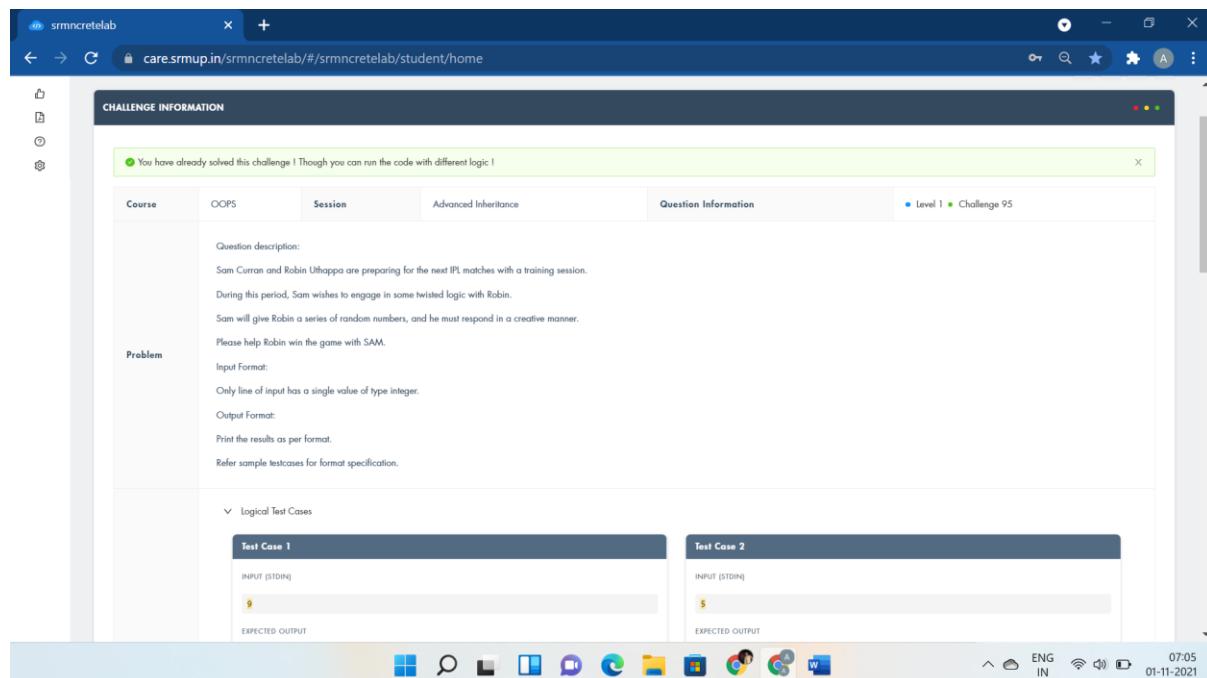
p.calorie();

p.dplan();

return 0;

}

```



```
#include <iostream>
```

```
using namespace std;

