

INPUT OUTPUT:-

```
#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;
}
```

Problem Description:
Arav and Aaron are participating in the Bike racing.
Arav have crossed some milestones earlier and Aaron crossed some milestones earlier during their racing,because they have changed their speeds at different times.
Both of them like to know the difference in speeds between them at different stages of racing.
Can you help finding the speed difference between Arav and Aaron?
Constraints:
20≤ aravspeed ≤100
20≤ aaronspeed ≤100
Input Format :
The first line of input represents the speed of Arav.
The second line of input represents the speed of Aaron.
Output Format:
Print difference between the driving speed of two participants in a single line.

Logical Test Cases

Test Case 1	Test Case 2
INPUT [STDIN] 74 51	INPUT [STDIN] 76 89
EXPECTED OUTPUT	EXPECTED OUTPUT

```
#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;
}
```

Problem Description:
Three brothers want to take a photo with family members. The photographer is capturing the photo from a long distance.
Some of the family members are standing behind that brothers and those people are not visible to the photographer.
So the photographer gets confused with the heights of three brothers.
To get clarity, he asked, "who is the tallest person among those three brothers? But no one responded clearly.
Can you help the photographer in finding the tallest among the three brothers?"

Constraint:

Problem

60 ≤ bro1 ≤ 80
60 ≤ bro2 ≤ 80
60 ≤ bro3 ≤ 80

Input Format :

The only line of input has three numbers bro1,bro2 and bro3 of type integers separated by a space which represents the height of three brothers.

Output Format:

Print the height of the tallest person among three brothers.

Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN) 61 79 79	INPUT (STDIN) 65 66 80

```
#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;
}
```

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 4
--------	------	---------	---------------	----------------------	-----------------------

Problem Description:
Siva and Guru are playing a mathematical game.
Guru says a random numbers to Siva and he needs to convert the numbers to words.
Since Guru is very fast in telling the numbers, Siva cant able to cope up with his friend in converting it to words.
Can you help Siva in converting the particular number to words by creating a simple programming logic.

Constraints:
1≤N≤1000

Input Format:
The Only line of input has a single integer representing the number said by Guru.

Output Format:
In the only line of output print the number in words.
Refer the sample test cases for formating.

Logical Test Cases

Test Case 1	Test Case 2
INPUT [STDIN] 1894	INPUT [STDIN] 7631

Type here to search

25°C 13:43 30-09-2021

```
#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;  
};
```

```

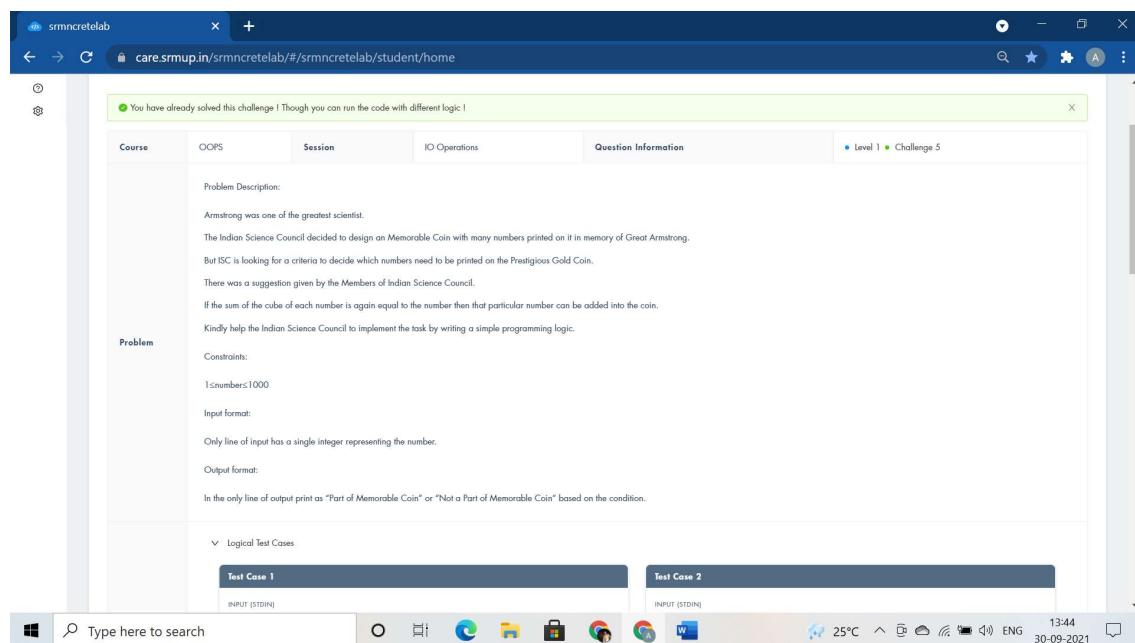
dig/=10;

}

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";

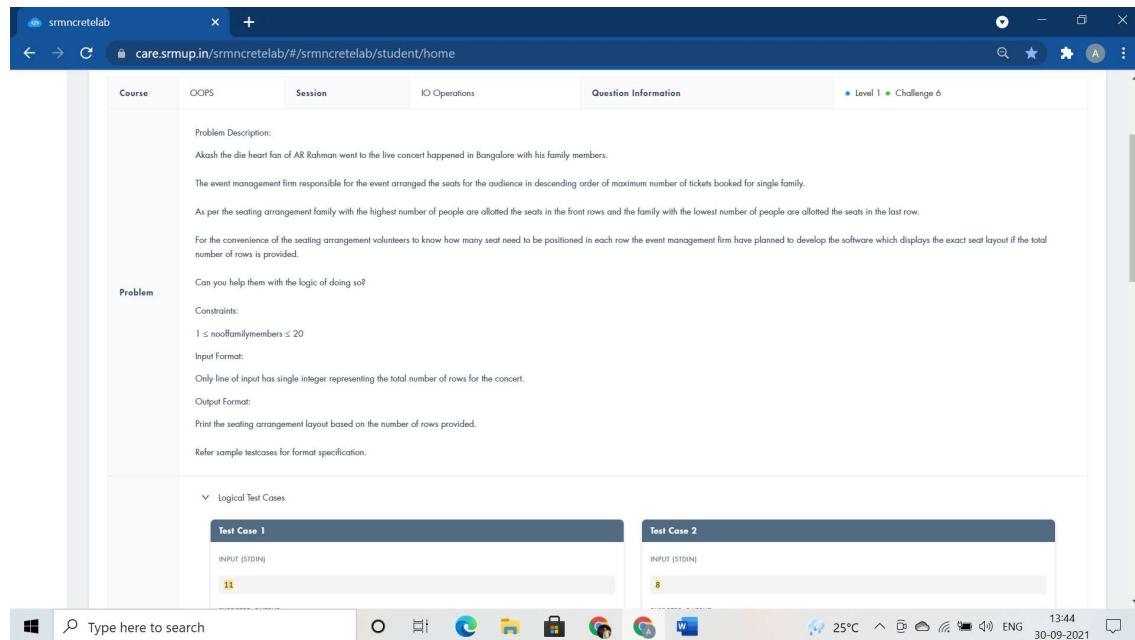
else

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--)

    {

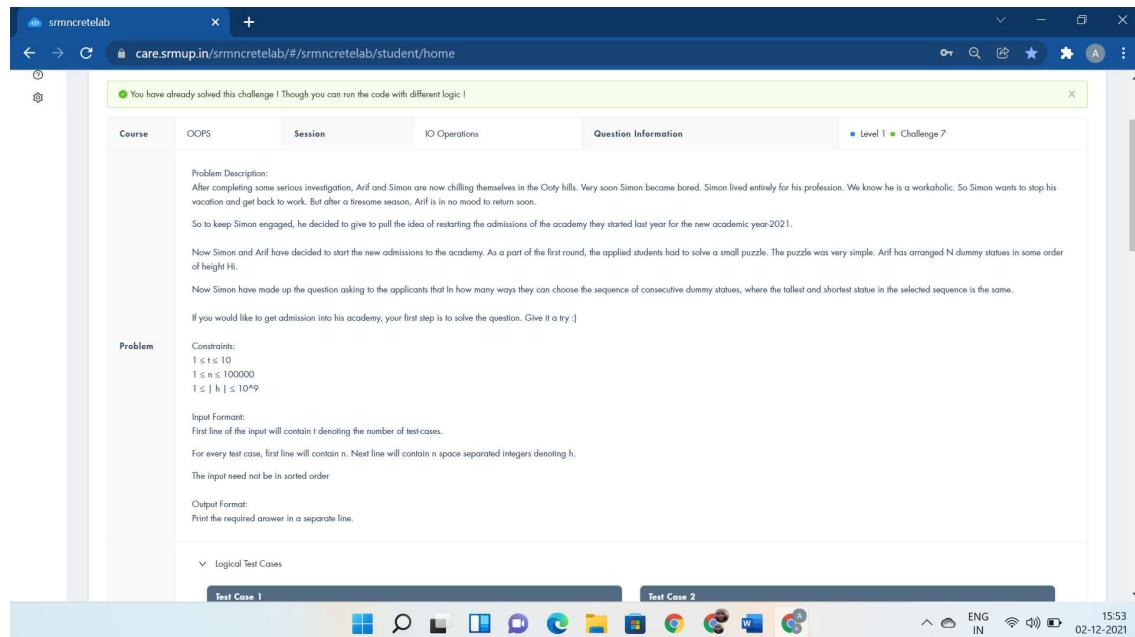
```

```

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

return 0;
}

```



```

#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++;

    }

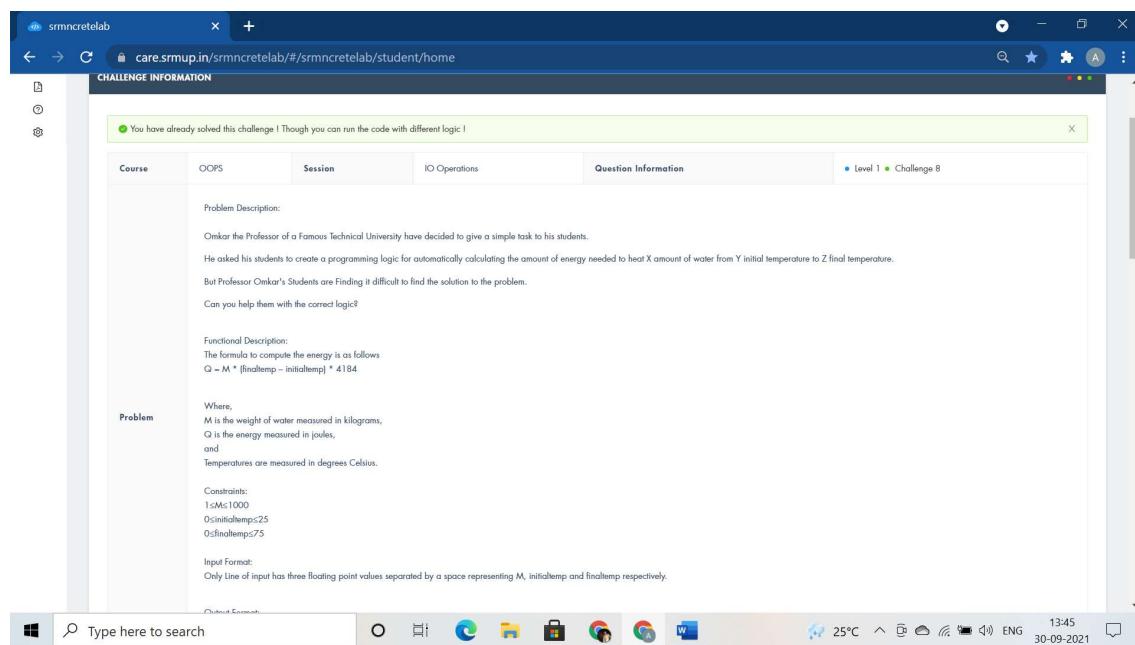
}

cout<<count<<endl;

}

return 0;
}

```



```

#include <iostream>

using namespace std;

int main()

{

```

```

int M,initialtemp,finaltemp;

float Q;

cin>>M>>initialtemp>>finaltemp;

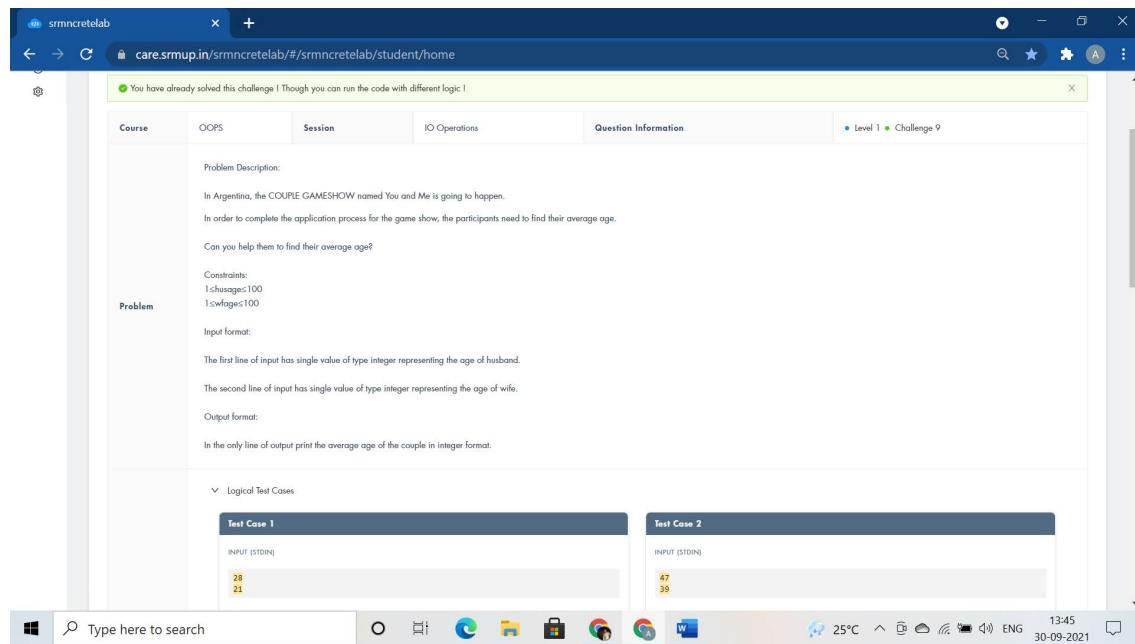
Q=M*(finaltemp - initialtemp)*4184;

cout<<""<<Q;

return 0;

}

```



```

#include <iostream>

using namespace std;

int main()

{
    int husage,wfage,coupleavgage;

    cin>>husage>>wfage;

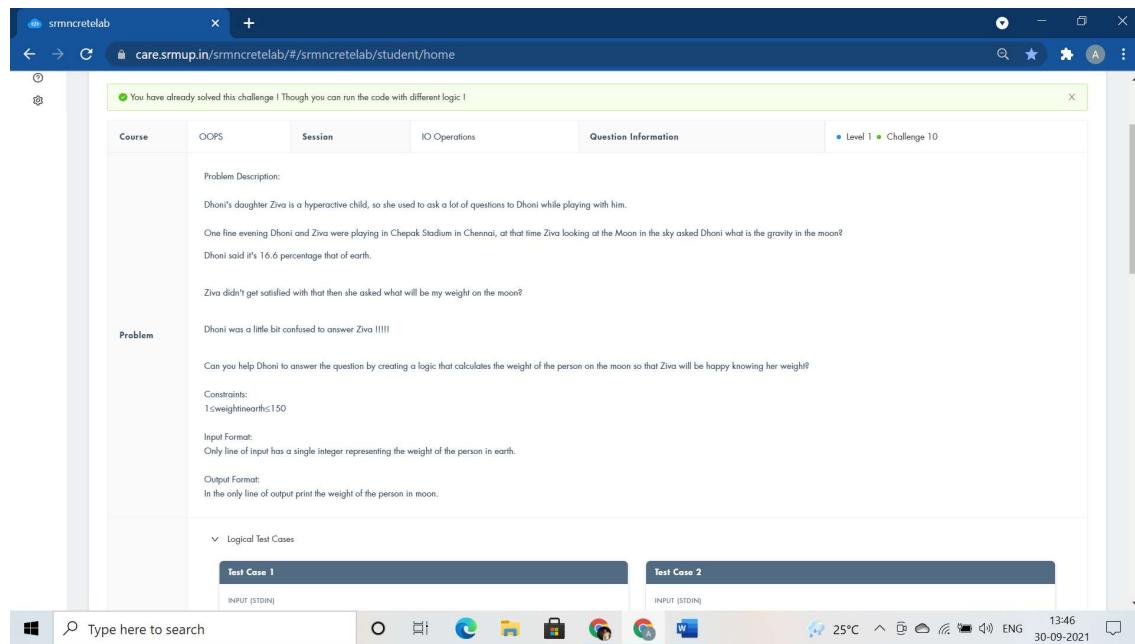
    coupleavgage=(husage+wfage)/2;

    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 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.

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;

};

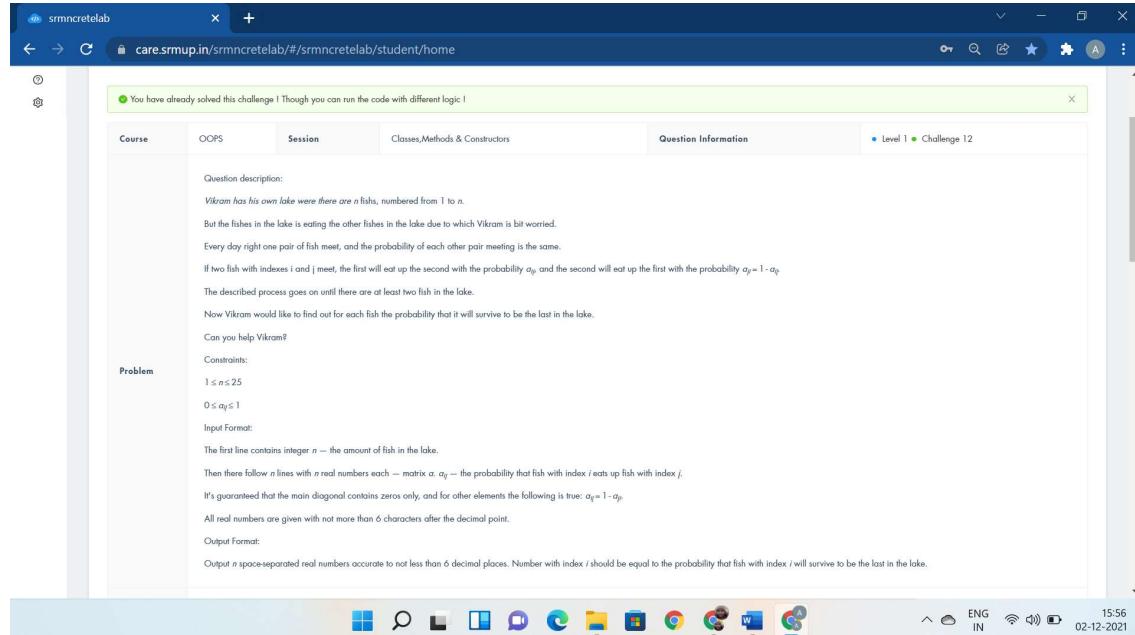
int main(){

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++)

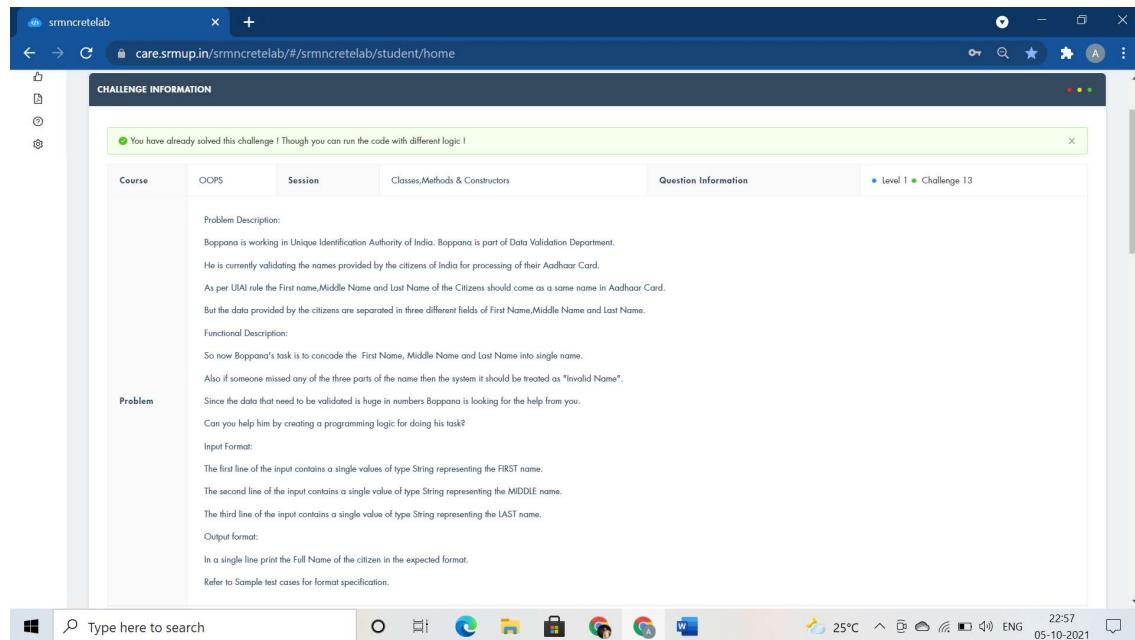
```

```

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)
{

