

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

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

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

**Problem**

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

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

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

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

Logical Test Cases

```
#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();

}

```

CHALLENGE INFORMATION

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

Course	OOPS	Session	Classes, Methods & Constructors	Question Information
				Level 1 Challenge 13

**Problem**

**Problem Description:**  
 Boppana is working in Unique Identification Authority of India. Boppana is part of Data Validation Department. He is currently validating the names provided by the citizens of India for processing of their Aadhaar Card. As per UIDAI rule the First name, Middle Name and Last Name of the Citizens should come as a same name in Aadhaar Card. But the data provided by the citizens are separated in three different fields of First Name, Middle Name and Last Name.

**Functional Description:**  
 So now Boppana's task is to concatenate the First Name, Middle Name and Last Name into single name. Also if someone missed any of the three parts of the name then the system it should be treated as "Invalid Name". Since the data that need to be validated is huge in numbers Boppana is looking for the help from you. Can you help him by creating a programming logic for doing his task?

**Input Format:**  
 The first line of the input contains a single value of type String representing the FIRST name.  
 The second line of the input contains a single value of type String representing the MIDDLE name.  
 The third line of the input contains a single value of type String representing the LAST name.

**Output format:**  
 In a single line print the Full Name of the citizen in the expected format.  
 Refer to Sample test cases for format specification.

```
#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;
}
```

srnmcretelab

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

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 1.4

**Problem**

Problem Description:  
Central Government Toll Booth is located at Pamban Bridge.  
A Car passing by the booth is expected to pay a toll.  
The tollbooth keeps the track of the number of cars that gone by and the total amount of cash collected.

Constraints:  
 $1 \leq T \leq 15$

Input Format:  
First line of input represents the Number of Testcases T.  
Next T lines has two values of type String and Double representing Vehicle number and Toll amount collected respectively.

Output Format:  
In the First Line of output print the number of cars passed.  
In the Second Line of output print the total toll amount collected.  
Refer sample testcases for Format Specification.

Logical Test Cases

Test Case 1	Test Case 2
INPUT (STDIN) 3 TMA01 39.5 PVA01 88.9	INPUT (STDIN) 5 UP00 68.5 MD01 88.9

Type here to search

25°C

22:57  
05-10-2021

```
#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(){
```

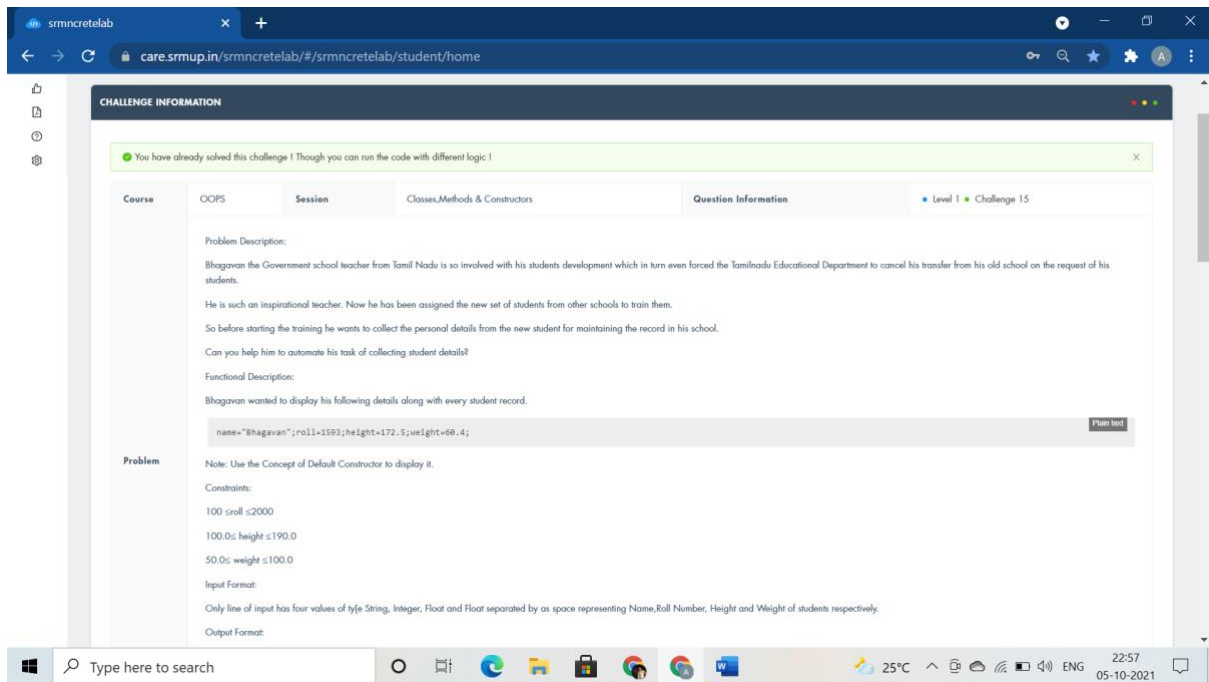
```
cars++;
```

```
}
```

```
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;
}