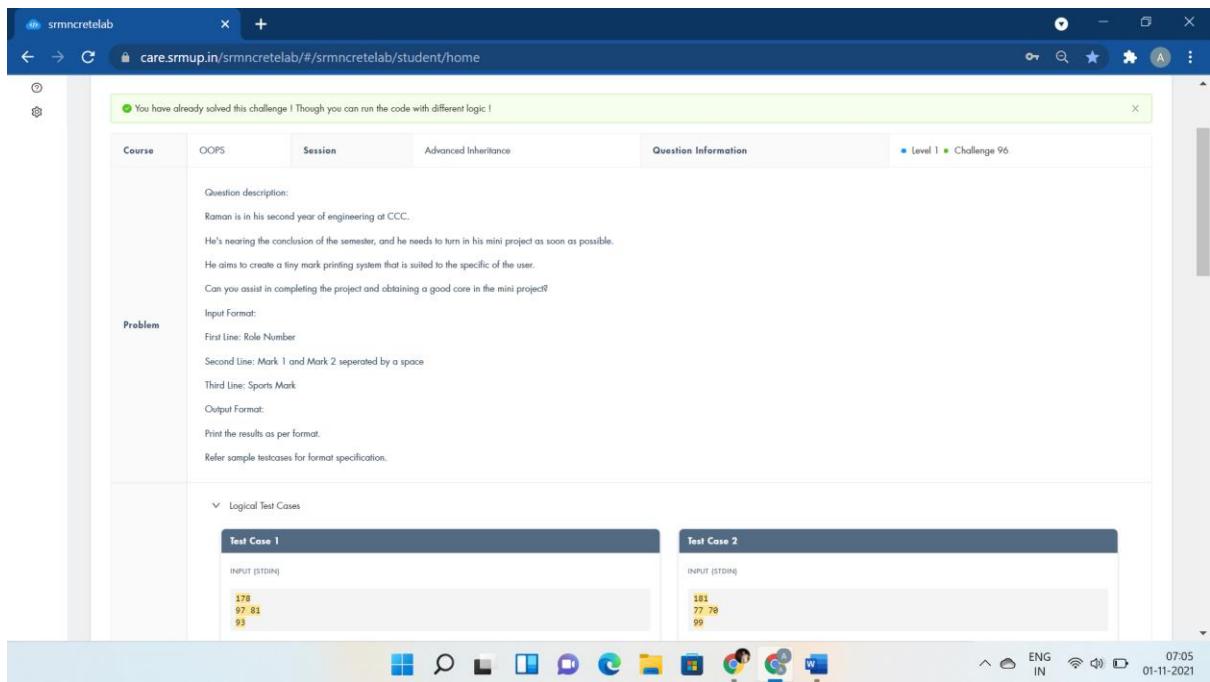
class Sam{

};

class Robin:public Sam{

public:
    int rows;
    void read(int y){
        rows=y;
    }
    void display(){
        for(int i=0;i<rows;i++){
            for(int j=0;j<rows;j++){
                cout<<"* ";
            }
            cout<<endl;
        }
    }
};

int main()
{
    Robin obj;
    int y;
    cin>>y;
    obj.read(y);
    obj.display();
    return 0;
}
```



```
#include <iostream>

using namespace std;

class student{

public:

int roll,m1,m2;

void get(){

cin>>roll>>m1>>m2;

}

};

class sports{

public:

int sp;

void getsm(){

cin>>sp;

}

};

class statement : public student, public sports{

public:

void display(){

cout<<"Roll No:"<<roll<<endl;

cout<<"Total:"<<m1+m2+sp<<endl;

cout<<"Average:"<<(m1+m2+sp)/3<<endl;

