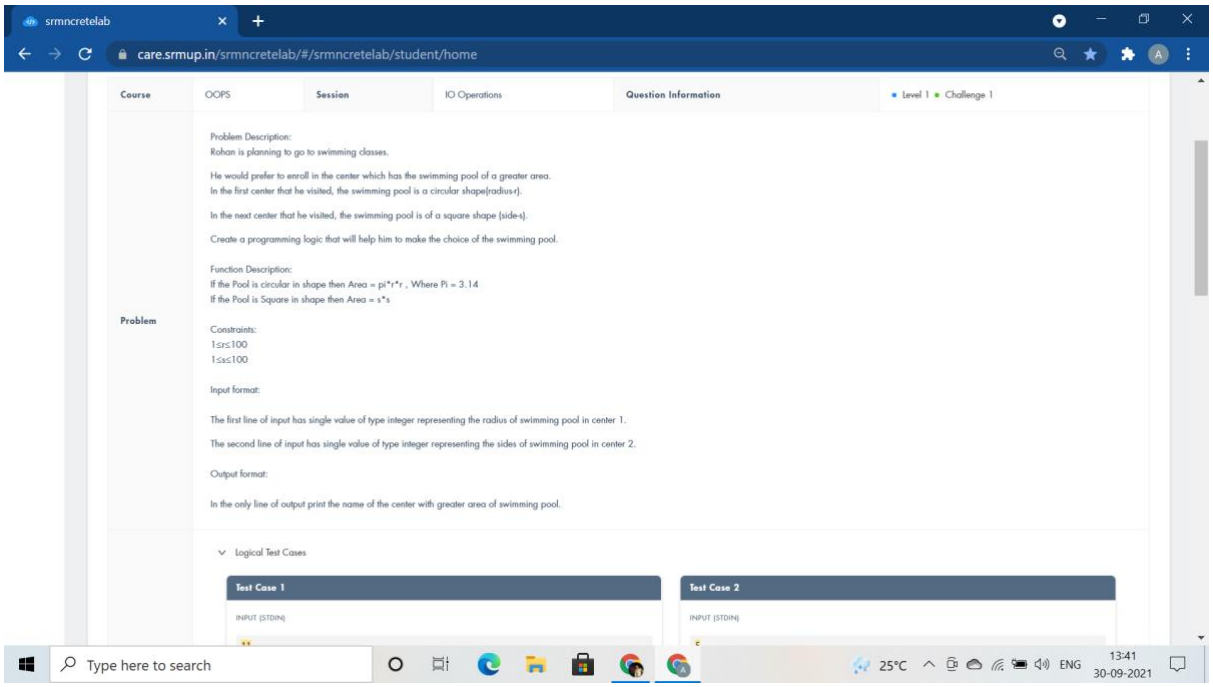
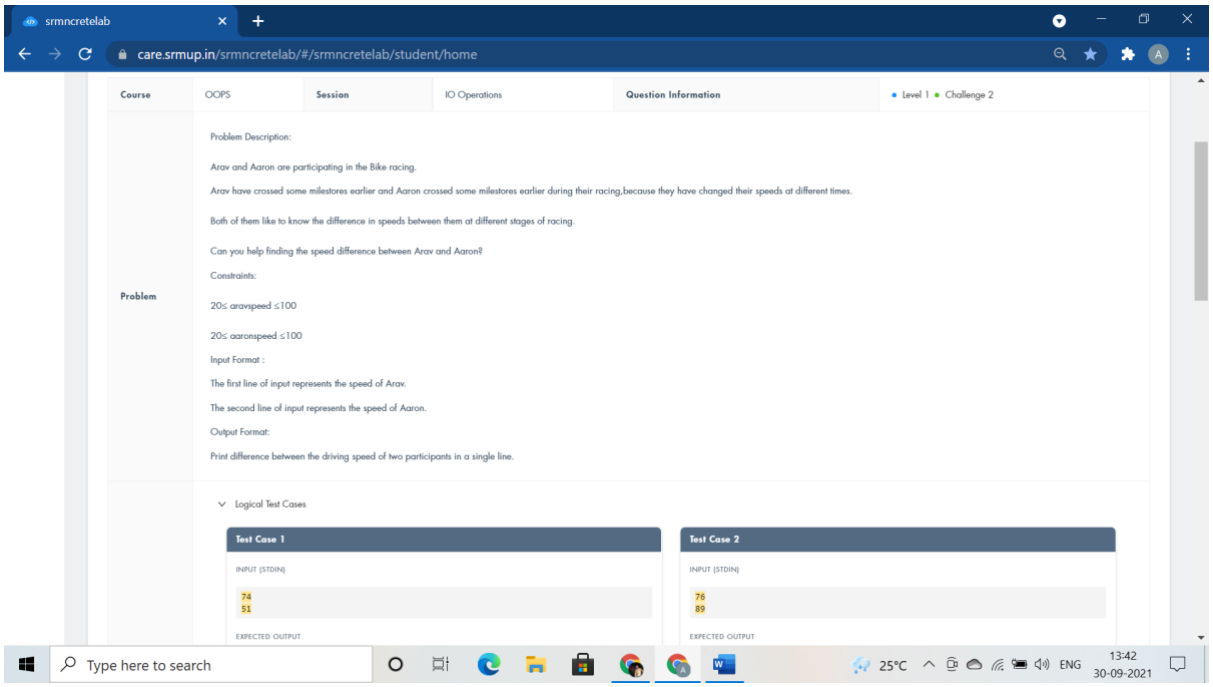
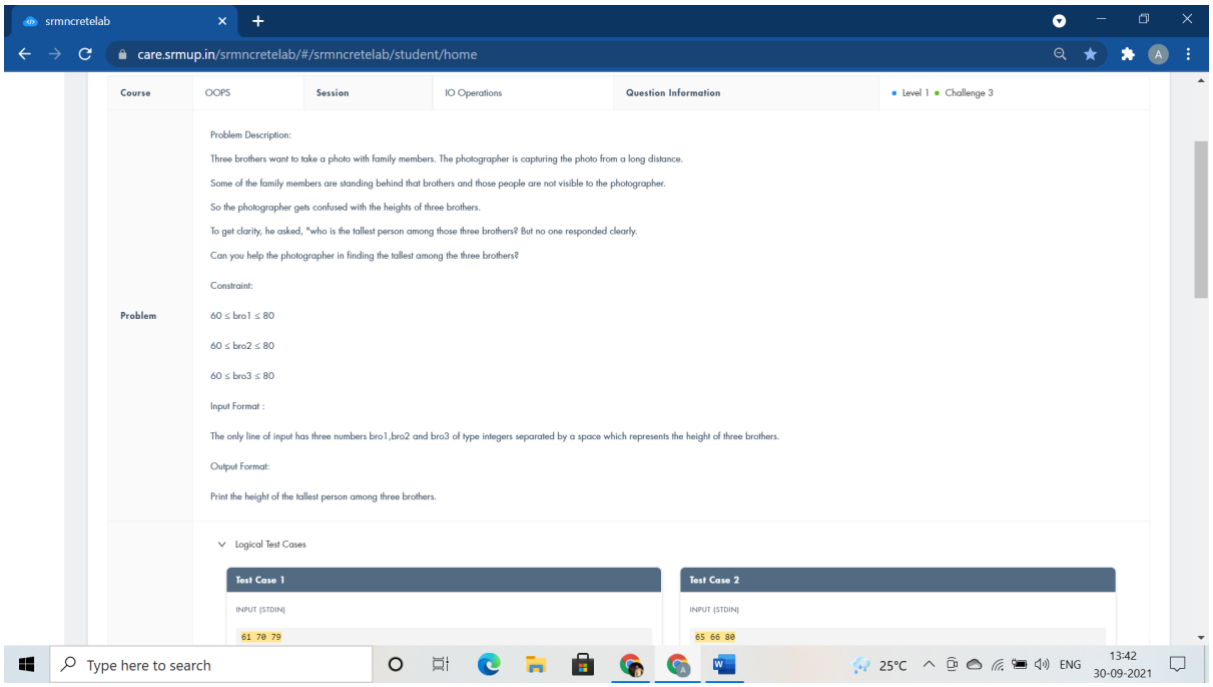
INPUT OUTPUT:-



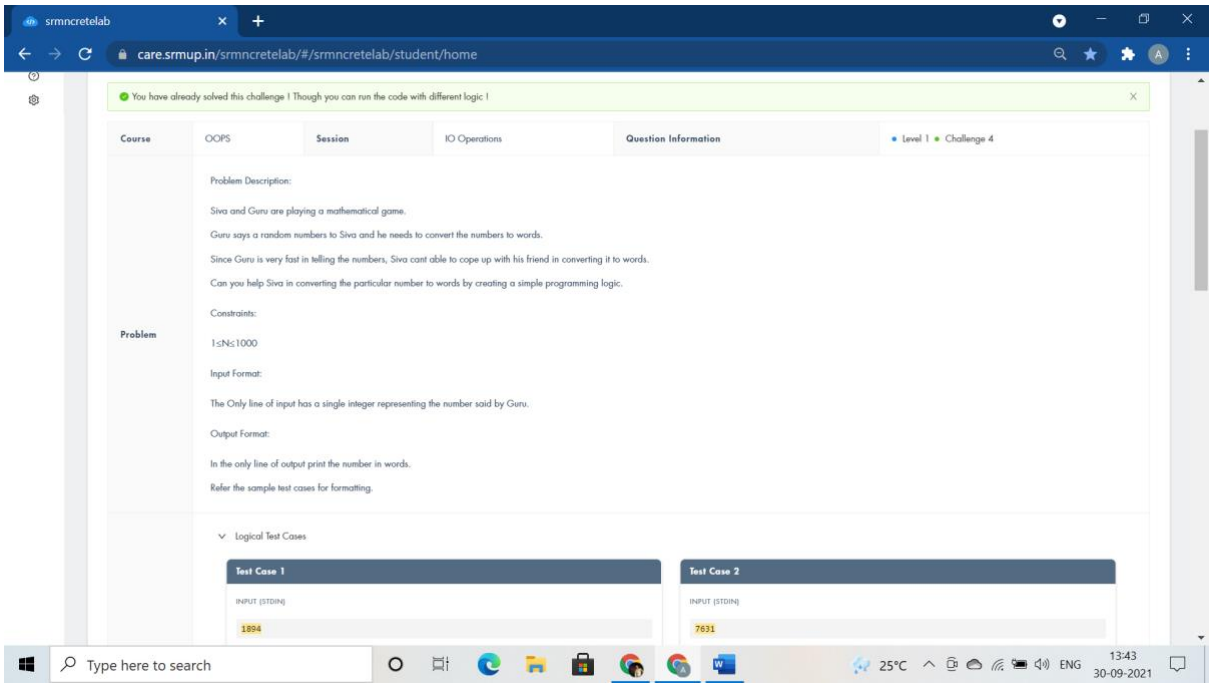

```
#include <iostream>
using namespace std;
int main()
{
   int r,s,cpool,spool;
   cin>>r>>s;
   cpool=3.14*r*r;
   spool=s*s;
   if(cpool>spool)
   cout<<"I Prefer Centre 1";
   else
   cout<<"I Prefer Centre 2";
          return 0;
}
```




```
#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;
}
```



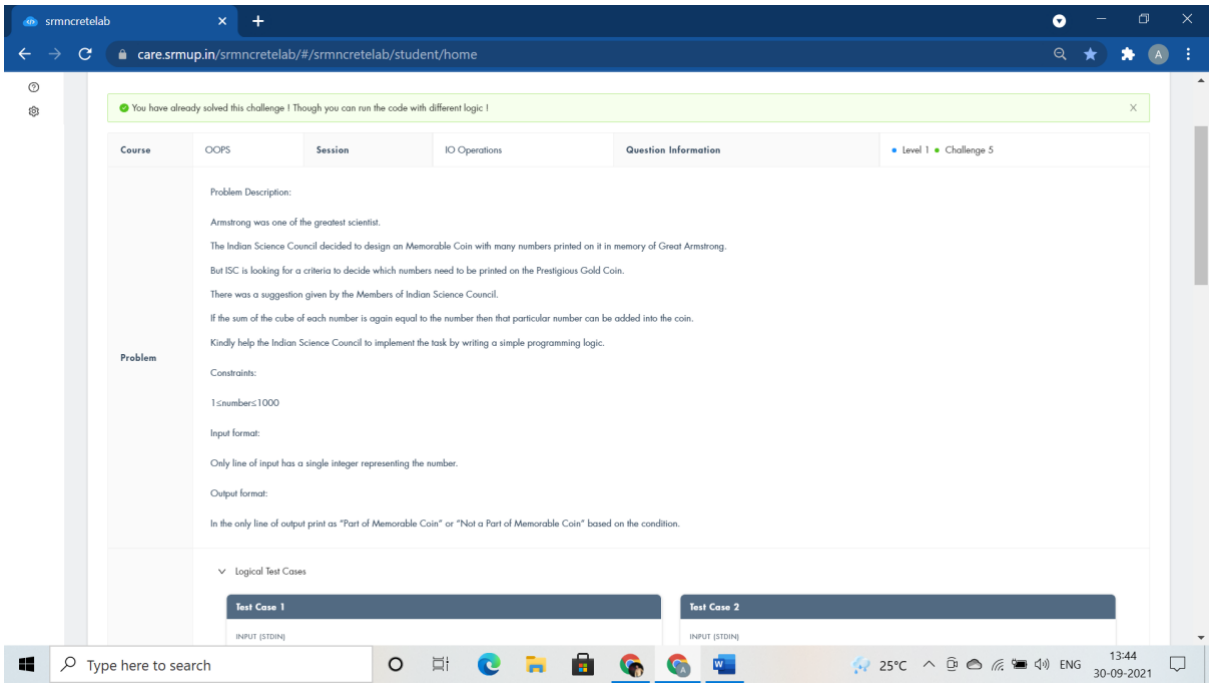

```
#include <iostream>
using namespace std;
int main() {
   int bro1,bro2,bro3;
   cin>>bro1>>bro2>>bro3;
   if(bro1>bro2)  {
      if(bro1>bro3)
      cout<<bro1;
      else
      cout<<bro3;
   }
   else if(bro2>bro3)
   cout<<bro2;
   else
   cout<<bro3;
         return 0;
}
```




```
#include <iostream>
using namespace std;
int main()
{
   int n,dig=0,rem;
   cin>>n;
   while(n!=0)
   {
      rem=n%10;
      dig=dig*10+rem;
      n/=10;
   }
   while(dig!=0)
   {
      rem=dig%10;
      switch(rem)
      {
```

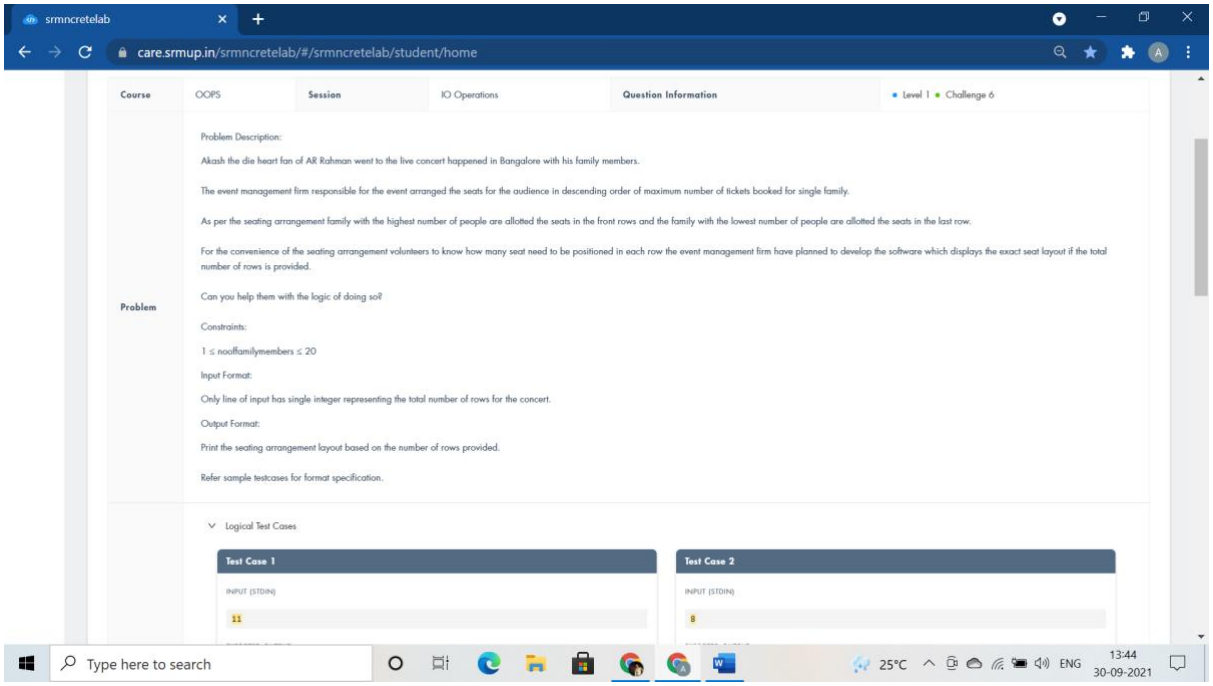
```
    case 0:
    cout<<"Zero ";
    break;
    case 1:
    cout<<"One ";
    break;
    case 2:
    cout<<"Two ";
    break;
    case 3:
    cout<<"Three ";
    break;
    case 4:
    cout<<"Four ";
    break;
    case 5:
    cout<<"Five ";
    break;
    case 6:
    cout<<"Six ";
    break;
    case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

```
    dig/=10;
  }
        return 0;
}
```



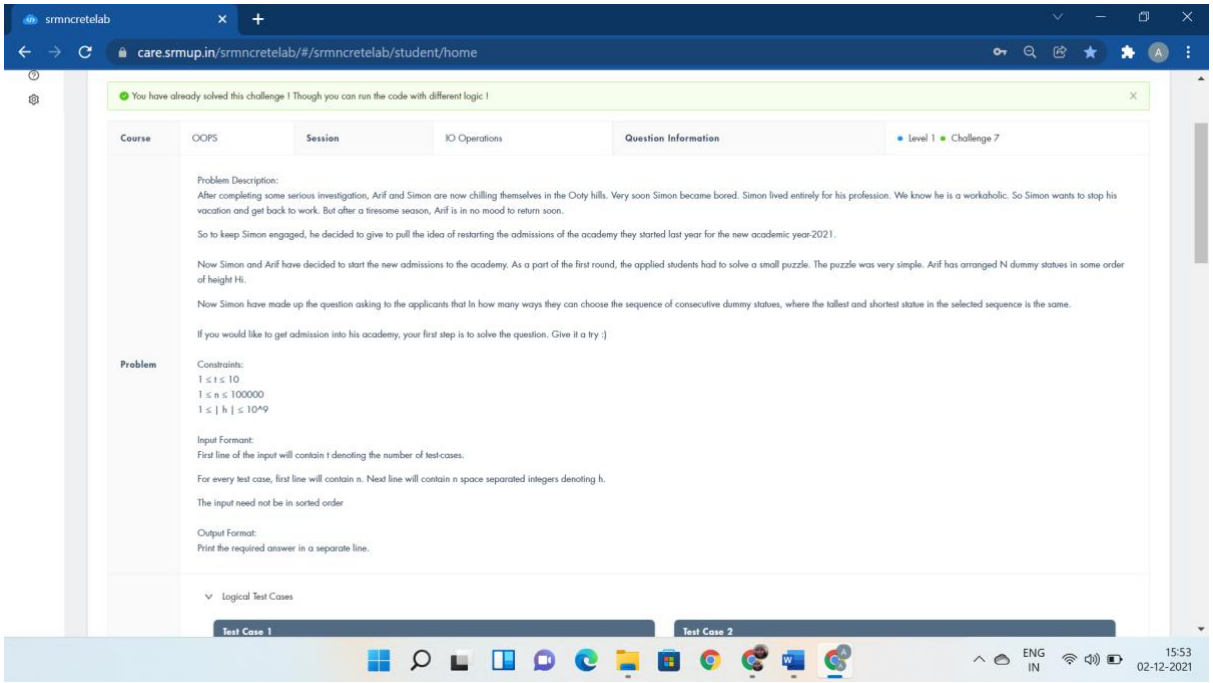

```
#include <iostream>
using namespace std;
int main()
{
   int number,num,rem,result=0;
   cin>>number;
   num=number;
   while(num!=0) {
      rem = num%10;
      result+=rem*rem*rem;
```

```
    num/=10;
  }
  if(result==number)
  cout<<"Part of Memorable Coin";
  else
  cout<<"Not a Part of Memorable Coin";
        return 0;
}
```



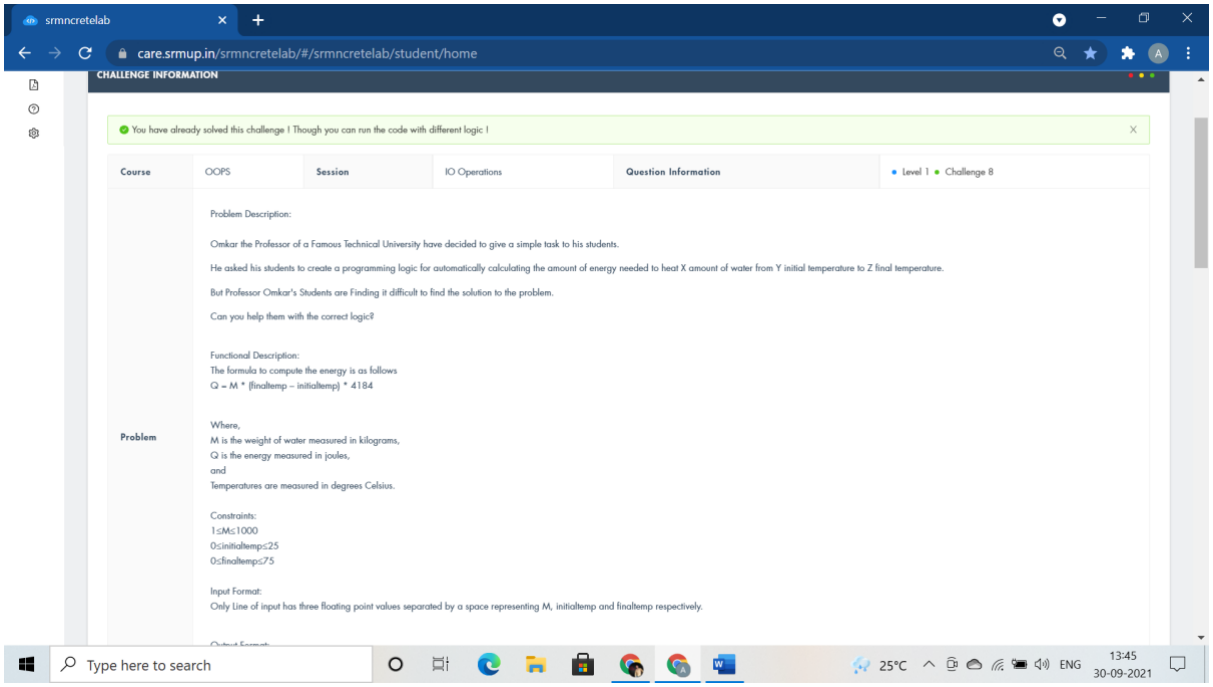

```
#include <iostream>
using namespace std;
int main()
{
  int nooffamilymembers,i,j;
  cin>>nooffamilymembers;
  for(i=nooffamilymembers;i>0;i--)
  {
```

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



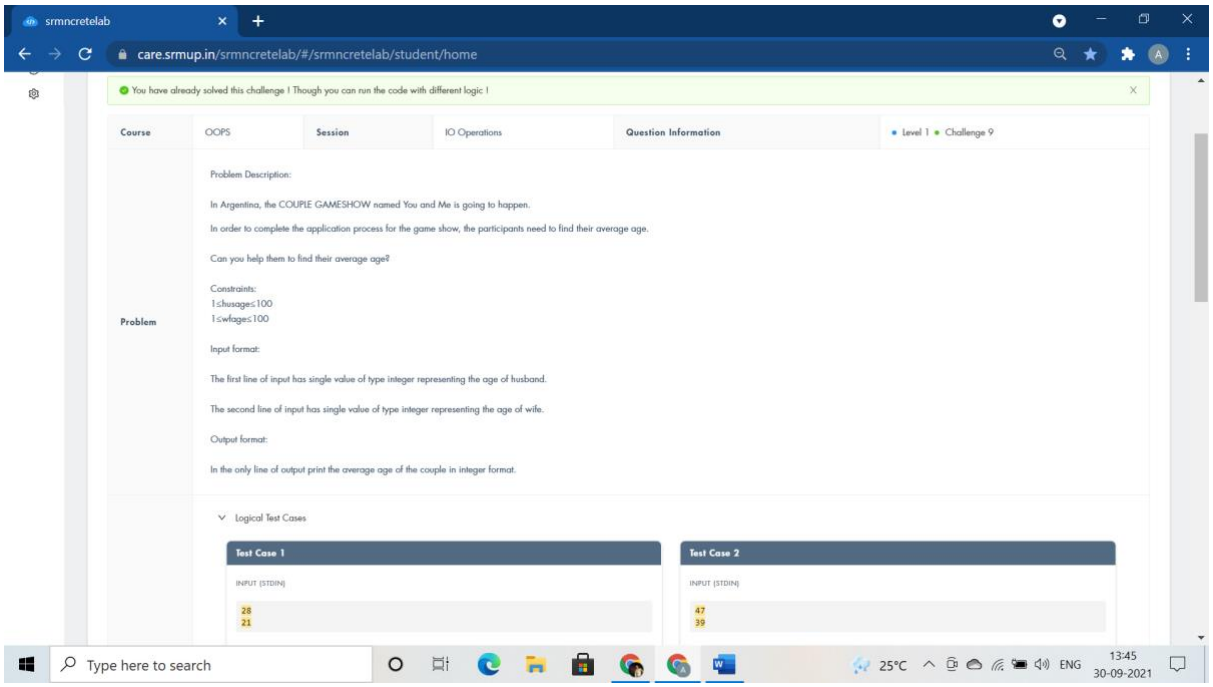

```
#include <iostream>
using namespace std;
int main()
{
  int t,n,h,i,l=1,count;
  cin>>t;
  while(t--)
  {
    l=1;
    count=0;
    cin>>n;
    for(i=1;i<=n;i++) {
```

```
        cin>>h;
        if(h==l) {
           count+=2;
        }
        if(h>l) {
           l=h;
           count++;
        }
      }
      cout<<count<<endl;
   }
   return 0;
}
```



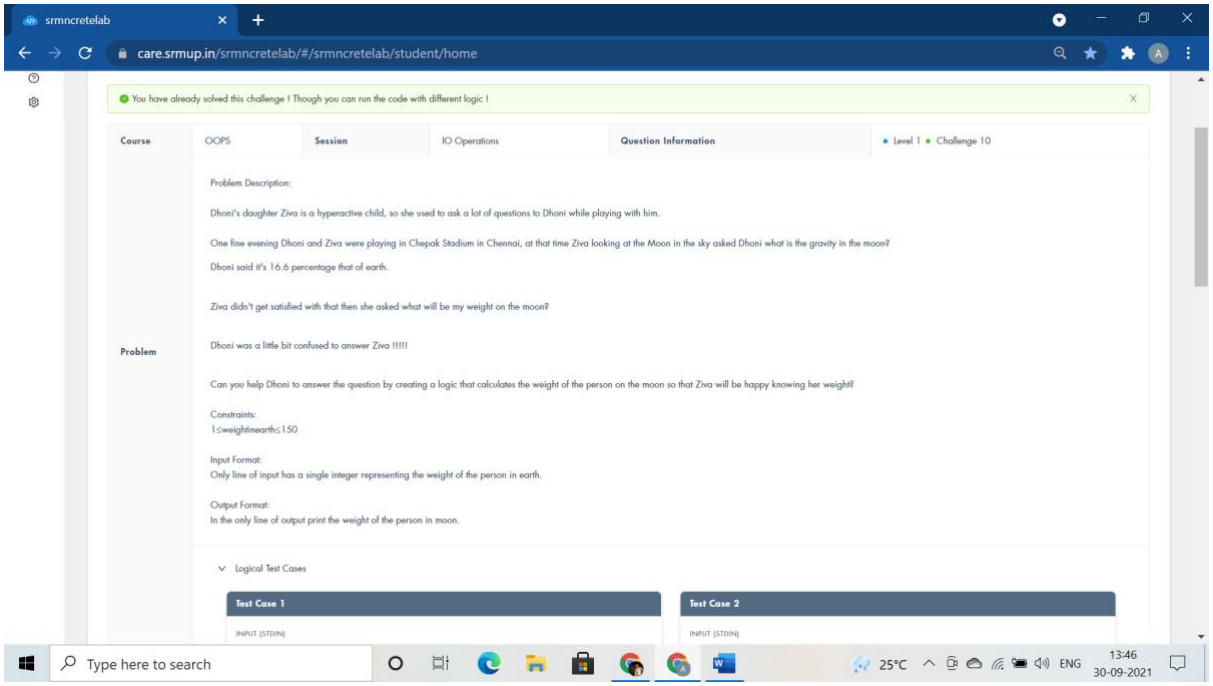