```

**CHALLENGE INFORMATION**

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

Course	Session	Question Information
OOPS	Classes, Methods & Constructors	Level 1 Challenge 19

**Problem**

**Problem Description:**  
 Fahad is the owner of one of the biggest Super Market in the City.  
 Since the day Fahad has taken charge of the Super Market from his father he is trying hard to save unproductive time of their workers.  
 Workers of his super market is spending lots of time in calculating the prices of items purchased by the customers so the long people queue keeps forming.  
 So now is planned to create a software which gets the number of items, item code and Price as from the staff and provide them the largest price among the items purchased and the sum of prices of all the items.  
 Since Fahad is not aware of the technical stuffs of implementation, can you help him with the programming logic for the software?

**Constraints:**  
 1 ≤ no\_items ≤ 10  
 100 ≤ itemcodes ≤ 500  
 1 ≤ itemprice ≤ 1000

**Input Format:**  
 The first line of the input contain a single value of type integer representing no. of items N.  
 The next N lines contain two values of type integer and float separated by a space representing item code and item Price respectively.

**Output Format:**  
 Print the largest price among all items, the total price of all items, item code and price of all the items in the expected format.  
 Refer Sample Testcases for format specification.

Logical Test Cases

```

#include <iostream>

using namespace std;

class ITEM
{
public:
    int n;

    float large=0,summ=0;

```

```

float arr[100],code[100];

void getdata(int b){
    n=b;
    for(int i=0;i<n;i++)
        cin>>code[i]>>arr[i];
}

void largest(){
    for(int i=0;i<n;i++)
    {
        if(arr[i]>=large)
            large=arr[i];
    }
}

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

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

};

using namespace std;

int main()
{
    ITEM order;
    int b;

```



```

cin>>b;

order.getdata(b);

order.largest();

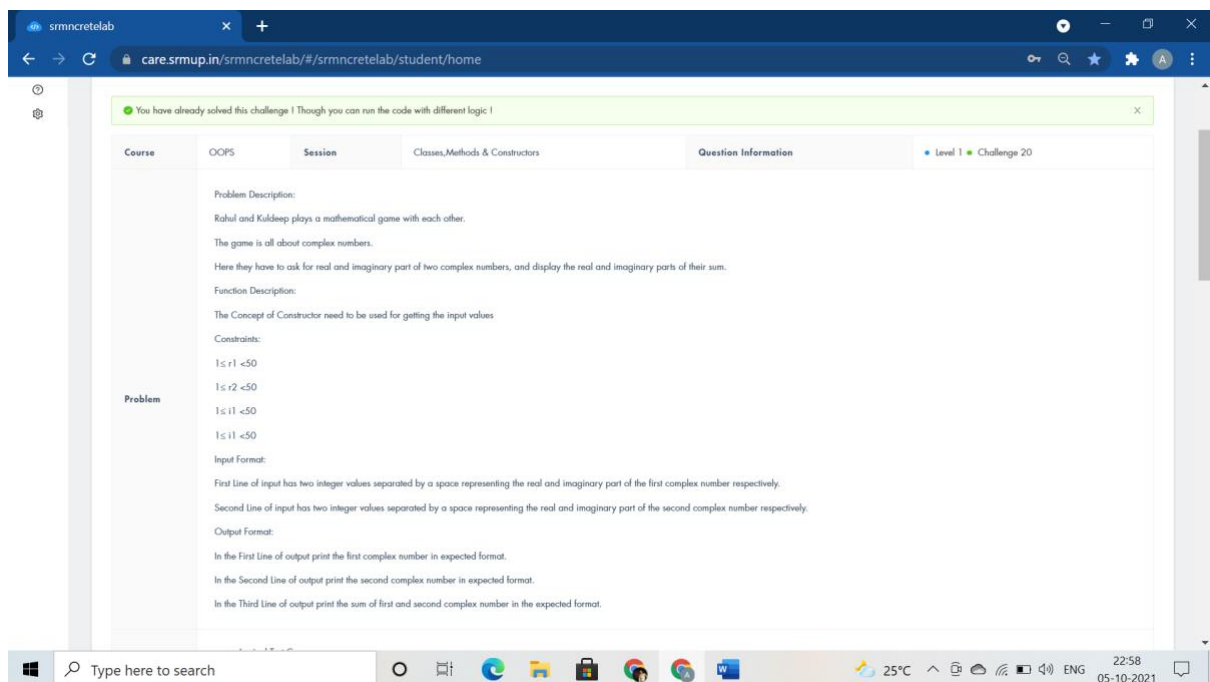
order.sum();

order.displayitems();

    return 0;

}