```

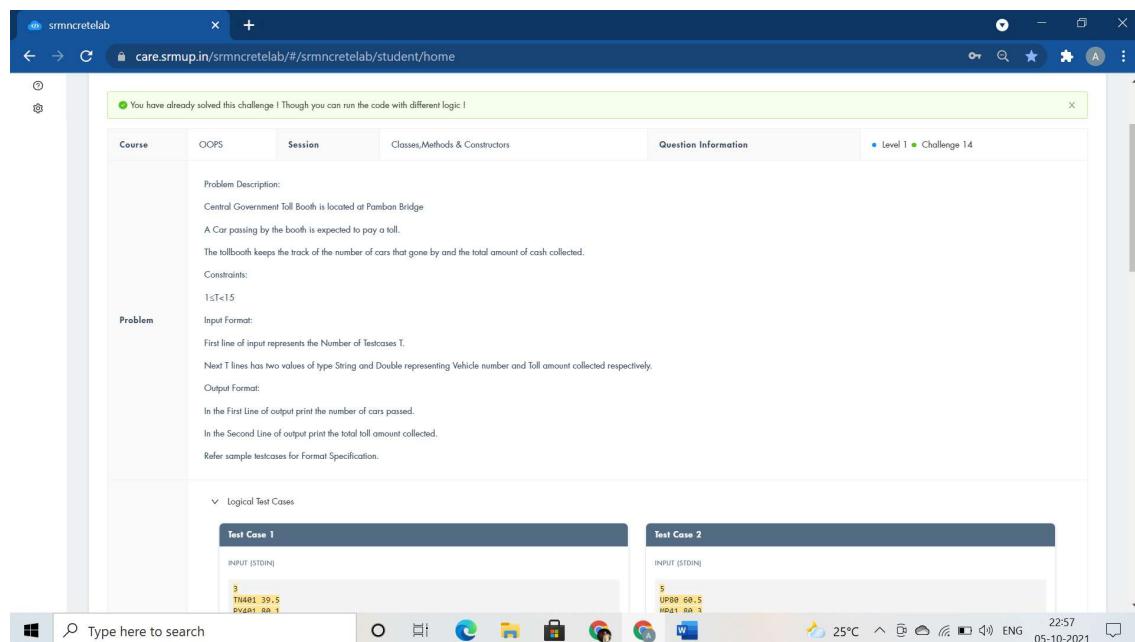
```

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;
}

```



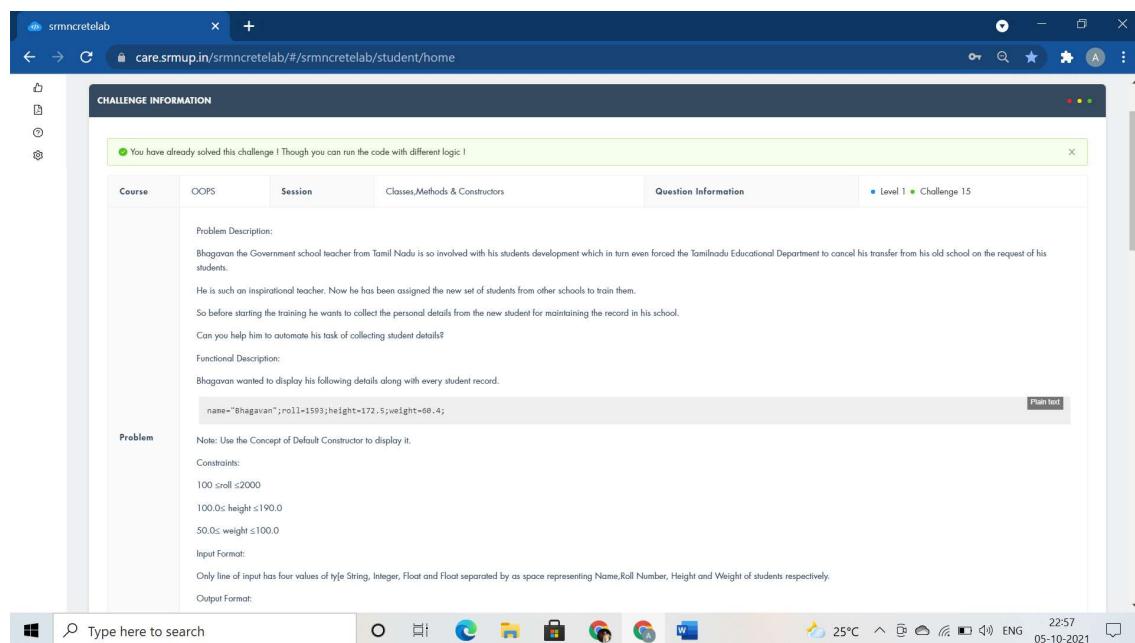
```
#include <iostream>
using namespace std;
class TollBooth
{
public:
int cars;
float tollcollected;
TollBooth(){
    cars=0;
    tollcollected=0;
}
void payingcar(double pay){
    cars++;
    tollcollected+=pay;
}
void nonpayingcar(){
    cars++;
}
void display(){
    cout<<cars<<endl<<tollcollected<<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();

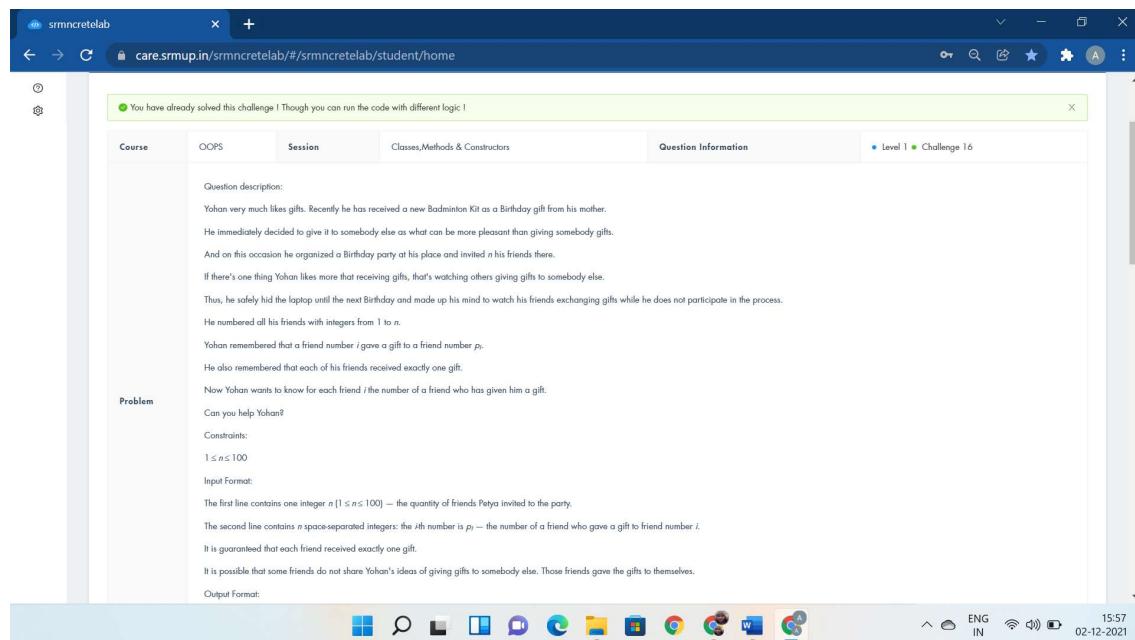
    s1.displaydata();

    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();

}
```

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 17

Question description

To celebrate the Reunion of 96 Batch of the Famous School the Ram and Jannu the organizers of the event decided to buy n liters of Fruit Drinks.

However, an unexpected difficulty occurred in the shop: it turned out that Fruit Drinks is sold in bottles 0.5, 1 and 2 liters in volume.

At that, there are exactly a bottles 0.5 in volume, b one-liter bottles and c of two-liter ones.

The organizers have enough money to buy any amount of Fruit Drinks.

What did cause the heated arguments was how many bottles of every kind to buy, as this question is pivotal for the distribution of Fruit Drinks among the Friends.

Your task is to count the number of all the possible ways to buy exactly n liters of Fruit Drinks and persuade the organizers that this number is too large.

All the bottles of Fruit Drinks are considered indistinguishable, i.e. two variants of buying are different from each other only if they differ in the number of bottles of at least one kind.

Constraints:

$1 \leq n \leq 10000$

$0 \leq a, b, c \leq 5000$

Input Format:

The first line contains four integers representing n , a , b , c respectively.

Output Format:

Print the unique number representing the solution to the problem.

If it is impossible to buy exactly n liters of Fruit Drinks, print 0.

Logical Test Cases

Test Case 1 Test Case 2

15:58 02-12-2021 ENG IN

```
#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(){

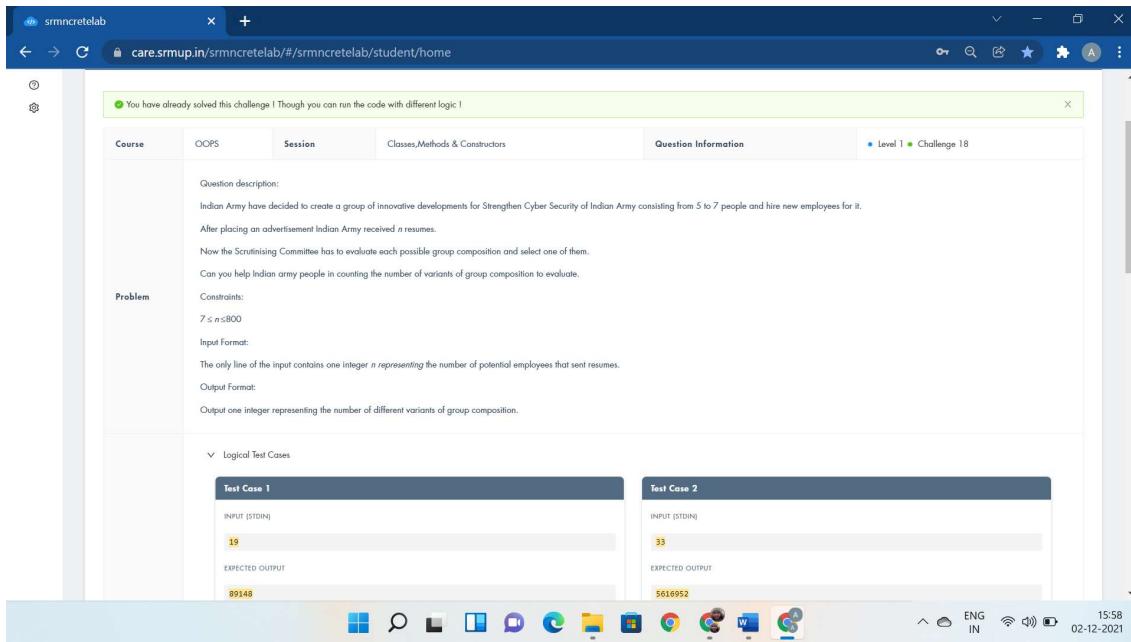
    Drinks Buy;

    Buy.Shop();

    Buy.display();

}
```

}



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

using namespace std;

class IndianArmy

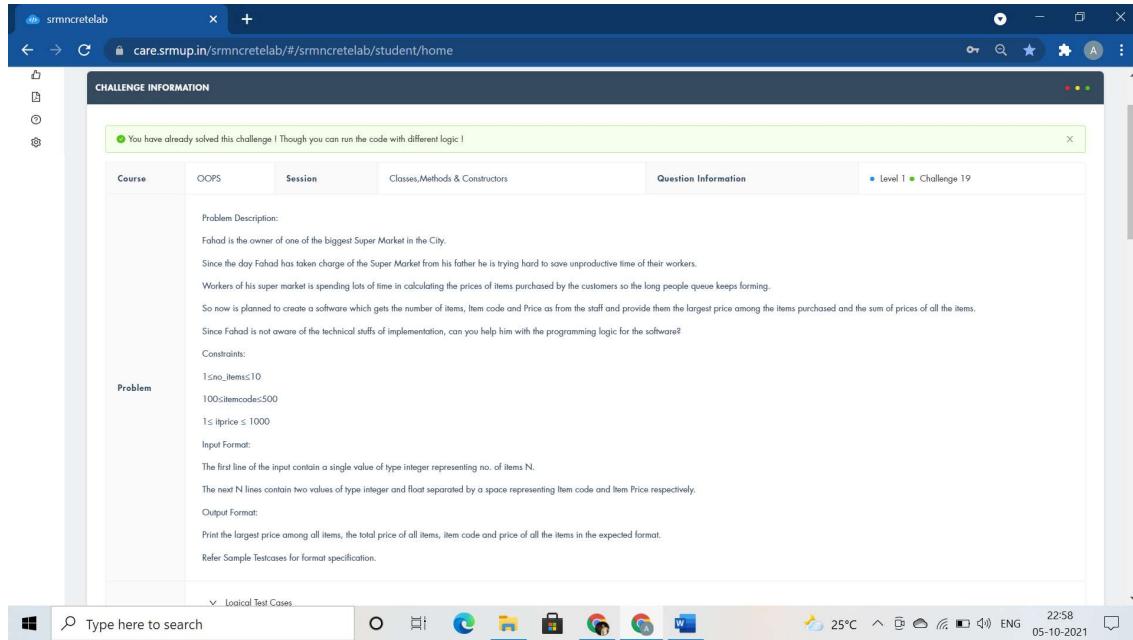
{
public:int ResumesofCamdicates(){

    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);

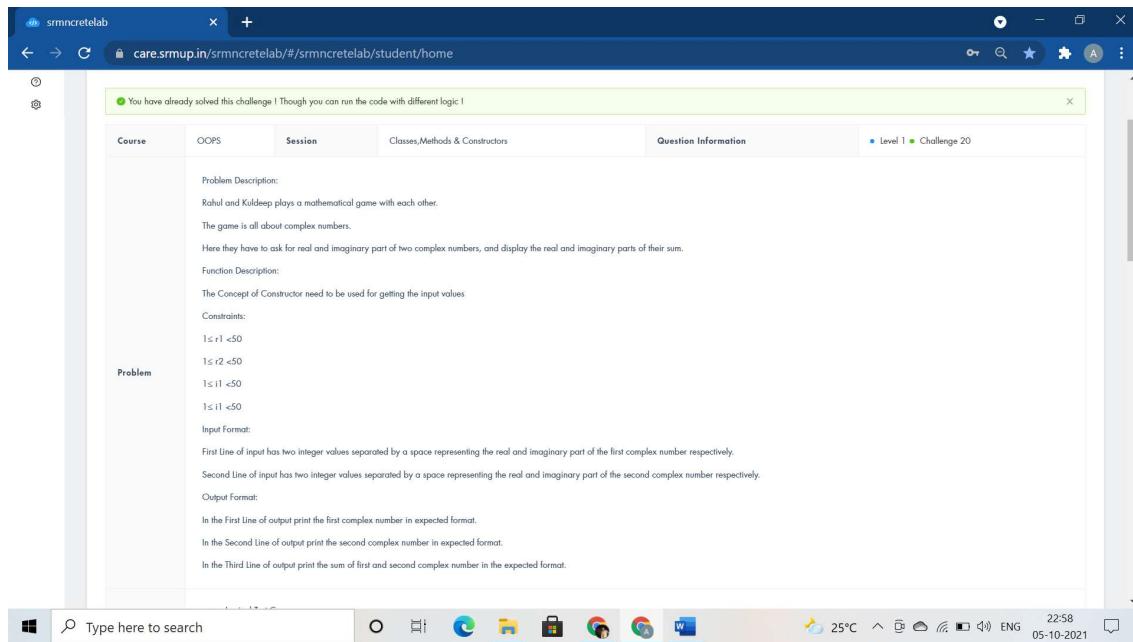
    order.largest();

    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();

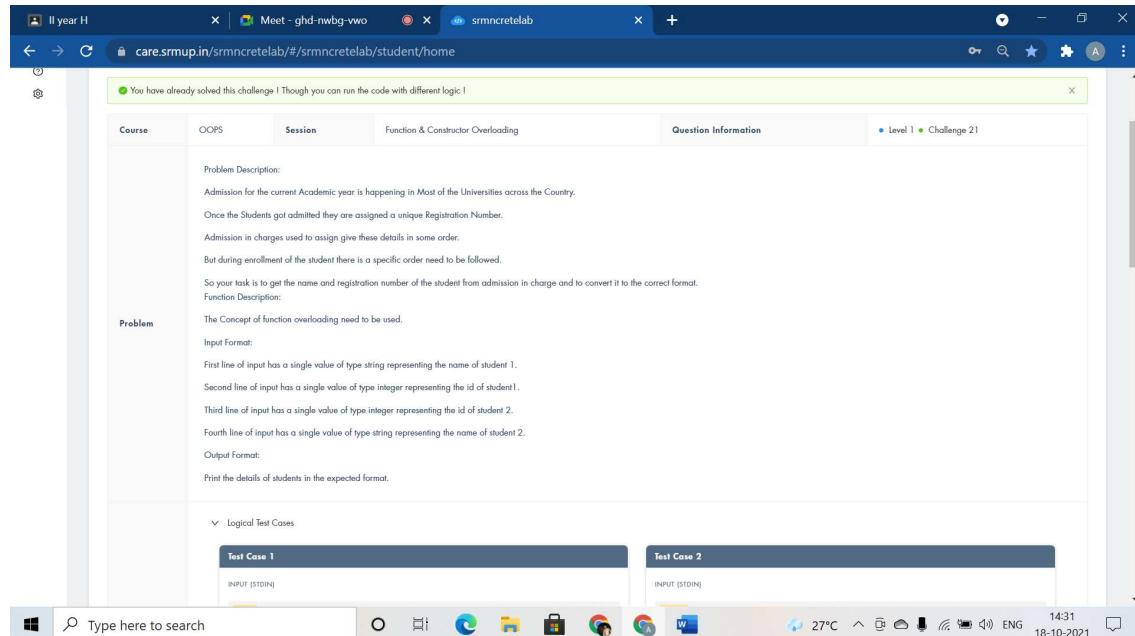
calculate.displaycomplex();