}

};
```

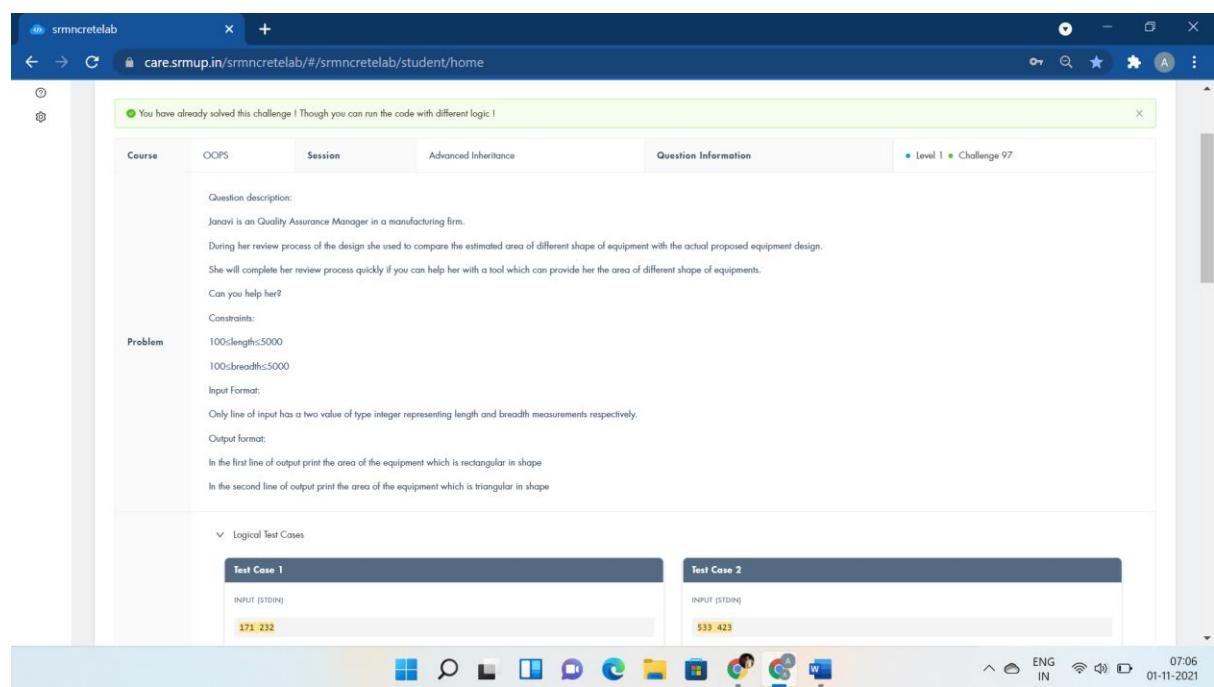
```

};

int main()
{
    statement obj;
    obj.get();
    obj.getsm();
    obj.display();

    return 0;
}

```



```

#include <iostream>

using namespace std;

class Shape{
public:
    int len,wid;
    void input(int l,int b){
        len=l;
        wid=b;
    }
};

class Rectangle: public Shape{
public:

```

```
void output(){
    cout<<len*wid<<endl;
}

};

class Triangle: public Shape{
public:
void output(){
//if((len*wid)%2==0)
    cout<<0.5*len*wid<<endl;
//else
//cout<<len*wid/2+1<<endl;
}

};

int main()
{
    int l,b;
    cin>>l>>b;
    Rectangle rect;
    Triangle tri;
    rect.input(l,b);
    tri.input(l,b);
    rect.output();
    tri.output();

    return 0;
}
```

The screenshot shows a web browser window with the URL [care.srmup.in/srmncretelab/#/srmncretelab/student/home](http://care.srmup.in/srmncretelab/#/srmncretelab/student/home). The page displays a challenge titled "Challenge 98" under the "Question Information" tab. The challenge description is as follows:

You have already solved this challenge! Though you can run the code with different logic!

**Course:** OOPS  
**Session:** Advanced Inheritance  
**Question Information:** Level 1 • Challenge 98

**Question description:**

In a bank, different customers have savings account.

Some customers may have taken a loan from the bank. So bank always maintains information about bank depositors and borrowers.

Design a Base class Customer [name, phone-number].

Derive a class Depositor[accno, balance] from Customer.

Again, derive a class Borrower [loan-no, loan-amnt] from Depositor.

Write necessary member functions to read and display the details of 'n'

**Input Format:**

First Line: N representing number of testcases  
Second Line: Customer name  
Third Line: Customer mobile number  
Forth Line: Customer Acc number  
Fifth Line: Customer balance  
Sixth Line: Customer Loan number  
Seventh Line: Loan amount  
**Output Format:**  
Print the results as per format.  
Refer sample testcases for format specification.

The browser interface includes a toolbar with various icons and a status bar at the bottom showing the date and time (01-11-2021, 07:06).

```
#include <iostream>

using namespace std;

class customer{

public:
    int no;
    long long int mobile;
    string name;

    void acceptc(){
        cin>>name>>mobile>>no;
    }
};

class deposit:public customer{
public:
    int bal;
    void acceptd(){
        cin>>bal;
    }
    void dispd(){
        cout<<"Customer Name:"<<name<<endl;
        cout<<"Customer Phone No:"<<mobile<<endl;
        cout<<"Customer A/c No:"<<no<<endl;
        cout<<"Balance:"<<bal<<endl;
    }
}
```