```



```

#include<iostream>

using namespace std;

class Complex{
public:
    int r1,i1,r2,i2,r3,i3;

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

    void addcomplex(){
        r3=r1+r2;
    }
}

```

```

        i3=i1+i2;
    }

    void displaycomplex(){
        cout<<r1<<" "<<i1<<"i"<<endl;
        cout<<r2<<" "<<i2<<"i"<<endl;
        cout<<r3<<" "<<i3<<"i"<<endl;
    }
};

int main(){
    Complex calculate;

    calculate.addcomplex();
    calculate.displaycomplex();

    return 0;
}

```

**CHALLENGE INFORMATION**

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

Course	OOPS	Session	Classes, Methods & Constructors	Question Information
				Level 1 Challenge 12

**Problem**

Question description:  
 Vikram has his own lake where there are  $n$  fishes, numbered from 1 to  $n$ .  
 But the fishes in the lake are eating the other fishes in the lake due to which Vikram is bit worried.  
 Every day right one pair of fish meet, and the probability of each other pair meeting is the same.  
 If two fish with indexes  $i$  and  $j$  meet, the first will eat up the second with the probability  $a_{ij}$  and the second will eat up the first with the probability  $a_{ji} = 1 - a_{ij}$ .  
 The described process goes on until there are at least two fish in the lake.  
 Now Vikram would like to find out for each fish the probability that it will survive to be the last in the lake.  
 Can you help Vikram?

Constraints:  
 $1 \leq n \leq 25$   
 $0 \leq a_{ij} \leq 1$

Input Format:  
 The first line contains integer  $n$  — the amount of fish in the lake.  
 Then there follow  $n$  lines with  $n$  real numbers each — matrix  $a$ .  $a_{ij}$  — the probability that fish with index  $i$  eats up fish with index  $j$ .  
 It's guaranteed that the main diagonal contains zeros only, and for other elements the following is true:  $a_{ij} = 1 - a_{ji}$ .  
 All real numbers are given with not more than 6 characters after the decimal point.

Output Format:

```

#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]);

}
```