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);

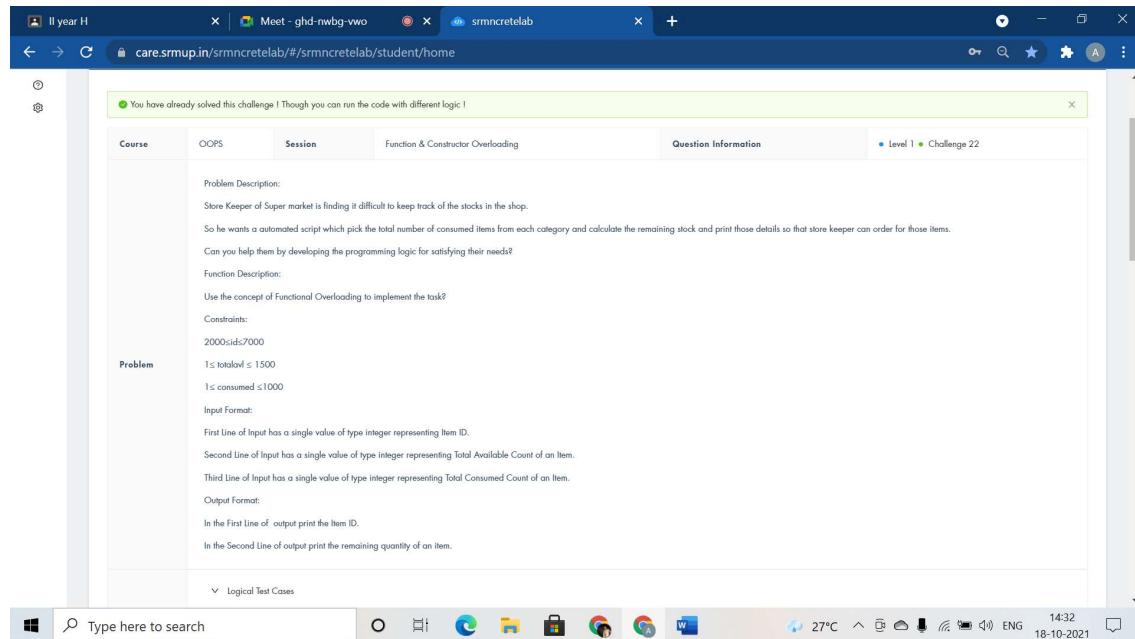
    cin>>id>>name;

    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;
}
```

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 23

Problem Description:
Gozal has n coins, the value of the i th coin is a_i .
Gozal wants to distribute all the coins between his pockets, but he cannot put two coins with the same value into the same pocket.
For example, if Gozal has six coins represented as an array $a=[1,2,4,3,3,2]$, he can distribute the coins into two pockets as follows: $\{1,2,3\}, \{2,3,4\}$.
Gozal wants to distribute all the coins with the minimum number of used pockets.

Can you Help him in doing that?

Constraints:
 $1 \leq n \leq 100$
 $1 \leq a_i \leq 100$

Input Format:
The first line of the input contains one integer n the number of coins.
The second line of the input contains n integers a_1, a_2, \dots , values of coins.

Output Format:
Print only one integer representing the minimum number of pockets Gozal needs to distribute all the coins so no two coins with the same value are put into the same pocket.

Logical Test Cases

Test Case 1	Test Case 2
INPUT [STDIN] 6 1 2 4 3 3 2	INPUT [STDIN] 4 1 2 2 2

```
#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){

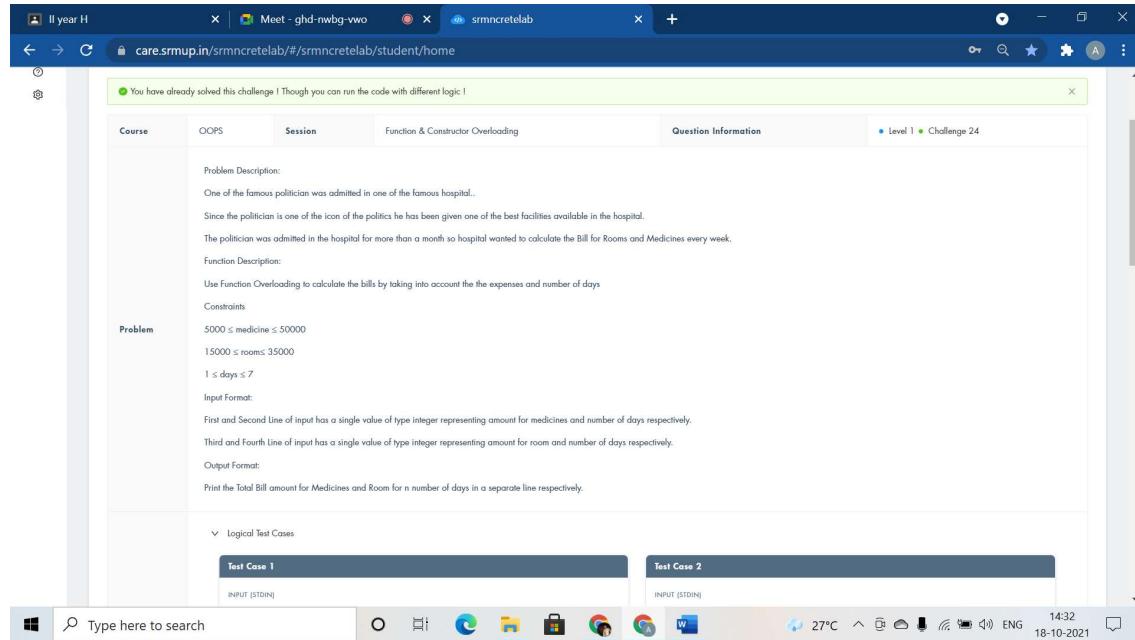
    if(n%mx==1 && n%11!=0)
        return mx+1;

    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);

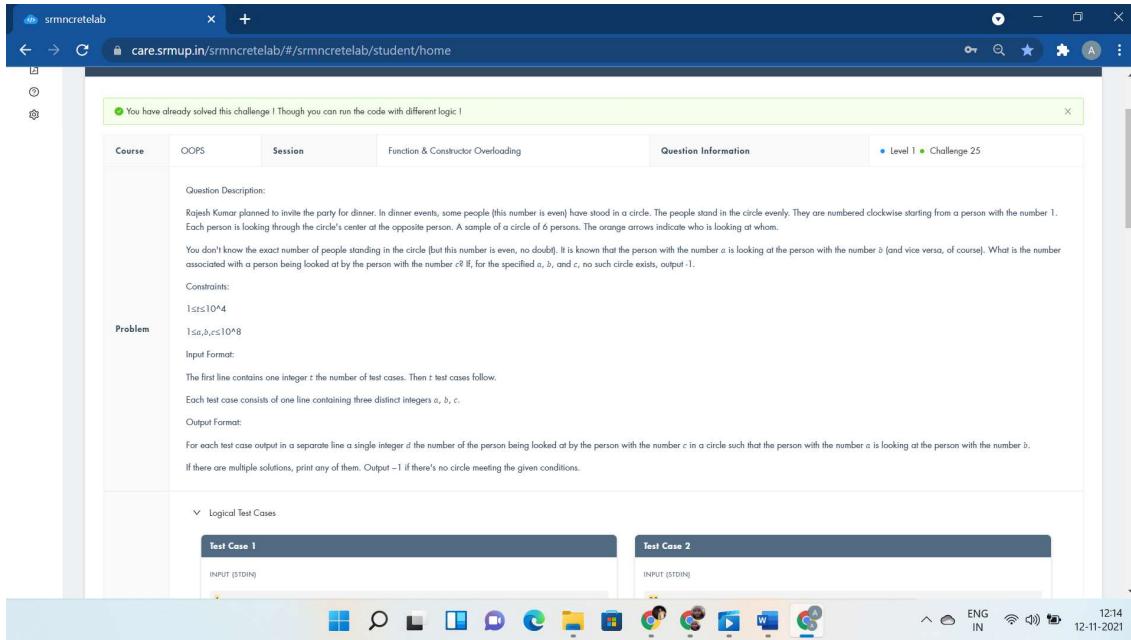
    cin>>roomrent>>days;

    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. The page displays a challenge titled "Function & Constructor Overloading". A message 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", "Session", and "Function & Constructor Overloading". The "Question Information" section indicates "Level 1" and "Challenge 26". The "Problem Description" section details a scenario about an athlete named Ram practicing for the Olympics. It specifies that Ram practices for 5 days a week, covering distances D1 through D5. The challenge requires using function overloading to calculate the total distance covered by Ram. Constraints are given as $1 \leq D1 \leq 100$, $1 \leq D2 \leq 100$, $1 \leq D3 \leq 100$, $1 \leq D4 \leq 100$, and $1 \leq D5 \leq 100$. The "Input Format" section describes the input as three lines of integers representing D1, D2, and D3 respectively. The browser's taskbar at the bottom shows various pinned icons and the system status bar indicating the date as 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;  
}
```

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 27

Question Description:
Valentina has given a multiset that means a set that can contain multiple equal integers containing $2n$ integers.
Determine if you can split it into exactly n pairs consists each element should be in exactly one pair.
So that the sum of the two elements in each pair is odd is divided by 2, the remainder is 1.

Constraints:
The input consists of multiple test cases. The first line contains an integer t the number of test cases. The description of the test cases follows.
The first line of each test case contains an integer n .

Input Format:
1 ≤ t ≤ 100
1 ≤ n ≤ 100
0 ≤ a_i ≤ 100

Output Format:
For each test case, print "Yes" if it can be split into exactly n pairs so that the sum of the two elements in each pair is odd, and "No" otherwise. You can print each letter in any case.

Logical Test Cases

Test Case 1 Test Case 2

INPUT [STDIN]

12:14 ENG IN 12-11-2021

```
#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)

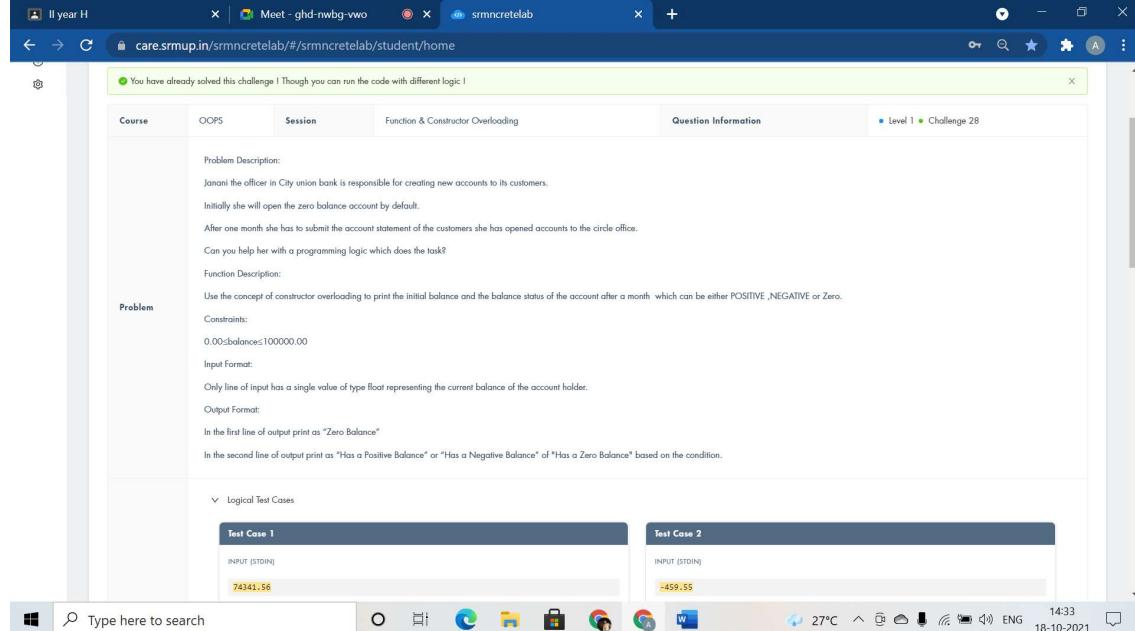
        cout<<"Yes"<<endl;

    else

        cout<<"No"<<endl;

    return 1;
}

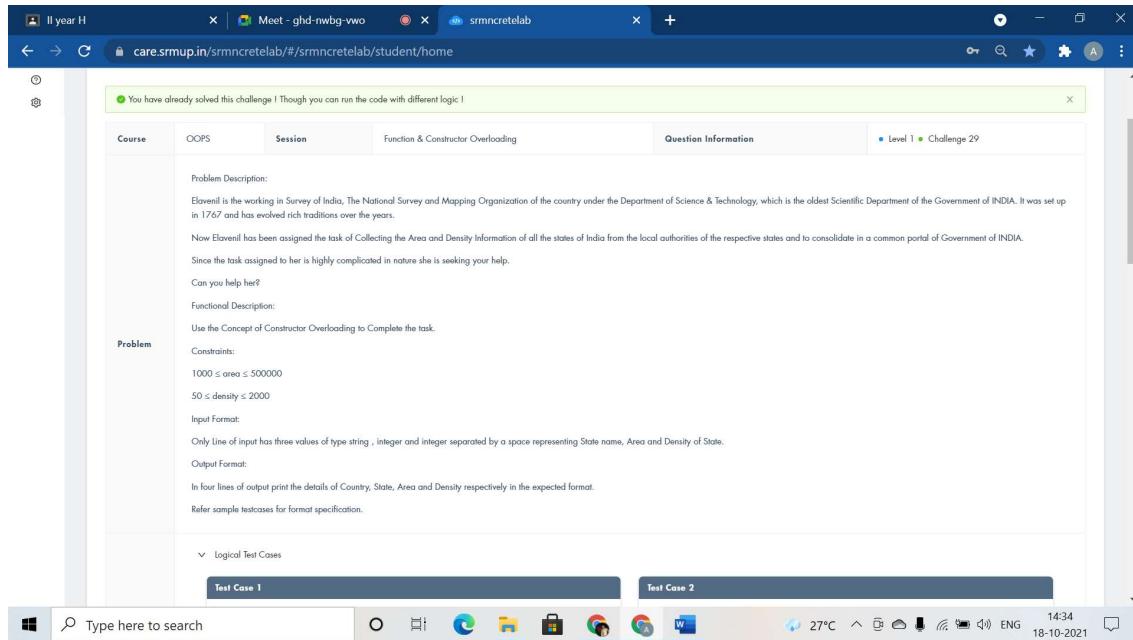
```



```
#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];

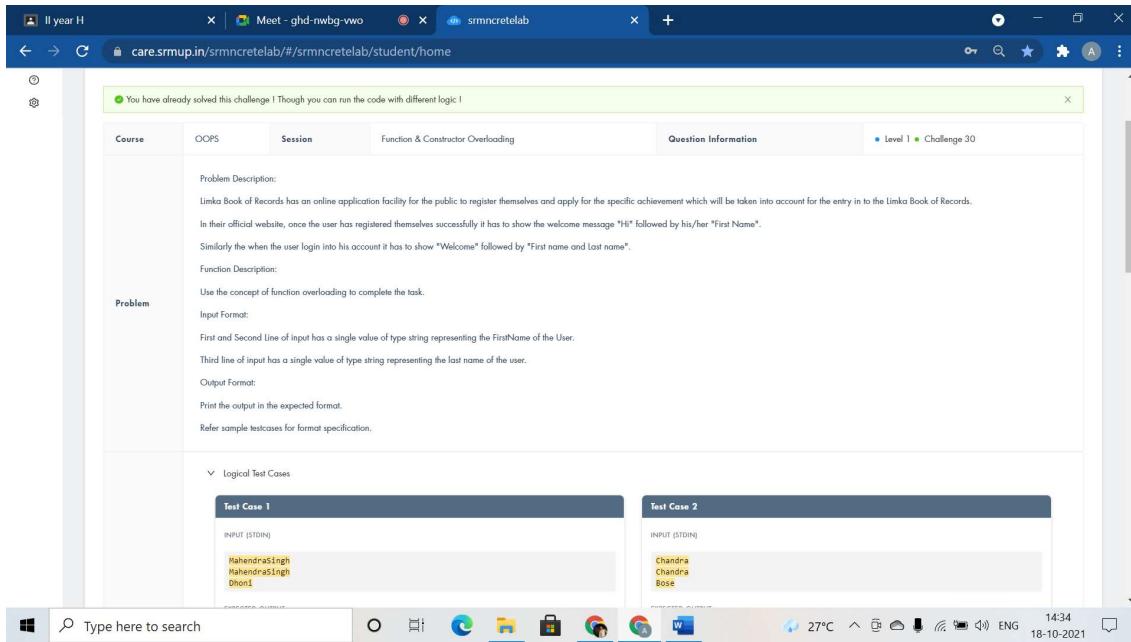
    int area,density;

    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:-

The screenshot shows a web browser window with a blue header bar. The address bar contains the URL `care.srmup.in/srmncretelab/#/srmncretelab/student/home`. The main content area has a yellow banner at the top stating "You have already solved this challenge! Though you can run the code with different logic!". Below this, there are tabs for "Course", "OOPS", "Session", "Operator Overloading" (which is selected), and "Question Information". The "Question Information" tab shows "Level 1" and "Challenge 31". The "Operator Overloading" section contains the following text:

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≤num, deno≤10⁴

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.