```
#include <iostream>
using namespace std;
int main()
{
```

```
    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;
```

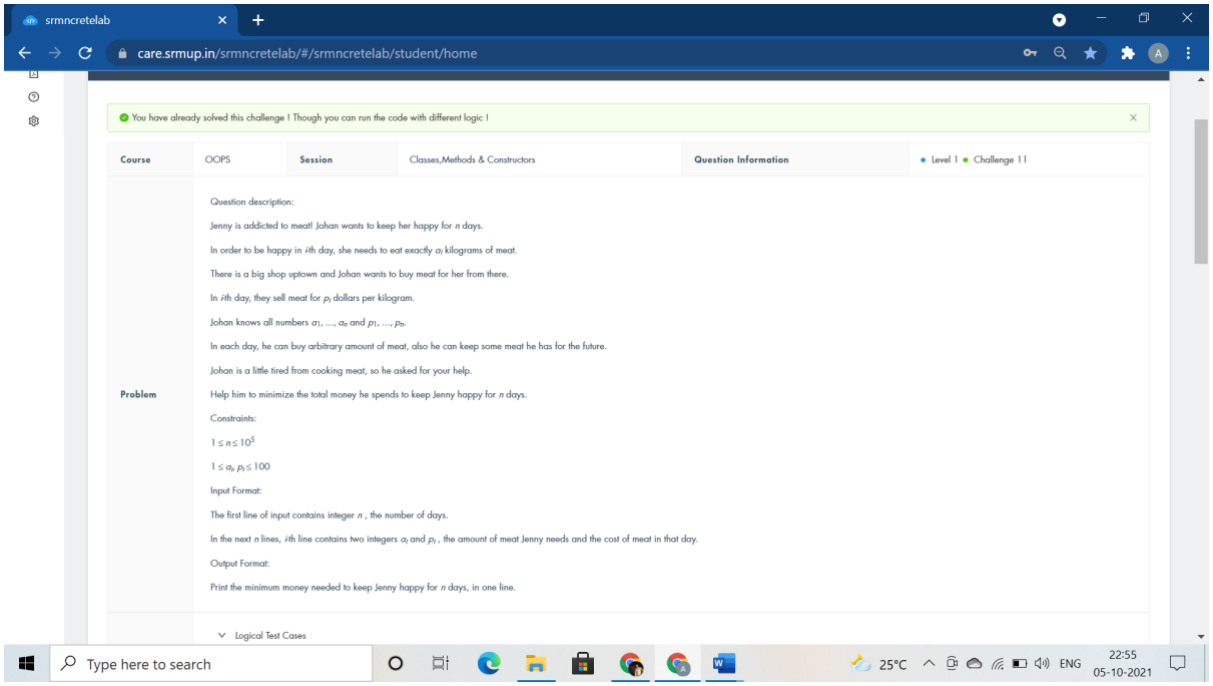
}




```
#include <iostream>
using namespace std;
int main()
{
   int weightinearth;
   cin>>weightinearth;
   float weightinmoon;
   weightinmoon=0.166*weightinearth;
   cout<<weightinmoon;
          return 0;
```

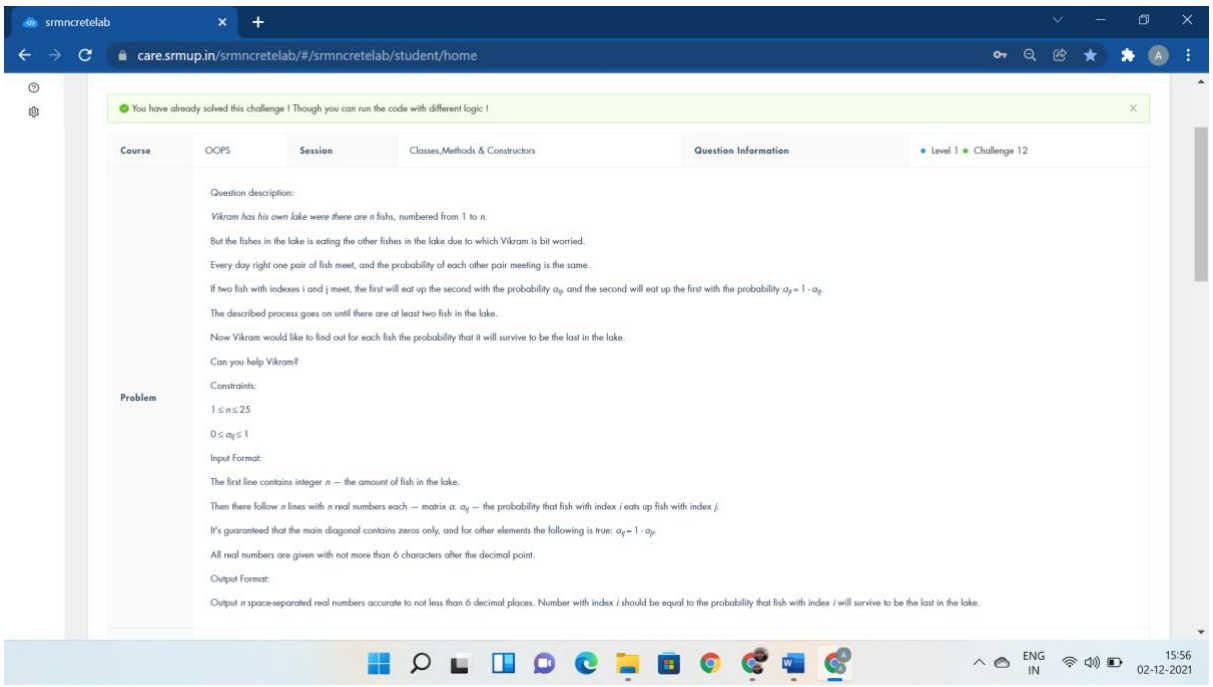
}

CLASSES METHODS AND CONSTRUCTORS:-



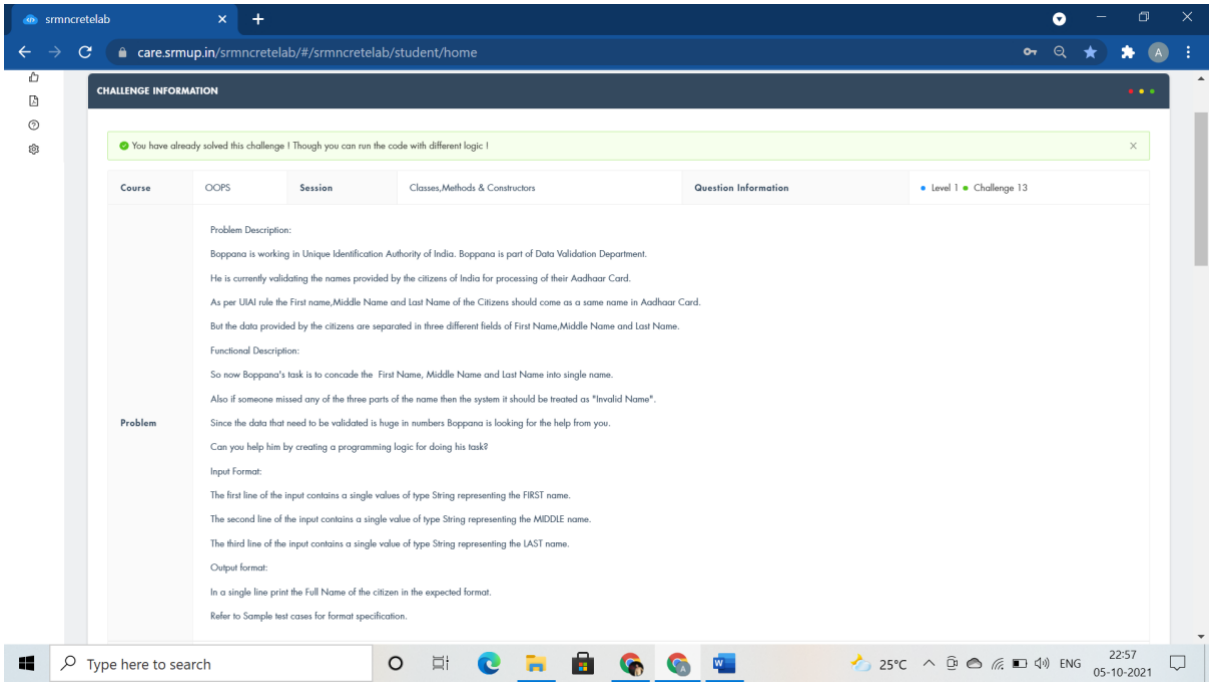

```
#include <iostream>
using namespace std;
class Happiness{
   public:int Meat(){
       int n,a,b,max=100,sum=0;
       cin>>n;
       while(n--)
       {
       cin>>a>>b;

          //max=b;
          if(b>=max)

          sum+=a*max;
         //  cout<<max<<endl;
```

```
        // cout<<sum<<endl;


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


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



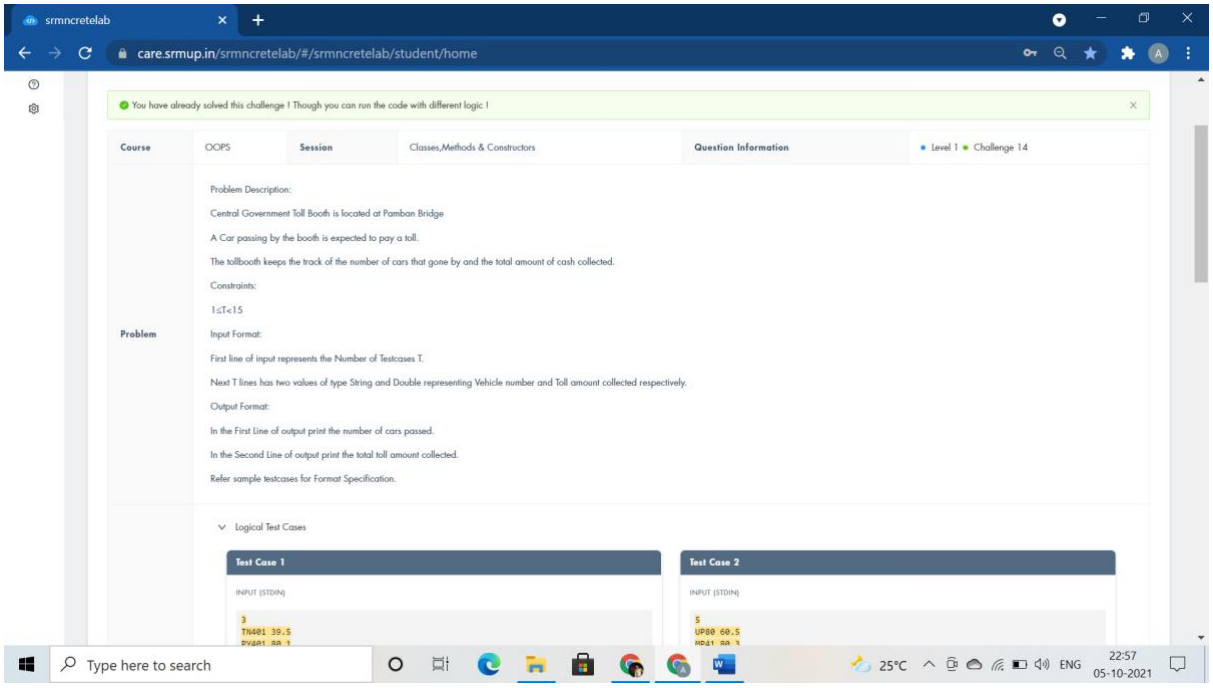
```
#include <iostream>
#include <string.h>
#include <stdio.h>
using namespace std;
double a[18][18], b[1 << 18];
int fun(int x) {
   int s = 0;
   while (x)
   {
      s += x & 1;
      x >>= 1;
   }
         return s;
   }
   int main() {
      if(0)
      cout<<"class Lake public:void survival() fish.survival();";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

```
            if (i & (1 << t))
                b[i - (1 << t)] += b[i] * a[r][t] / c;
    }
    for (r = 0; r < n - 1; r++)
        printf("%.6lf ", b[1 << r]);
    printf("%.6lf\n", b[1 << r]);
}
```



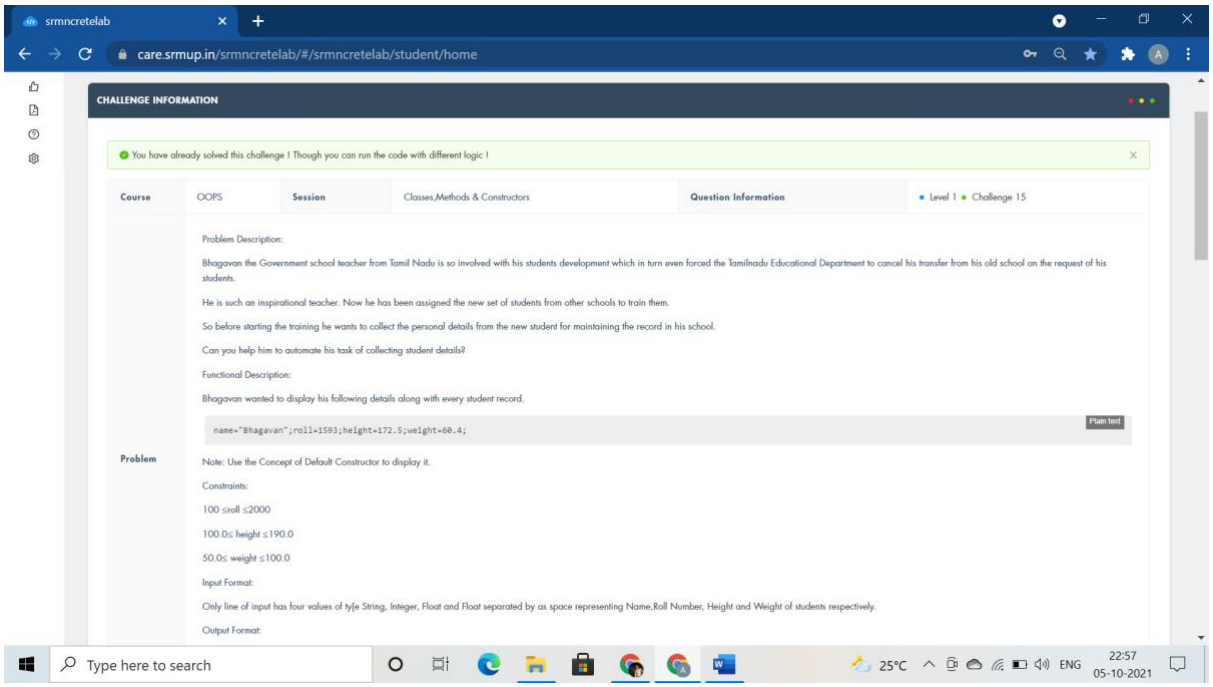

```
#include <iostream>
#include<cstring>
#include<string>
using namespace std;
class aadhaar
{
   public:
   void NameofCitizen(string fn,string mn,string ln)
   {
```

```
    if(fn.empty() || mn.empty() || ln.empty() )
    {
       cout<<"Invalid Name";
    }
    //cout<<"Invalid name"; exit(0) :
    else
    cout<<fn<<mn<<ln;
  }
};
int main()
{
  aadhaar Card;
  string fn,mn,ln;
  cin>>fn>>mn>>ln;
  Card.NameofCitizen(fn,mn,ln);
        return 0;
}
```



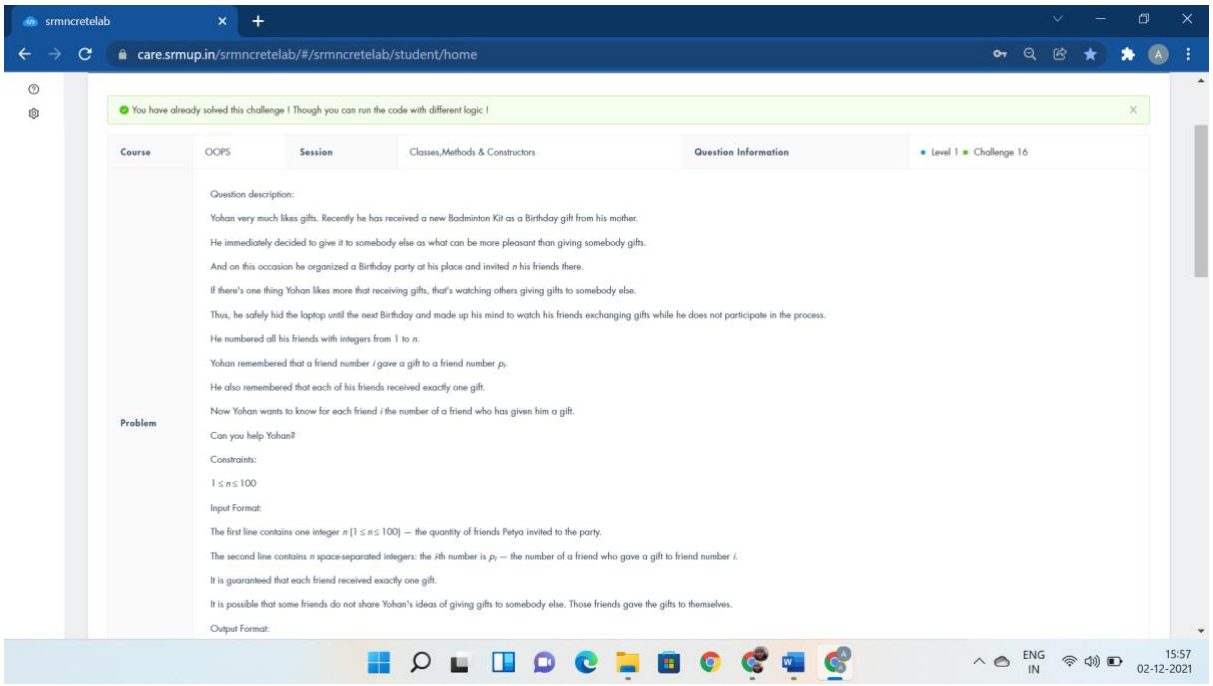
```
#include <iostream>
using namespace std;
class TollBooth
{
   public:
   int cars;
   float tollcollected;
   TollBooth(){
      cars=0;
      tollcollected=0;
   }
   void payingcar(double pay){
      cars++;
      tollcollected+=pay;
   }
   void nonpayingcar(){
      cars++;
   }
   void display(){
      cout<<cars<<endl<<tollcollected<<endl;
   }
};
int main()
{
   TollBooth obj;
   char VehicleNo[10];
   float TollAmt;
   int carpassed,i;
   cin>>carpassed;
   for(i=0;i<carpassed;i++)
   {
```

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



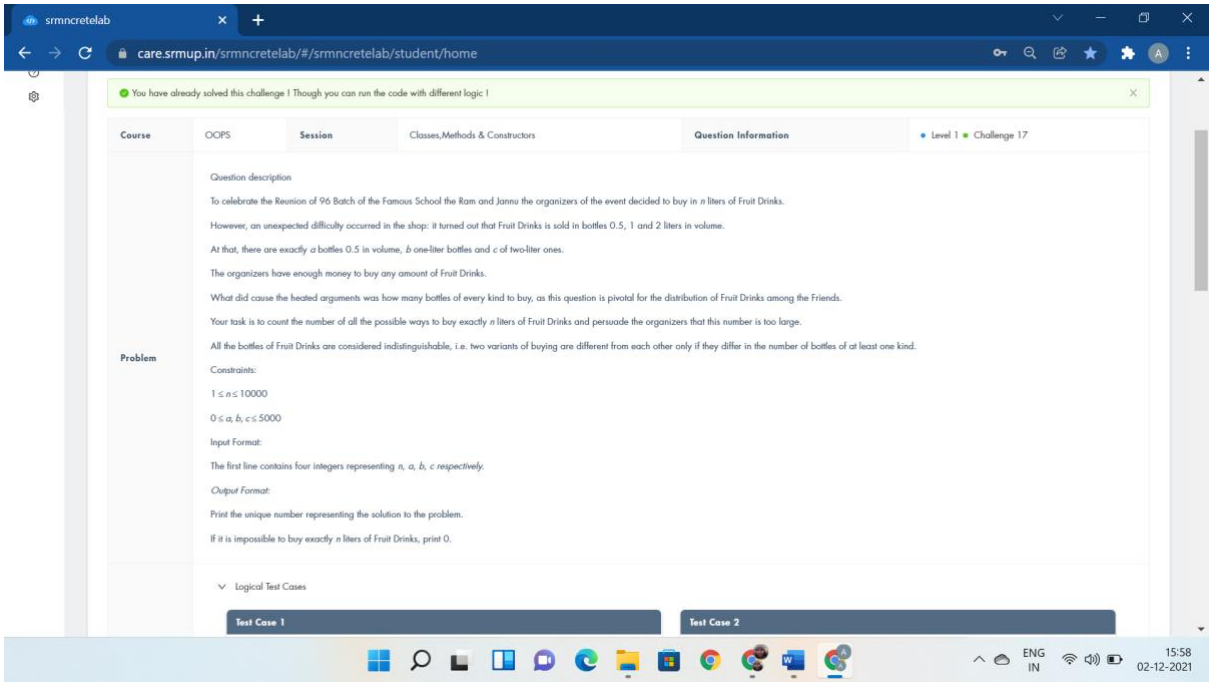

```
#include <bits/stdc++.h>
//#include<iomanip>
//#include<string>
using namespace std;
class student
{
    string name;
    int roll;
    float height,weight;
```

```
    public:
    student(){name="Bhagavan";roll=1593;height=172.5;weight=60.4;}
    void getdata() {
        cin>>name>>roll>>height>>weight;
    }
    void displaydata(){
        cout<<name<<" "<<roll<<" "<<height<<" "<<weight<<endl;
    }
};
int main()
{
    student s1,s2;
    s1.getdata();
    s1.displaydata();
    s2.displaydata();
            return 0;
}
```



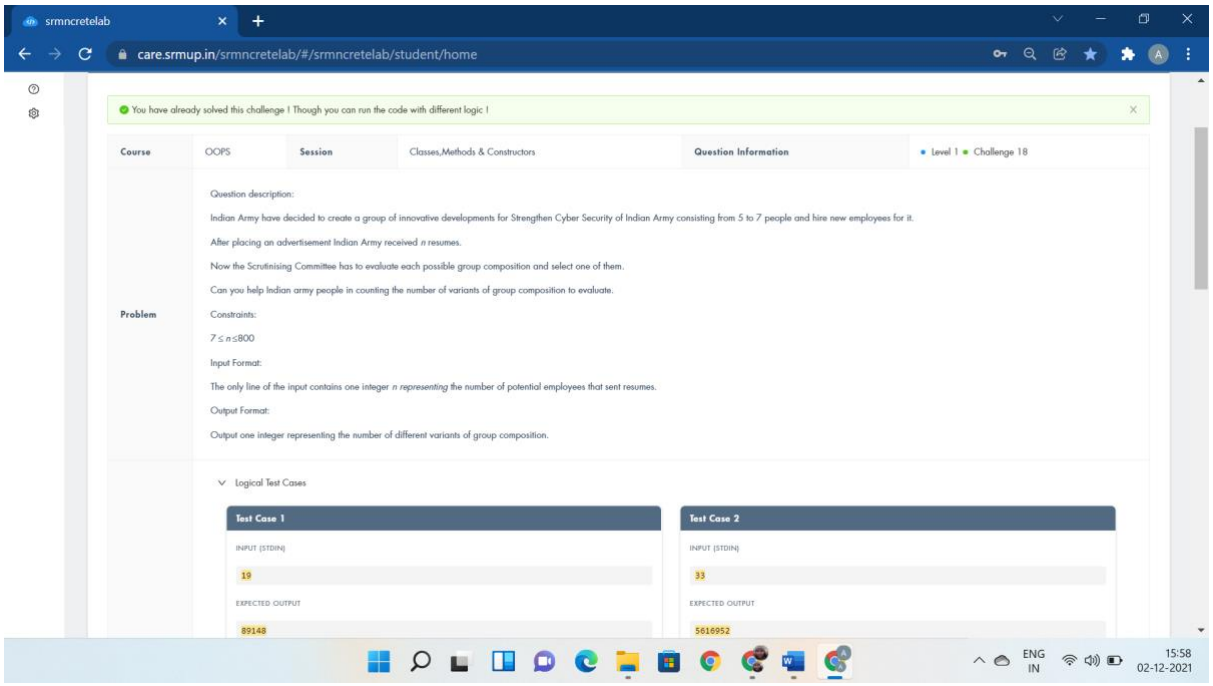

```
#include <iostream>
```

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

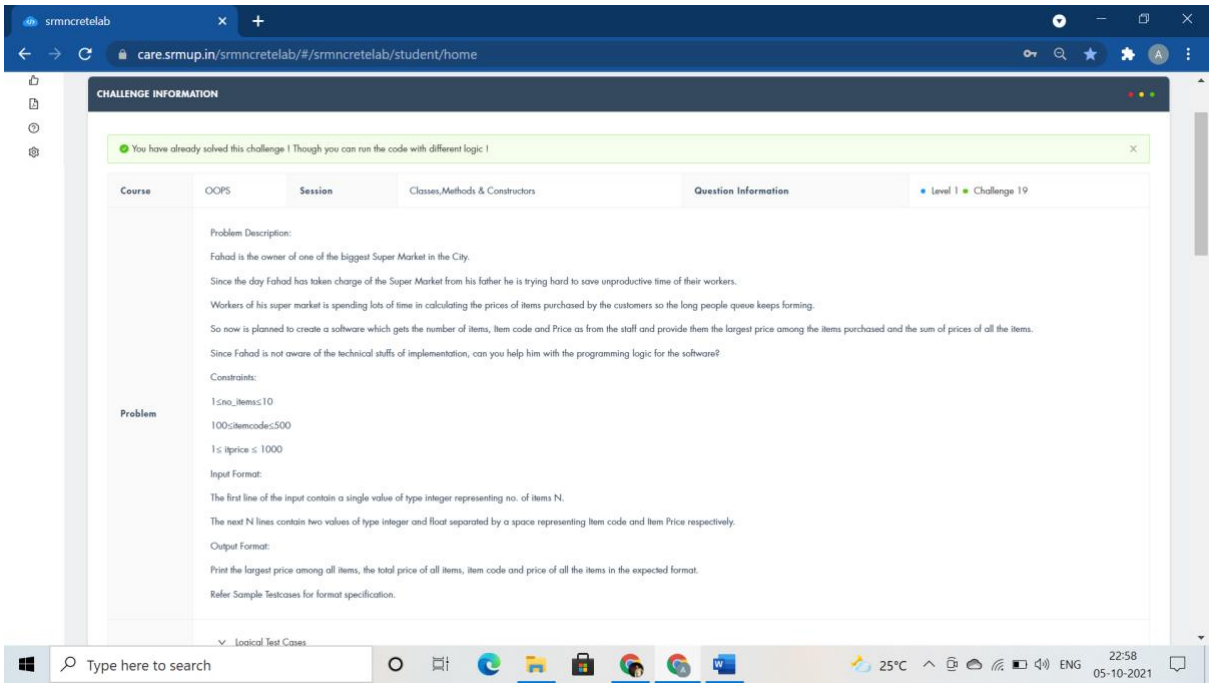



```
#include<bits/stdc++.h>
using namespace std;
class Drinks{
   int n,a,b,c,t,ans=0;
   public:void Shop(){
      cin>>n>>a>>b>>c;
   }
   void display(){
      for(int i=0;i<=b;i++)
                  for(int j=0;j<=c;j++)
                  if(2*(n-i-j*2)>=0&&2*(n-i-j*2)<=a)
                  ans++;
                  cout<<ans;
   }
};
int main(){
   Drinks Buy;
   Buy.Shop();
   Buy.display();
```

}

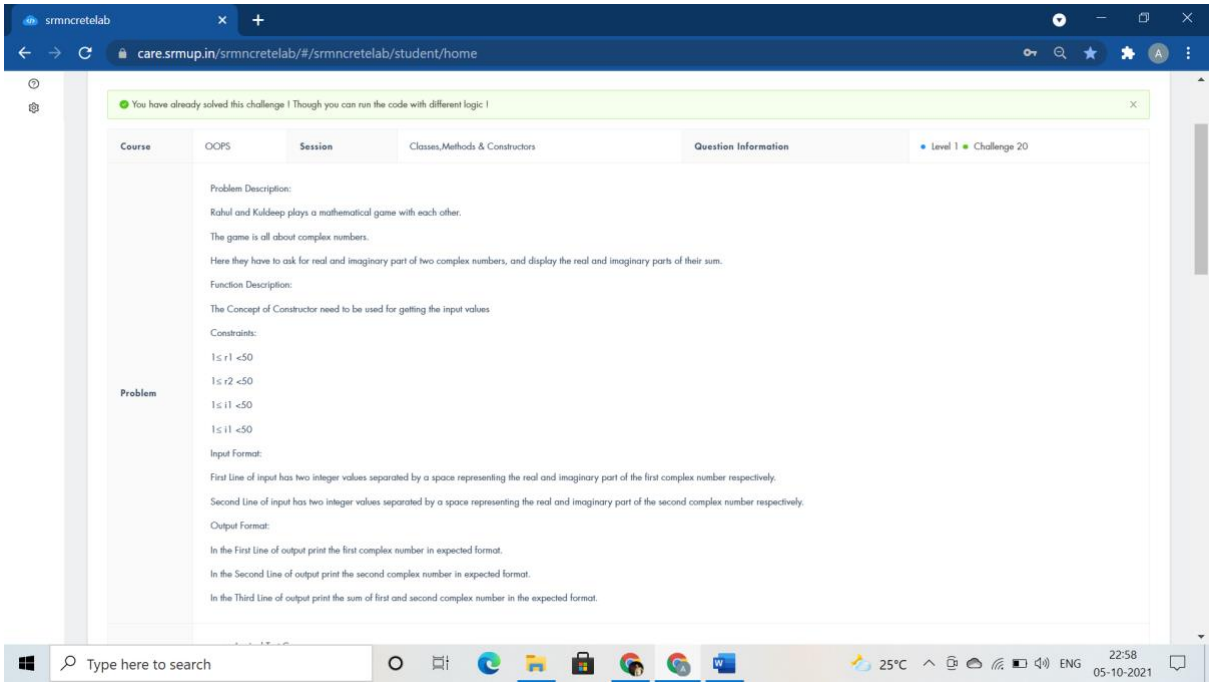