The screenshot shows a web browser window with the URL `care.srmmp.edu.in/srmmpetelab/#/srmmpetelab/student/home`. The page is titled 'CHALLENGE INFORMATION' and displays details for a course named 'OOPS'. A green notification bar at the top states: 'You have already solved this challenge! Though you can run the code with different logic!'. Below this, a table shows the course details, including the session 'Classes, Methods & Constructors' and the question information 'Level 1 Challenge 13'. The problem description is visible, detailing a scenario where Yohan gives gifts to his friends and asks for help to find out how many friends gave him a gift.

```
#include <iostream>

using namespace std;

class Friends
{
public: void Gifts(){
    int i, n, a, b[50] = { 0 };

    cin >> n;

    for (i = 1; i < n+1; i++)
    {
        cin >> a;

        b[a] = i;
    }

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

}

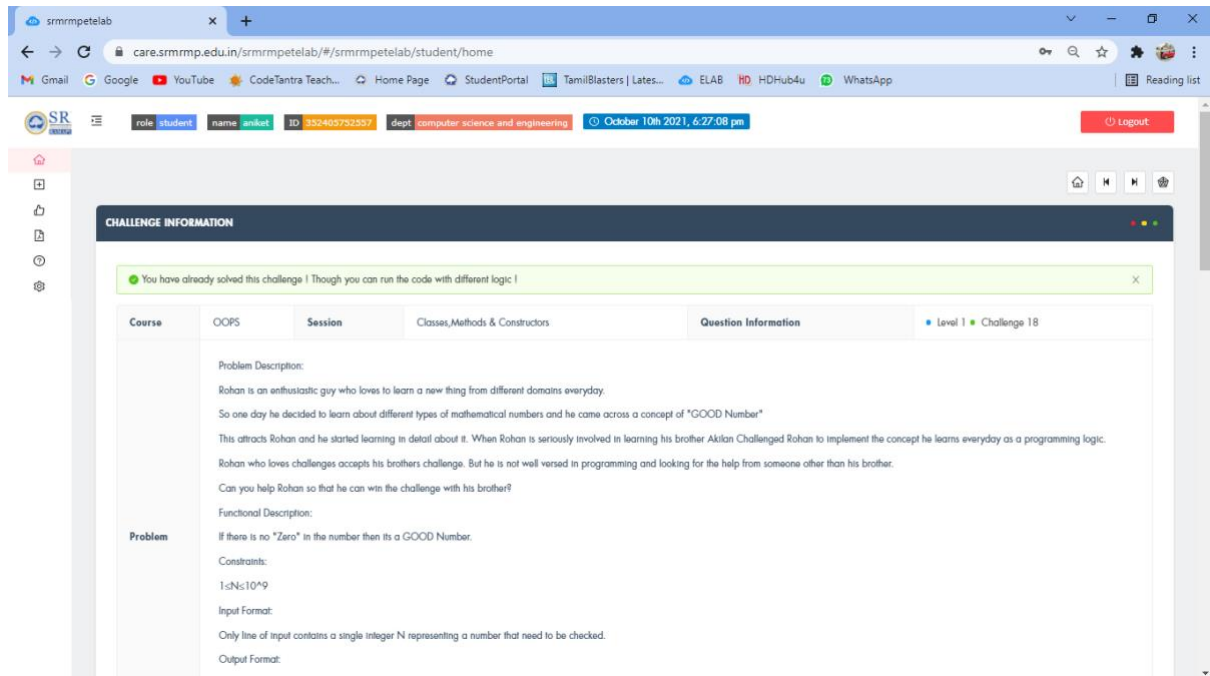
};

int main()
```

```

{
    Friends Sharing;
    Sharing.Gifts();
}

```



```

#include <iostream>

using namespace std;

class GoodNum
{
public:
    void check(int tNum)
    {
        int cnt=0;
        int rem;
        while(tNum>0)
        {
            rem=tNum%10;
            if(rem==0)
                cnt++;
            tNum/=10;
        }
    }
}