Below this, there is a section titled "Logical Test Cases" with two tabs: "Test Case 1" and "Test Case 2". Both tabs show the placeholder "INPUT [STDIN]". At the bottom of the browser window, there is a taskbar with various icons and system status indicators.

```

#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;
        }
    }
}
}

```

```

        }

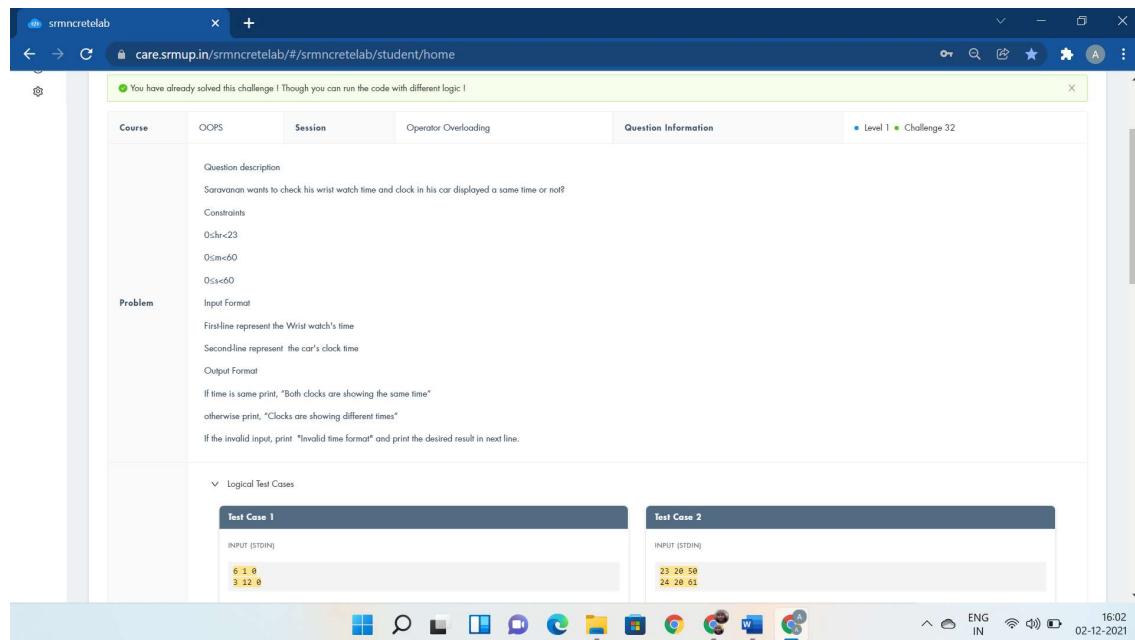
    }

    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

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 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.

Problem
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

17:21 27-10-2021

```
#include <iostream>

using namespace std;

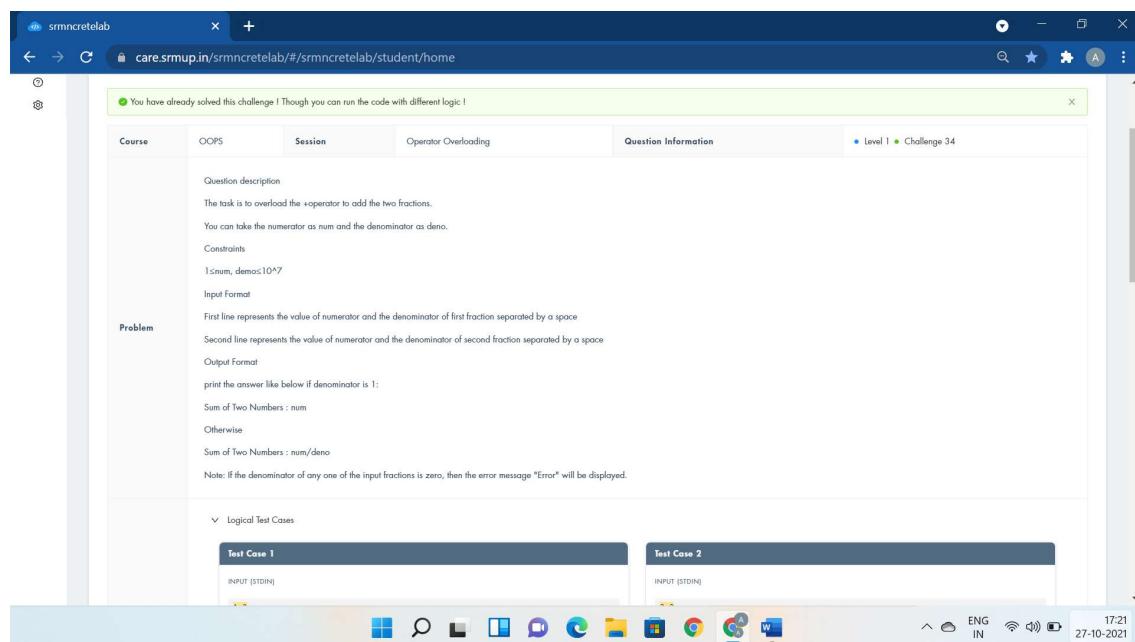
class Scrum{

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;
}
```

srmncretelab care.srmup.in/srmncretelab/#/srmncretelab/student/home

CHALLENGE INFORMATION

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	Question description Subash is a computer science student. He had a task from the Mathematics professor that to overload - operator as to find the determinant of 2 x 2 matrices. Can you help him to find the determinant of the square matrices of order 2. Constraints -10 <= a,b,c,d <= 10				Logical Test Cases
	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				

Test Case 1
INPUT [STDIN]
1 2
3 4
EXPECTED OUTPUT

Test Case 2
INPUT [STDIN]
2 8
3 2
EXPECTED OUTPUT

17:22 27-10-2021

```
#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;

}
```

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 36

Problem: 1sa,b,cs10^5

Question description: The math assignment says you will be given numbers, mostly with imaginary additions, that means complex numbers, and you need to add them and tell the answer in your answer script. You told your friend John that you don't know the addition of complex numbers, so John will write a program, which you can write in order to get the results of addition.

John knows Object oriented programming enough to complete the task.

Constraints: 1 ≤ a, b ≤ 10^5

Input Format: Three integers a b and c

Output format: First print the complex number a+bi
Next line print a + bi + c as i2.
Next line i2+a+bi

Logical Test Cases:

Test Case 1	Test Case 2
INPUT [STDIN] 2 5 3	INPUT [STDIN] 4 5 2
EXPECTED OUTPUT 2 + 5i	EXPECTED OUTPUT 4 + 5i

```
#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;
    }

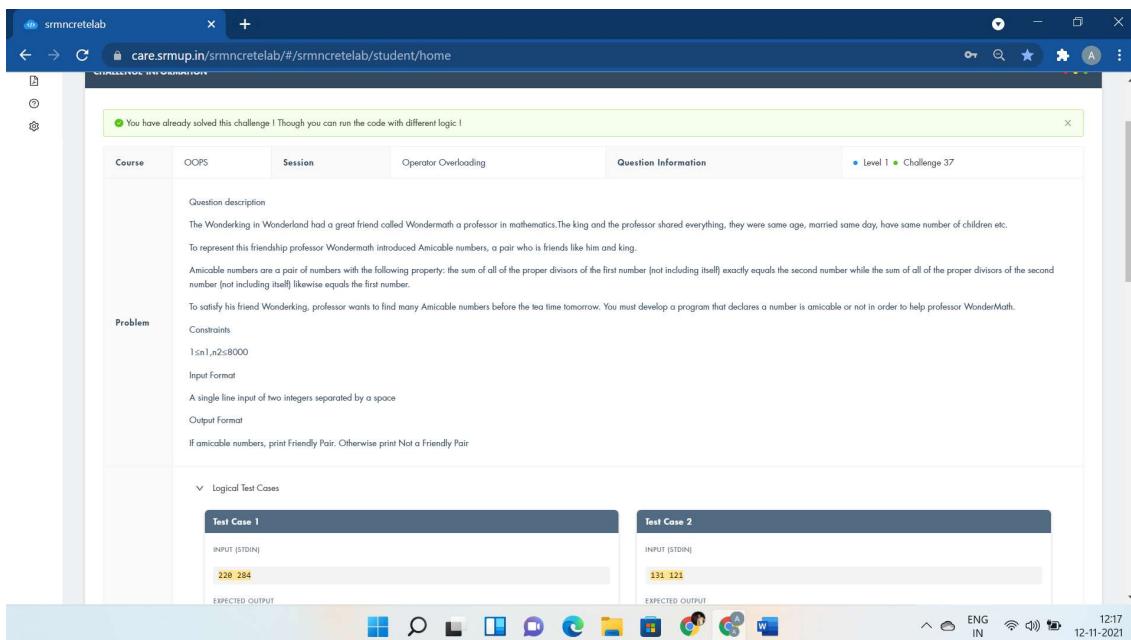
    Complex operator+(Complex obj) {
        Complex res;
        res.real = real + obj.real;
        res.imag = imag + obj.imag;
        return res;
    }

    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: COOPS Session: Operator Overloading Question Information Level I | 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 ≤ n ≤ 100
Function Description
Create a class Diff with a member functions sumofsquare and squareofsum with int datatype and use insertion overloading
Constraints
1 ≤ n ≤ 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;

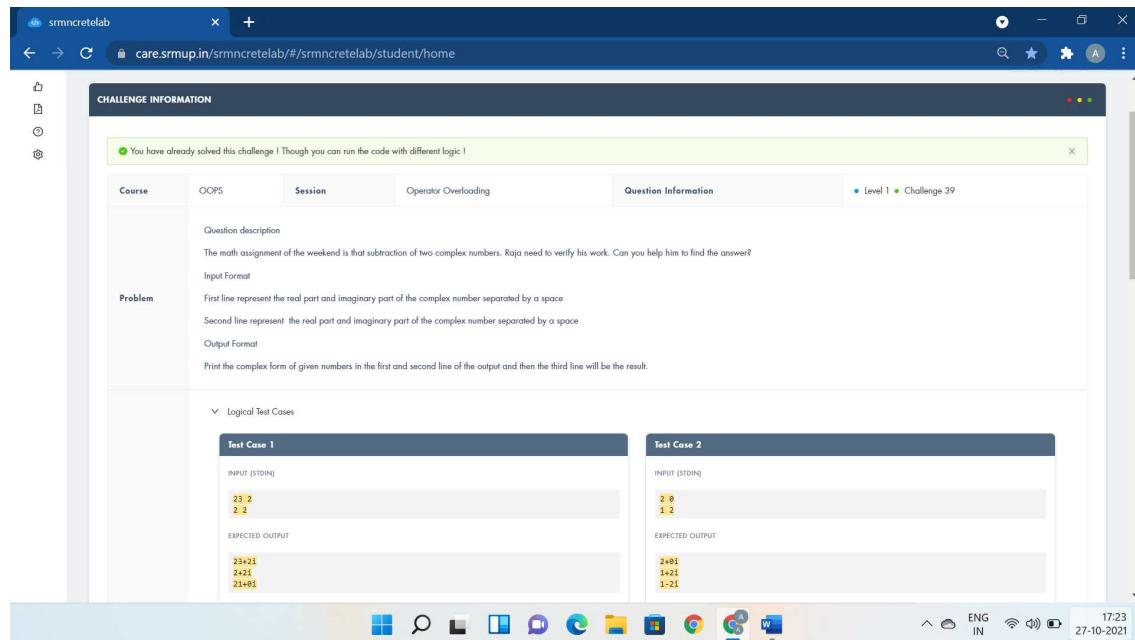
//int sq2=n*(n+1)*(2*n+1)/6;

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;
}

```

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

Course	Session	Operator Overloading	Question Information
			Level 1 • Challenge 40

Problem

Question description
The task is to overload the +operator to subtract the two fractions.
You can take the numerator as num and the denominator as deno.
Constraints
1≤num, deno≤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)	INPUT (STDIN)

```
#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:-

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

Course: OOPS **Session:** Inheritance **Question Information:** Level 1 • Challenge 41

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?

Problem:
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.

```
#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;

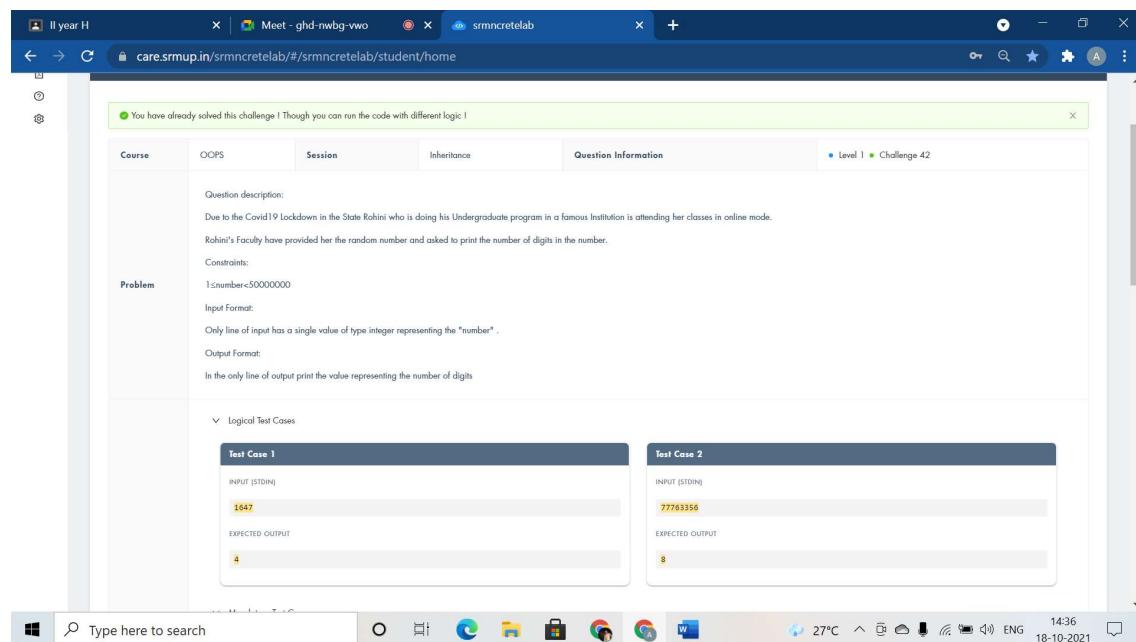
t.getdata();

t.display();

return 0;

}

```



```
#include <iostream>
```

```
using namespace std;

class Assignement{

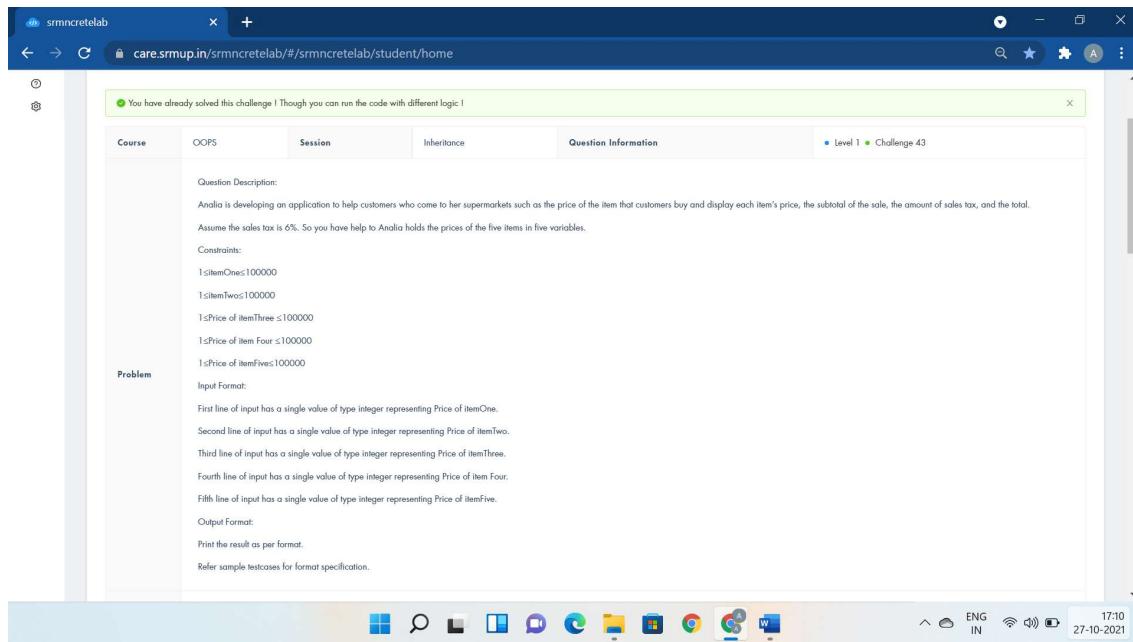
public:
    int num;

    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;
}
```



```
#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;

}

};

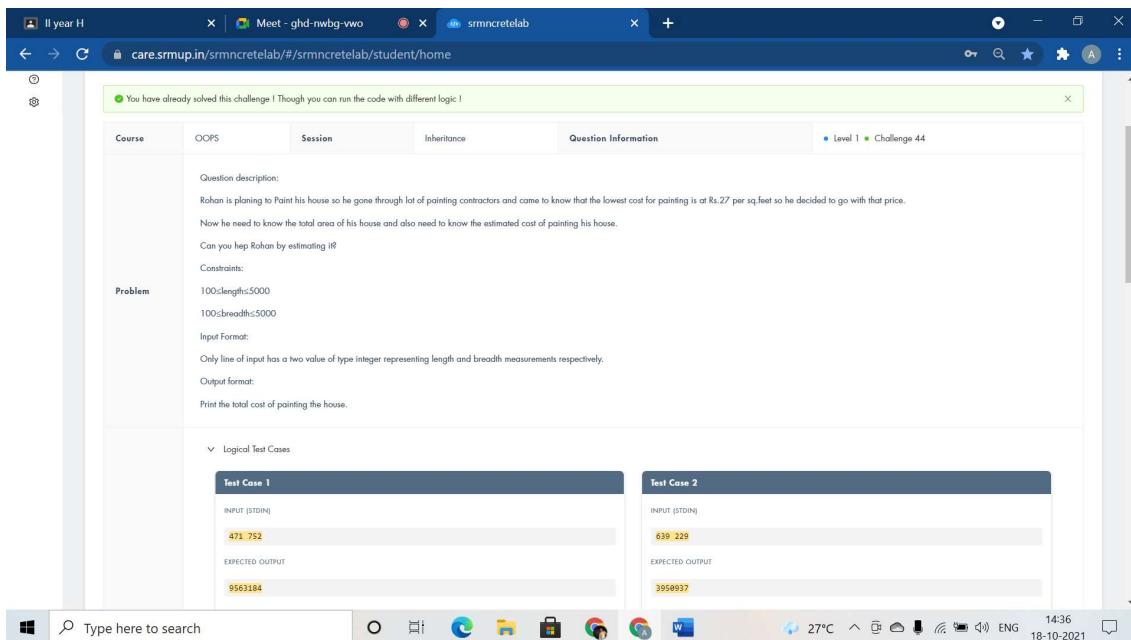
class customer:public market{
```

```

};

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{

};

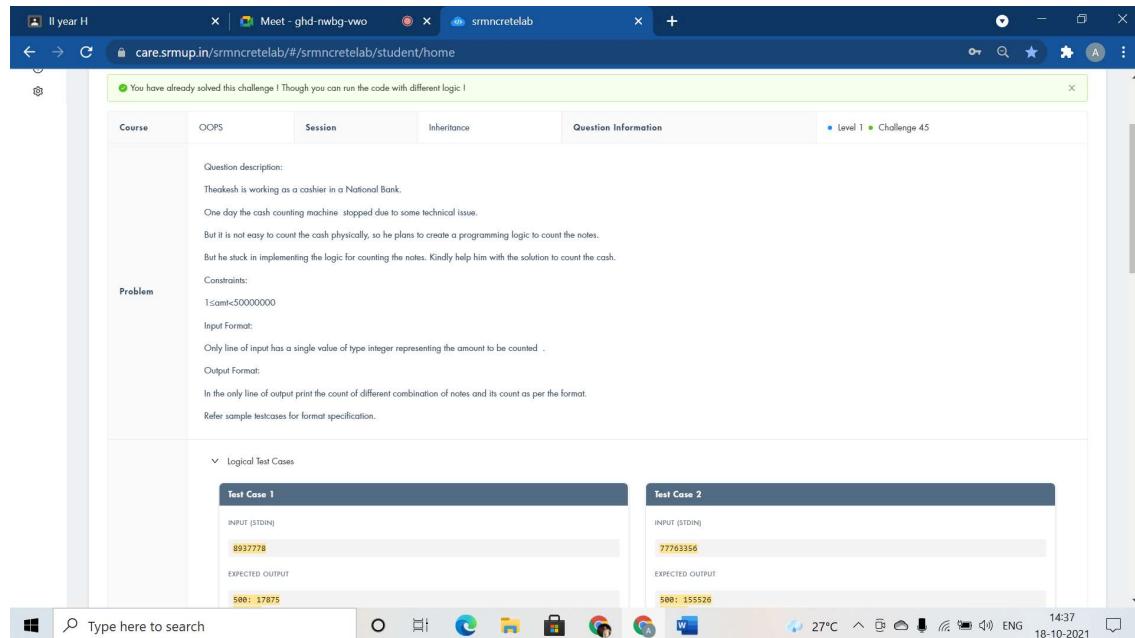
int main()

{
    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;  
}
```