```
#include <bits/stdc++.h>
using namespace std;
class IndianArmy
{
   public:int ResumesofCamdidates(){
            long long n;
            cin>>n;
            long long k=n*(n-1)*(n-2)*(n-3)*(n-4)/120;
            cout<<k+k*(n-5)/6+k*(n-5)*(n-6)/42;
   return 1;
   }
};
int main(){
   IndianArmy GroupingofResumes;
   GroupingofResumes.ResumesofCamdidates();
            return 0;
}
```




```
#include <iostream>
using namespace std;
class ITEM
{
   public:
   int n;
   float large=0,summ=0;
   float arr[100],code[100];
   void getdata(int b){
      n=b;
      for(int i=0;i<n;i++)
      cin>>code[i]>>arr[i];
   }
   void largest(){
      for(int i=0;i<n;i++)
      {
```

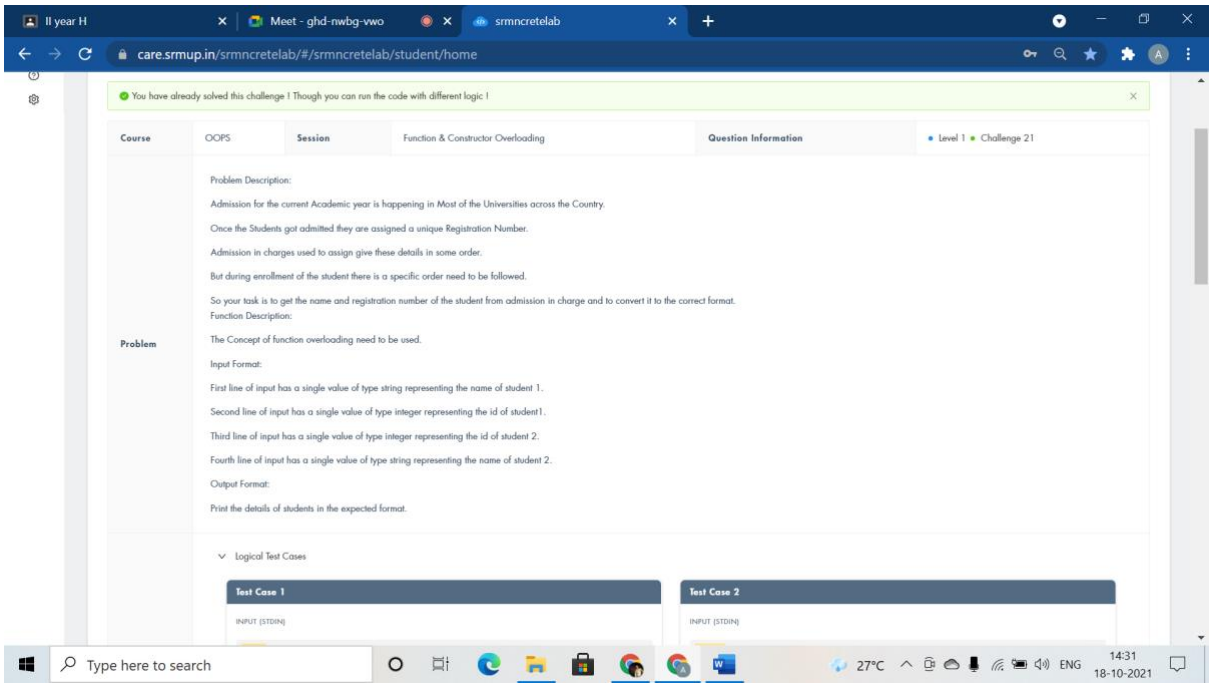
```
        if(arr[i]>=large)
        large=arr[i];
      }
   }
   void sum(){
      for(int i=0;i<n;i++)
      summ+=arr[i];
   }
   void displayitems(){
      cout<<"Largest Price="<<large<<endl;
      cout<<"Sum of Prices="<<summ<<endl;
      cout<<"Code and Price"<<endl;
      for(int i=0;i<n;i++)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

};
using namespace std;
int main()
{
   ITEM order;
   int b;
   cin>>b;
   order.getdata(b);
   order.largest();
   order.sum();
   order.displayitems();
          return 0;
}
```




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

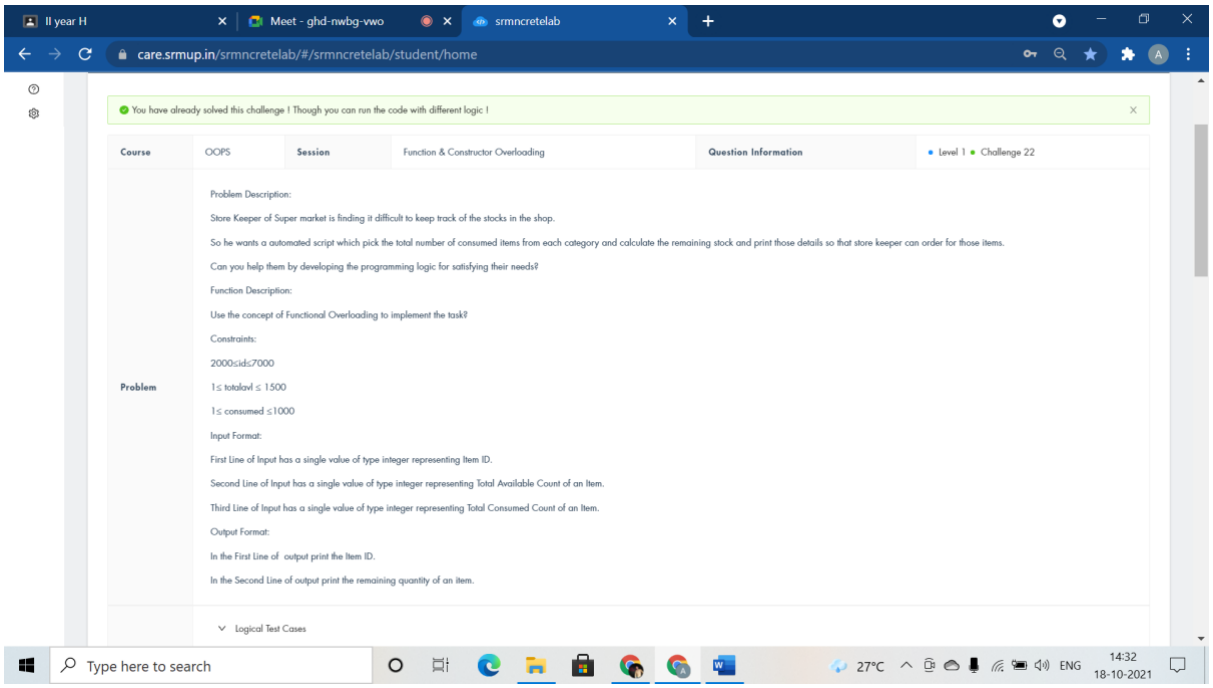
```
    Complex calculate;
   calculate.addcomplex();
   calculate.displaycomplex();
    return 0;
}
```

Constructor Overloading:-

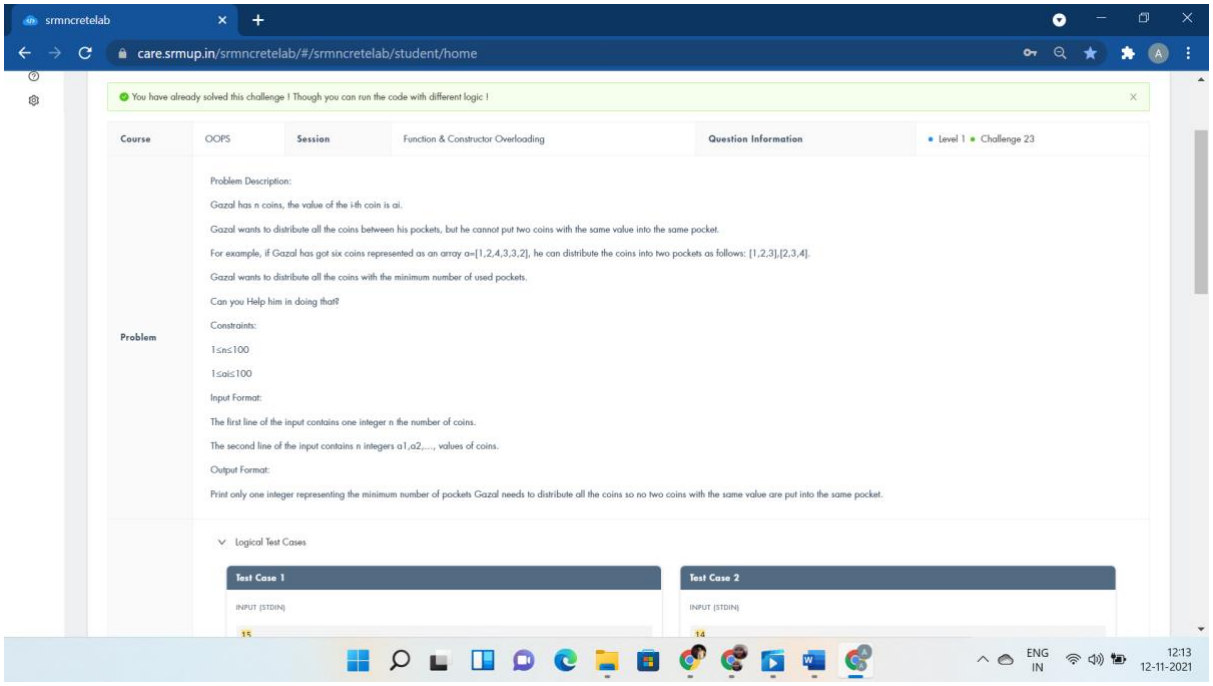



```
#include <iostream>
using namespace std;
class Student
{
   public:
   void Identity(string name,int id){
      cout<<name<<" "<<id<<endl;
```

```
    }
    void Identity(int id,string name){
        cout<<name<<" "<<id<<endl;
    }
};
int main()
{
    Student Details;
    string name;
    int id;
    cin>>name>>id;
    Details.Identity(name,id);
    cin>>id>>name;
    Details.Identity(id,name);
            return 0;
}
```

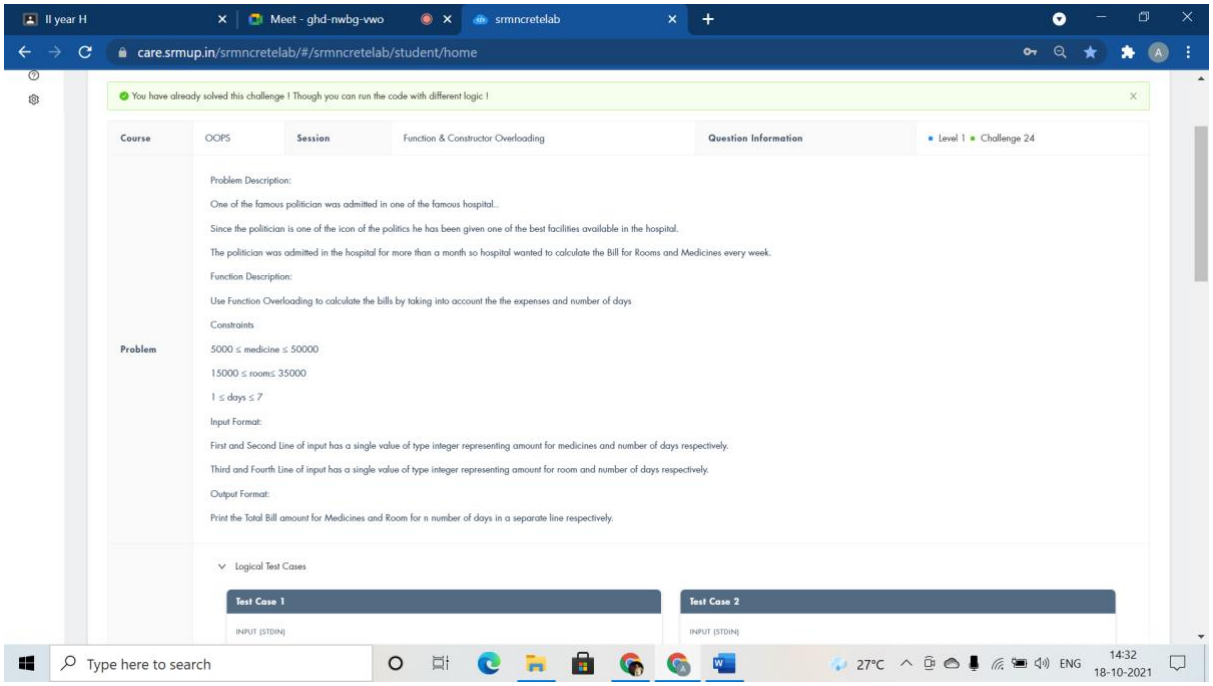


```
#include <iostream>
using namespace std;
class Store{
   public:
   void itemcount(int id){
      cout<<id<<endl;
   }
   void itemcount(int totalavl,int consumed){
      cout<<totalavl - consumed<<endl;
   }
};
int main()
{
   Store purchase;
   int id,totalavl,consumed;
   cin>>id>>totalavl>>consumed;
   purchase.itemcount(id);
   purchase.itemcount(totalavl,consumed);
         return 0;
}
```



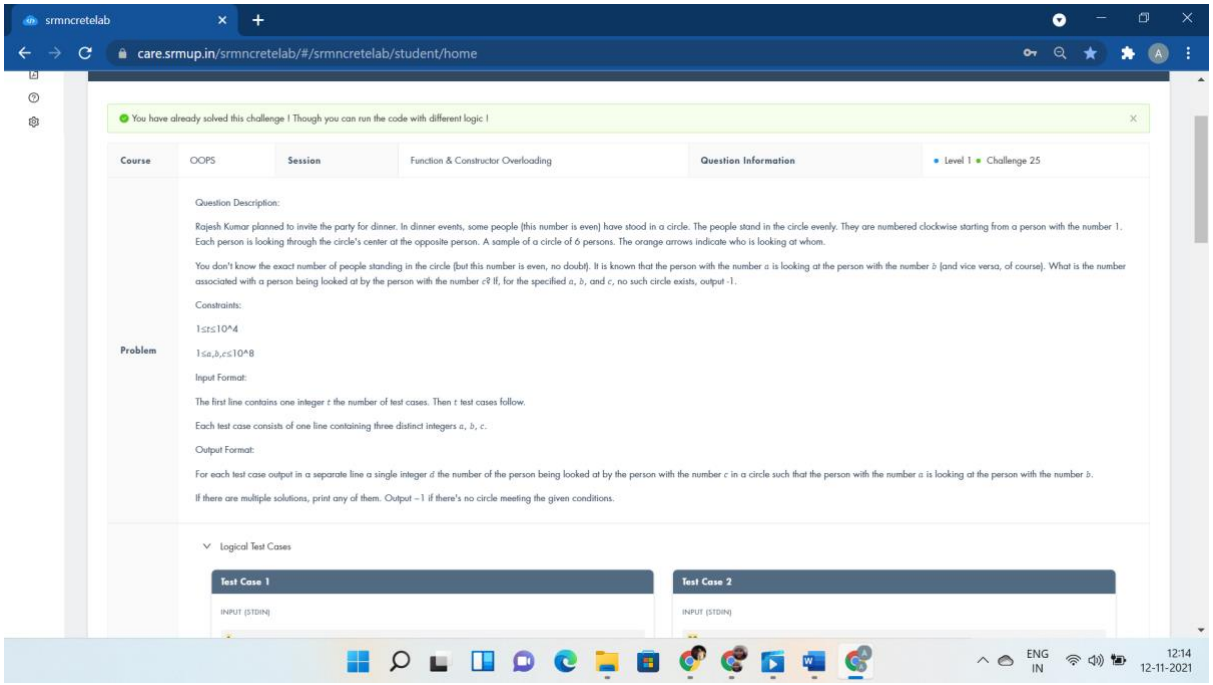

```
#include<bits/stdc++.h>
using namespace std;
int i,n,a,mx=INT_MIN,c[1000];
int res(int n);
int dis(int n,int mx);
int main(){
        cin>>n;
        mx=res(n);
        cout<<dis(n,mx);
        return 0;
        cout<<"int* GazalCoin(int arr[],int n) int* GazalCoin(int arr[],int n,int i) GazalCoin(arr,n,0);";
}
int res(int n){
   for(i=0;i<n;i++){
                cin>>a;
                c[a]++;
                mx=max(mx,c[a]);
        }
        return mx;
```

```
}
int dis(int n,int mx){
   if(n%mx==1 && n%11!=0)
            return mx+1;
         if(n%mx==1 && n%11 == 0)
            return mx;
         if(n%mx==2)
            return mx+1;
         return mx;
}
```



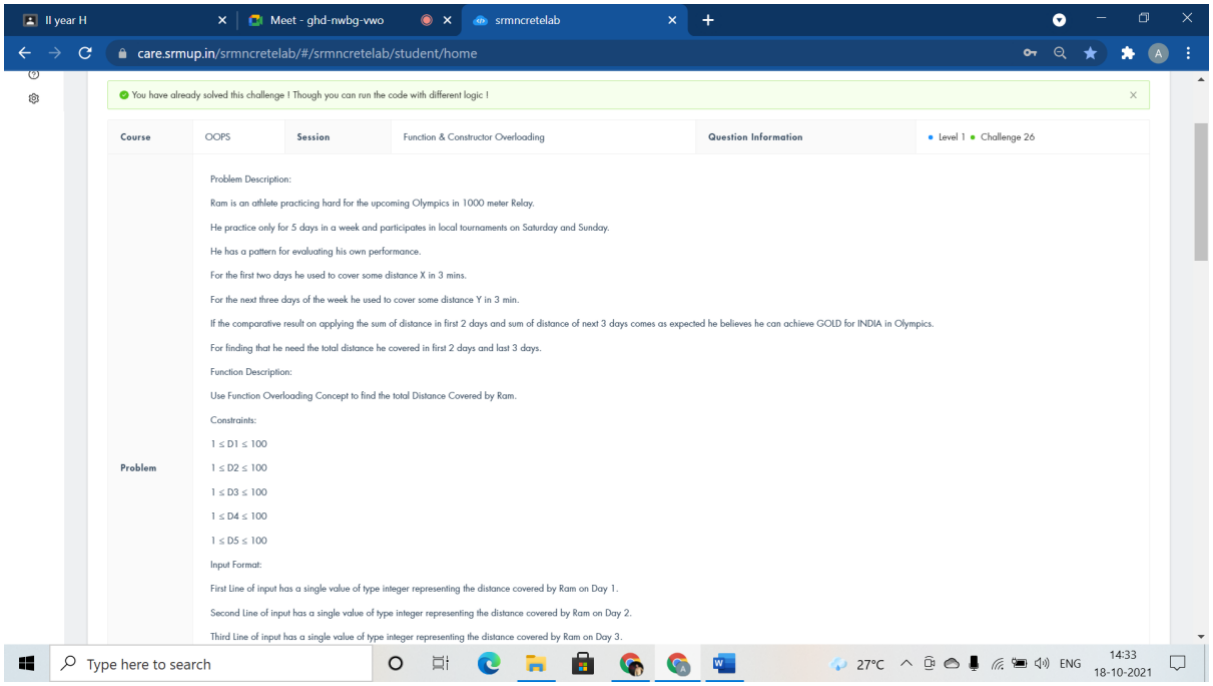

```
#include <iostream>
using namespace std;
class Hospital{
   public:
   void bill(long int mdeicinebill,int days){
      cout<<mdeicinebill*days<<endl;
   }
```

```
    void bill(int roomrent,int days){
        cout<<roomrent*days;
    }
};
int main()
{
    Hospital ob;
    long int mdeicinebill,days;
    int roomrent;
    cin>>mdeicinebill>>days;
    ob.bill(mdeicinebill,days);
    cin>>roomrent>>days;
    ob.bill(roomrent,days);
            return 0;
}
```



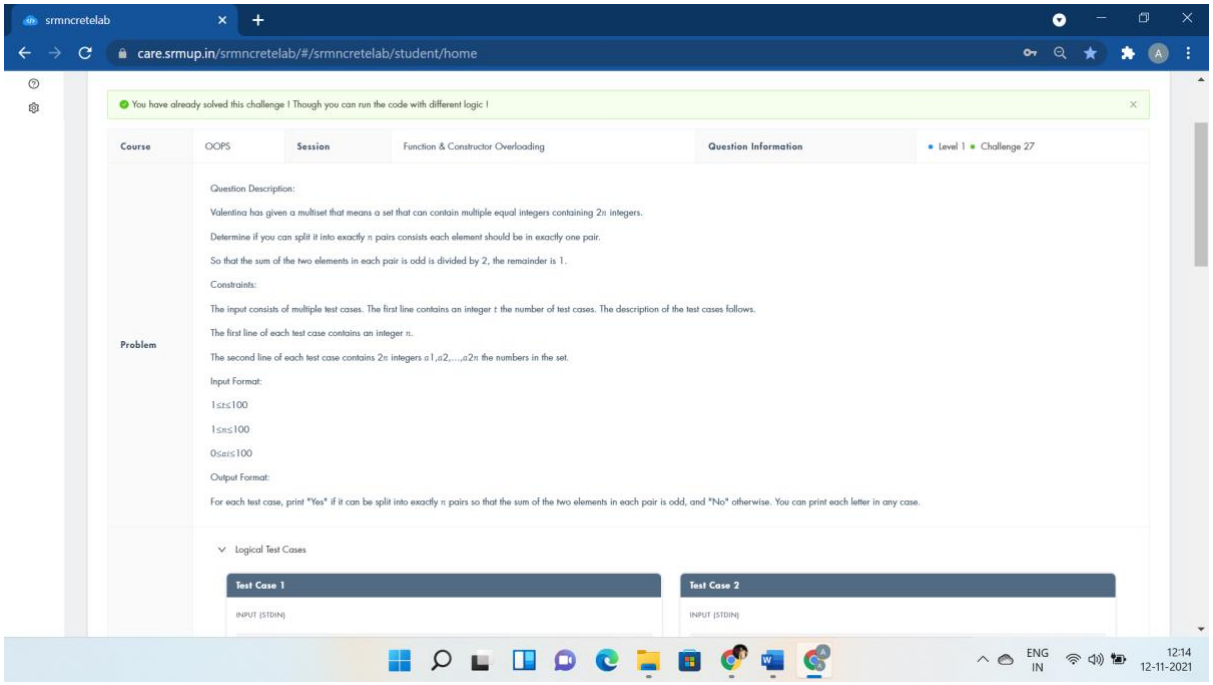

```
#include<bits/stdc++.h>
using namespace std;
int i,T,a,b,c,n;
#define f(i,a,n) for(i=a;i<n;i++)
```

```
class solve{
   public:
   void get(){
      std::cin>>a>>b>>c;
      n=2*abs(a-b);
   }
   void get2(){
      if(c>n||max(a,b)>n)
      cout<<"-1"<<endl;
      else if(c>n/2)
      cout<<c-n/2<<endl;
      else
      cout<<c+n/2<<endl;
   }
};
int main(){
         cin>>T;
         solve p;
         f(i,0,T){
         p.get();
         p.get2();
         }
         return 0;
         cout<<"void pline(int v[],int n) void pline(int v) else if(x>n||x<=0)";
}
```



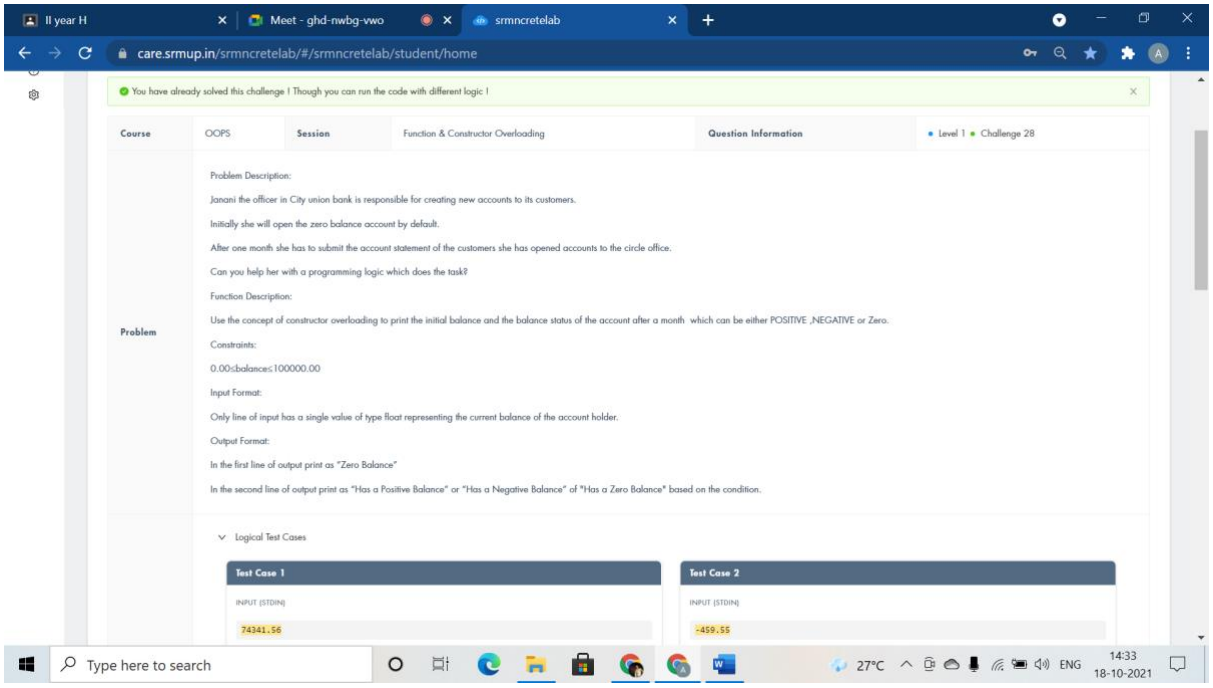

```
#include <iostream>
using namespace std;
class Olympic{
   public:
   void distance(int D1,int D2){
      cout<<D1+D2<<" meters"<<endl;
   }
   void distance(int D3, int D4, int D5){
      cout<<D3+D4+D5<<" meters"<<endl;
   }
};
int main()
{
   Olympic Medal;
   int D1,D2,D3,D4,D5;
   cin>>D1>>D2>>D3>>D4>>D5;
   Medal.distance(D1,D2);
   Medal.distance(D3,D4,D5);
```

```
        return 0;
}
```

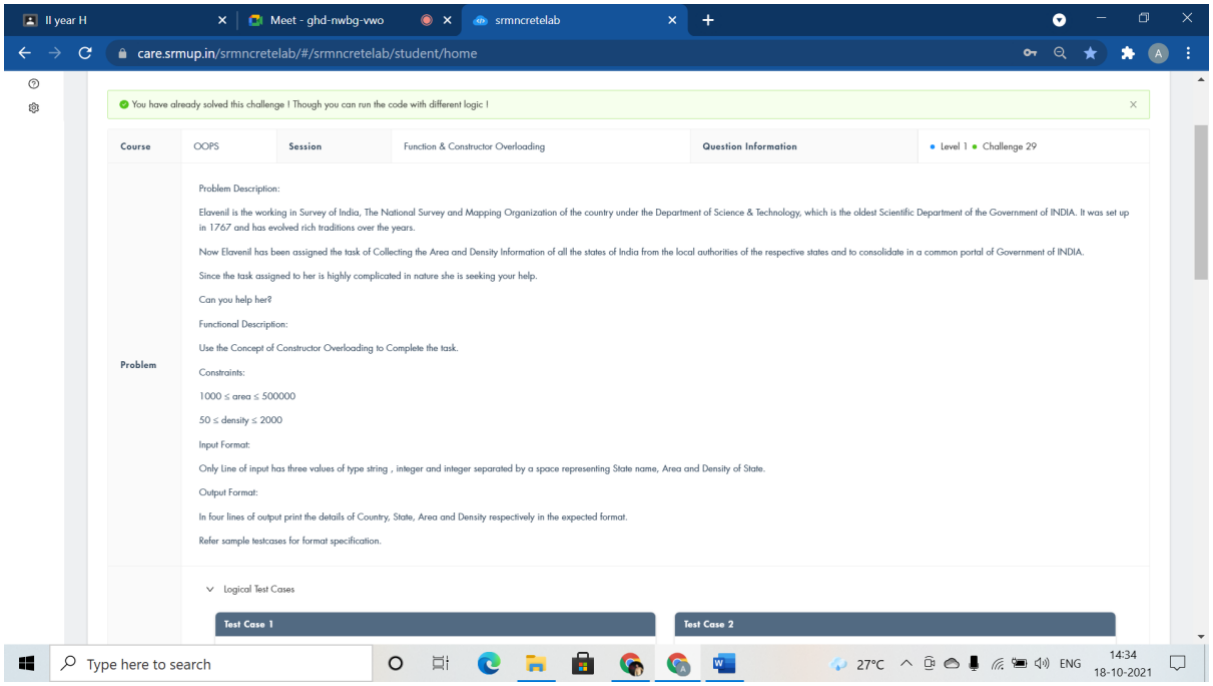



```
#include <iostream>
using namespace std;
int power(int x,int p);
int power(int x,int y,int p);
int main()
{
   int t;
   cin>>t;
   while(t--){
      int n,odd=0;
      cin>>n;
      int z=power(n,odd);
      //cout<<n<<z;
      power(n,z,1);
}
        return 0;
```