```

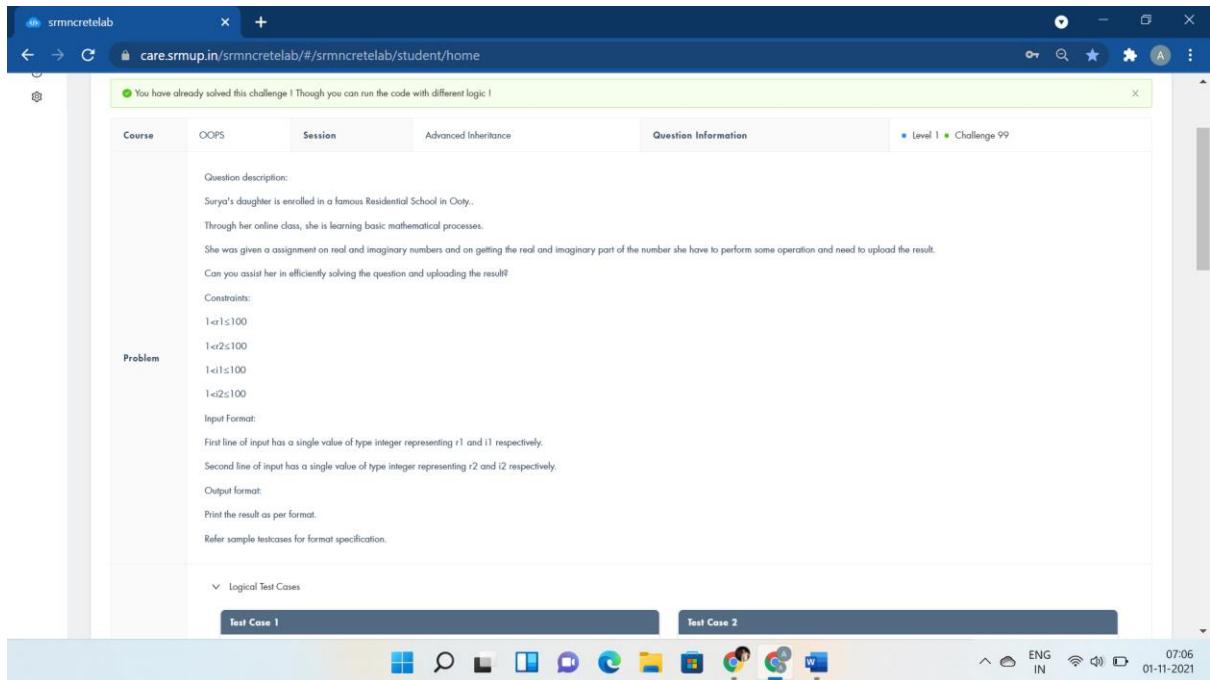
};

class borrow:public deposit{
public:
long long int loan_no,amt;
void acceptb(){
    cin>>loan_no>>amt;
}

void dispb(){
    cout<<"Loan No:"<<loan_no<<endl;
    cout<<"Loan Amount:"<<amt<<endl;
}

int main()
{
    int n;
    cin>>n;
    borrow b1[n];
    for(int i=0;i<n;i++){
        b1[i].acceptc();
        b1[i].acceptd();
        b1[i].acceptb();
        b1[i].dispd();
        b1[i].dispb();
    }
    return 0;
}

```



```
#include <iostream>

using namespace std;

class Receive{

public:

int r1,i1,r2,i2,r3,i3;

void getdata(){

    cin>>r1>>i1>>r2>>i2;

}

};

class Operate : public Receive{

public:

void add(){

    r3=r1+r2;

    i3=i1+i2;

}

};

class Present :public Operate{

public:

void output(){

    cout<<r1<<"+"<<i1<<"i"<<endl;

    cout<<r2<<"+"<<i2<<"i"<<endl;

    cout<<r3<<"+"<<i3<<"i"<<endl;

}

};
```

```

};

int main()
{
    Present calc;