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

Course	OOPS	Session	Inheritance	Question Information	Level 1 Challenge 46
Question description: Priya is an architect. During her designing process she used to note down lot of length and breadth details in her Design notebook. But it often leads to confusion. So to avoid confusion she decided to keep track of all her measurement digital. So she will be happy if you can help her with the logic which can get the measurements from here and display in the formatted manner. Can you do it? Constraints: 100≤length≤5000 100≤breadth≤5000 Input Format: Only one line of input has two values of type integer representing length and breadth measurements respectively. Output format: Print the details in the formatted manner. Refer sample testcases for format specification.					
Logical Test Cases					
Test Case 1			Test Case 2		
INPUT [STDIN]			INPUT [STDIN]		
103 179			273 653		
EXPECTED OUTPUT			EXPECTED OUTPUT		

```
#include <iostream>

using namespace std;

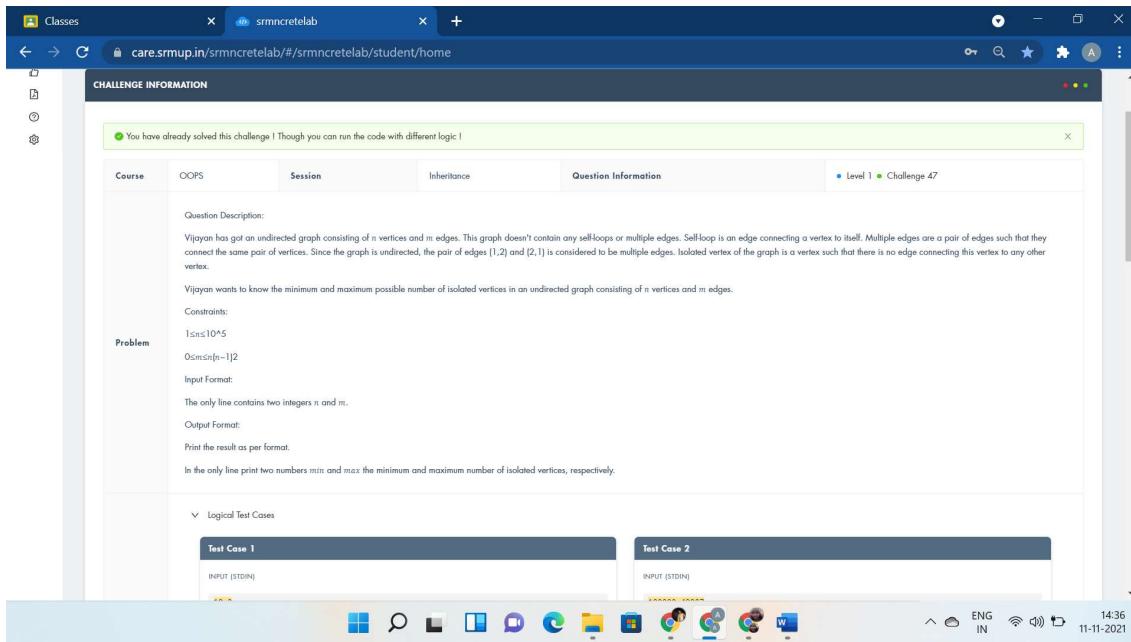
class ReceiveMesurement{

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;

}

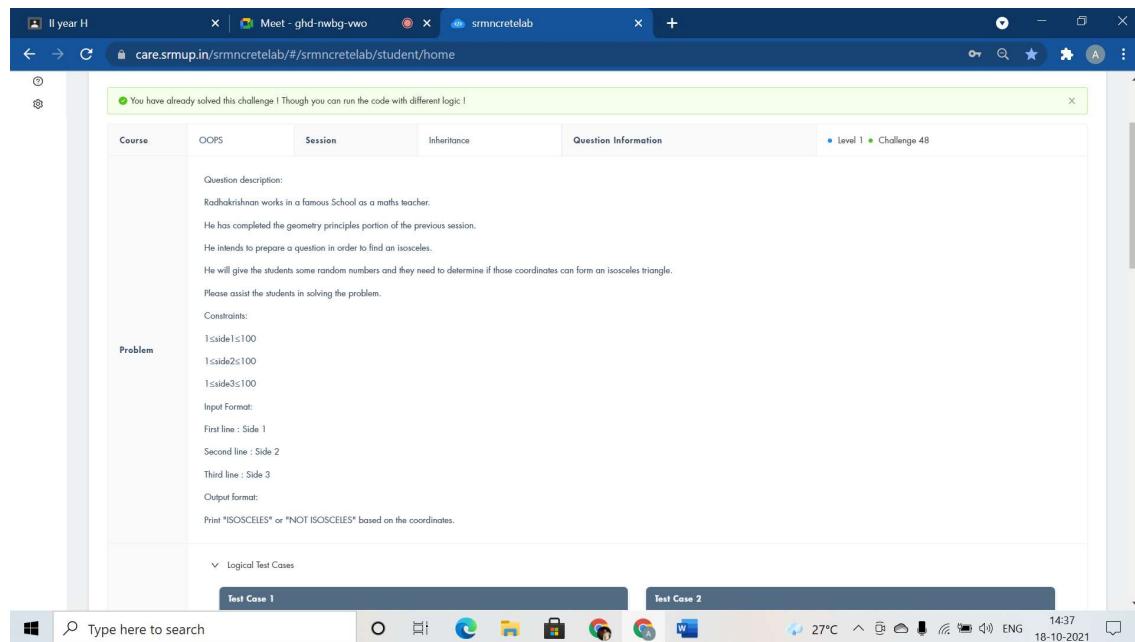
};

};
```

```

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";
}

};

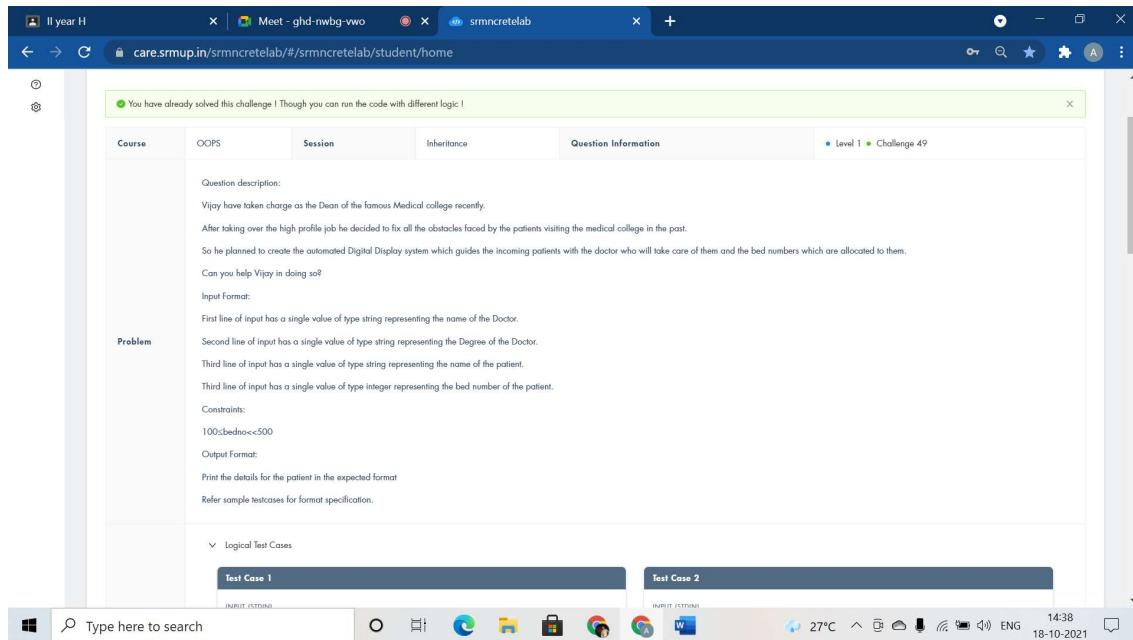
class isosceles : public triangle {

};

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

    return 0;
}

```



```

#include <iostream>

using namespace std;

class doctor{

public:

```

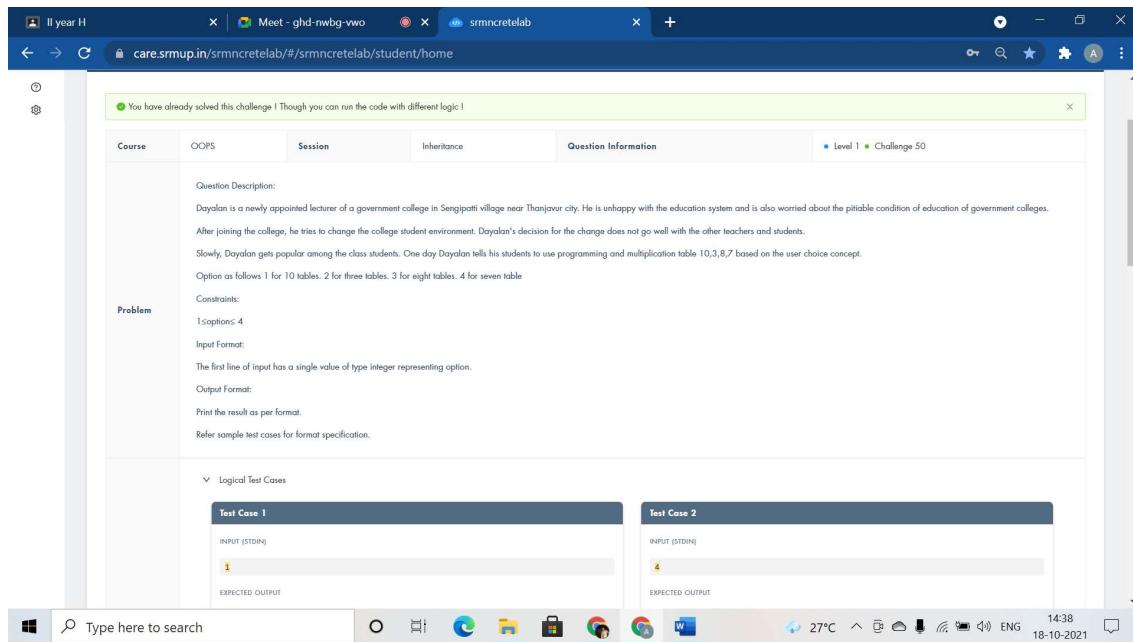
```

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;
}

```



```
#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:-

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 51**

Problem

Question description:
Vikram is an Data Collection Officer in Tamilnadu School Educational Department.
Recently Tamilnadu Government have announced the merit list of the 12th Grade students.
So the senior authority of Vikram have ordered him to collect the Name and Registration number of the student who have scores top 3 positions in each districts of the state to media release.
Vikram is collecting information from various districts and finally he need to consolidate the name and registration number in the format provided by his superior.
Can you help Vikram in final printing task?

Constraints:
2021100000<roll<202999999

Input Format:
First line of input has a single value of type integer representing the Registration number of the Student.
Second line of input has a single value of type string representing the name of the student.

Output Format:
Print the Student details as per the format
Refer sample testcases for Format Specification.

Logical Test Cases

Test Case 1 **Test Case 2**

INPUT [STDIN] INPUT [STDIN]

17:12 27-10-2021

```
#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;
}

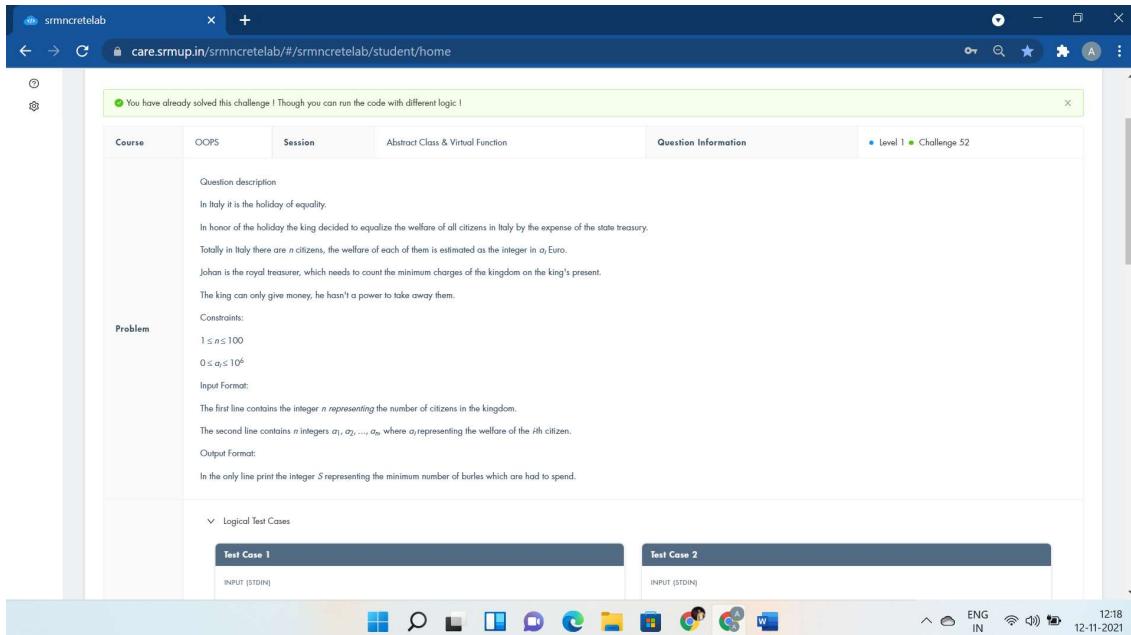
void District :: display(){
    cout<<"Student Name is: "<<name<<endl<<"Student Roll no is: "<<roll;
}
```

```

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 (){

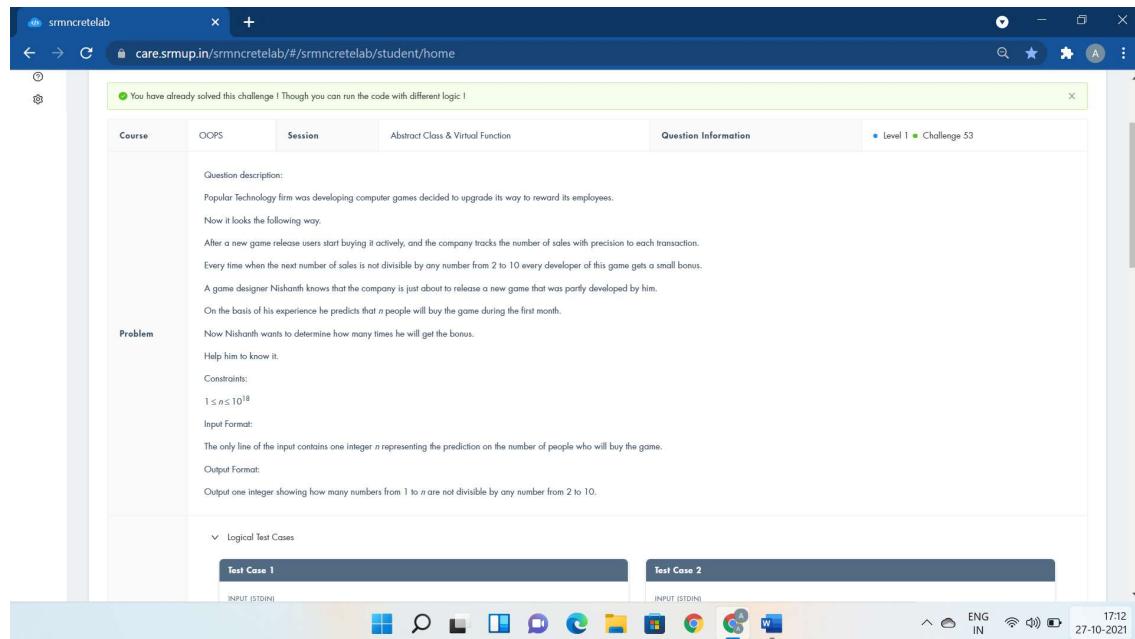
    Citizen obj;

    obj.Expenses();

    return 0;

}

```



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

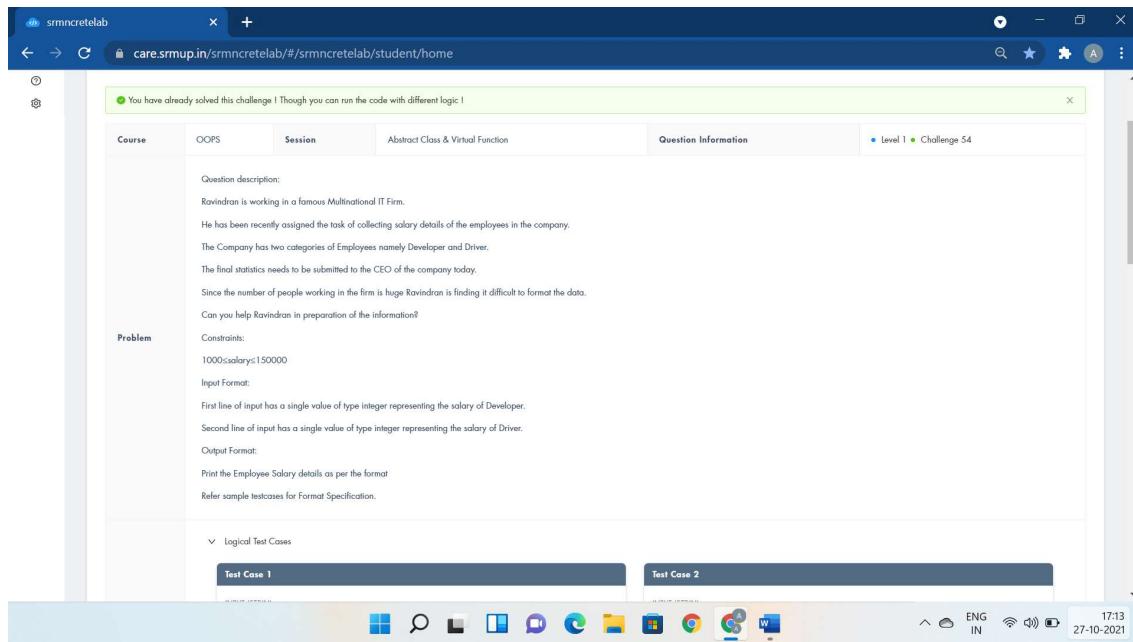
```
using namespace std;