```
}
int power(int x,int p){
   int a[2*x];
   for(int i=0;i<2*x;i++){
      cin>>a[i];
      if(a[i]%2==1)
      p++;
   }
   return p;
}
int power(int x,int y,int p){
   if(x==y)
   cout<<"Yes"<<endl;
   else
   cout<<"No"<<endl;
   return 1;
}
```

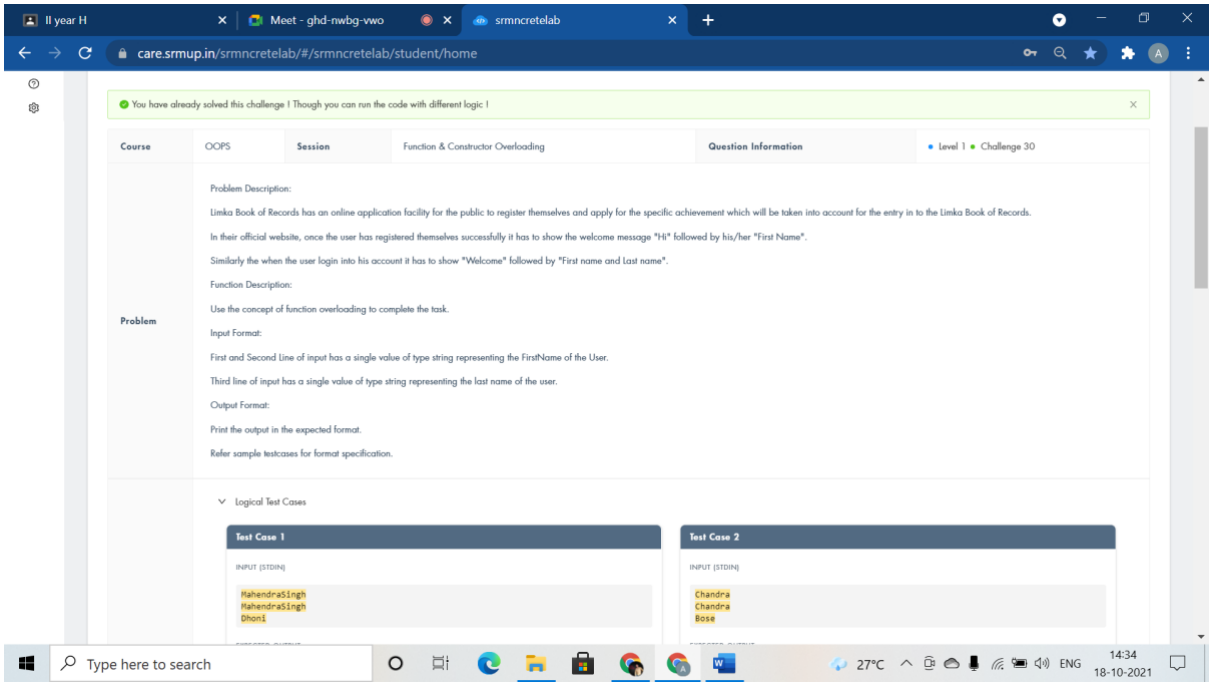


```
#include <iostream>
using namespace std;
class AccBalance{
   public:
   AccBalance(){cout<<"Zero Balance"<<endl;}
   AccBalance(int balance){
      if(balance<0)
      cout<<"Has a Negative Balance";
      else if(balance==0)
      cout<<"Has a Zero Balance";
      else
      cout<<"Has a Positive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




```
#include <iostream>
using namespace std;
class Country{
   public:
   Country(){cout<<"Country:INDIA"<<endl;}
   Country(char statename[100],int area,int density)
   {
      cout<<"State:"<<statename<<endl<<"Area:"<<area<<endl<<"Density:"<<density<<endl;
   }
};
int main()
{
   Country country;
   char statename[100];
   int area,density;
   cin>>statename>>area>>density;
   Country statesofindia(statename,area,density);
         return 0;
```

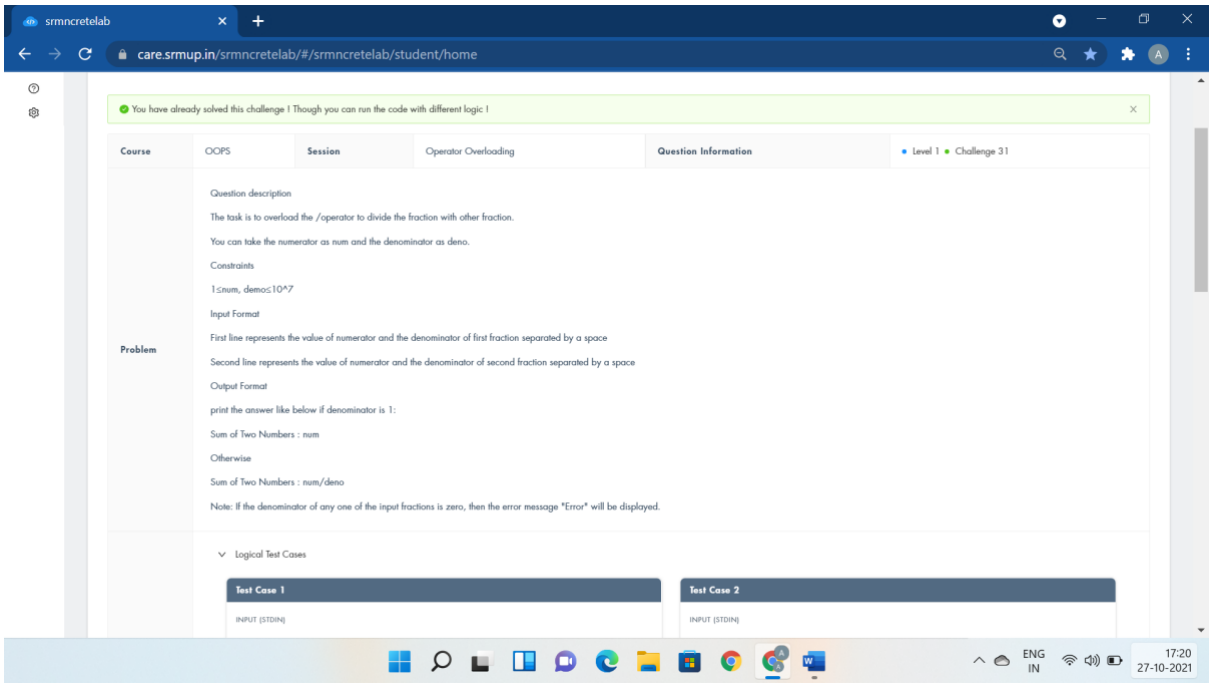
}




```
#include <iostream>
using namespace std;
class Welcomemsg{
   public:
   void msg(string fname){
      cout<<"Hi "<<fname<<endl;
   }
   void msg(string fname,string lname){
      cout<<"Welcome "<<fname<<" "<<lname;
   }
};
int main()
{
   Welcomemsg ob;
   string fname,lname;
```

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

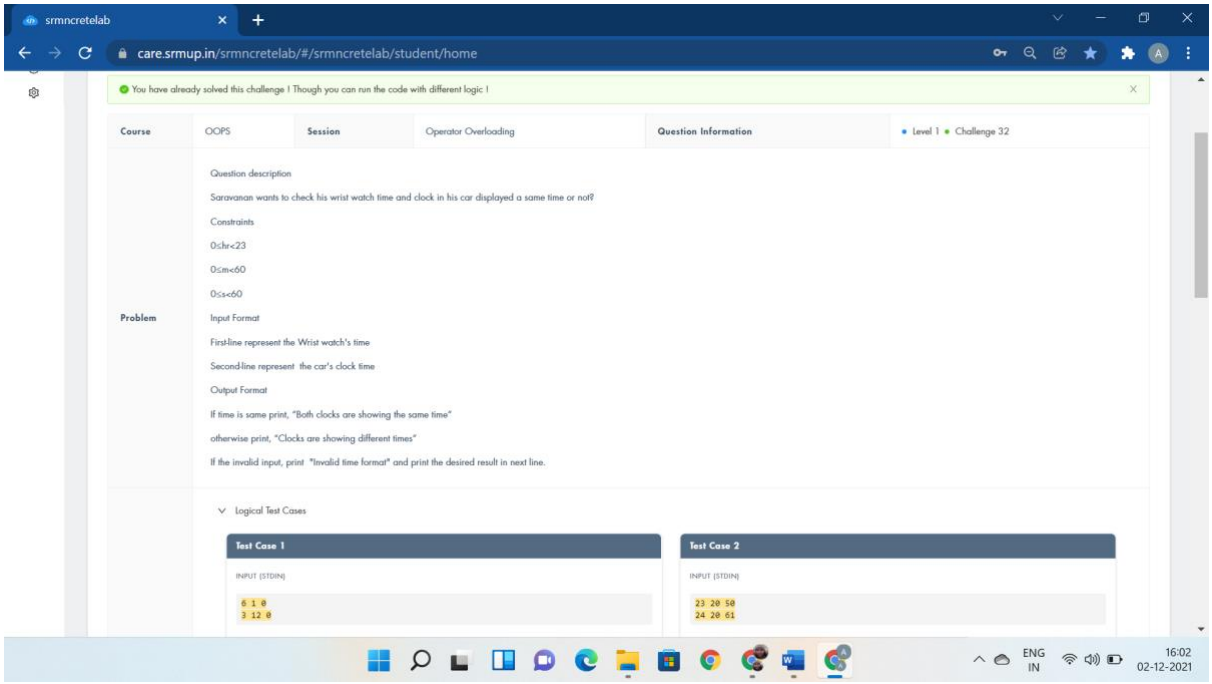
Operator Overloading:-




```
#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){
```

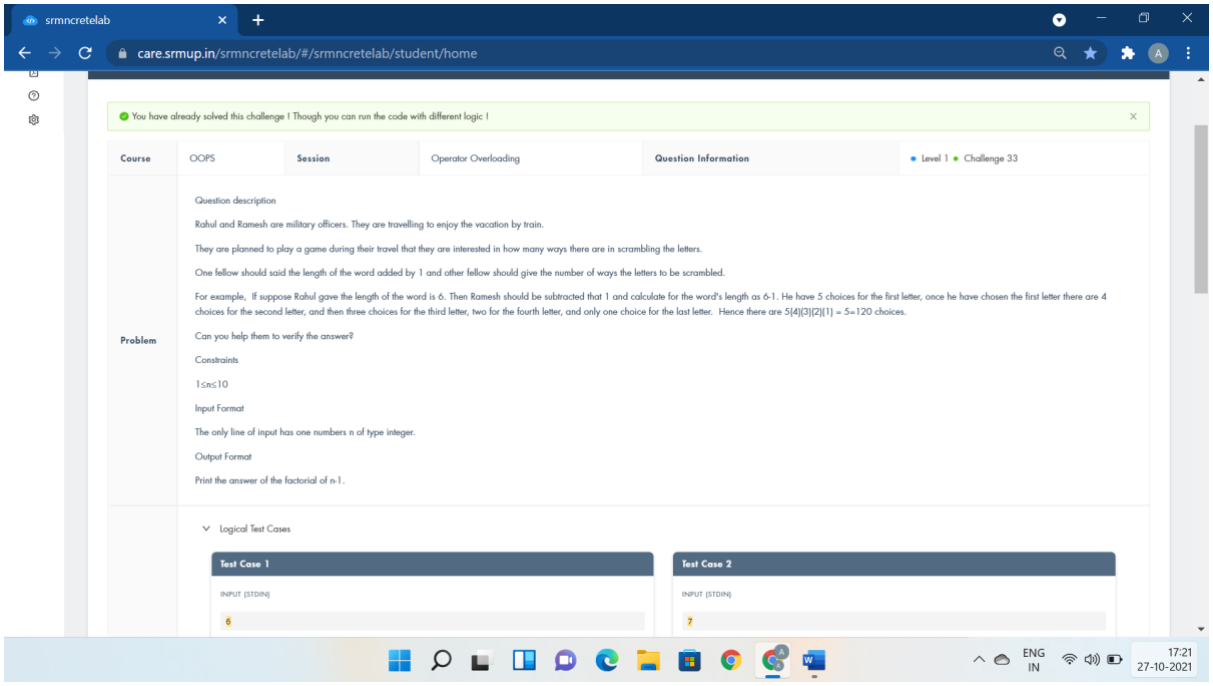
```
    Fraction res;
    res.num=num * obj.den;
    res.den=den * obj.num;
    return res;
  }
  void display1(){
    cout<<num/den;
  }
  void display2(){
    cout<<num<<"/"<<den;
  }
  void display3(){
    cout<<"Error";
  }
};
int main()
{
  int a,b,c,d;
  cin>>a>>b;
  cin>>c>>d;
  Fraction ob1(a,b), 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;
}
```



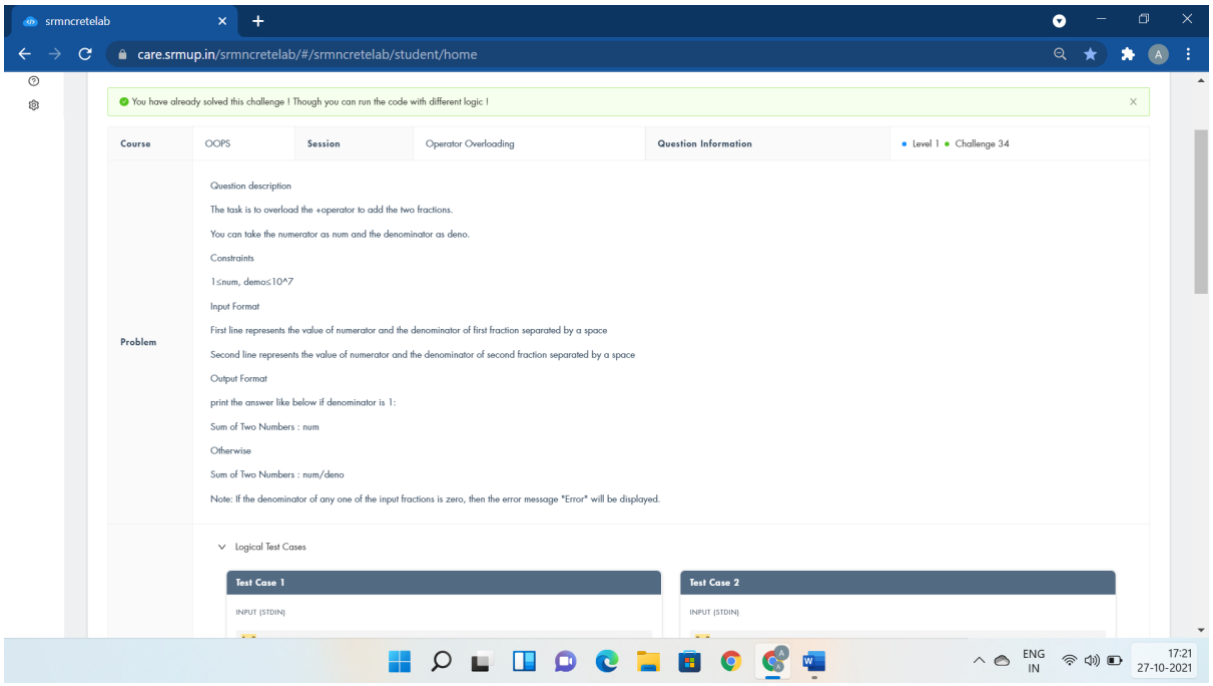

```
#include <iostream>
using namespace std;
class Time
{
  int h,m,s;
  public:
  Time()
  {
    cin>>h>>m>>s;
  }
  void check()
  {
    if(h>23 || m>59 ||s>59 )
```

```
        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;
}
```



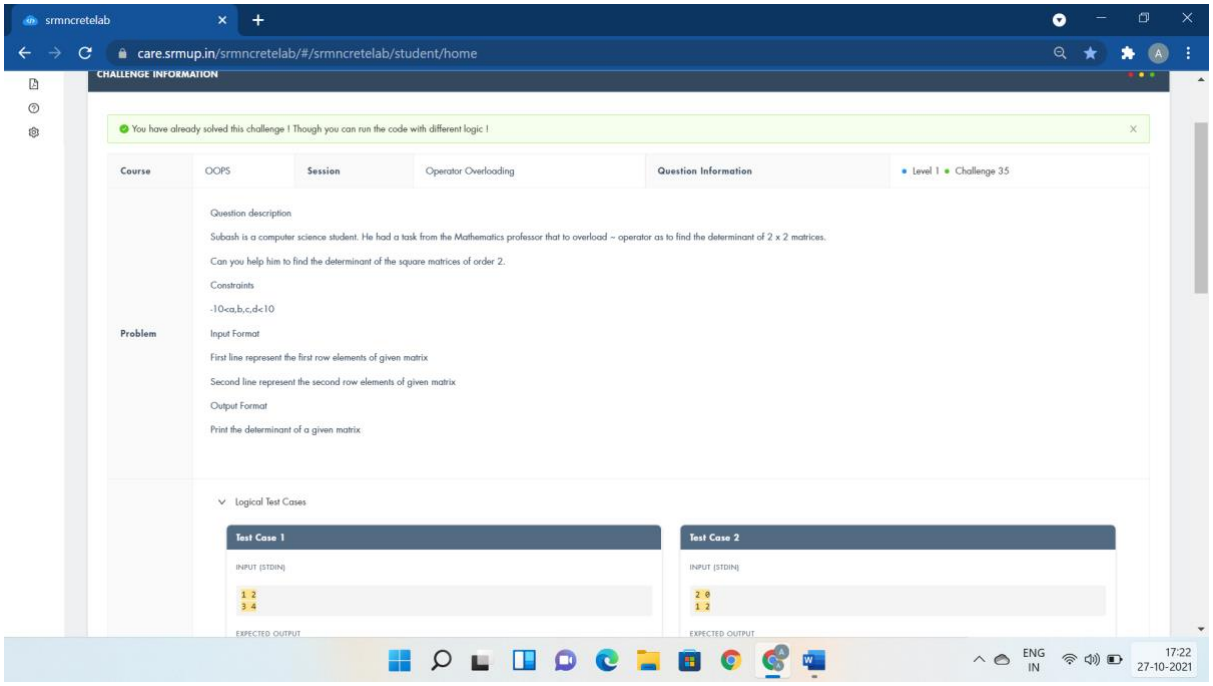

```
#include <iostream>
using namespace std;
class Scrum{
   public:
   int n;
   Scrum(int h)
   {
      n=h;
   }
   Scrum operator -- (int){
      Scrum T(int h);
      --n;
      return 1;
   }
   void display(){
      int res=1;
      for(int i=1;i<=n;i++){
         res=res*i;
      }
      cout<<res;
   }
};
```

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

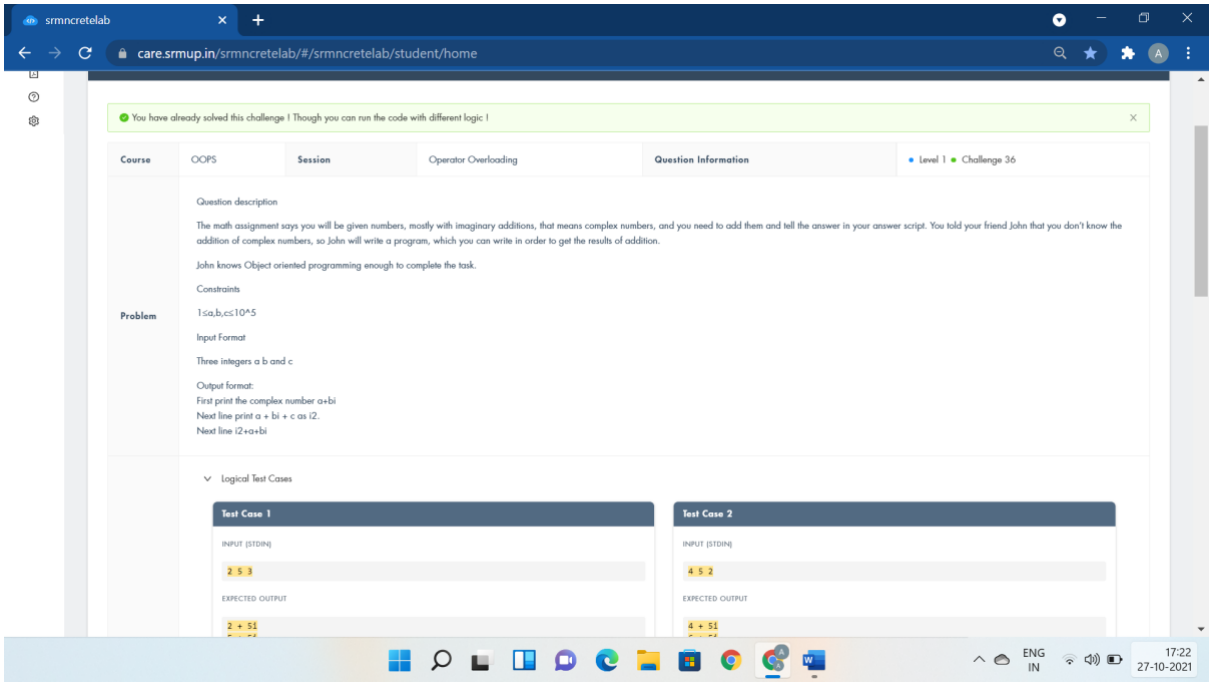



```
#include<iostream>
using namespace std;
class Fraction
{
   public:
   int num,den;

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;
}
```

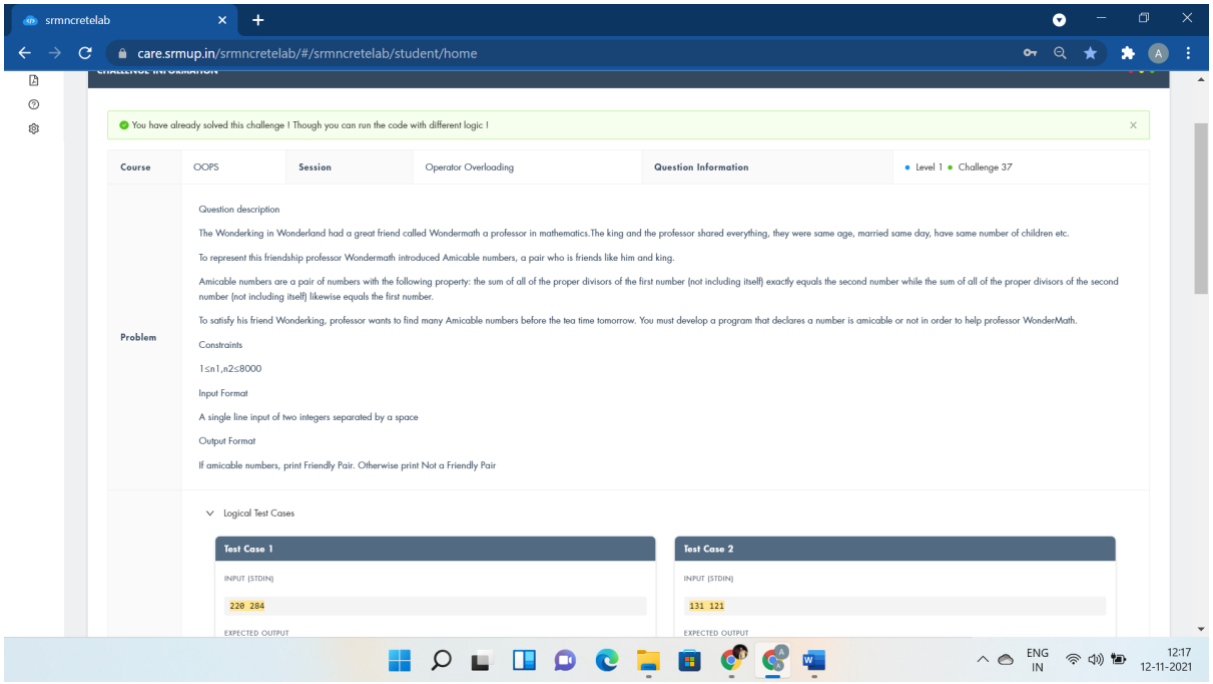



```
#include <iostream>
using namespace std;
class matrix{
   public:
   int operator ~(){
      int a,b,c,d;
      cin>>a>>b>>c>>d;
      return a*d-b*c;
   }
};
int main()
{
   matrix t;
   cout<<~t;
            return 0;
}
```



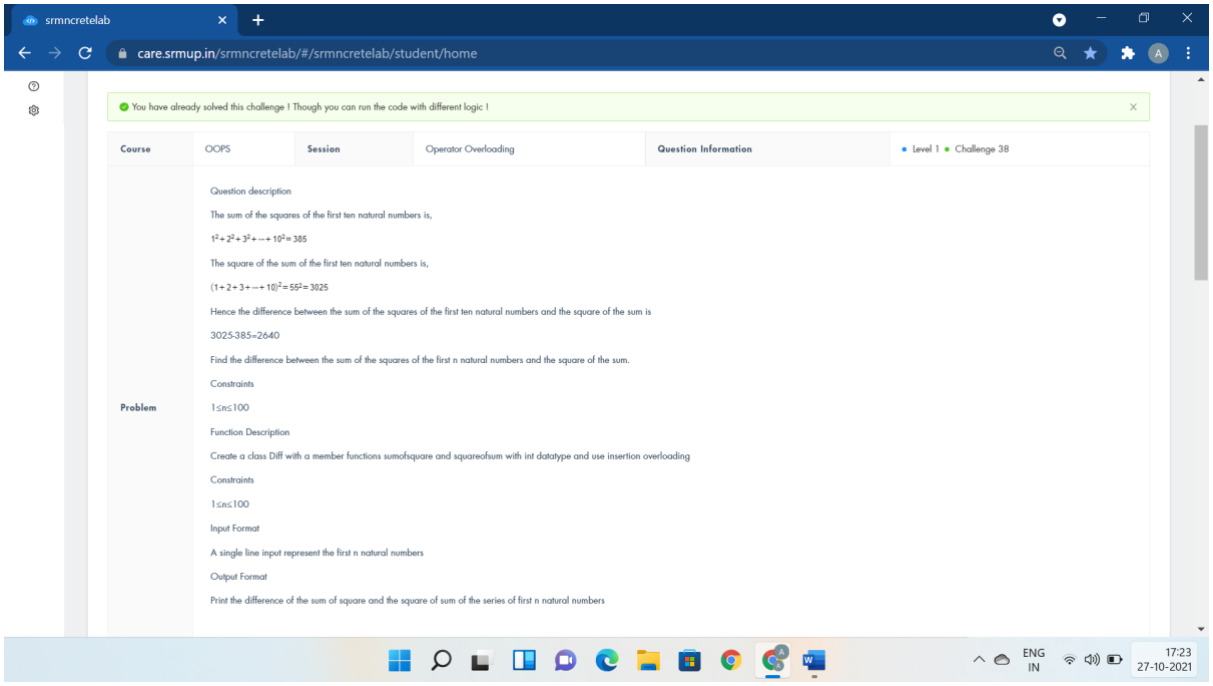

```
#include<iostream>
using namespace std;

class Complex {
private:
        int real, imag;
public:
        Complex(int r = 0, int i =0) {real = r; imag = i;}
         Complex operator+(int a) {
           Complex res;
           res.real = real + a;
           res.imag = imag;
           return res;
        }
        Complex operator+(Complex obj) {
                Complex res;
                res.real = real + obj.real;
                res.imag = imag + obj.imag;
                return res;
        }
        void print() { cout << real << " + " << imag <<"i"<< endl; }
};
```

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



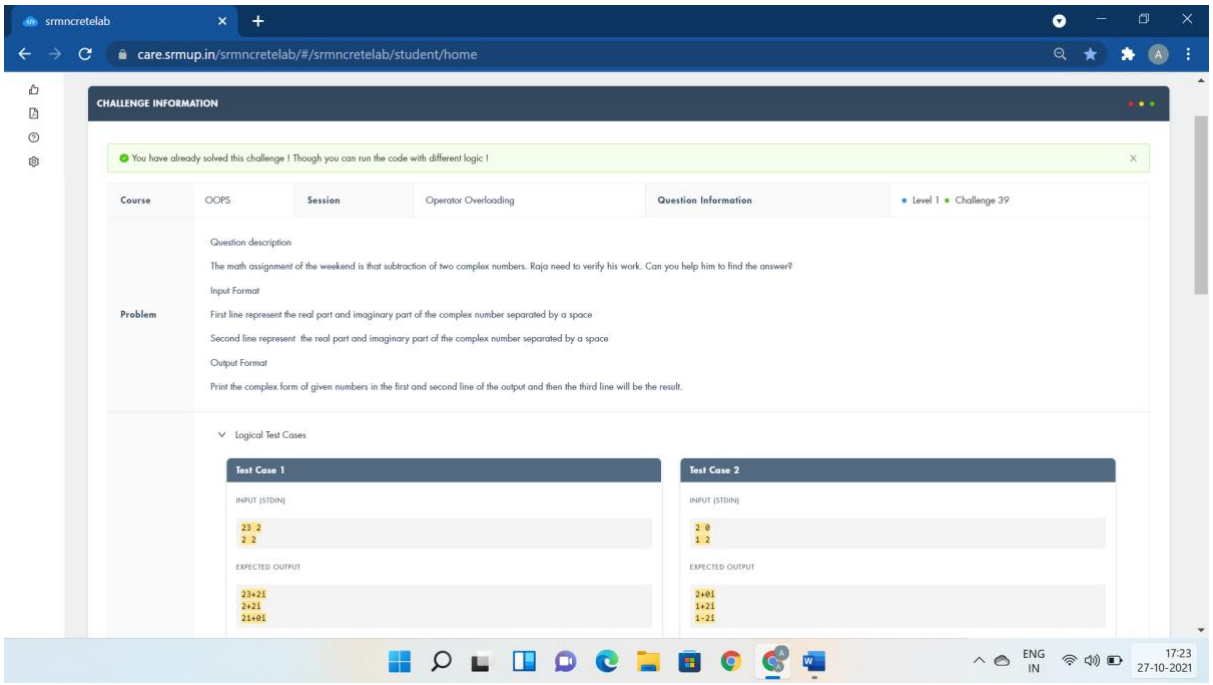

```
#include<iostream>
using namespace std;
class compare{
   public:
   int first,sum1=0;
   compare(int x){
      first=x;
   }
   void f(){
      //first1=first;
```