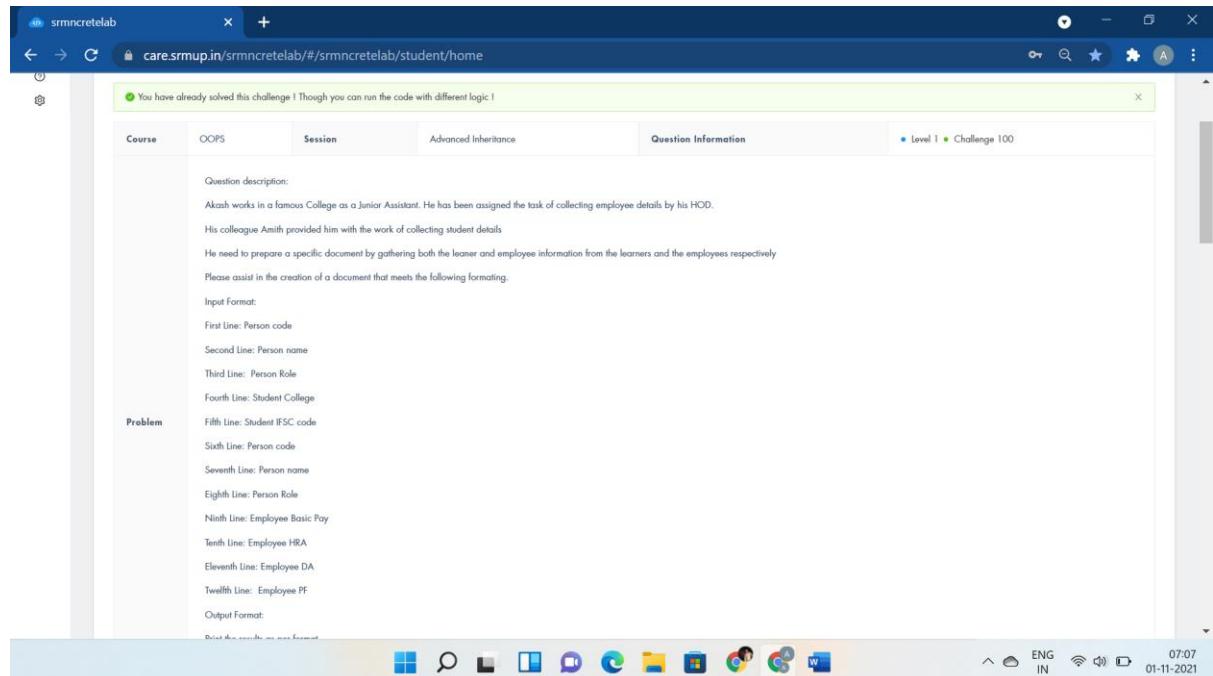
    calc.getdata();

    calc.add();

    calc.output();

    return 0;
}

```



```

#include <iostream>

using namespace std;

class Person{

};

class Employee : private Person{

};

class Student : private Person{

public:

    int n1,n2,basic,hra,da,pf;

    string name1,role1,col;ifsc,name2,role2;

    void getdetail(){

        cin>>n1>>name1>>role1>>col>>ifsc>>n2>>name2>>role2;

    }
}

```

```

void getEmployeeDetails(){
    cin>>basic>>hra>>da>>pf;
}

void student_display(){
    cout<<"Person number:"<<n1<<endl;
    cout<<"Person name:"<<name1<<endl;
    cout<<"Person Role:"<<role1<<endl;
    cout<<"Student college Name:"<<col<<endl;
    cout<<"Student IFSC:"<<ifsc<<endl;
    cout<<"Person number:"<<n2<<endl;
    cout<<"Person name:"<<name2<<endl;
    cout<<"Person Role:"<<role2<<endl;
}

void employee_display(){
    cout<<"Employee Basic pay:"<<basic<<endl;
    cout<<"Employee HRA:"<<hra<<endl;
    cout<<"Employee DA:"<<da<<endl;
    cout<<"Employee PF:"<<pf<<endl;
    cout<<"Employee Net Pay:"<<basic+hra+da-pf<<endl;
}

};

int main()
{
    Student e;
    e.getdetail();
    e.getEmployeeDetails();
    e.student_display();
    e.employee_display();

    return 0;
    cout<<"s.student_display();";
}

```