```

```

}

if(cnt==0)

cout<<"GOOD Number"<<endl;

else

cout<<cnt;

}

};

int main(){

    int N;

    cin>>N;

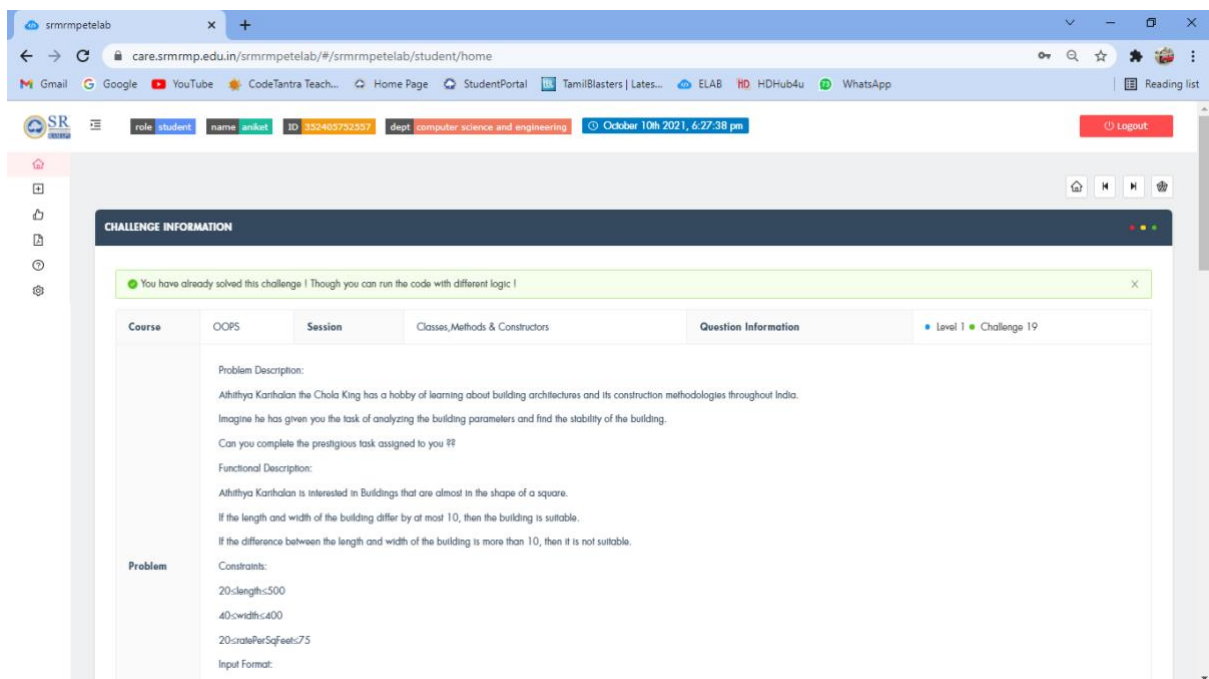
    GoodNum Learning;

    Learning.check(N);

    return 0;

}

```



```

#include <iostream>

#include <math.h>

using namespace std;

class Building
{

```

```

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

int main()
{
    Building construction;

```

```

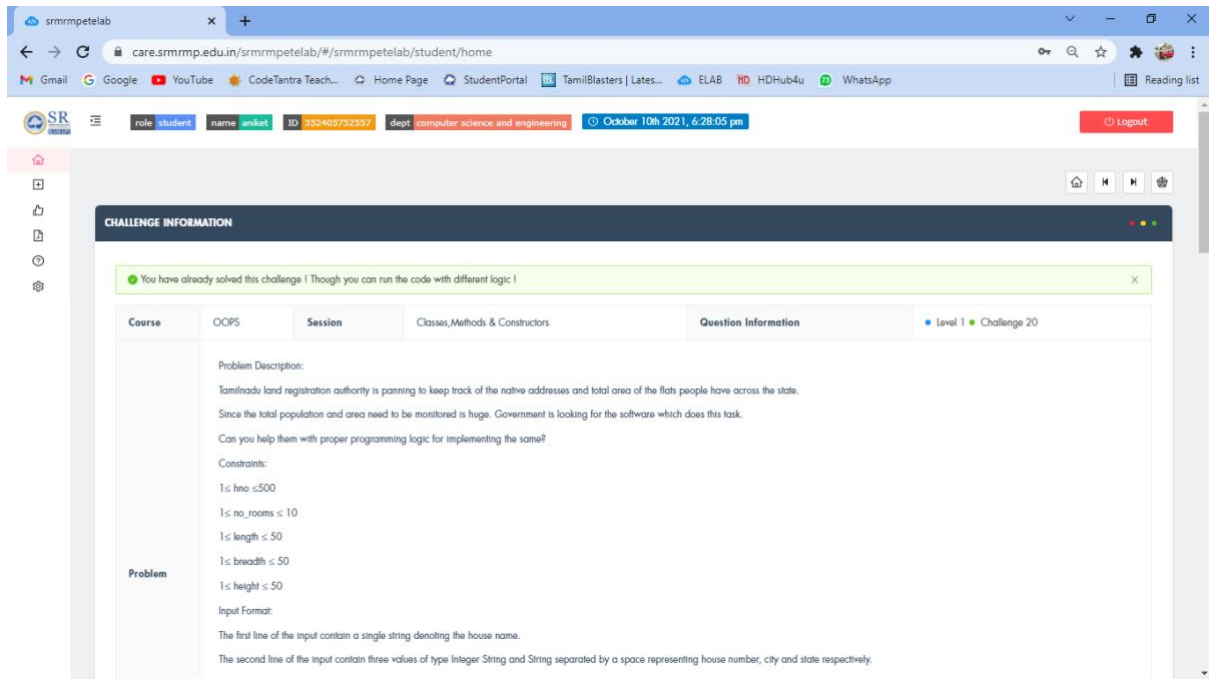
construction.calculateCost();

construction.determineSuitability();

return 0;

}