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


  void operator ==(compare t2){
    if(first==t2.sum1 && t2.first==sum1)
    cout<<"Friendly Pair";
    else
    cout<<"Not a Friendly Pair";
  }
};
  //main program
  int main()
  {
    int first,second;
    //user input
    cin>>first;
    //user input
    cin>>second;
    compare t1(first),t2(second);
    t1.f();
    t2.f();
    t1==t2;
    return 0;
  }
```



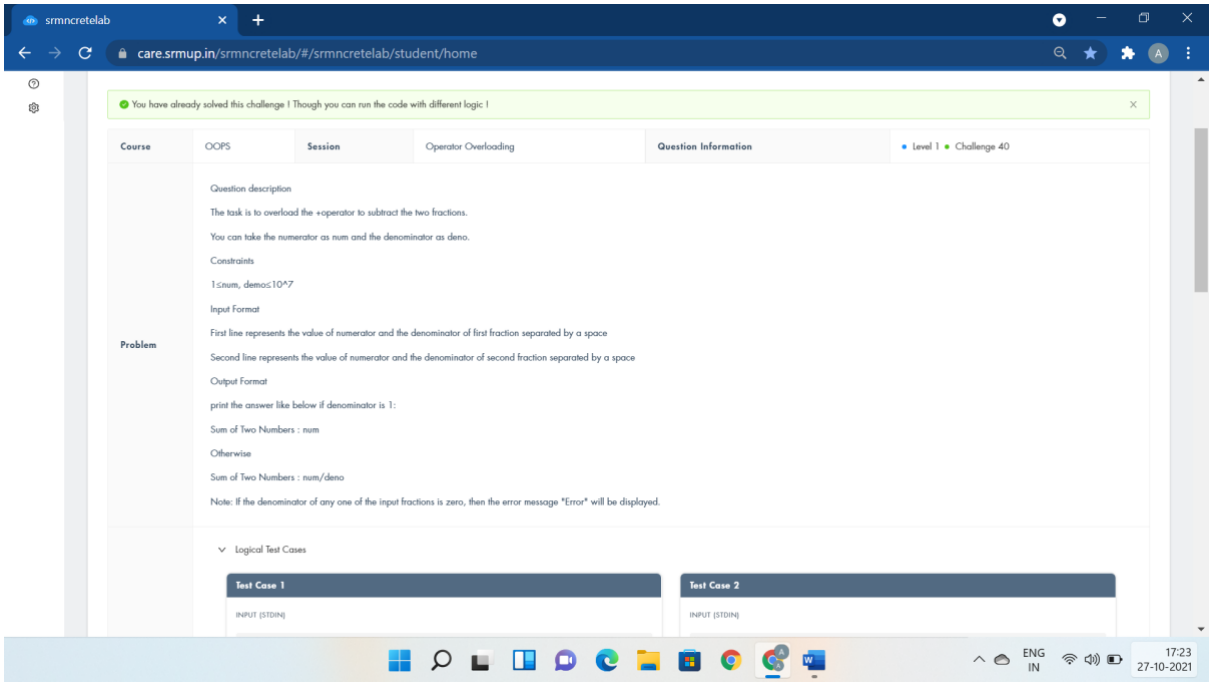

```
#include <iostream>
using namespace std;
class Diff{
   public:
   int n;
   void getdata(){
      cin>>n;
   }
   int sumofsquare();
   int sumofnumsq(){
      return n*(n+1)*(2*n+1)/6;
   }
};
int Diff :: sumofsquare(){
   return n*n*(n+1)*(n+1)/4;
}

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

```
    n.getdata();
    //int sq=n*n*(n+1)*(n+1)/4;
    //int sq2=n*(n+1)*(2*n+1)/6;
    cout<<n.sumofsquare()-n.sumofnumsq();
            return 0;
}
```




```
#include <iostream>
using namespace std;
class complex
{
   private:
     float real;
     float imag;
   public:
      complex() {cin>>real>>imag;}
      complex operator-(complex ob)
      {
         complex t;
         t.real = real - ob.real;
         t.imag = imag - ob.imag;
         return t;
```

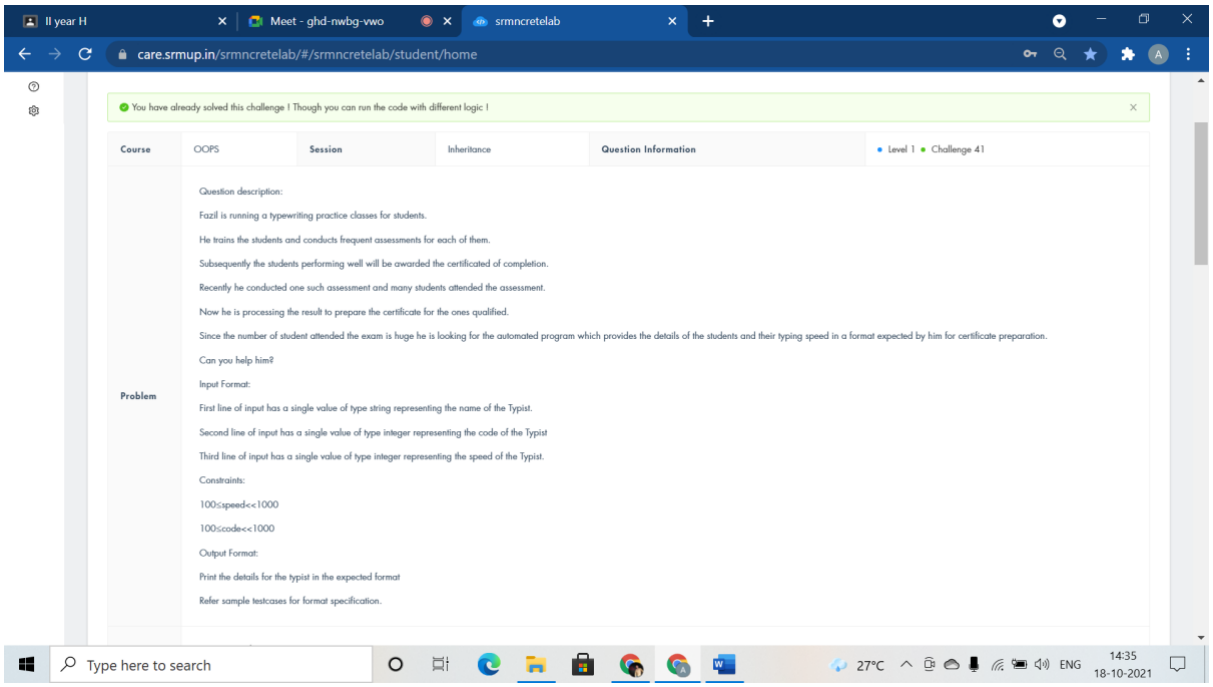
```
    }

    void output()
    {
        if(imag < 0)
            cout<< real << imag << "i"<<endl;
        else
            cout<< real << "+" << imag << "i"<<endl;
    }
};
int main()
{
   complex c1, c2;
   c1.output();
   c2.output();
   (c1 - c2).output();
   return 0;
}
```




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

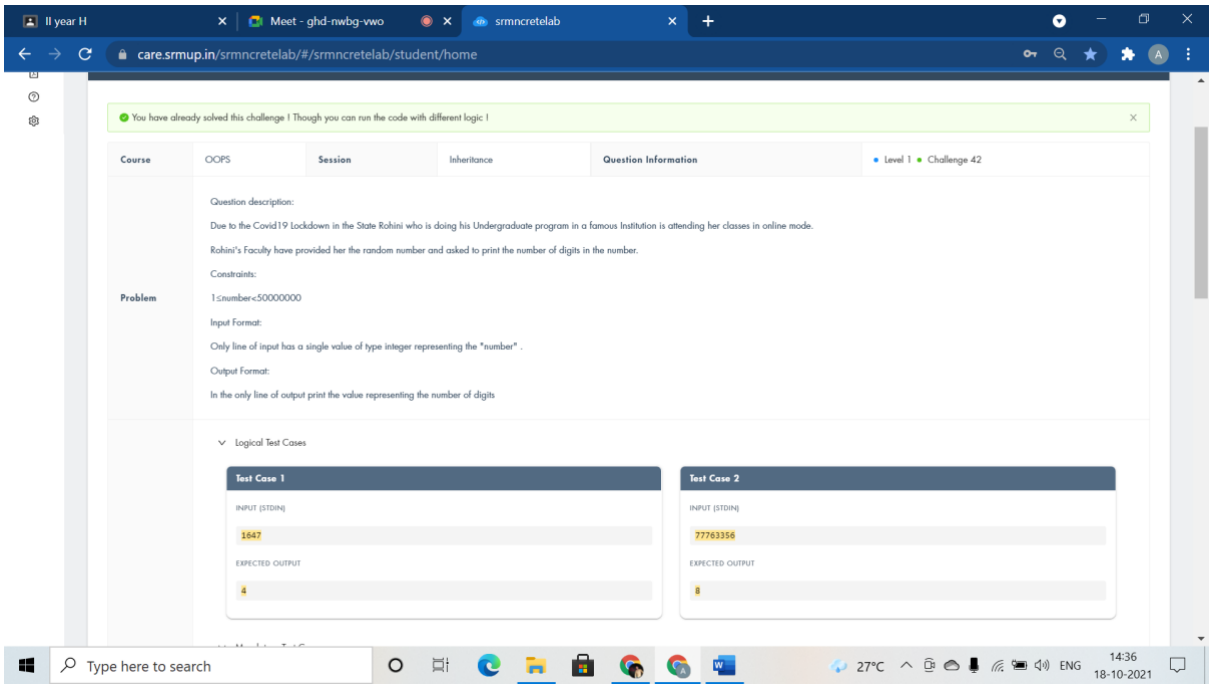
```
{
   public:
   int num,den;

Fraction()
{
   num=0;
   den=0;
}
void getinput()
{
   cin>>num>>den;
}
Fraction operator -(Fraction obj)
{
   Fraction temp;
   temp.num=(num*obj.den)-(den*obj.num);
   temp.den=den*+obj.den;
   return temp;
}
};
int main()
{
   Fraction f1,f2,add;
   f1.getinput();
   f2.getinput();
   add=f1-f2;
   if(add.den==0)
   cout<<"Error";
   else if(add.num%add.den == 0)
   cout<<add.num/add.den;
   else
   cout<<add.num<<"/"<<add.den;
   return 0;
}
```

Inheritance:-



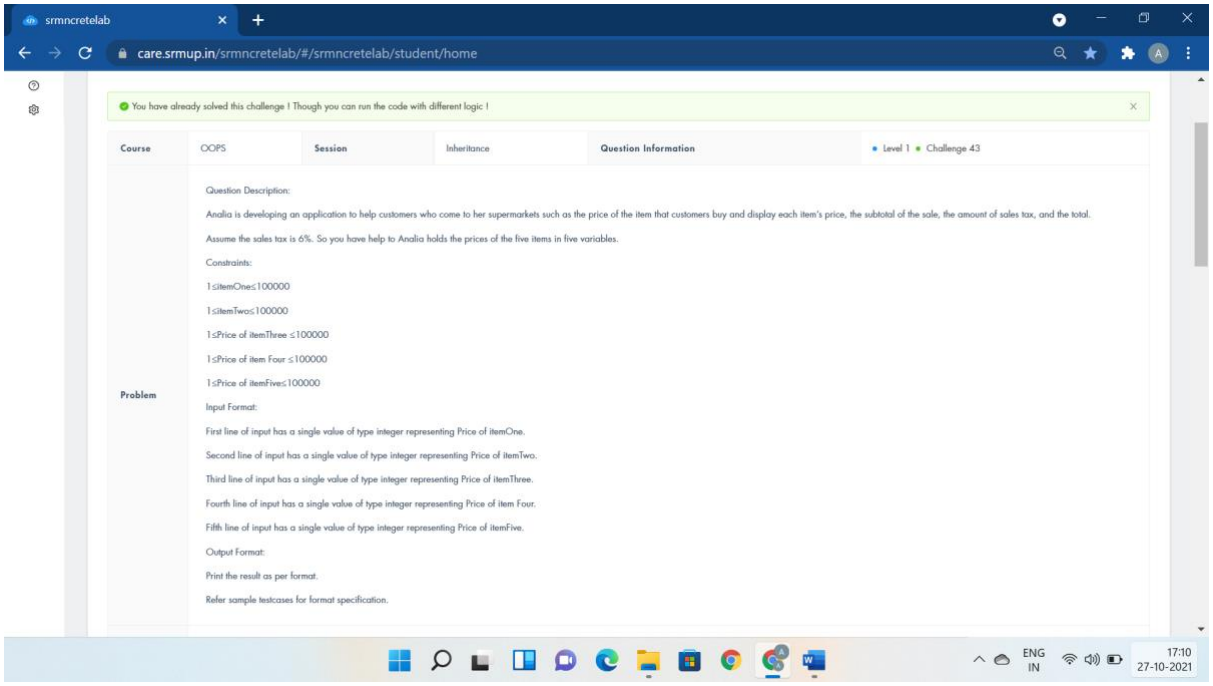

```
#include <iostream>
using namespace std;
class staff{
   public:
   int code,speed;
   string name;
   void getdata();
   void display();
};
void staff::getdata(){
   cin>>name>>code>>speed;
}
void staff::display(){
   cout<<"Name:"<<name<<endl<<"Code:"<<code<<endl<<"Speed"<<speed;
}
class typist: 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;
}
```



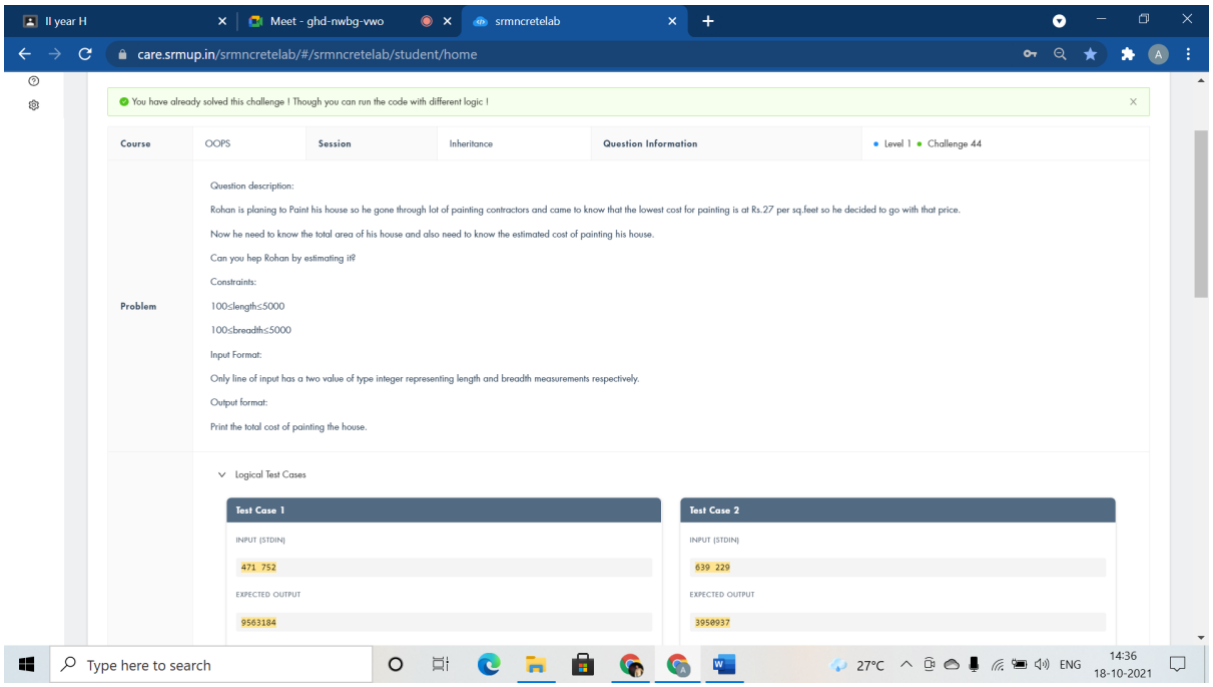

```
#include <iostream>
```

```
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;
}
```



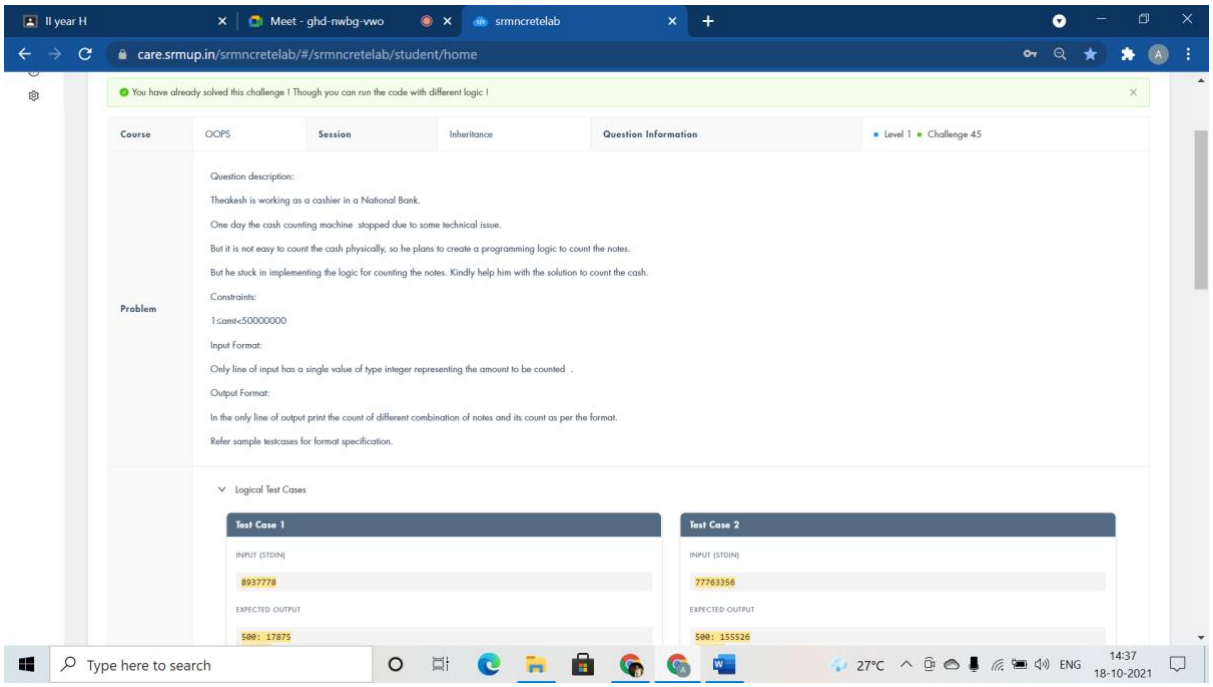

```
#include <iostream>
using namespace std;
class market{
   public:
   float i1,i2,i3,i4,i5;
   float Subtotal,tax;
   void items(){
      cin>>i1>>i2>>i3>>i4>>i5;
   }
   void buy(){
      Subtotal=(i1+i2+i3+i4+i5);
      cout<<"Subtotal=$"<<Subtotal<<endl;
      tax=0.06*i1+0.06*i2+0.06*i3+0.06*i4+0.06*i5;
      cout<<"Tax=$"<<tax<<endl;
      cout<<"Total=$"<<Subtotal+tax;
   }
};
class customer:public market{
```

```
};
int main()
{
   customer c;
   c.items();
   c.buy();
}
```



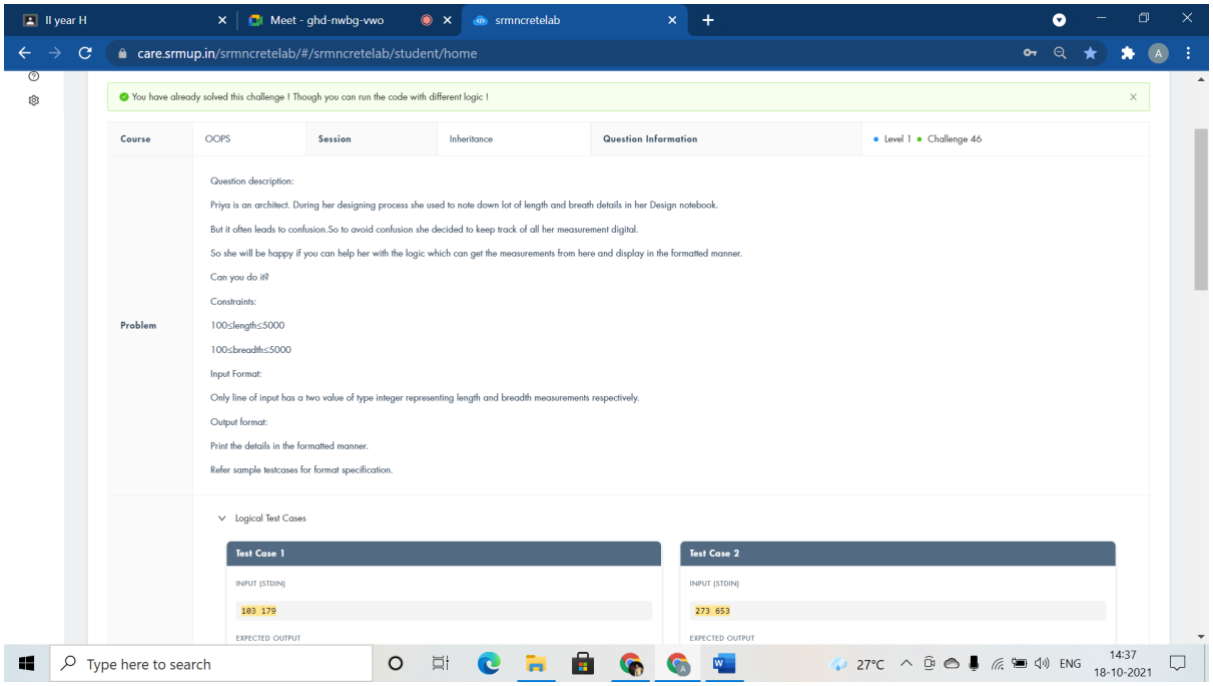

```
#include <iostream>
using namespace std;
class ReceiveMesurement{
   public:
   int l,b;
   void painingarea(){
   cin>>l>>b;
      cout<<l*b*27;
   }
```

```
};
class CalculateArea : public ReceiveMesurement{
};
int main()
{
   CalculateArea mt;
   mt.painingarea();
          return 0;
}
```



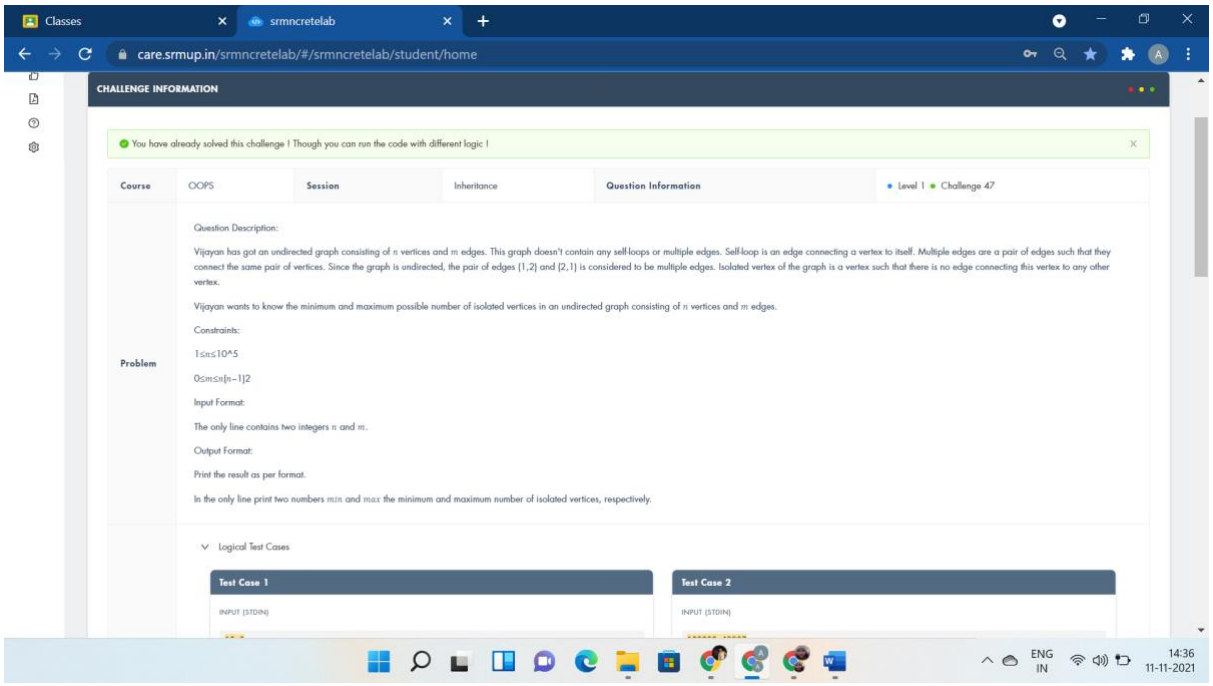

```
#include <iostream>
using namespace std;
class Bank{
   public:
   int n;
   void get(){
      cin>>n;
   }
   void display(){
      cout<<"500: "<<n/500<<endl;
```

```
        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;
}
```



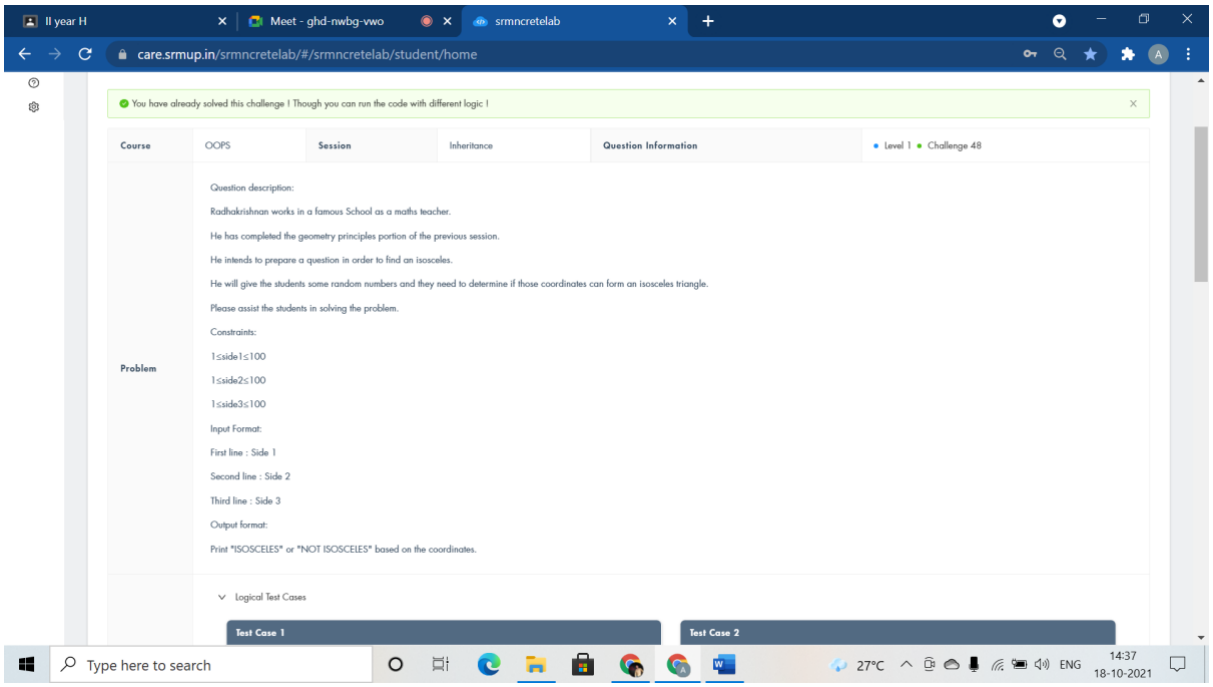

```
#include <iostream>
using namespace std;
class ReceiveMesurement{
   public:
   int l,b;
   void display(){
      cin>>l>>b;
      cout<<"Length:"<<l<<" metres"<<endl;
      cout<<"Breadth:"<<b<<" metres";
   }
};
class FormatMesurement : public ReceiveMesurement{
};
int main()
{
   FormatMesurement mt;
   mt.display();
         return 0;
```

}



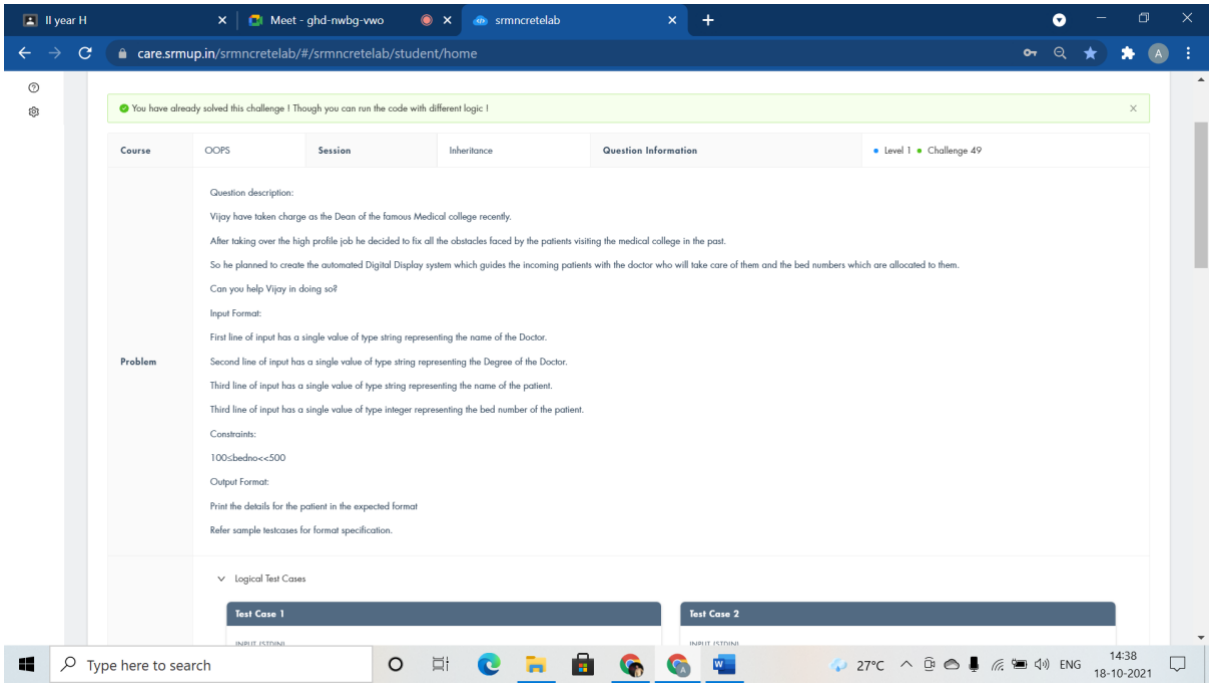

```
#include <bits/stdc++.h>
using namespace std;
class graph{
   public:
   void edge(){
   }
};
class pairs:public graph{
   public:
   long long int n,m,k=0;
   void vertex(){
      cin>>n>>m;
      cout<<max(0ll,n-2*m)<<" ";
      while(k*(k-1)/2<m) k++;
      cout<<n-k<<endl;
   }
};
```