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;

}

};
```

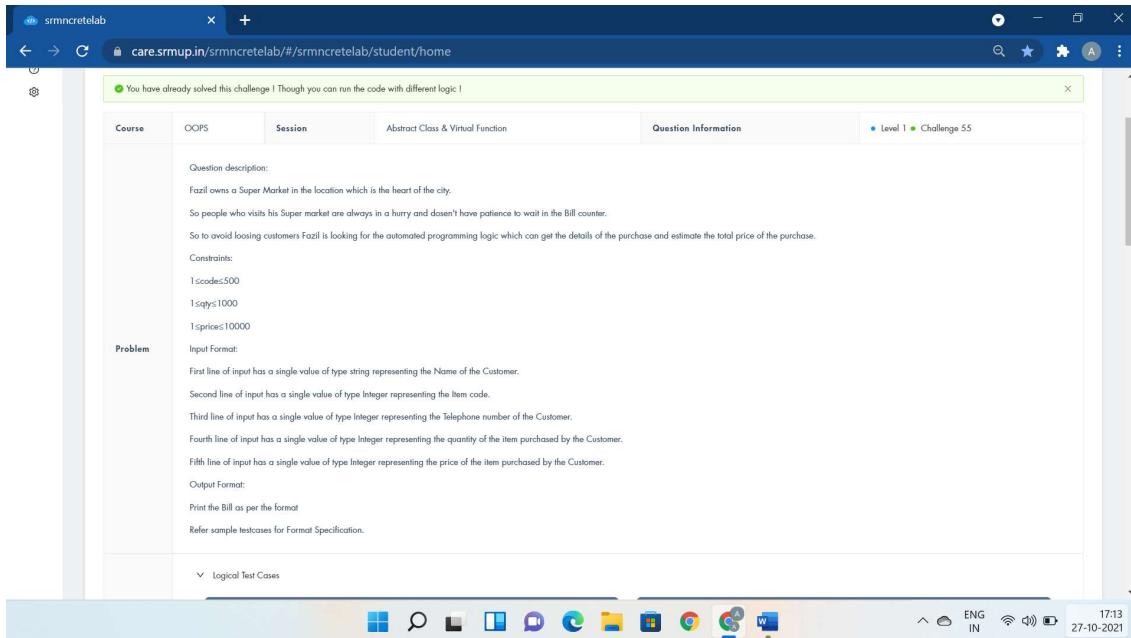
```

};

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;  
}
```

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 56

Question description:
Idumban Karri's friend Soman Santhanam given him two integers n and k .
Soman asked Idumban to find k th smallest divisor of n , or report that it doesn't exist.
Divisor of n is any such natural number, that n can be divided by it without remainder.

Constraints:
 $1 \leq n \leq 10^{15}$
 $1 \leq k \leq 10^9$

Input Format:
The first line contains two integers n and k

Output Format:if n has less than k divisors, output -1.

Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN) 4 2	INPUT (STDIN) 5 3
EXPECTED OUTPUT 2	EXPECTED OUTPUT -1

```
#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;

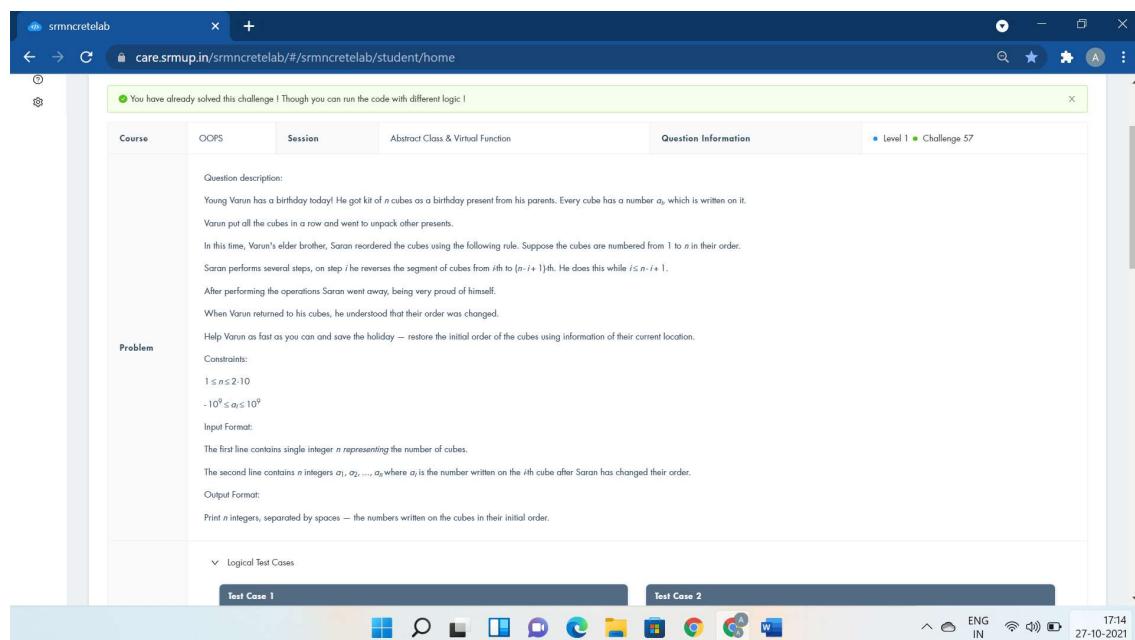
    obj.Divisor();

    obj.Display();

    return 0;

}

```



```
#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;
}
```

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 58

Question description:

Omkar is mad about coding, that is why he writes encoded messages.

He calls the *median letter* in a word the letter which is in the middle of the word.

If the word's length is even, the median letter is the left of the two middle letters.

In the following examples, the median letter is highlighted: contest, info.

If the word consists of single letter, then according to above definition this letter is the median letter.

Omkar encodes each word in the following way: he writes down the median letter of the word, then deletes it and repeats the process until there are no letters left.

Problem

You are given an encoding s of some word, your task is to decode it.

Constraints:

$1 \leq n \leq 2000$

Input Format:

The first line contains a positive integer n representing the length of the encoded word.

The second line contains the string s of length n consisting of lowercase English letters — the encoding.

Output Format:

Print the word that Omkar encoded.

Logical Test Cases

Test Case 1 Test Case 2

INPUT (STDIN) INPUT (STDIN)

17:14 27-10-2021

```
#include <iostream>
#include<string>
using namespace std;
class Decode{
public:virtual void Convert()=0;
};

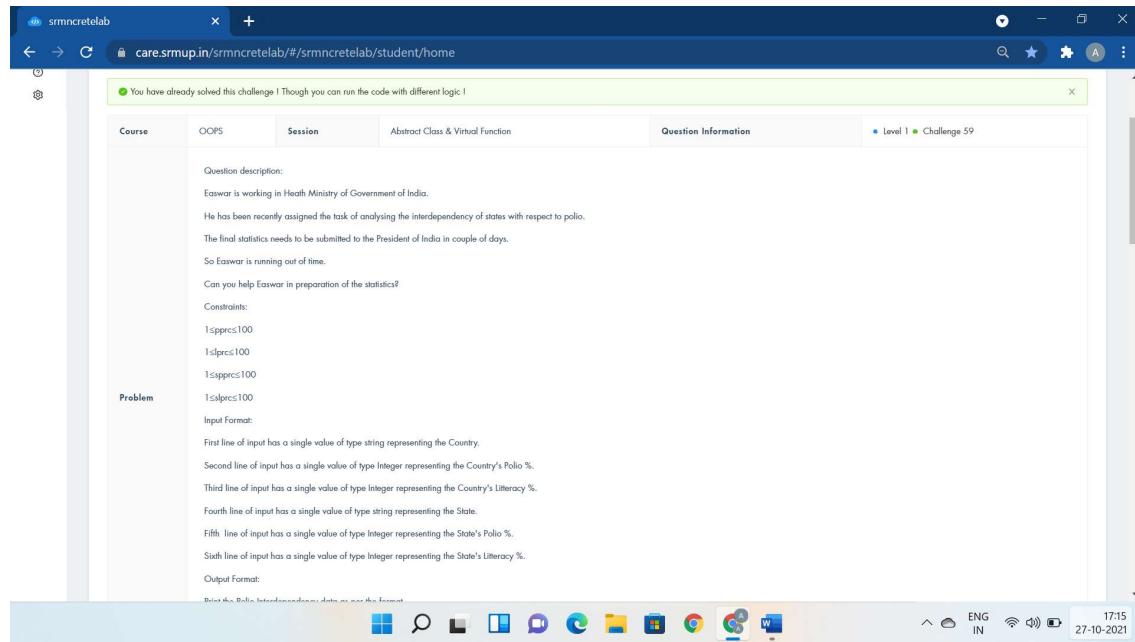
class Word:public Decode{
public:
string s1,s2;
int n;
void Convert(){
cin>>n>>s1;
for(int i=0;i<n;i++){
if((n-i)%2==1)
s2=s2+s1[i];
else
s2=s1[i]+s2;
}
cout<<s2;
}
```

```

    }
};

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 Infalk. 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 T 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.

Problem
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

12:20 12-11-2021

```
#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:-

```

#include <bits/stdc++.h>

using namespace std;

template <class Forest>

Forest Visit(Forest a,Forest b){

if(a>b)

cout<<"Kayaal\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!

Course **OOPS** **Session** **Templates** **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 inequality 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:

```

#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;
}

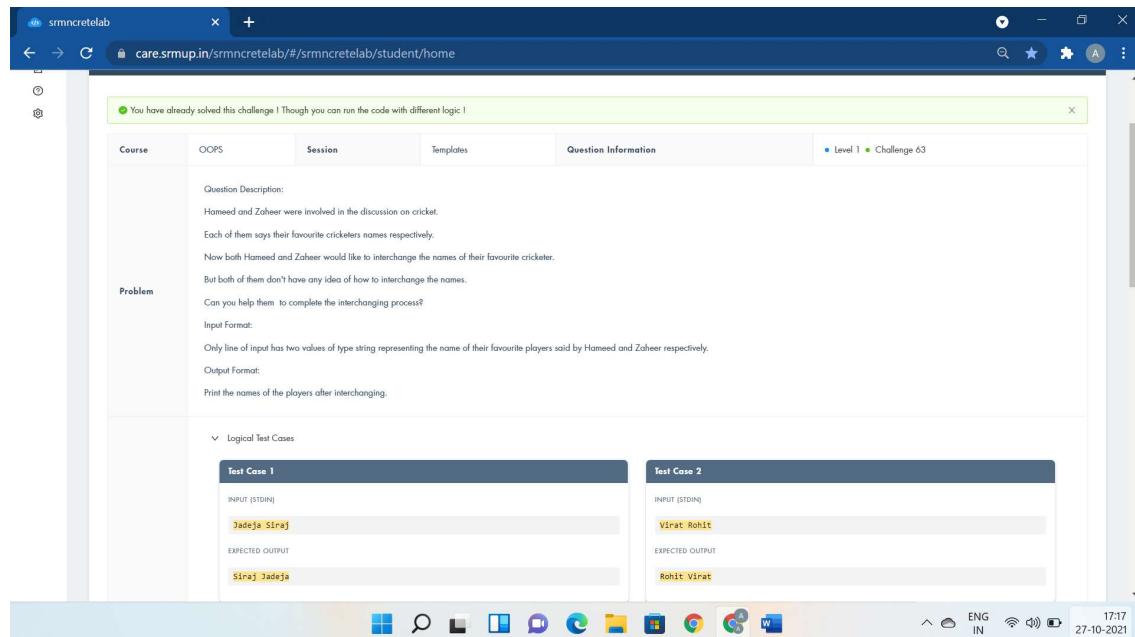
int main()
{

```

```

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);

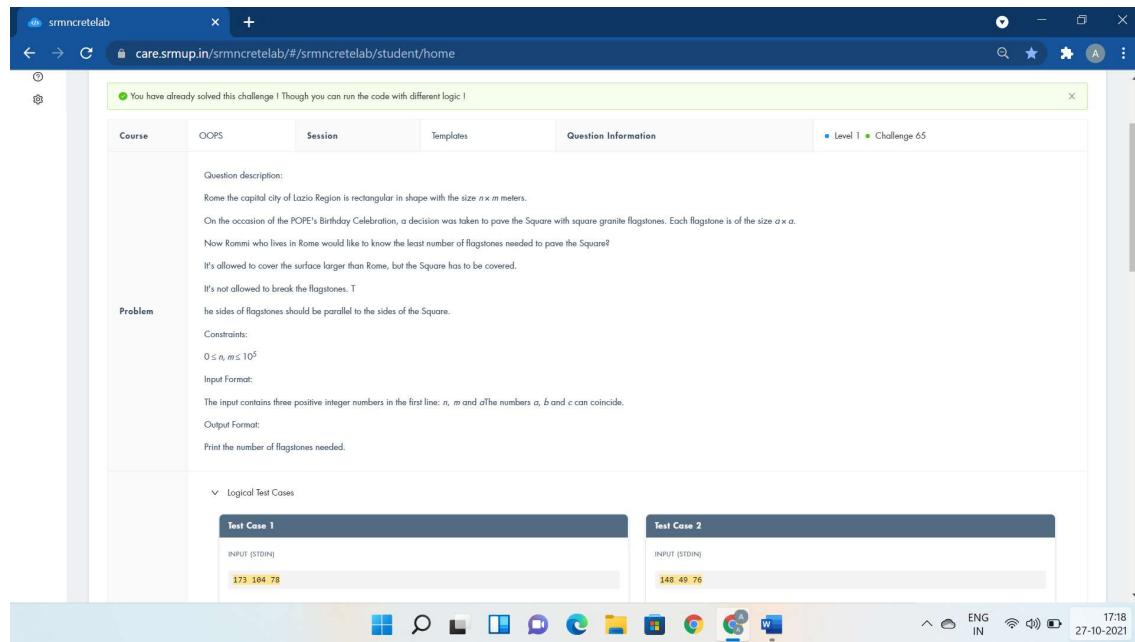
return 0;
}

```

```
}
```

```
#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;
    MagicClocl(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 engaged 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 the 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 Dog?
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.srmcup.in/srmncretelab/#/srmncretelab/student/home. The page title is "CHALLENGE INFORMATION". The main content area is titled "Challenge 68" and includes tabs for "Course", "OOPS", "Session", "Templates", "Question Information", and "Level 1". A message at the top says "You have already solved this challenge! Though you can run the code with different logic!". The "Question Information" tab is active, displaying a "Question description:" section with text about Rohan's interest in space research and a cube-shaped planet. It also contains sections for "Problem", "Input Format", "Output Format", and "Logical Test Cases" (Test Case 1 and Test Case 2). The bottom of the screen shows a Windows taskbar with various icons and the date/time 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 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

12:22
ENG IN
12-11-2021

```
#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:-

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 71

Question description:
Bharat loves to experiment with strings and one fine day he decided to check if two names matches with each other.
So he now tried to create a programming logic for the same but finding it difficult.
Can you help the students in doing the same?
Input Format:
First line of input has the first name
Second line of input has the second name
Output Format:
If name 1 = name 2 print name 1 is name 2
If name 1 != name 2 print name 1 is not name 2
And throw the error message "Inappropriate Input" if anything other than the string is provided as input.
Refer sample testcases for format specification.

Logical Test Cases

Test Case 1 Test Case 2

INPUT [STDIN] INPUT [STDIN]

Virat Dhoni Ishant Sacin

EXPECTED OUTPUT

```
#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";

    }

}
```

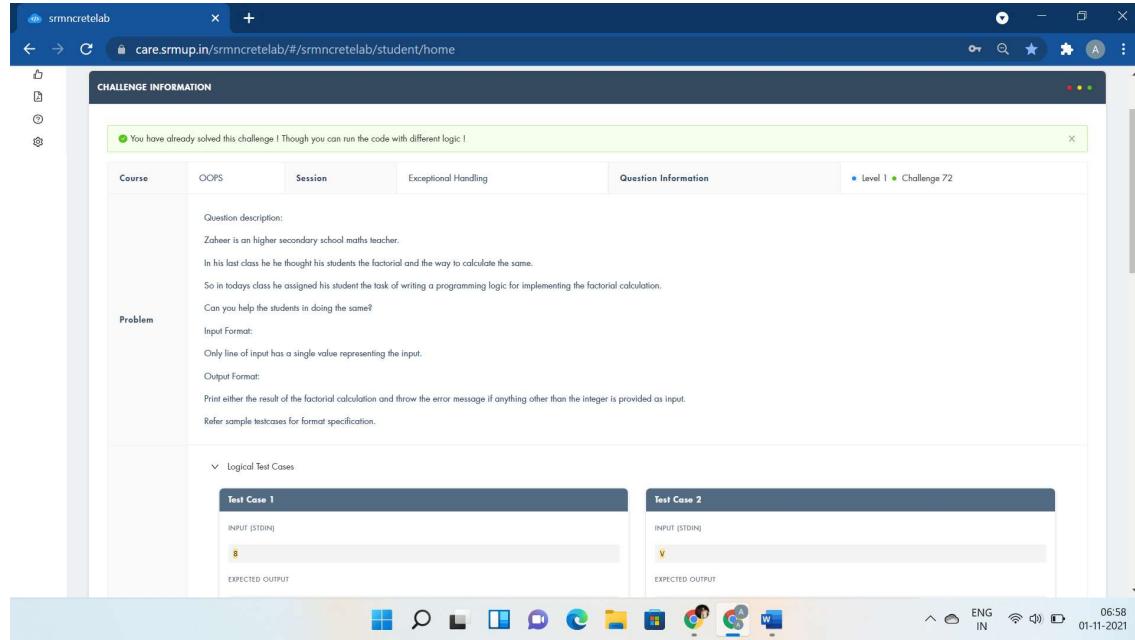
```

    }

    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";
    }
}