```



```

#include <iostream>

using namespace std;

class address
{
    int hno;

    char cty[20];

    char state[20];

public:

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

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

```



```

        cout<<"City="<<cty<<endl;

        cout<<"State="<<state<<endl;
    }
};

class house
{

    char housename[30];
    address a;

    int n;
public:
    void input();
};

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

    cin>>n;

    int lenght,widht,height;
    for (int i = 0; i < n; i++)
    {
        cin>>lenght>>widht>>height;

        cout<<"Detail of Room "<<i+1<<endl;

        cout<<"Length="<<lenght<<endl;
        cout<<"Breadth="<<widht<<endl;
        cout<<"Height="<<height<<endl;
    }
}

```

```

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

```

The screenshot shows a mobile application interface with a dark theme. At the top, the status bar displays the time 11:19 PM, signal strength, and battery level at 85%. The app's header shows a download speed of 3.7KB/s. The main content area is titled "Problem" and contains the following text:

**Problem Description:**

Abhilash wants to save money for his first car.

He puts money in the Bank every day.

He starts by putting in \$1 on Monday, the first day.

Every day from Tuesday to Sunday, he will put in \$1 more than the day before.

On every subsequent Monday, he will put in \$1 more than the previous Monday.

Given n, print the total amount of money he will have in the bank at the end of the n<sup>th</sup> day.

Can you Help Abhilash in knowing how much saving he have at the end of n<sup>th</sup> day?

**Constraints:**

$1 \leq n \leq 1000$

```

#include <iostream>

using namespace std;

class Bank
{
    int total;

    public:

    void totalMoney(int n)
    {
        int r;

        r = n%7;

        n/=7;

        total =(n*(49+(7*n)))/2 + r*(2*(n+1)+r-1)/2;
    }
}

```

```

        cout<<total;
    }
};

int main(){

    int n;

    cin>>n;

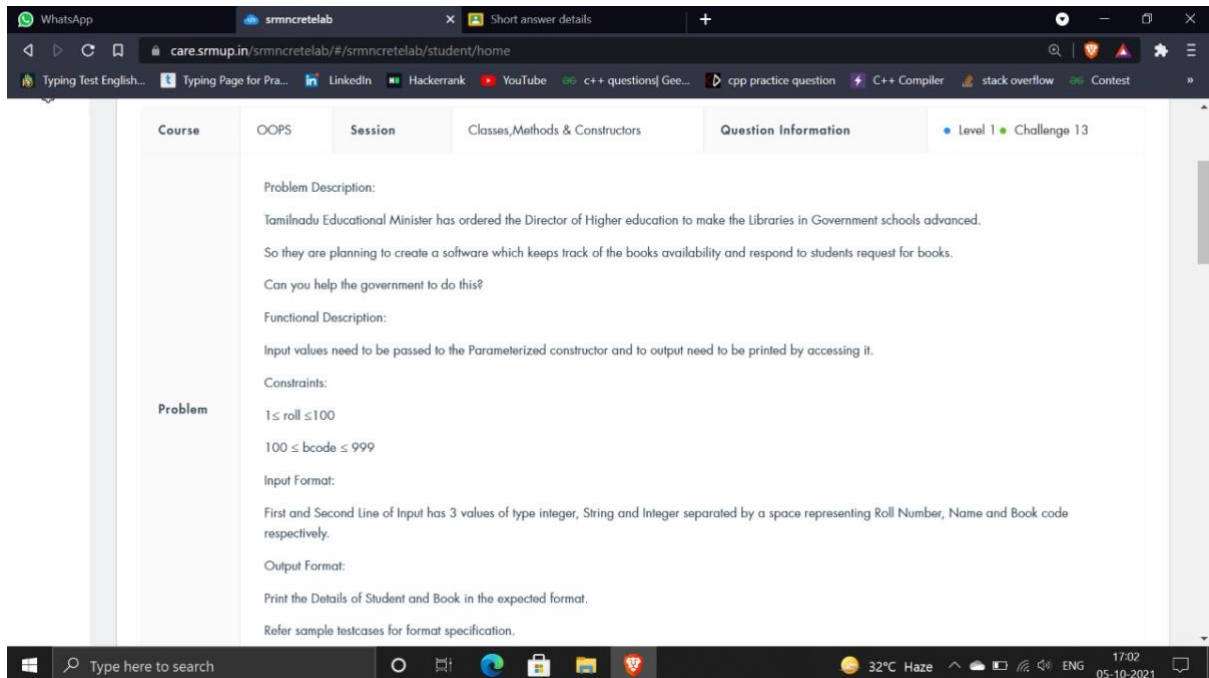
    Bank CalculateMoney;

    CalculateMoney.totalMoney(n);

    return 0;

}