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



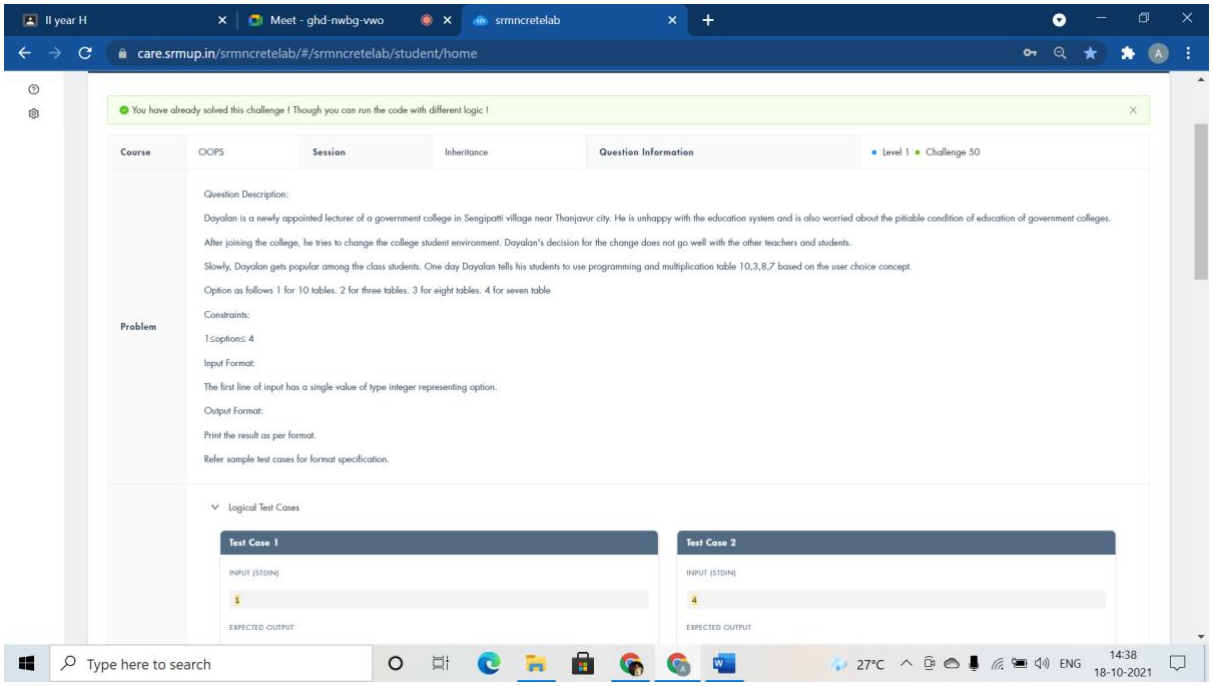

```
#include <iostream>
using namespace std;
class triangle{
   public:
   int a,b,c;
   void read(){
      cin>>a>>b>>c;
   }
   void check(){
      if(a==b || b==c || a==c)
```

```
        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:
```

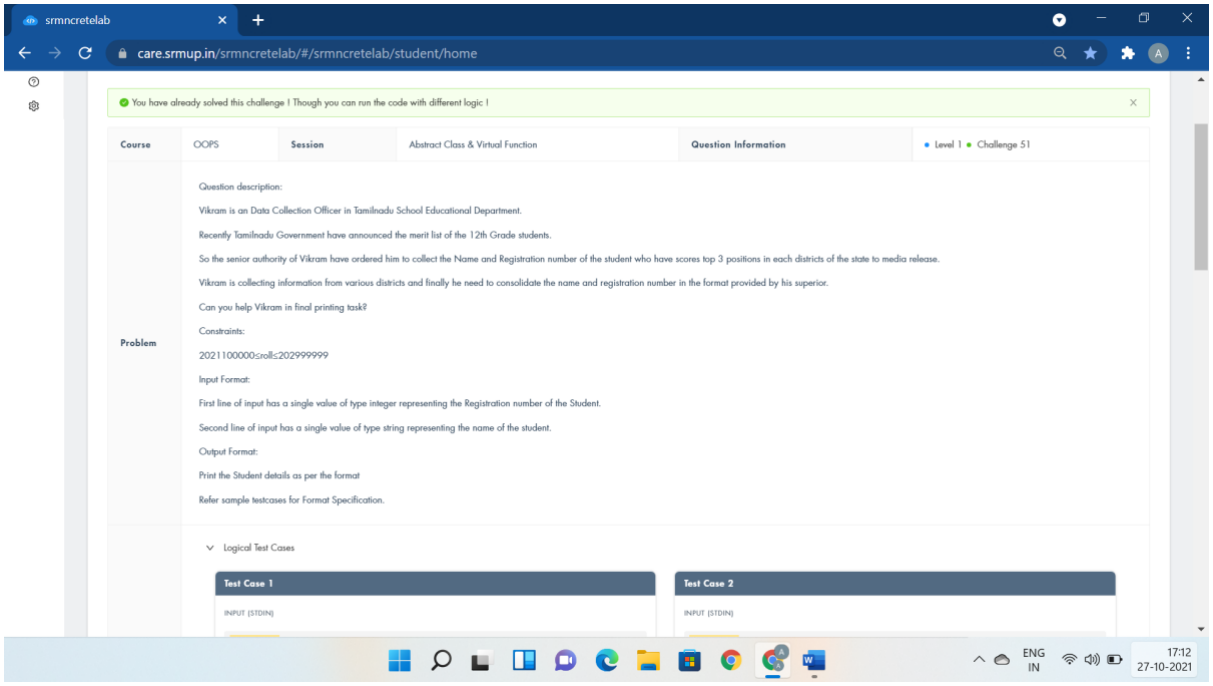
```
    string name,degree,pname;
    int no;
    void getedu(){
        cin>>name>>degree>>pname;
    }
    void getdata(){
        cin>>no;
    }
    void dispedu(){
        cout<<"Doctor Name:"<<name<<endl<<"Doctorate Degree:"<<degree<<endl<<"Patient
Name:"<<pname<<endl;
    }
    void dispdata(){
        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;
```

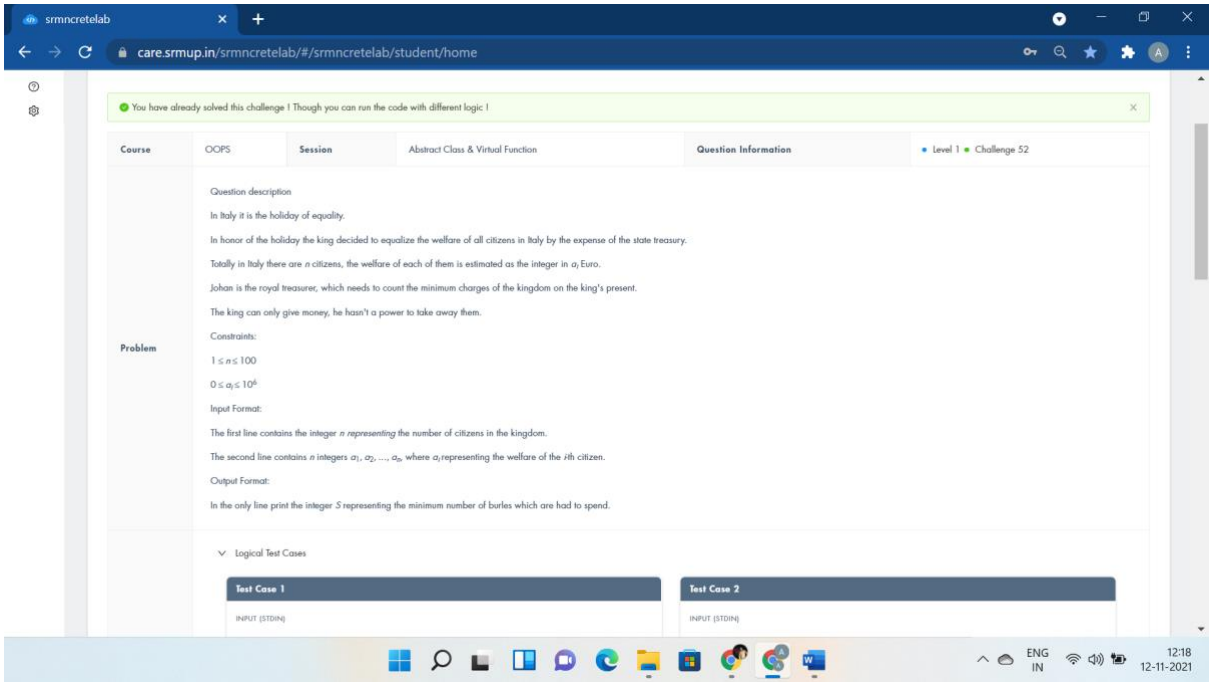
```
    }
    void tentable(){
        for(int i=1;i<=10;i++)
        cout<<num<<"*"<<i<<"="<<num*i<<endl;
    }
};
class ten:public teacher{
};
class three:public teacher{
};
class eight:public teacher{
};
class seven:public teacher{
};

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:-



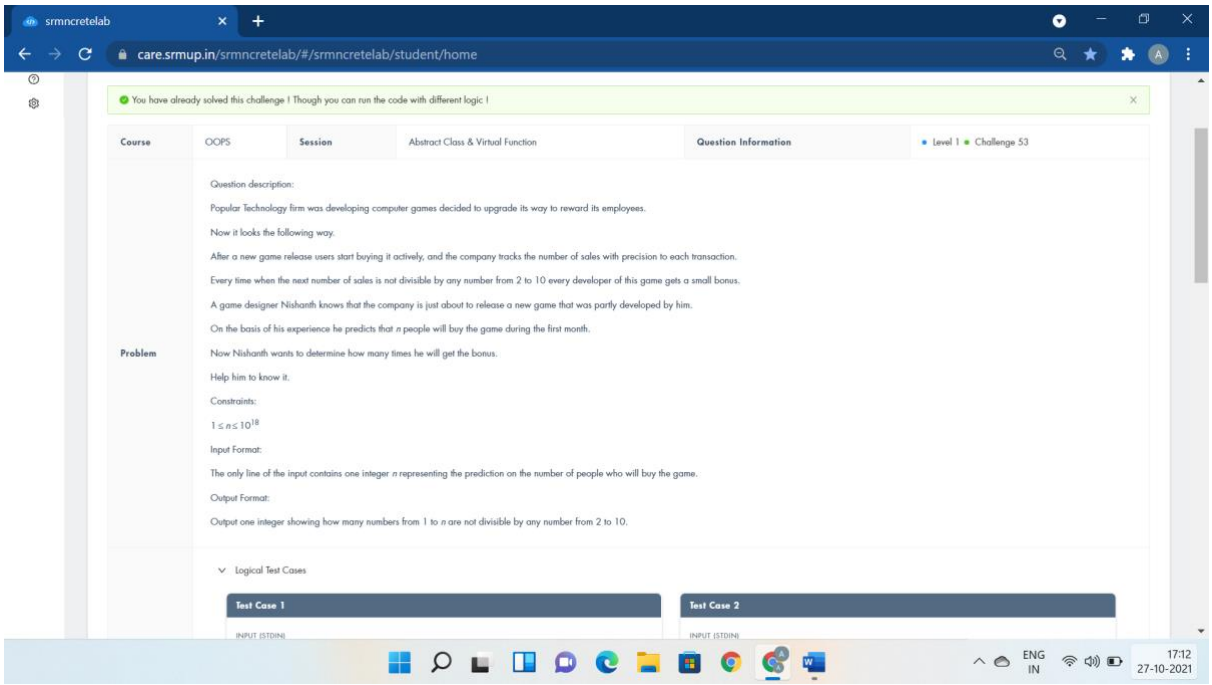

```
#include <iostream>
using namespace std;
class School{
   public:
   int roll;
   string name;
   virtual void getdata(){};
   virtual void display(){};
};
class District : public School{
   void getdata();
   void display();
};
void District :: getdata(){
      cin>>roll>>name;
   }
void District :: display(){
      cout<<"Student Name is: "<<name<<endl<<"Student Roll no is: "<<roll;
   }
```

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



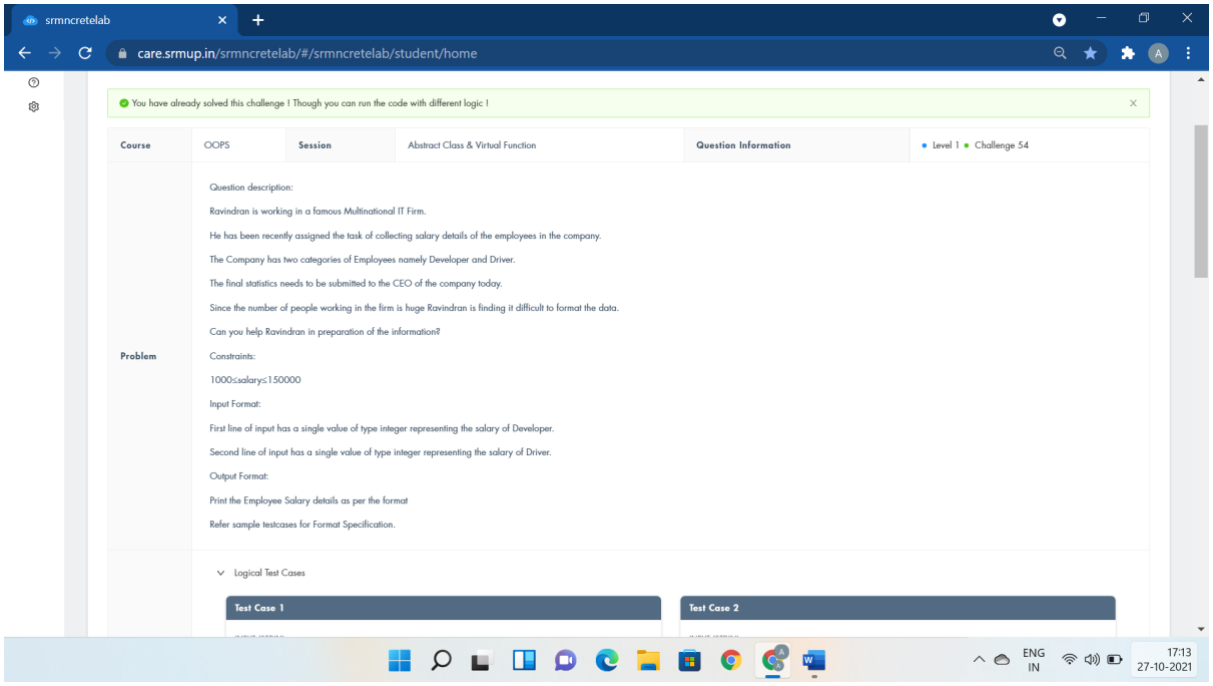

```
#include <bits/stdc++.h>
using namespace std;
int a,b,c,d,i;
class Holiday{
   public:virtual void Expenses()=0;
};
class Citizen:public Holiday{
   public:
   void Expenses(){
```

```
        cin>>c;
        for (i=0; i<c; i++){
            cin>>a;
            if (d<a) d=a;
            b=b+a;
        }
        cout<<d*c-b;
    }
};
int main (){
    Citizen obj;
    obj.Expenses();
    return 0;
}
```



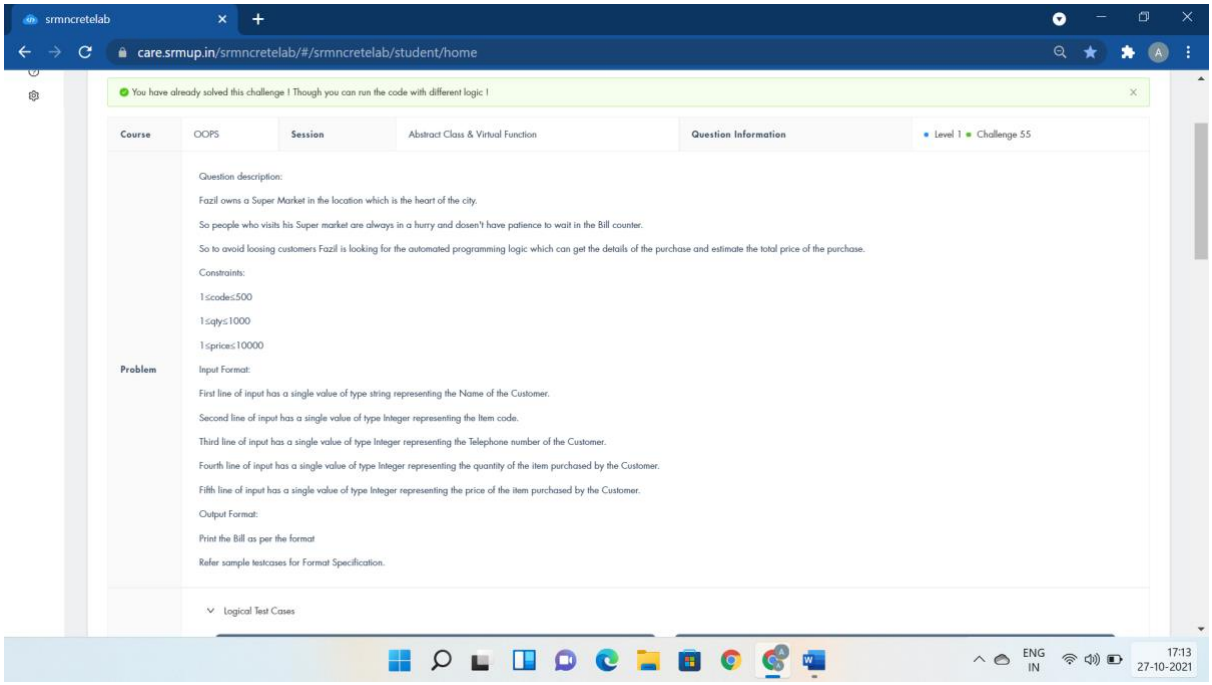

```
#include <bits/stdc++.h>
```

```
using namespace std;
class Employees{
   public:virtual void BuyingGame()=0;
};
class Reward:public Employees{
   public:
   int n;
   void BuyingGame(){
      cin>>n;
      cout<<n - n / 2 - n / 3 - n / 5 - n / 7
                  + n / 6 + n / 10 + n / 14 + n / 15 + n / 21 + n / 35
                  - n / 30 - n / 42 - n / 70 - n / 105 + n / 210;

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



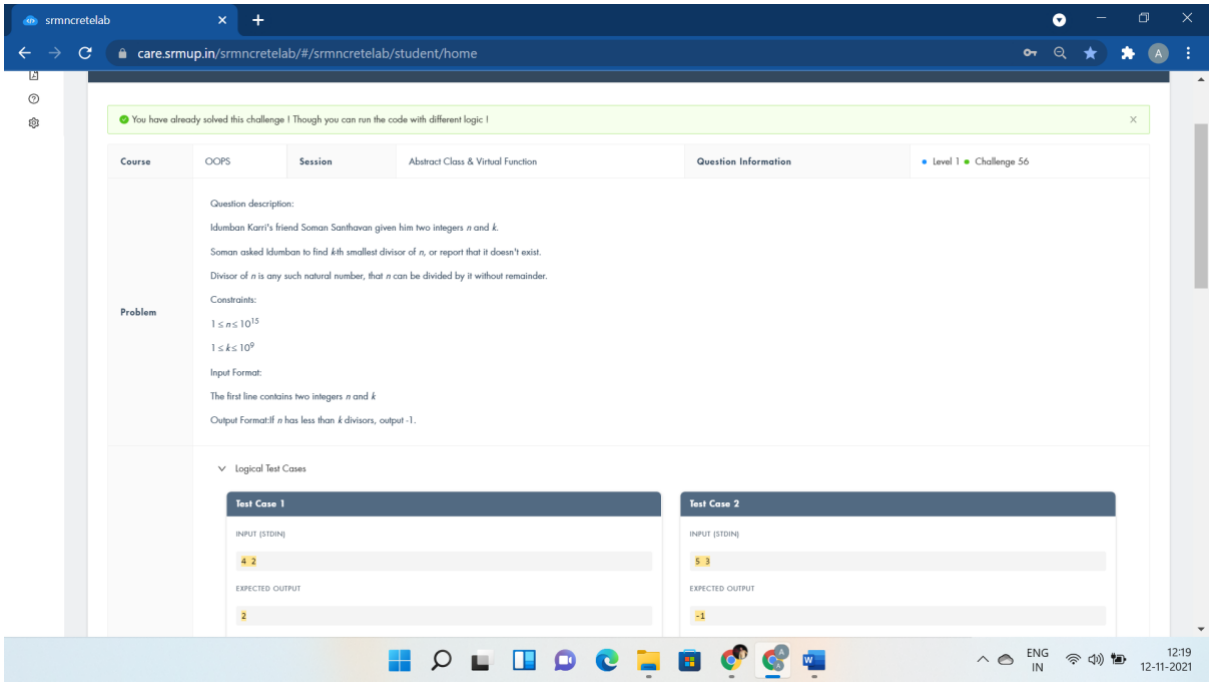

```
#include <iostream>
using namespace std;
class Employee{
   public:
   int s1,s2;
};
class Developer : public Employee{
   public:
   void getSalary(){
      cin>>s1;
      cout<<"Salary of Developer:"<<s1<<endl;
   }
};
class Driver : public Employee{
   public:
   void getSalary(){
      cin>>s2;
      cout<<"Salary of Driver:"<<s2<<endl;
   }
```

```
};
int main()
{
   Developer d1;
   Driver d2;
   d1.getSalary();
   d2.getSalary();
         return 0;
}
```



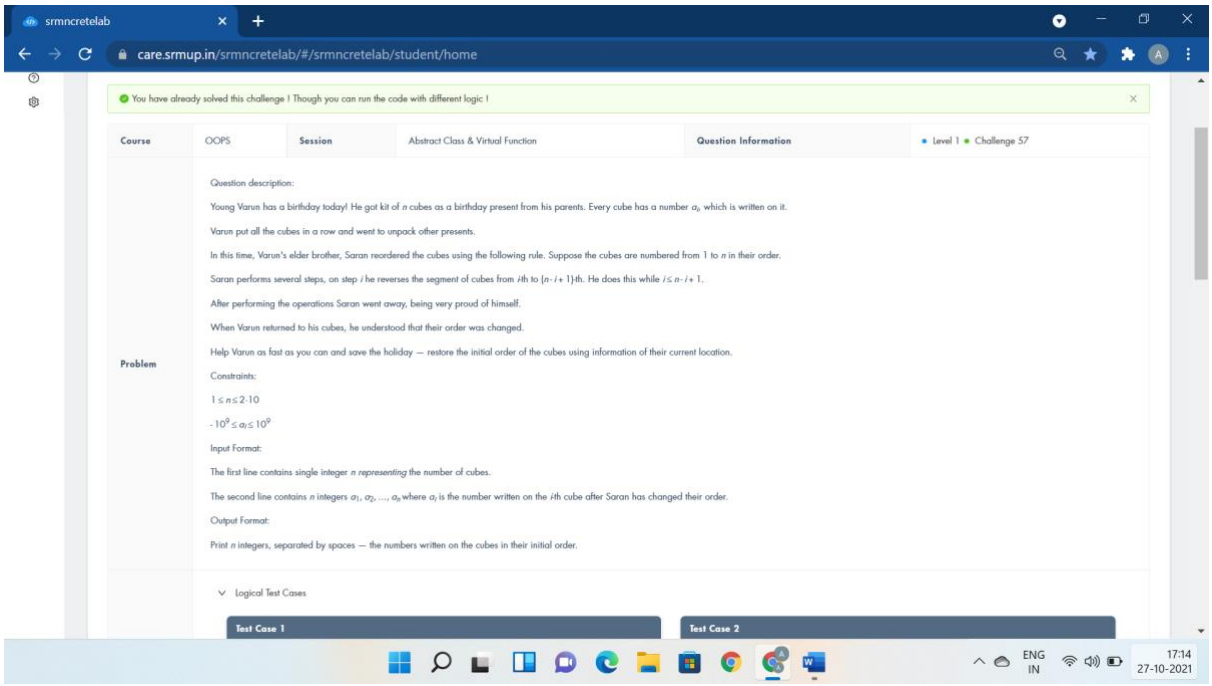

```
#include <iostream>
using namespace std;
class consumer{
   public:
   string name;
   virtual void getdata()=0;
   virtual void display()=0;
};
class transaction: public consumer{
```

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



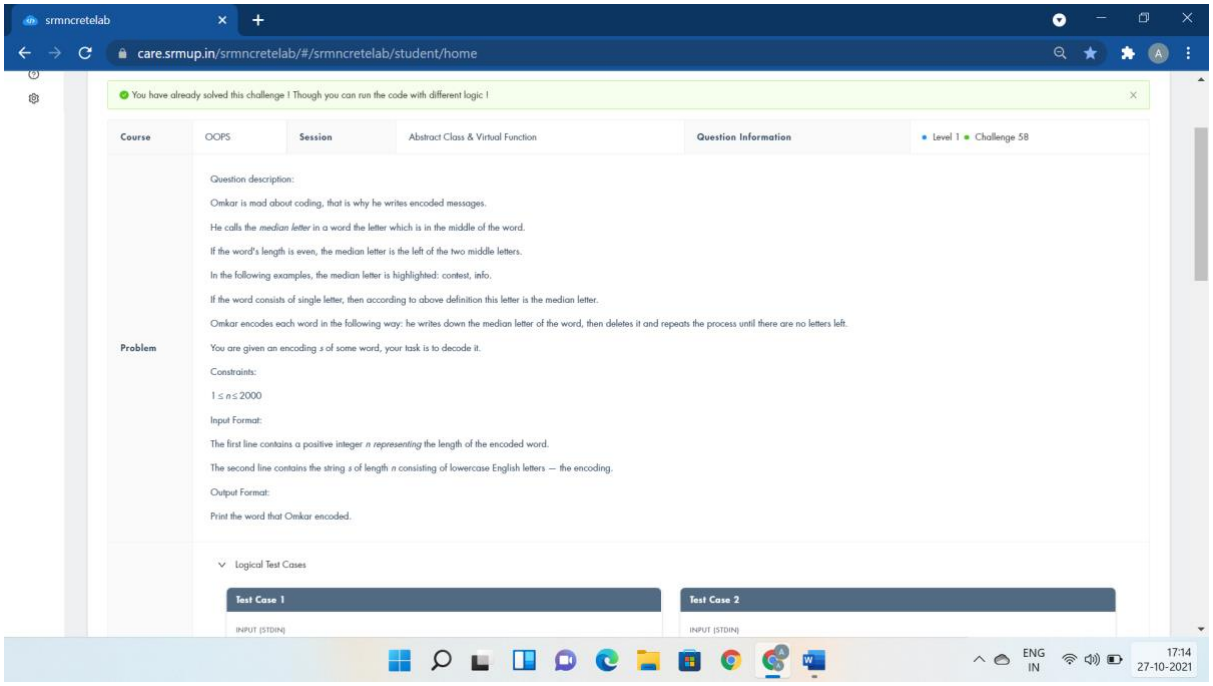

```
#include<iostream>
using namespace std;
class Problem {
   public:virtual void Divisor()=0;
};
class Calculation:public Problem{
   public:
      int n,k,i;
      void Divisor(){
      cin>>n>>k;
      }
      int Display()
      {
         int count;
      for(i=1;i<=n;++i)
      {
         if(n%i==0)
         {
            count++;
```

```
            if(count==k){
                cout<<i;
                return 1;
                }
          }
       }
       cout<<-1;
       return 1;
   }
};
int main()
{
   Calculation obj;
   obj.Divisor();
   obj.Display();
   return 0;
}
```