```

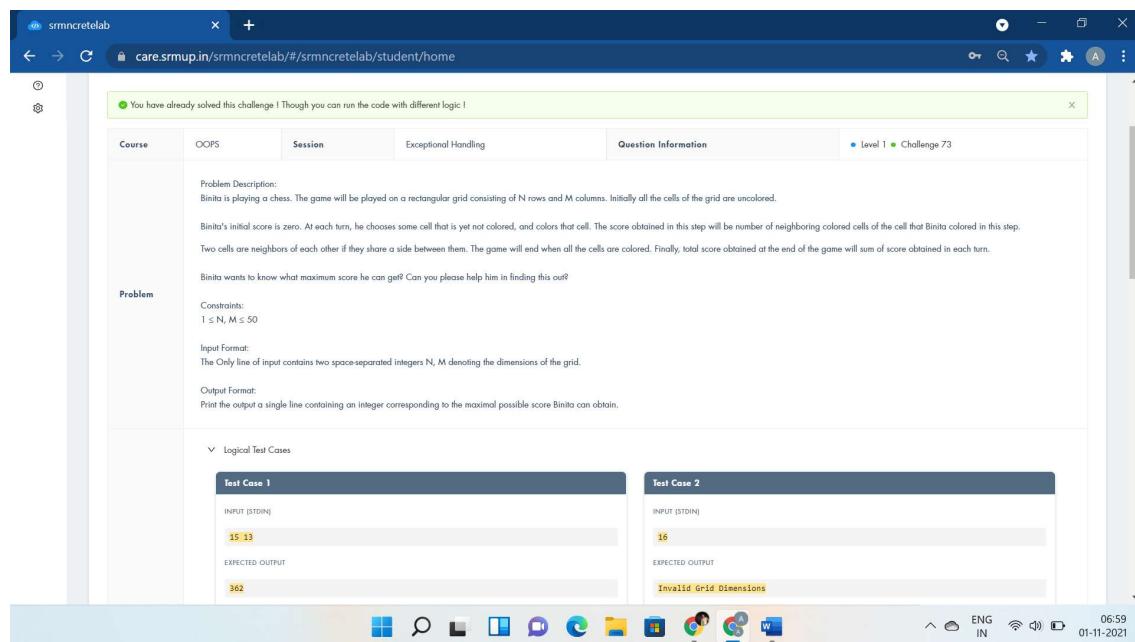
```

    }

    return 0;
}

}

```



```

#include <iostream>

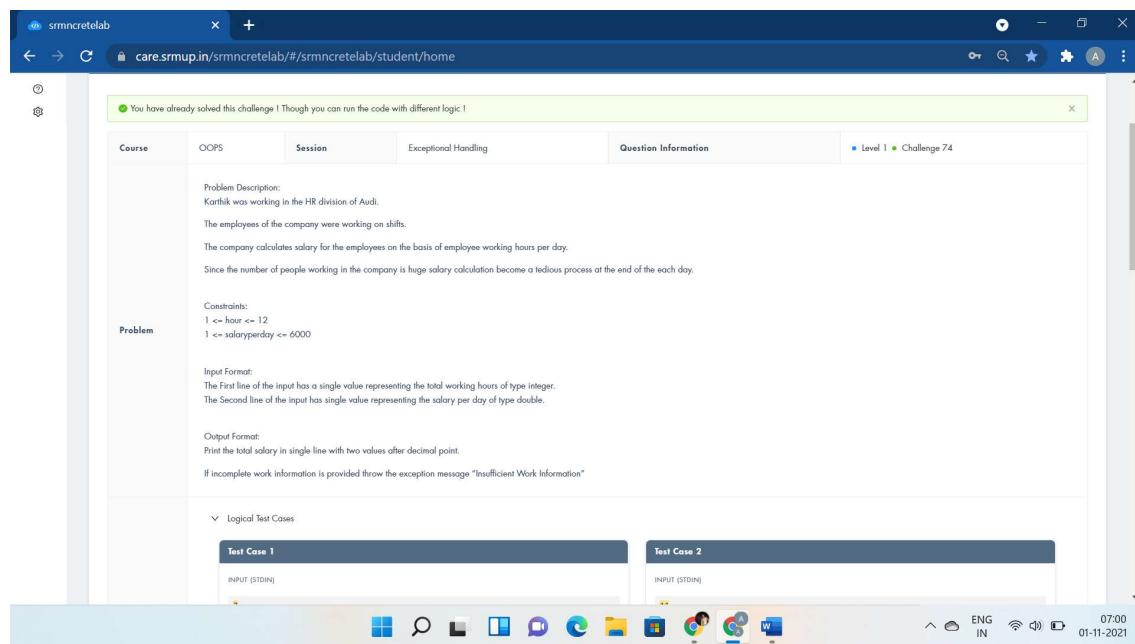
using namespace std;

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;  
}
```

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 75**

Question description:
Amithan has the practice of feeding the unprivileged kids with the money he has.
He usually provides them the donuts and the glass of milk.
Amithan would like to know if he has sufficient milk and donuts for providing it to children without shortage.
Can you help Amithan for determining the same?

Constraints
 $0 \leq \text{donuts}, \text{milk} \leq 1000$

Problem

Input Format:
First line of input has a single value of type integer representing the number of donuts.
Second line of input has a single value of type integer representing the number of glasses of milk.

Output Format:
Print the relevant message according to the availability of milk.
In the result print fine values after decimal point
Refer sample testcases for format specification.

Logical Test Cases

Test Case 1
INPUT (STDIN)
1000
0

Test Case 2
INPUT (STDIN)
1000
1000

```
#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 !

Course **OOPS** **Session** **Exceptional Handling** **Question Information** **Level 1** **Challenge 76**

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 SIDOHAR to perform the operations.

Problem
Constraints:
1≤m≤100
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

01-11-2021 07:00

```
#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";
    }
}
```

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 77

Problem Description:
The Electricity Officer has mentioned the total counts of unit and amount. The officer inform the customer the bill amount in a unique format.
The format given by electricity officer as follow:
But customers are finding the difficult to find the exact amount that needs to be paid.
Can you help the customers?
Functional Description:
Total Bill Amount = unitconsumed ^ costperunit
Constraints:
 $1 \leq \text{unitconsumed} \leq 500$
 $2 \leq \text{costperunit} \leq 10$

Input Format:
The first line of input represents the integer value of unitconsumed
The second line of input represents the integer value of costperunit

Output Format:
Print the total Bill amount in single line.

Logical Test Cases

Test Case 1	Test Case 2
INPUT: 100 2	INPUT: 100 2
OUTPUT: 1024	OUTPUT: 1024

07:01 ENG IN 01-11-2021

```
#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:
 $1 \leq n \leq 10^4$
 $1 \leq m \leq 10^4$
 $1 \leq a \leq 10^4$

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 VB	INPUT [STDIN] 1 Ge
EXPECTED OUTPUT	EXPECTED OUTPUT

01-11-2021 07:02

```
#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 a 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: [Windows Taskbar icons] ENG IN 01-11-2021 07:02

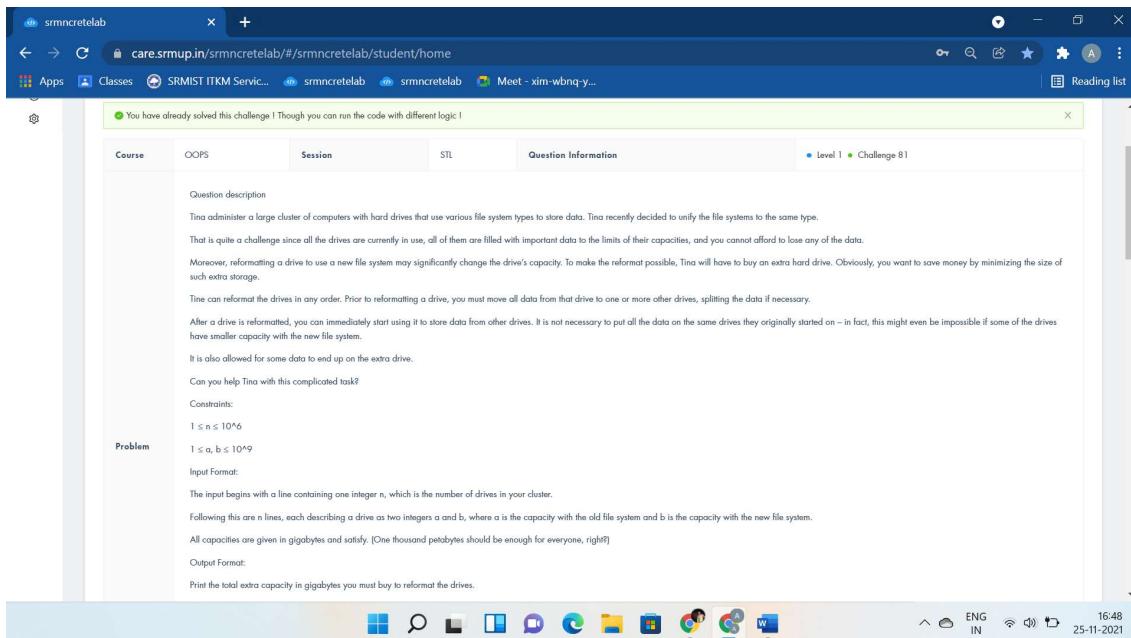
```
#include <iostream>

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, reformating 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:

$1 \leq n \leq 10^6$
 $1 \leq a, b \leq 10^9$

Problem:

Input Format:

The input begins with a line containing one integer n , which is the number of drives in your cluster.

Following this 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.

```
#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;

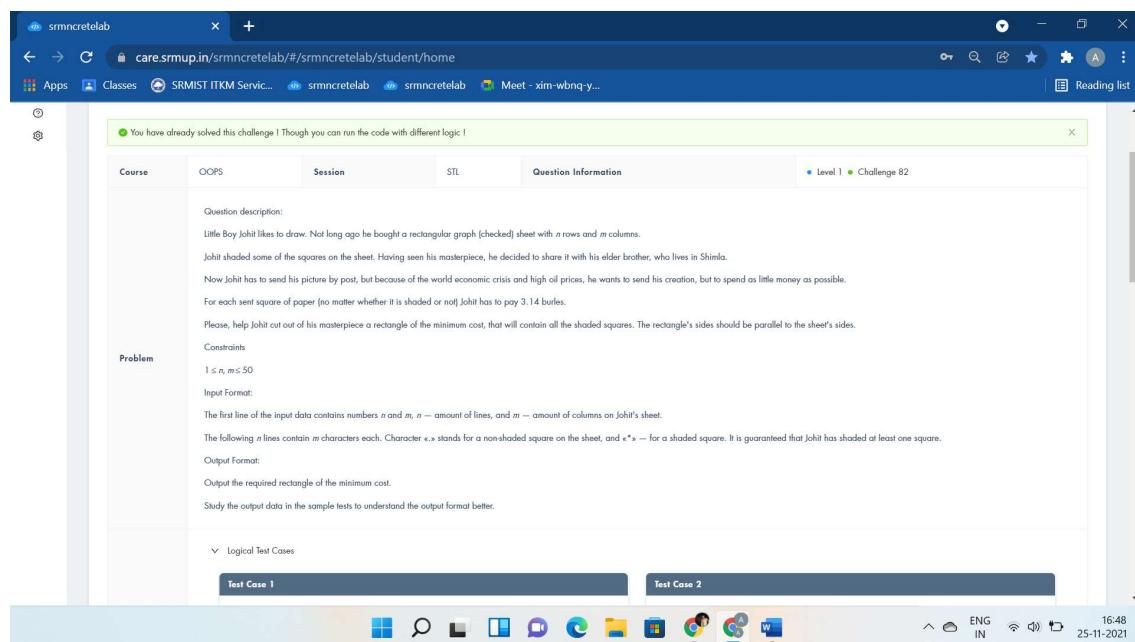
}

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);

            }

        }

    }

}

```

```

    }

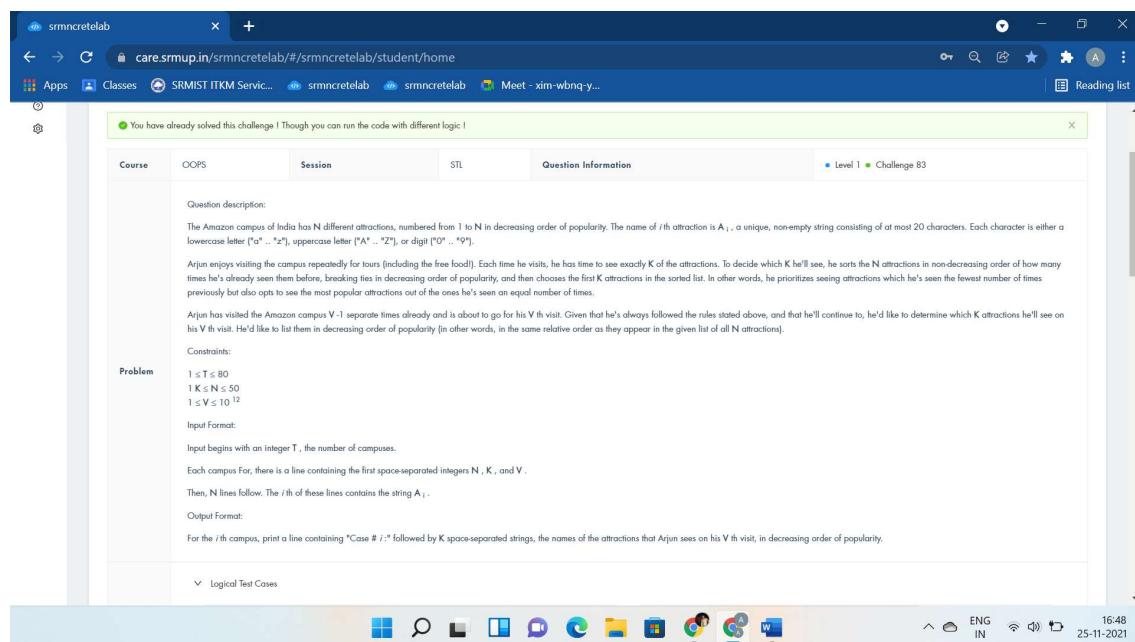
    for(int i=sx;i<=x;i++){
        for(int j=sy;j<=y;j++){
            cout<<a[i][j];
        }
        cout<<endl;
    }

    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];

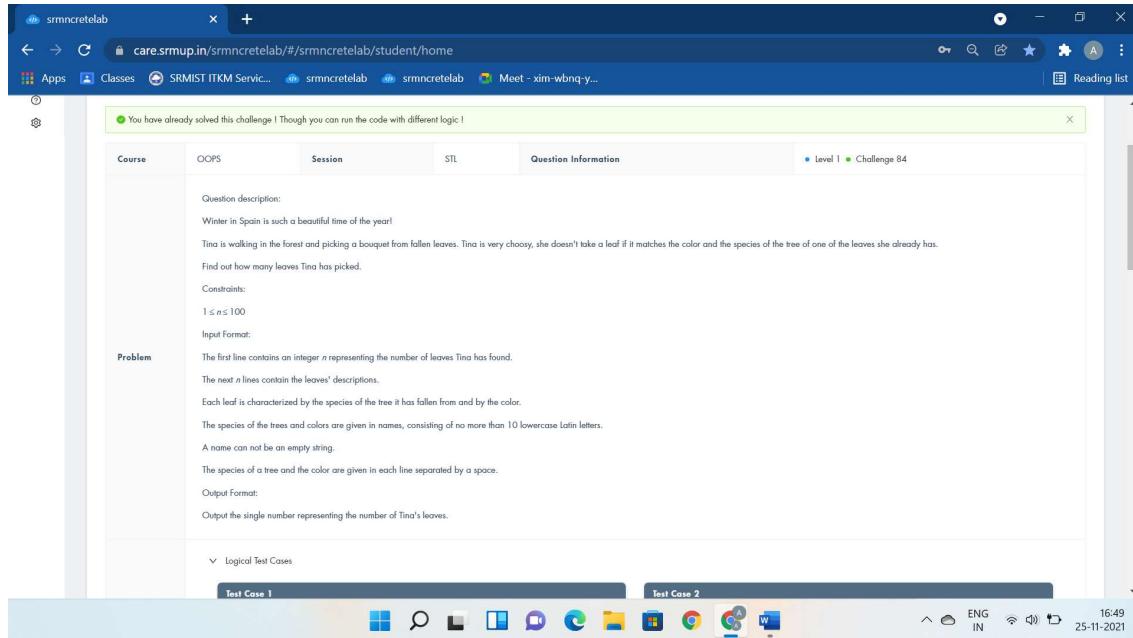
int main(){
    LL t; cin>>t; while(t--){
        cin>>n>>k>>v;
        for(int i=0; i<n; i++)
            cin>>name[i];
        LL st=((v-1)*k)%n;

```

```

//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));

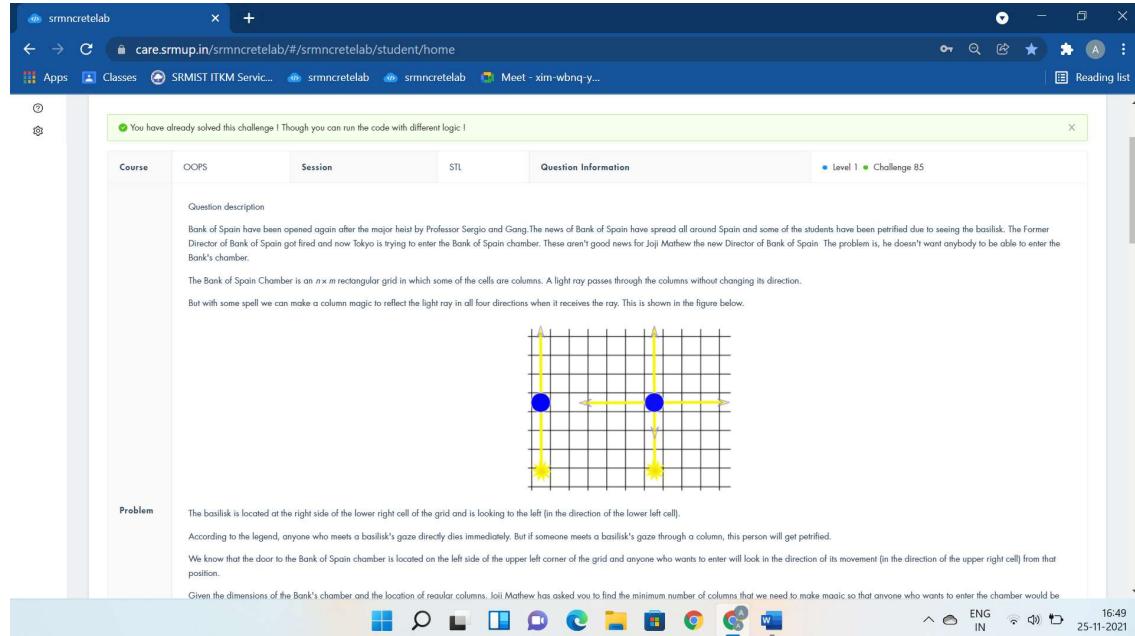
}

cout<<Descriptionofleaves.size();

return 0;

}

```



```

#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] == 'B')
                use[i].push_back(j);
        }
    }
    int ans = 0;
    for(int i=0;i<n;i++)
    {
        if(use[i].size() == 0)
            continue;
        sort(use[i].begin(),use[i].end());
        int l = use[i].size();
        for(int j=0;j<l;j++)
        {
            if(j == 0)
                ans += use[i][j];
            else if(j == l-1)
                ans += use[i][j];
            else
                ans += use[i][j] - use[i][j-1];
        }
    }
    cout<<ans;
}

```

```

        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!

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 dances 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.

Problem

Constraints:

$3 \leq n \leq 10^5$

$1 \leq m \leq 10^5$

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++;

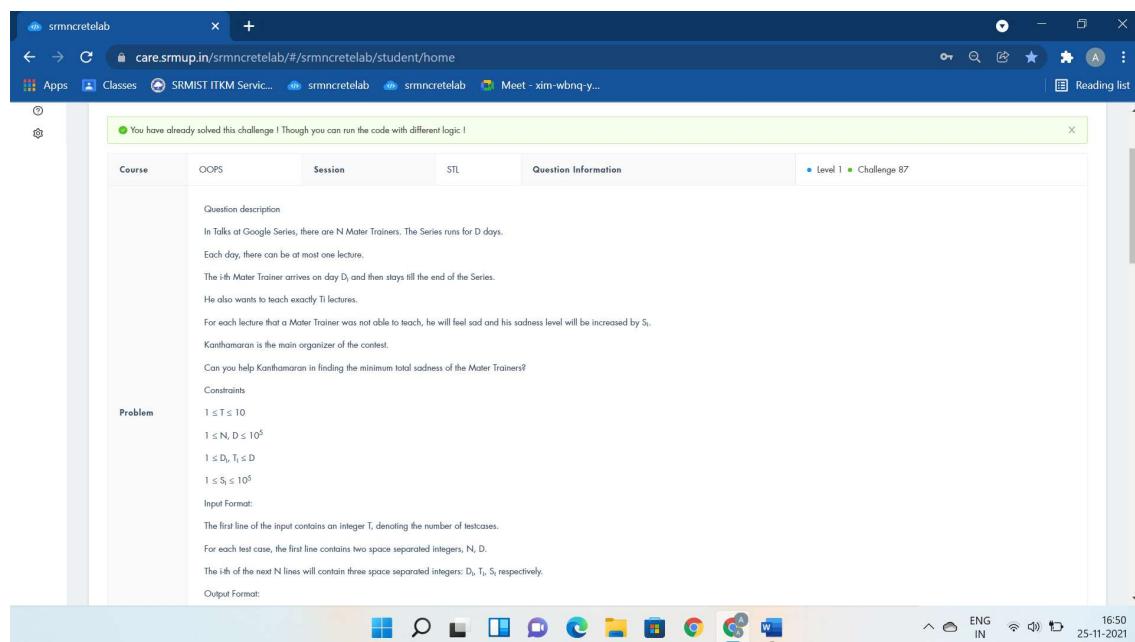
            if(a[c[k]]==0)
```

```

    {
        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 ";

```

```
#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;
}

```

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

Question Information

Level 1 • Challenge 89

Course **OOPS** **Session** **STL**

Question description

Little Madura's has a famous zoo where there is an enclosure with palm civets. It is known that palm civets like to spit. Suruli watched these interesting animals for the whole day and registered in his notepad where each animal spit. Now he wants to know if in the zoo there are two palm civet, which spit at each other. Help him to solve this task.