```



```

#include <iostream>

using namespace std;

class student
{
    string name;

    int roll;

    float height, weight;

public:

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

```

```

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

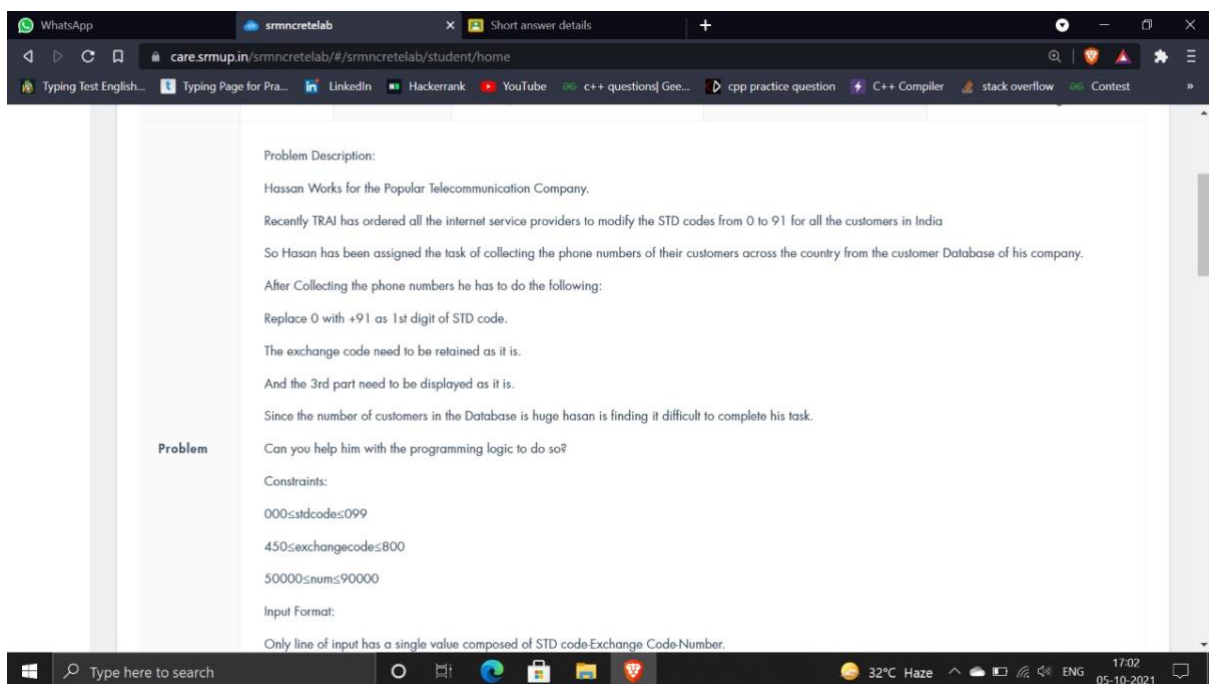
void displaydata()
{
    cout<<name<<" "<<roll<<" "<<height<<" "<<weight<<endl;
}

};

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

    return 0;
}

```



```
#include <iostream>

using namespace std;
```

```
class Phone
{
    public:
    char n[14];
    void change()
    {
        cin>>n;

        n[0]='1';

        cout<<'9'<<n;
    }
};
```

```
int main()
{
    Phone obj;

    obj.change();

    return 0;
}
```