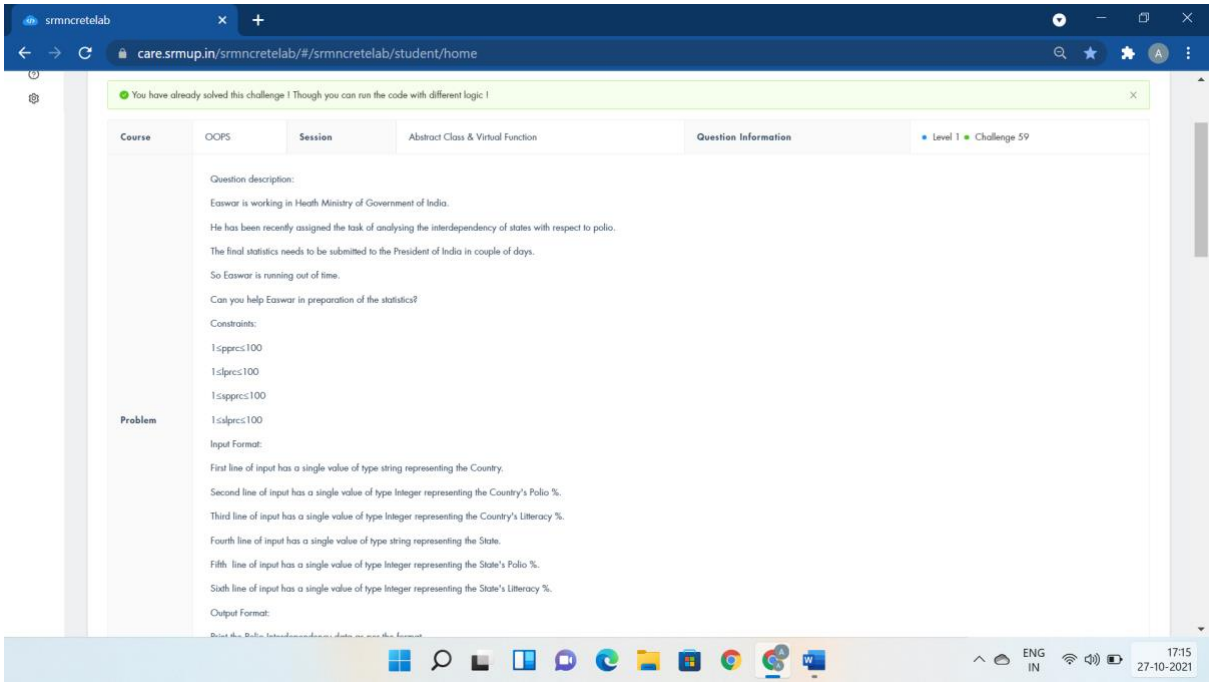

```
#include <iostream>
```

```
using namespace std;
class Gift {
   public:virtual void Cubes()=0;
};
class Birthday:public Gift{
   public:
   int a[10],n;
   void Cubes(){
      cin>>n;
      for(int i=0;i<n;i++)
      cin>>a[i];
      for(int i=0;i<n/2;i+=2)
         /*int temp=a[i];
         a[i]=a[n-i-1];
         a[n-i-1]=temp;*/
         swap(a[i],a[n-i-1]);
      for(int i=0;i<n;i++)
      cout<<a[i]<<" ";
   }
};
int main()
{
   Birthday obj;
   obj.Cubes();
            return 0;
}
```



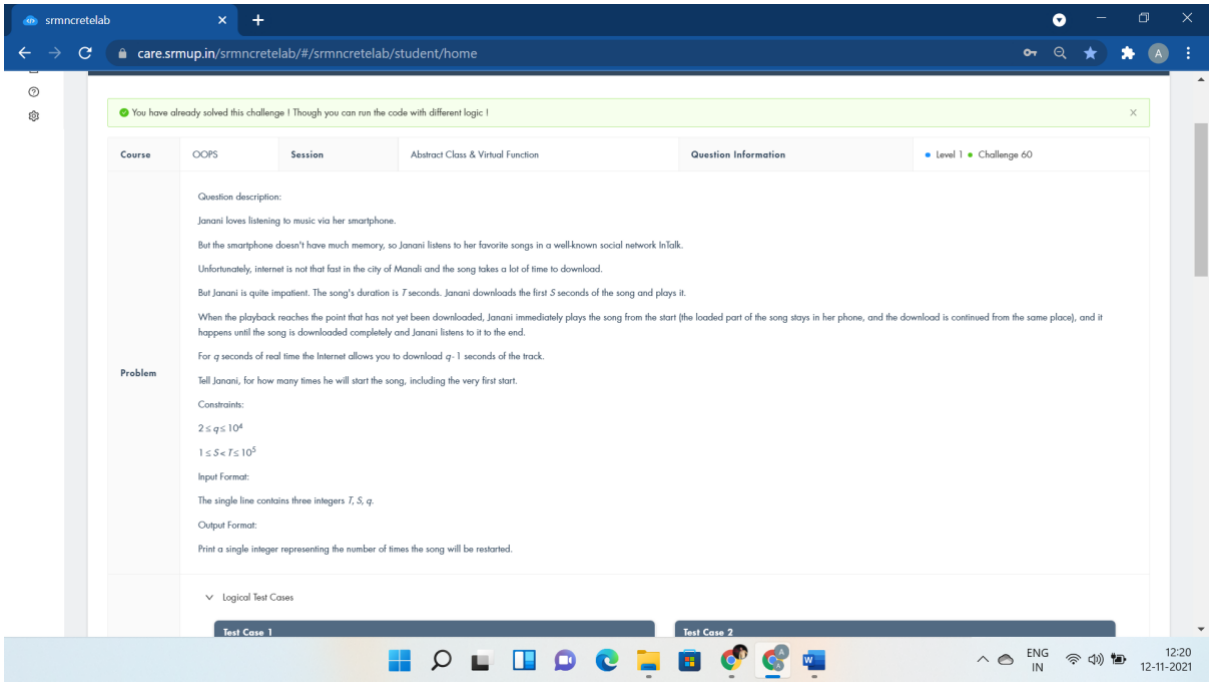

```
#include <iostream>
#include<string>
using namespace std;
class Decode{
   public:virtual void Convert()=0;
};
class Word:public Decode{
   public:
   string s1,s2;
   int n;
   void Convert(){
      cin>>n>>s1;
      for(int i=0;i<n;i++){
         if((n-i)%2==1)
                              s2=s2+s1[i];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

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




```
#include <iostream>
using namespace std;
class country
{
  public:
  virtual void getdata() = 0;
  virtual void display() = 0;
};
class state:public country
{
```

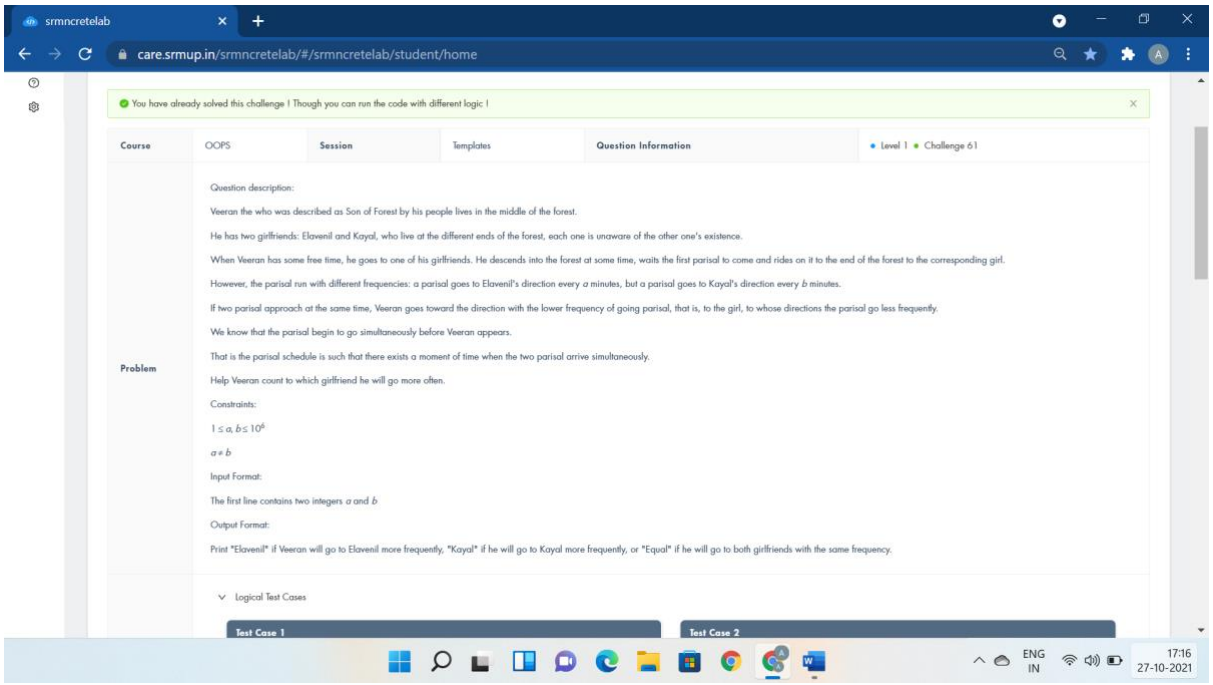
```
public:
char a[20];
   int b,c;
   char d[20];
   int e,f;
   void getdata(){
    cin>>a>>b>>c>>d>>e>>f;
   }
   void display()
   {
      cout<<"Country:"<<a<<endl<<"Country's Polio %:"<<b<<endl;
      cout<<"Country Literacy %:"<<c<<endl<<"Interdependency Rate:"<<(float)b/c<<endl;
      cout<<"State Name:"<<d<<endl<<"% of Polio of State:"<<e<<endl;
      cout<<"% of Literacy of State:"<<f<<endl<<"Interdependency Rate:"<<(float)e/f;
   }
};
int main() {
   if(0)
   cout<<"country::getdata();";
   country *o1;
   state o2;
   o1=&o2;
   o1->getdata();
   o2.display();
return 0;
}
```




```
#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()
{
```

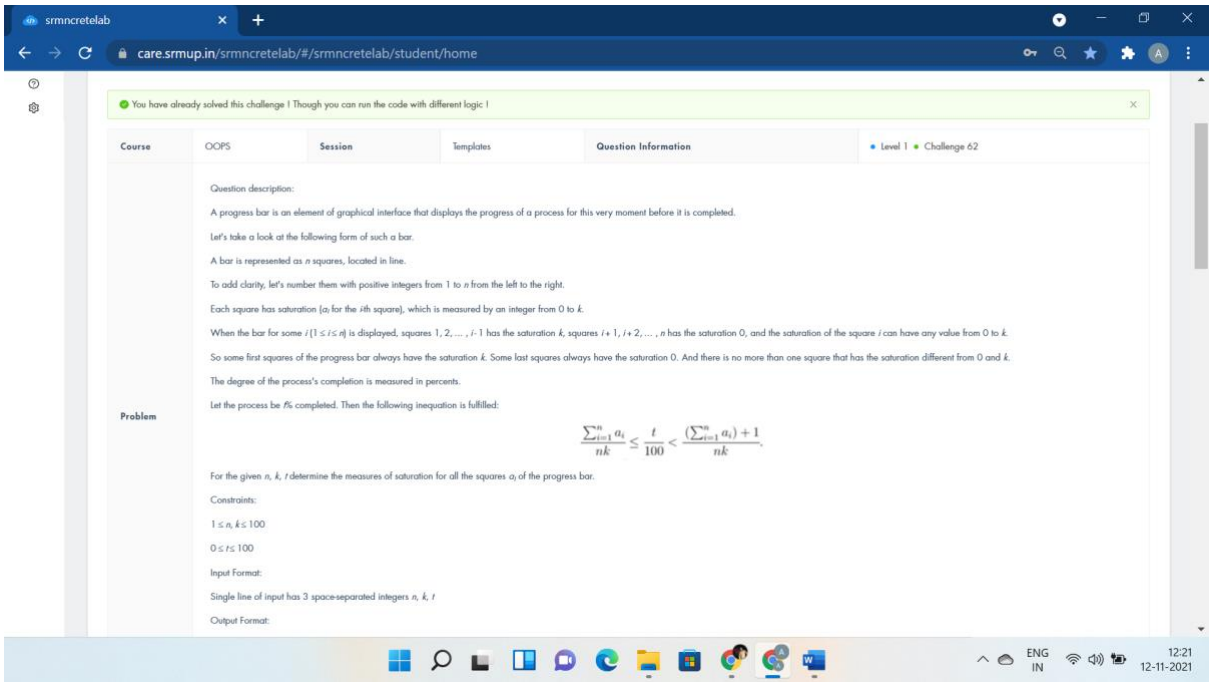
```
    LoveForMusic obj;
    obj.Listening();
    return 0;
}
```

Templates:-



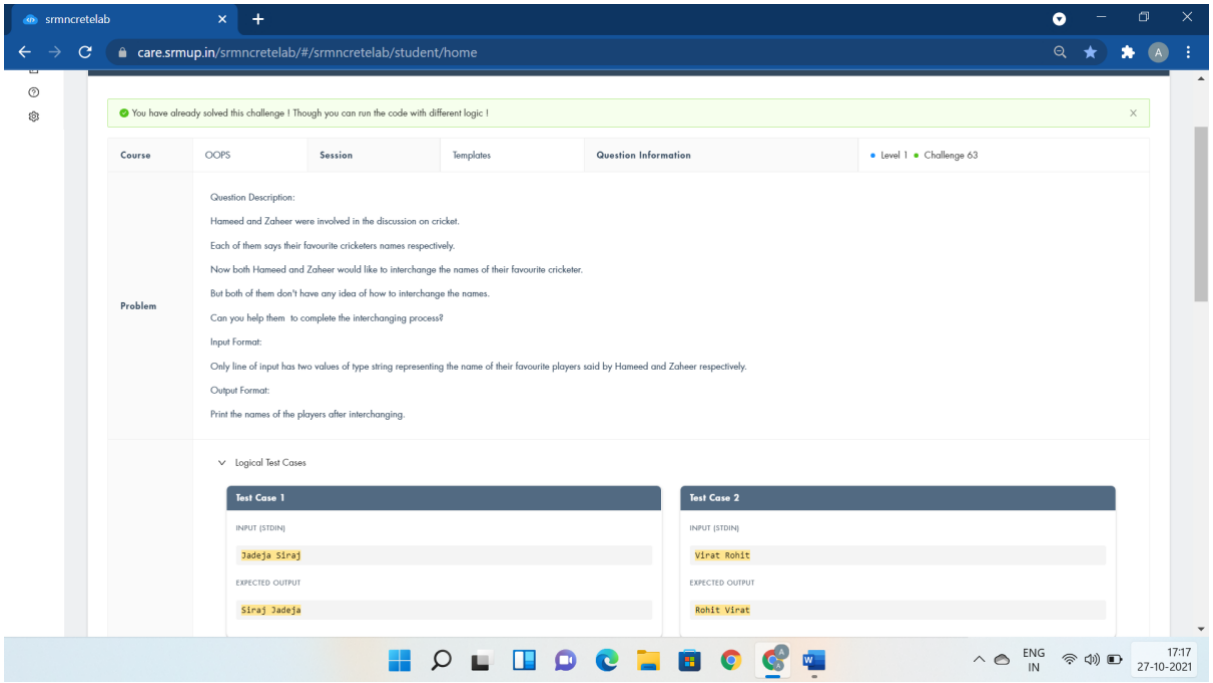

```
#include <bits/stdc++.h>
using namespace std;
template <class Forest>
Forest Visit(Forest a,Forest b){
   if(a>b)
      cout<<"Kayal\n";
   else
      cout<<"Elavenil\n";
   return 1;
}
int main()
{
   int a,b;
   cin>>a>>b;
```

```
    if(a%(a-b)==0 && b%(a-b)==0)
    cout<<"Equal\n";
    else
    Visit(a,b);
    return 0;
}
```



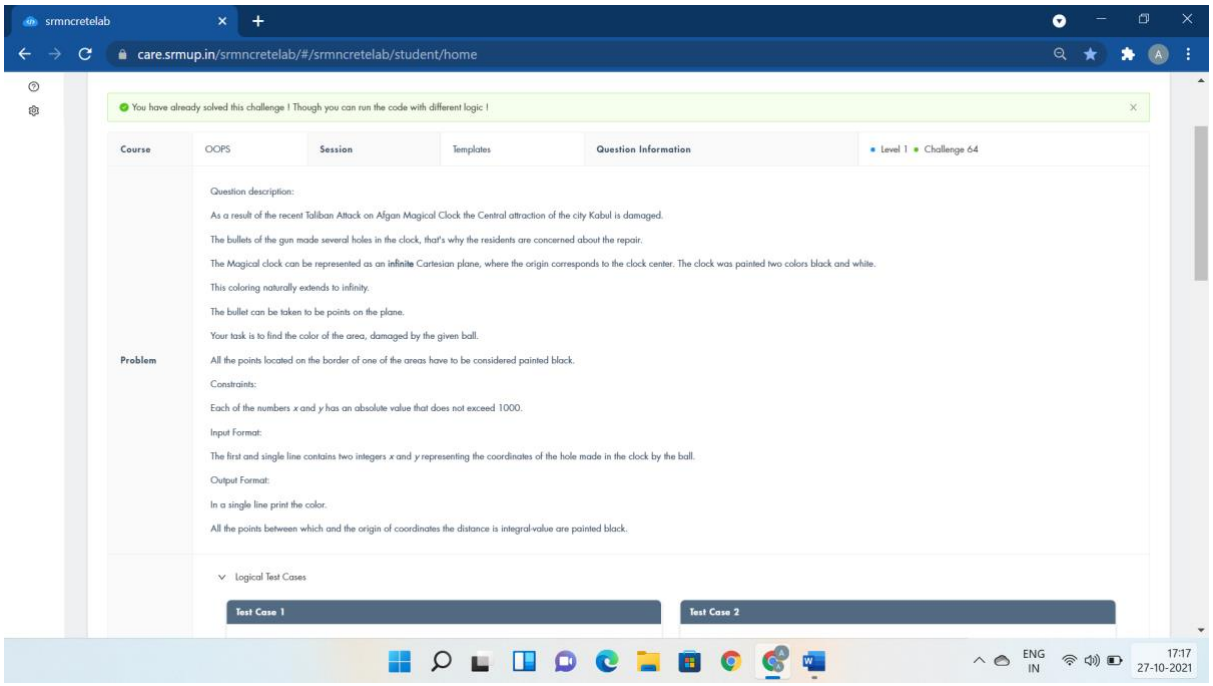

```
#include <iostream>
using namespace std;
template <class Interface>
Interface Bar(Interface n,Interface k,Interface t){
    t = t*k*n/100.0;
    while(n--){
        cout<<min(t,k)<<" ";
        t-=min(t,k);
    }
    return 1;
}
int main()
{
```

```
    int n,k,t;
    cin>>n>>k>>t;
    Bar(n,k,t);
        return 0;
}
```



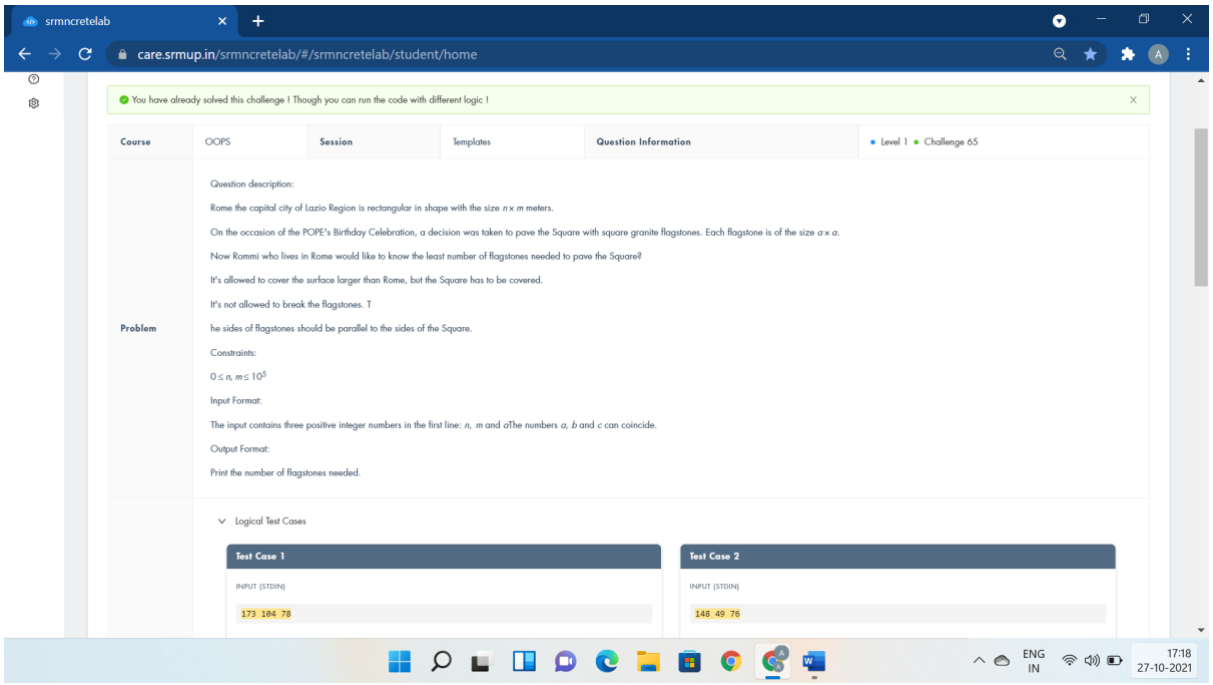

```
#include <iostream>
using namespace std;
template <class T>
void InterchangeFavPlayers(T &player1,T &player2){
    cout<<player2<<" "<<player1;
}
int main()
{
    string player1,player2;
    cin>>player1>>player2;
    InterchangeFavPlayers(player1,player2);
        return 0;
```

}

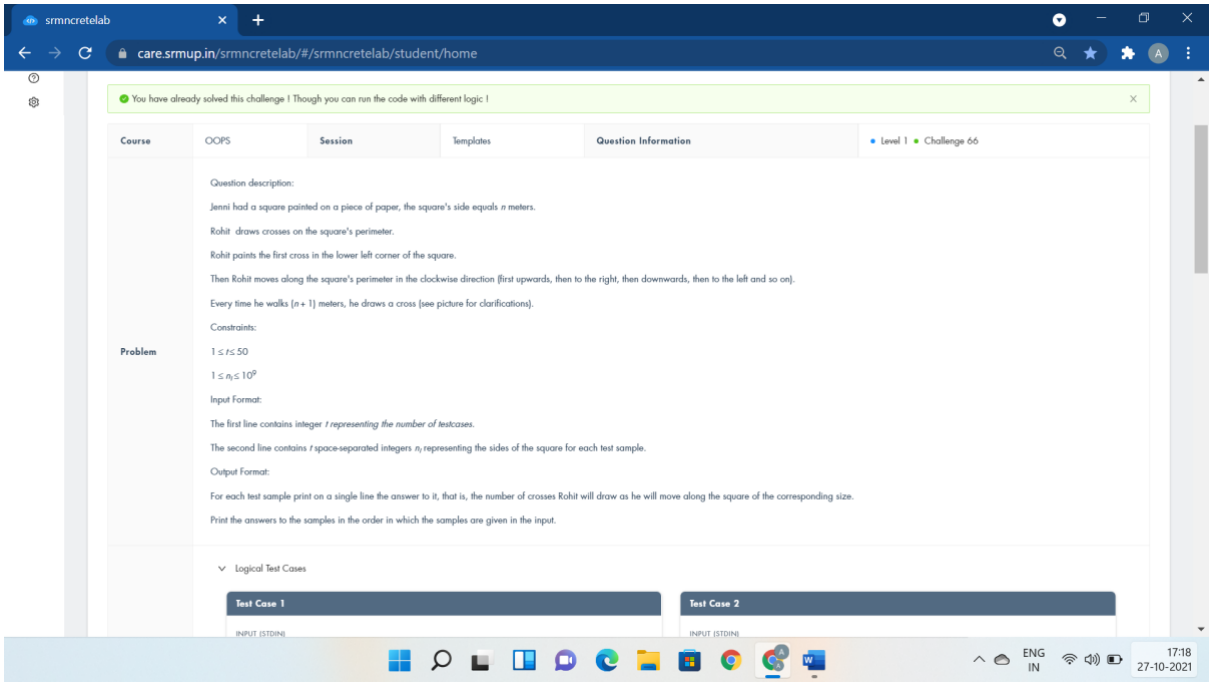



```
#include <iostream>
#include<cmath>
using namespace std;
template <class Hole>
Hole MagicClocl(Hole x,Hole y){
   int c;
            c=sqrt(x*x+y*y);
            if(c*c==x*x+y*y){
                        cout<<"black\n";
                        return 0;
            }
            if(x*y<0)
     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

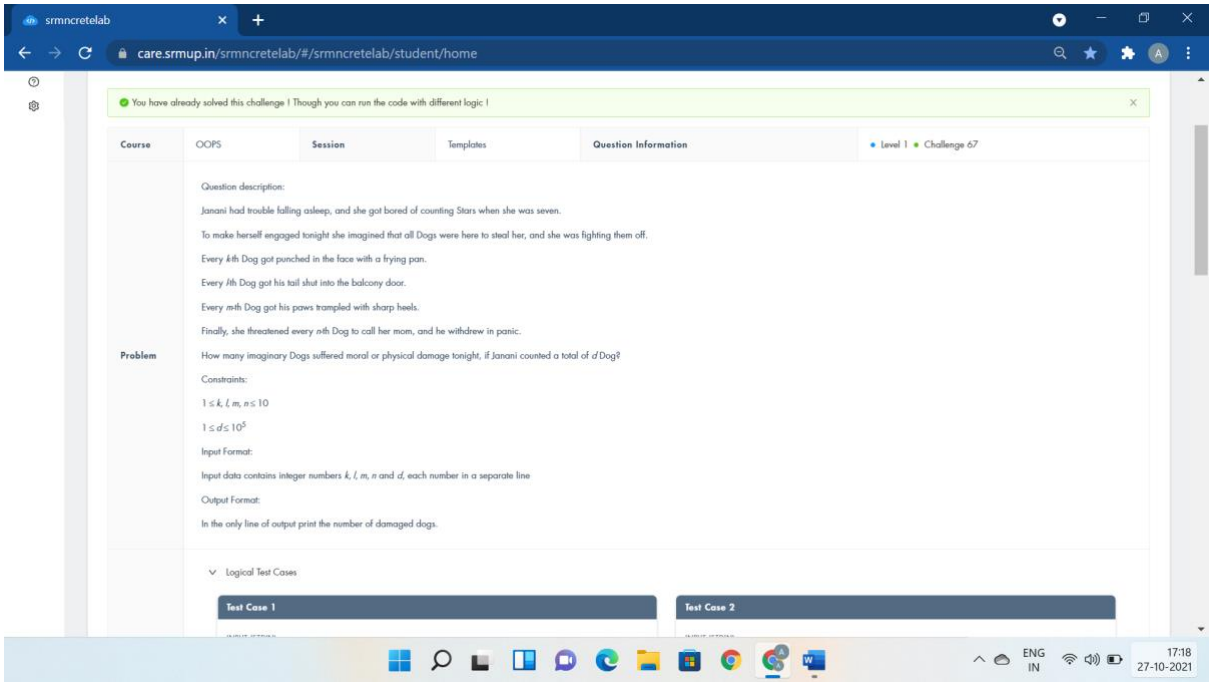
```
{
   int x,y;
   cin>>x>>y;
   MagicClocl(x,y);
            return 0;
}
```



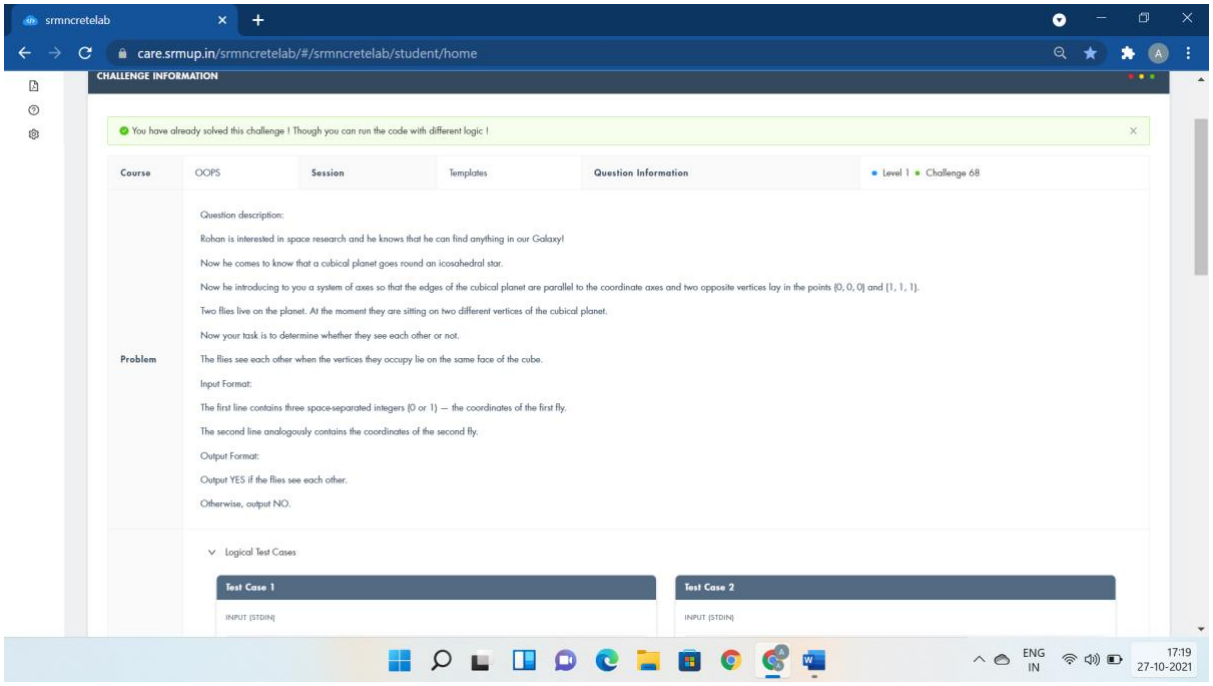

```
#include <iostream>
using namespace std;
template <class Celebration>
Celebration Rome(Celebration a,Celebration b,Celebration c){
   cout<<((b+c-1)/c)*((a+c-1)/c);
   return 1;
}
int main()
{
   int a,b,c;
   cin>>a>>b>>c;
   Rome(a,b,c);
            return 0;
}
```