The trajectory of a comet's spit is an arc, i.e. if the palm civet in position x splits d meters right, he can hit only the palm civet in position $x+d$, if such a palm civet exists.

Constraints:

- $1 \leq n \leq 100$
- $-10^4 \leq x_i \leq 10^4$
- $1 \leq |d| \leq 2 \cdot 10^4$

Problem

Input Format:

The first line contains integer n the amount of palm civet in the zoo.

Each of the following n lines contains two integers x_i and d_i records in Suruli's notepad.

x_i is a position of the i th camel, and d_i is a distance at which the i th palm civet spit.

Positive values of d_i correspond to the splits right, negative values correspond to the splits left.

No two palm civet may stand in the same position.

Output Format:

If there are two palm civet, which spit at each other, output YES. Otherwise, output NO.

Logical Test Cases

```

#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";

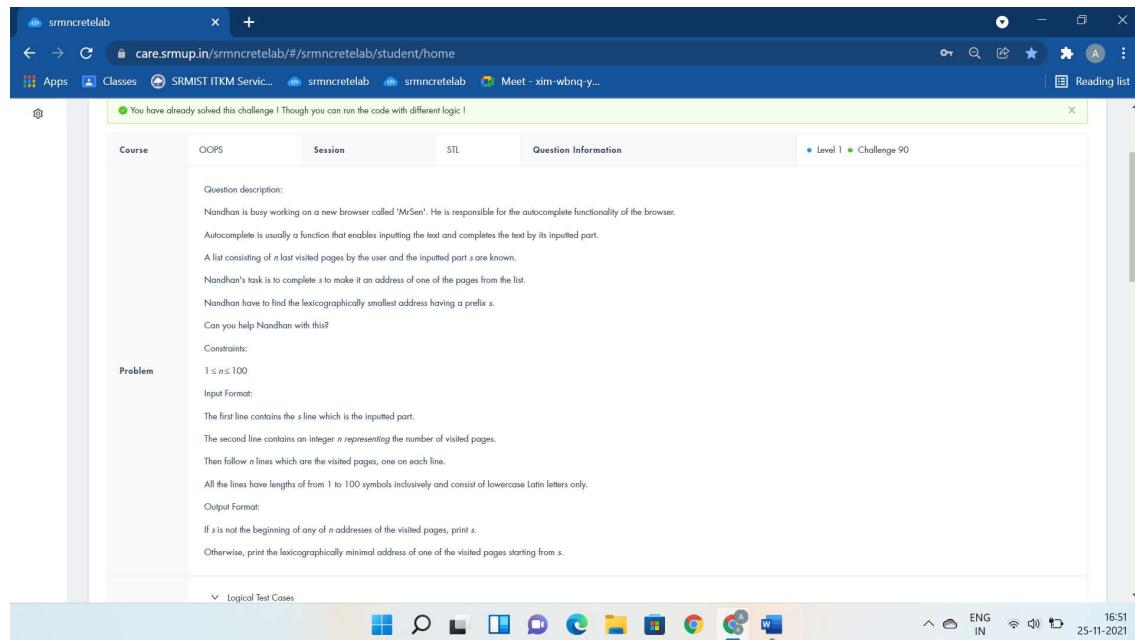
    return 0;

    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:
Ravindra 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 Ravindra 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 DA
Sixth Line: Employee PF
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;
}
```

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 92

Question description
Sivakumar is working as a Principal of the famous school.
In his school currently 11th Grade admissions are going on.
Based on the marks scored by the students in 10th Grade different groups have been assigned to them along with their class in-charge names.
Manually doing this process is tedious so Sivakumar planned to create a programming logic for doing the same.
Can you help him with the logic?
Constraints
100 ≤ id ≤ 999
Input Format:
The first line of input has a single value of type integer representing the number of students.
For each test case that follows:
The First line of input has five values of type integer representing the id of the student.
The Second line of input has a single value of type string representing name of the student.
The Third line of input has a single value of type string representing the group allocated to the student.
The Fourth line of input has a single value of type string representing name of class in-charge.
Output Format:
Print the output in the expected format.
Refer sample testcases for format specification.

```
#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();

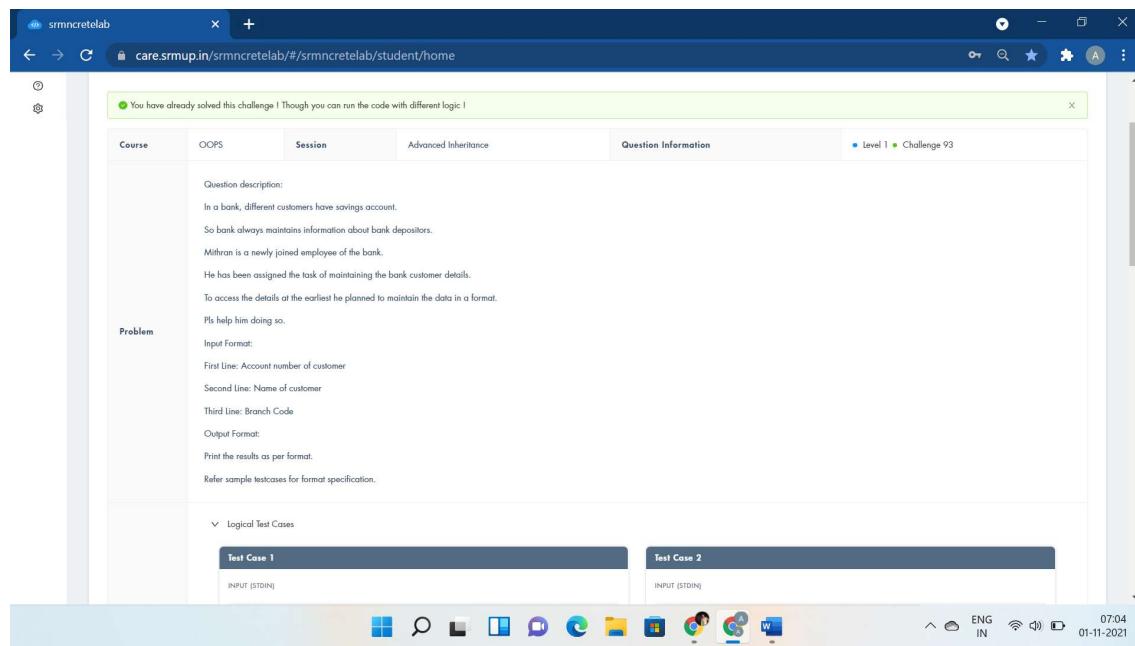
}

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;
}

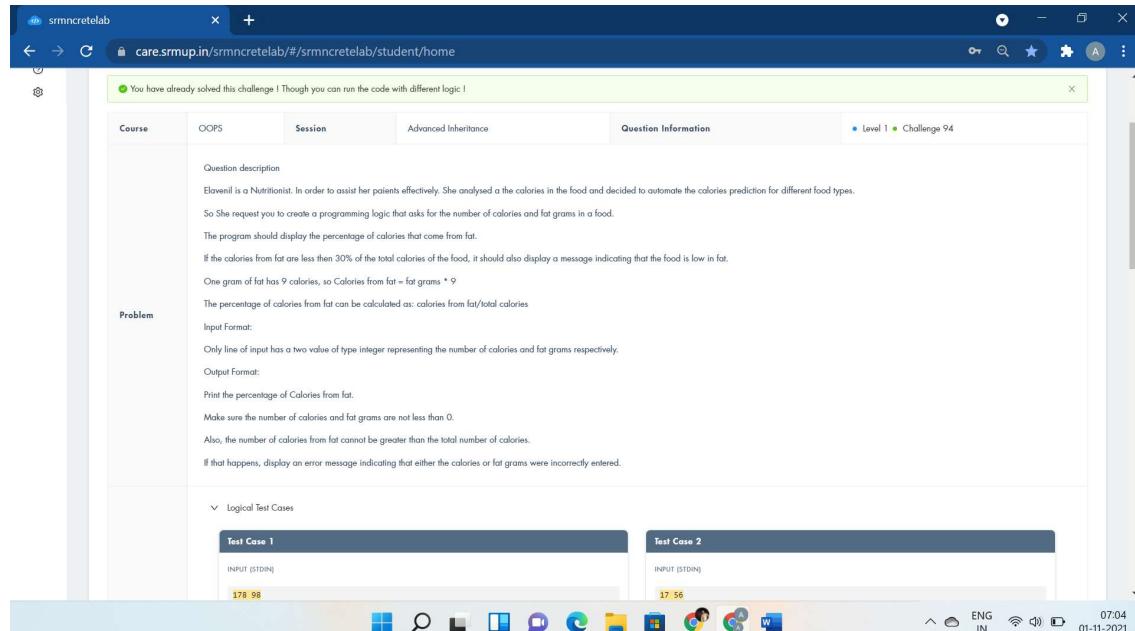
};

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;

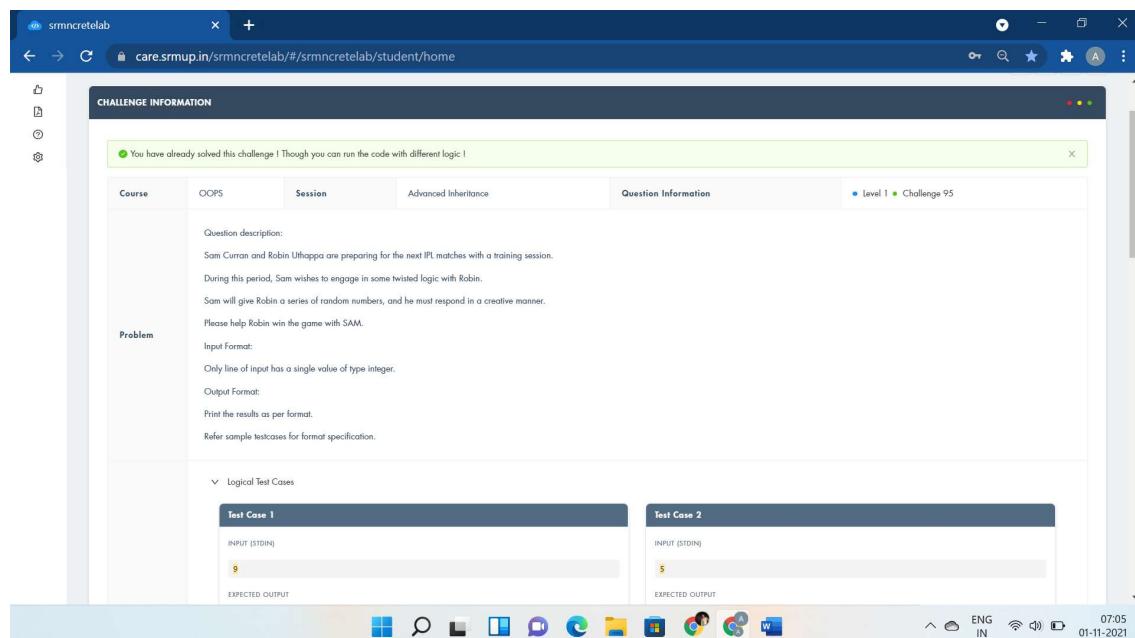
p.calorie();

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;
}
```

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 96

Question description:

Romon is in his second year of engineering at CCC.

He's nearing the conclusion of the semester, and he needs to turn in his mini project as soon as possible.

He aims to create a tiny mark printing system that is suited to the specific of the user.

Can you assist in completing the project and obtaining a good score in the mini project?

Input Format:

First Line: Roll Number
Second Line: Mark 1 and Mark 2 separated by a space
Third Line: Sports Mark

Output Format:

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

Logical Test Cases

Test Case 1 Test Case 2

INPUT [STDIN]

178
97 81
93

INPUT [STDIN]

181
77 78
99

```
#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;

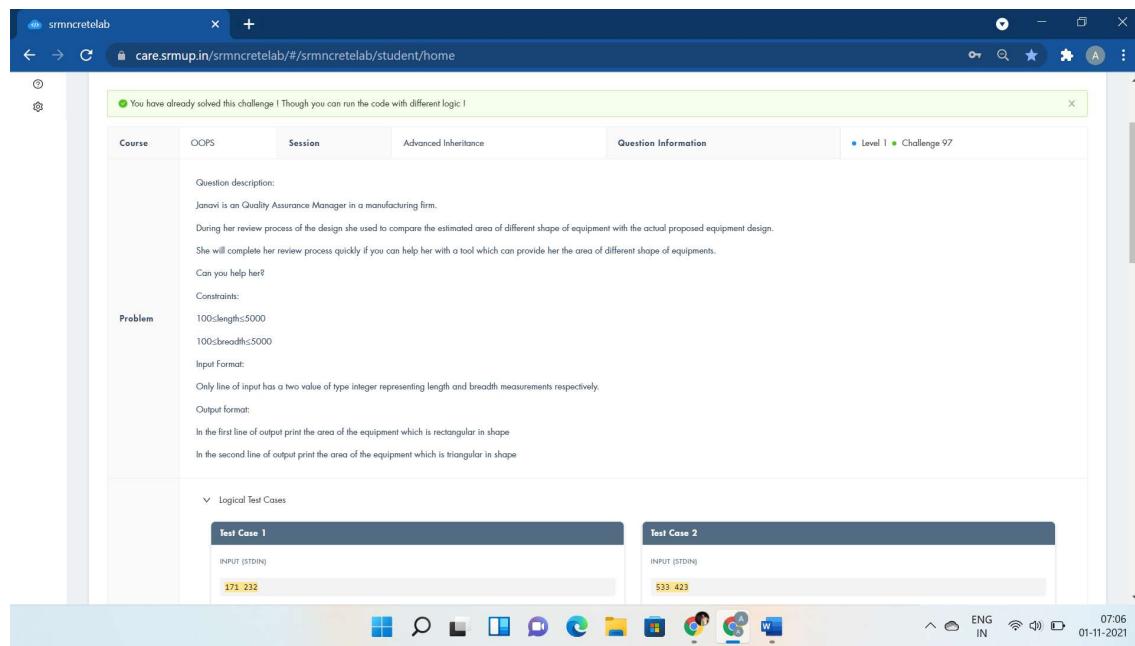
    obj.get();

    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;
}
```

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-amt] 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.

```
#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;
}

```

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 99**

Problem

Question description:
Surya's daughter is enrolled in a famous Residential School in Ooty.
Through her online class, she is learning basic mathematical processes.
She was given a assignment on real and imaginary numbers and on getting the real and imaginary part of the number she have to perform some operation and need to upload the result.
Can you assist her in efficiently solving the question and uploading the result?

Constraints:
 $1 \leq r1 \leq 100$
 $1 \leq r2 \leq 100$
 $1 \leq i1 \leq 100$
 $1 \leq i2 \leq 100$

Input Format:
First line of input has a single value of type integer representing $r1$ and $i1$ respectively.
Second line of input has a single value of type integer representing $r2$ and $i2$ respectively.

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

Logical Test Cases

Test Case 1 Test Case 2

01-11-2021 07:06

```
#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;

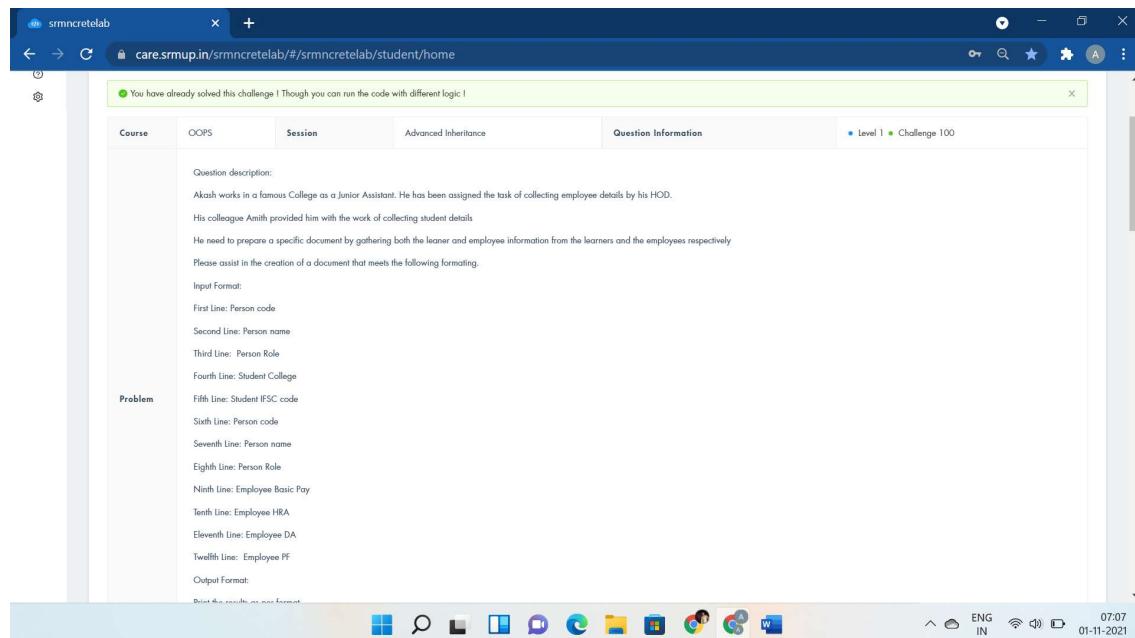
    calc.getdata();

    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();";
}

```