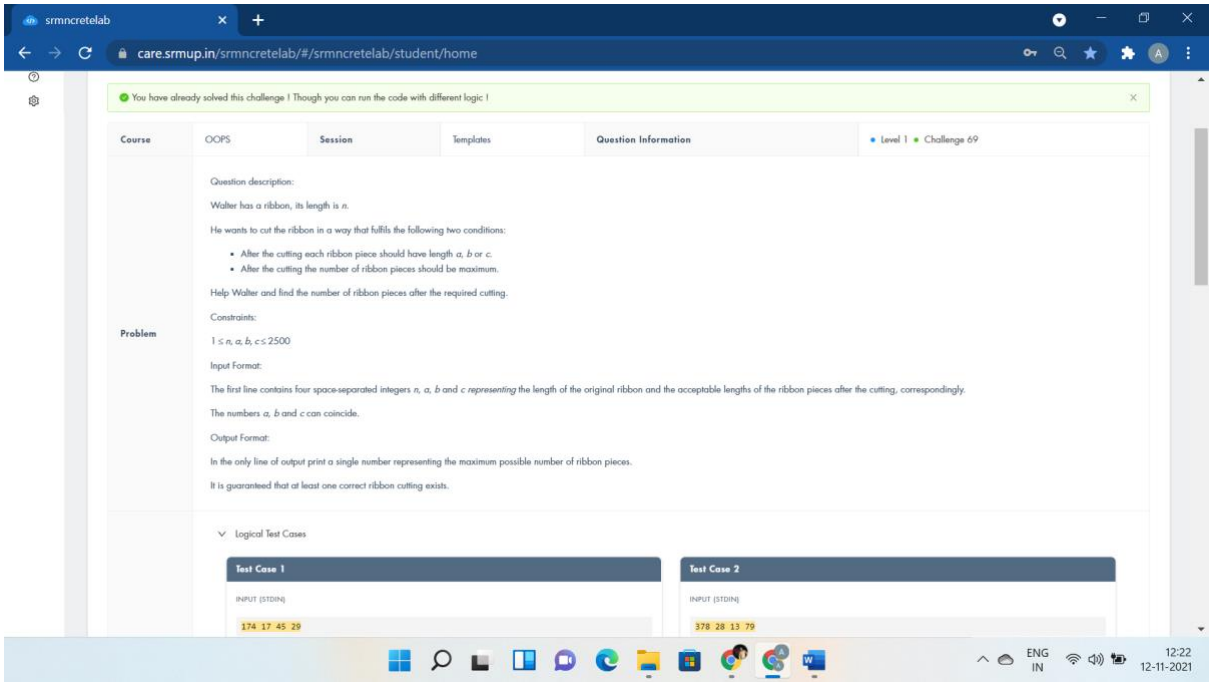

```
#include <iostream>
using namespace std;
template <class Paper>
Paper Square(Paper T){
   if(T%2==0)
   return 4*T+1;
   else if(T%4==1)
   return 2*T+1;
   else
   return T+1;
}
int main()
{
   int T,n;
   cin>>T;
   while(T--){
      cin>>n;
      cout<<Square(n)<<endl;
   }
            return 0;
}
```



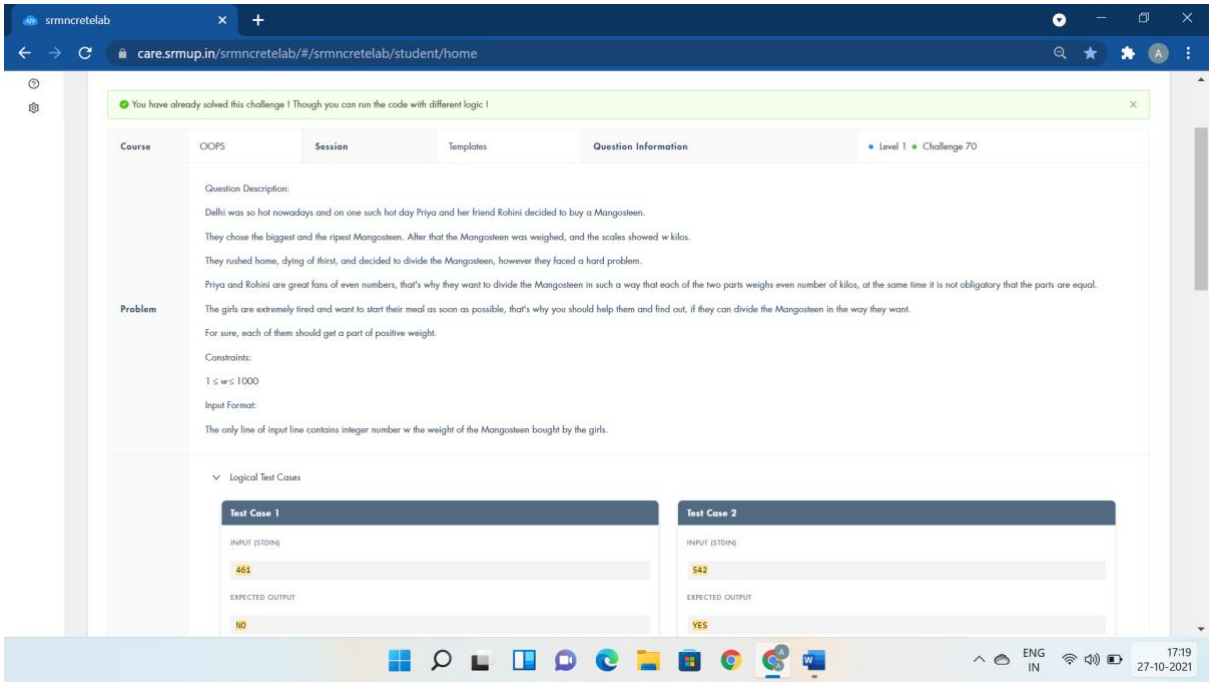

```
#include <iostream>

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;

}
```




```
#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;
}
```

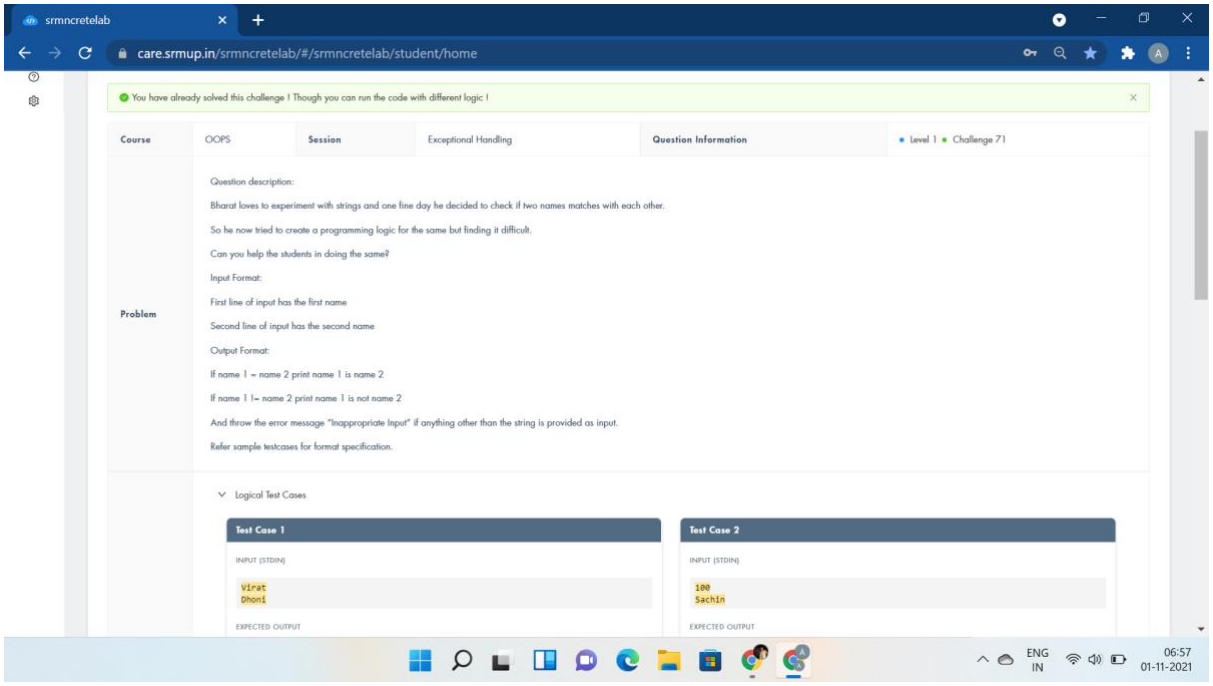



```
#include<bits/stdc++.h>
using namespace std;
template <class Ribbon>
Ribbon Pieces(Ribbon n,Ribbon a,Ribbon b,Ribbon c){
   int d=1,e,i,j;
   for(i=0;i<=4000;i++)
   for(j=0;j<=4000;j++) {
      e=n-a*i-b*j;
      if(e>=0&&e%c==0)
      d=max(d,i+j+e/c);
   }
   cout<<d;
   return 1;
}
int main(){
int n,a,b,c;
cin>>n>>a>>b>>c;
Pieces(n,a,b,c);
}
```



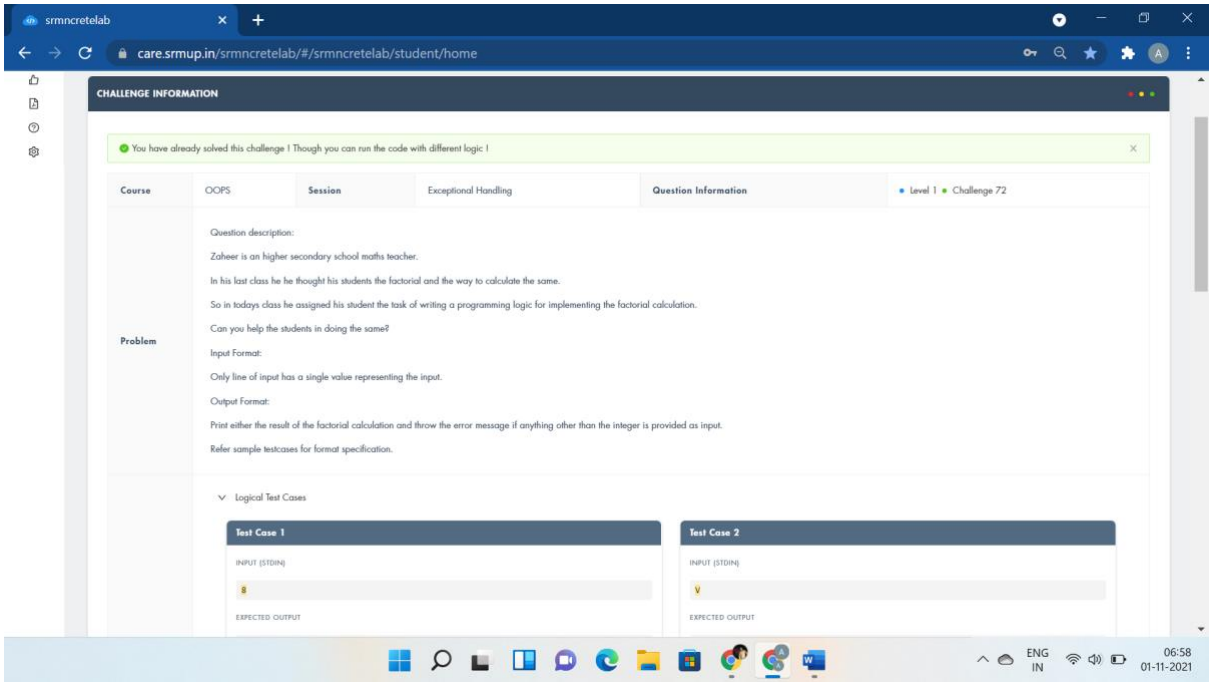

```
#include <iostream>
using namespace std;
template<class T>
T DivideMangosteen(T PurchasedWeight){
   if(PurchasedWeight%2==0)
   cout<<"YES";
   else
   cout<<"NO";
   return 1;
}
int main()
{
   int PurchasedWeight;
   cin>>PurchasedWeight;
   DivideMangosteen(PurchasedWeight);
            return 0;
}
```

Exceptional Handling:-



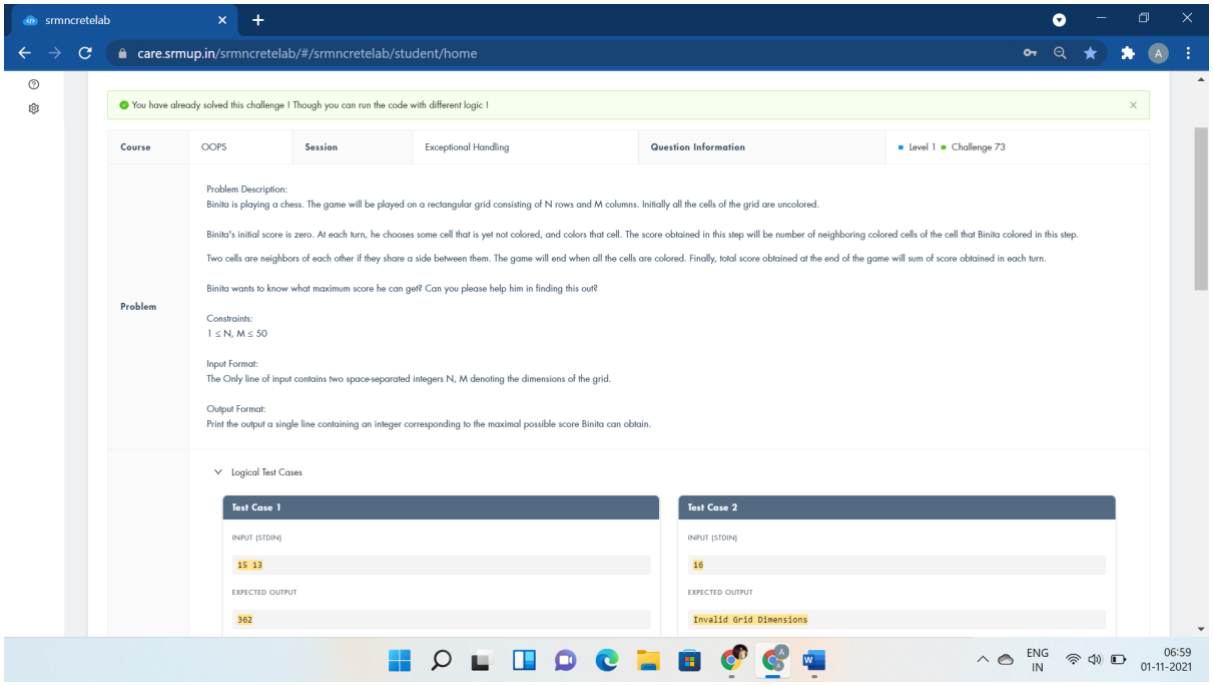

```
#include <iostream>

using namespace std;

int main()

{

  string str1,str2;

  try{

    cin>>str1>>str2;

    int count, n=str1.size();

    if(cin){

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

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

        throw 0;

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

        count++;

      }

      if(count!=n)

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

      else

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

    }

  }

  catch (int i){

    cout<<"Inappropriate Input";
```

```
    }
            return 0;
}
```



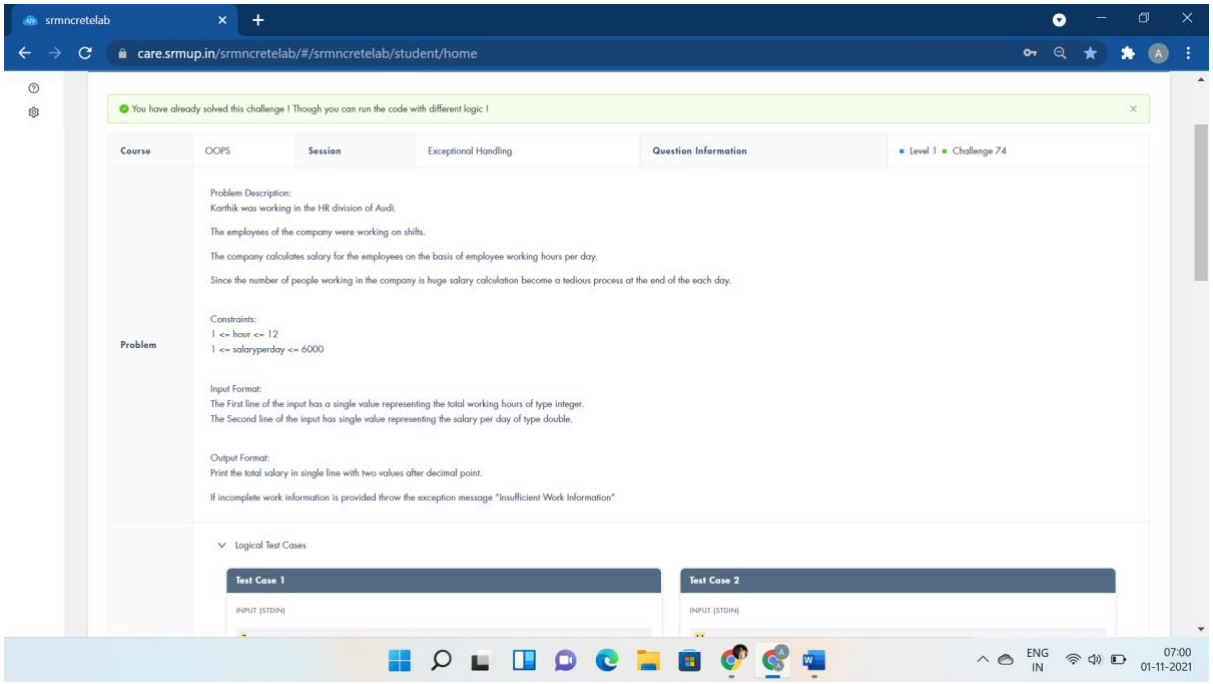

```
#include <bits/stdc++.h>
#include <string.h>
using namespace std;
int main()
{
    int k;
  try{
    cin>>k;
    if(cin)
    cout<<fixed<<setprecision(0)<<tgamma(k+1);
    else
    throw "e";
  }
  catch (int i){
  }
  catch (const char *exp){
    cout<<"Input should be a Integer";
```

```
    }
        return 0;
}
```

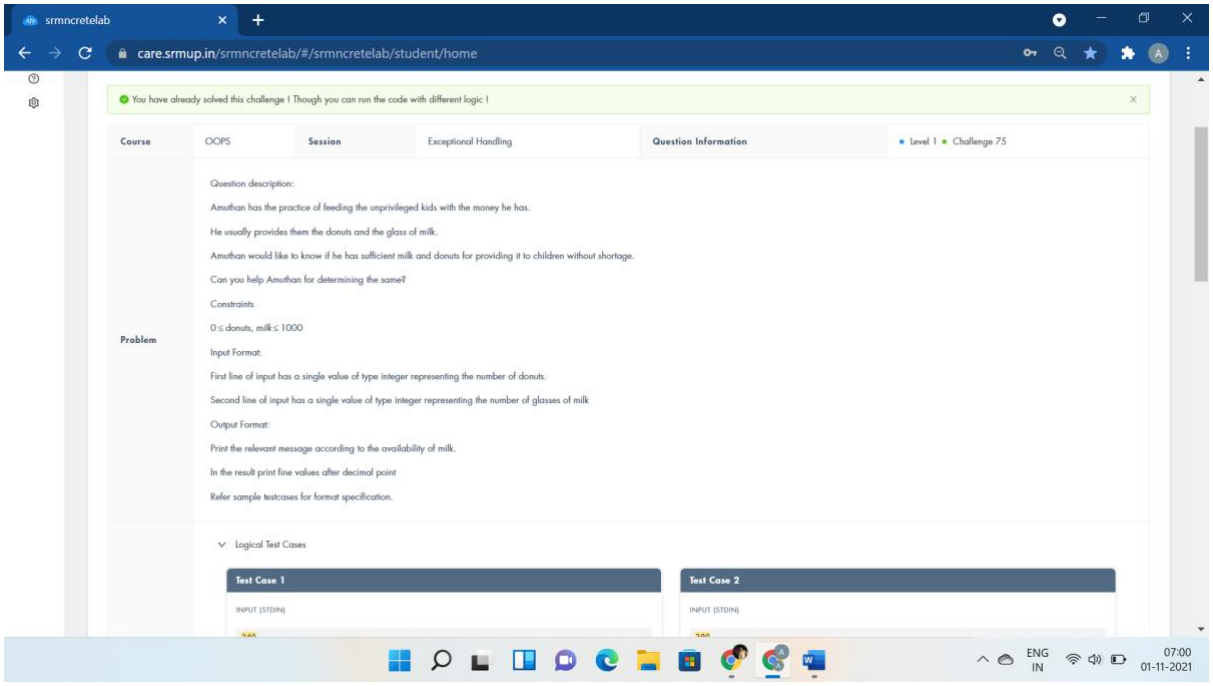



```
#include <iostream>
using namespace std;
int main()
{
   int n,m;
   try{
   cin>>n;
   cin>>m;
   if(cin){
      cout<<n-1+(1+2*(n-1))*(m-1);
   }
   else
   throw 0;
   }
   catch(int griddimensions)
   {
      cout<<"Invalid Grid Dimensions";
   }
```

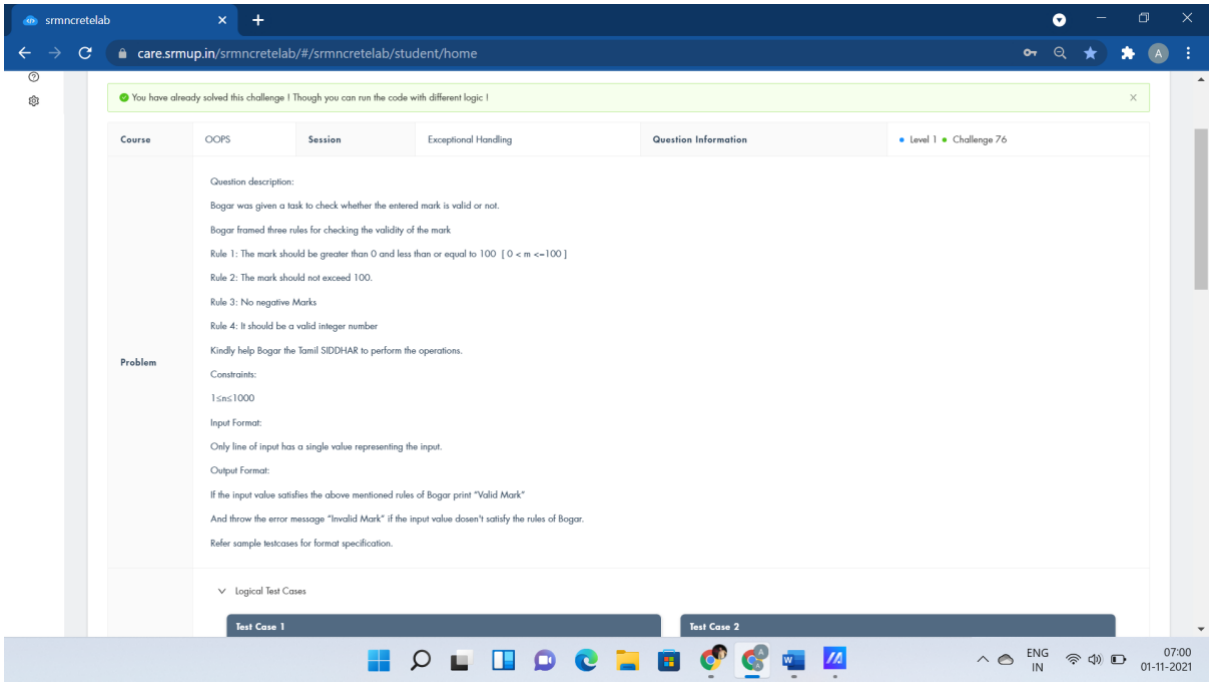
```
        return 0;
}
```



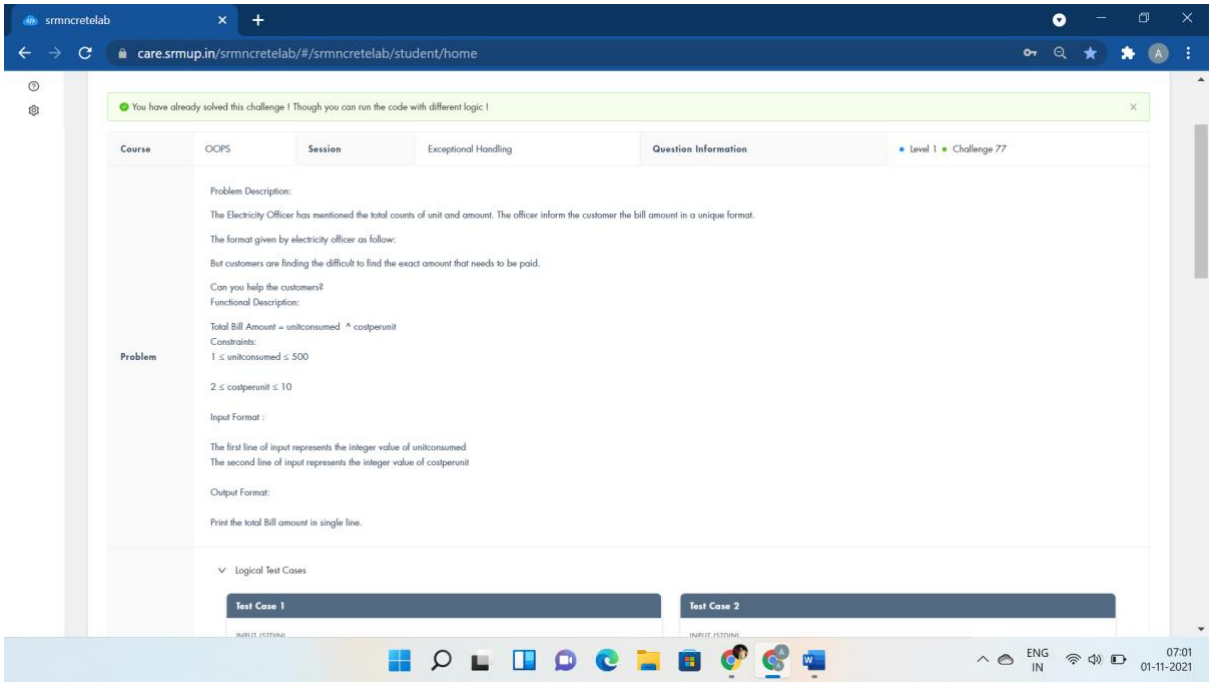

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
   float hour,salaryperday;
   try{
   cin>>hour;
   cin>>salaryperday;
   if(cin){
      cout<<fixed<<setprecision(2)<<hour*salaryperday;
   }
   else
   throw 0;
   }
   catch(int workstatus)
   {
      cout<<"Insufficient Work Information";
   }
        return 0;
}
```



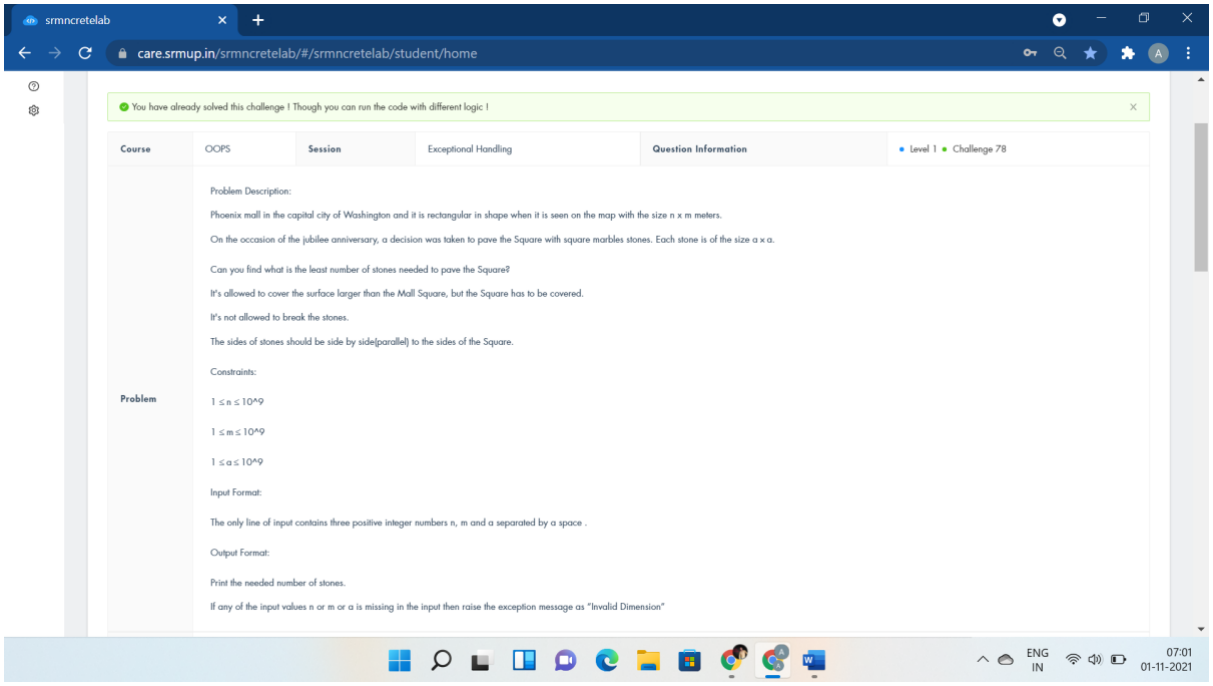

```
#include <iostream>

using namespace std;

int main()

{

   int donuts,milk;

   try{

      cin>>donuts;

      cin>>milk;

      if(milk==0)

      throw donuts;

      else

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

   }

   catch(int e){

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

   }

            return 0;

}
```



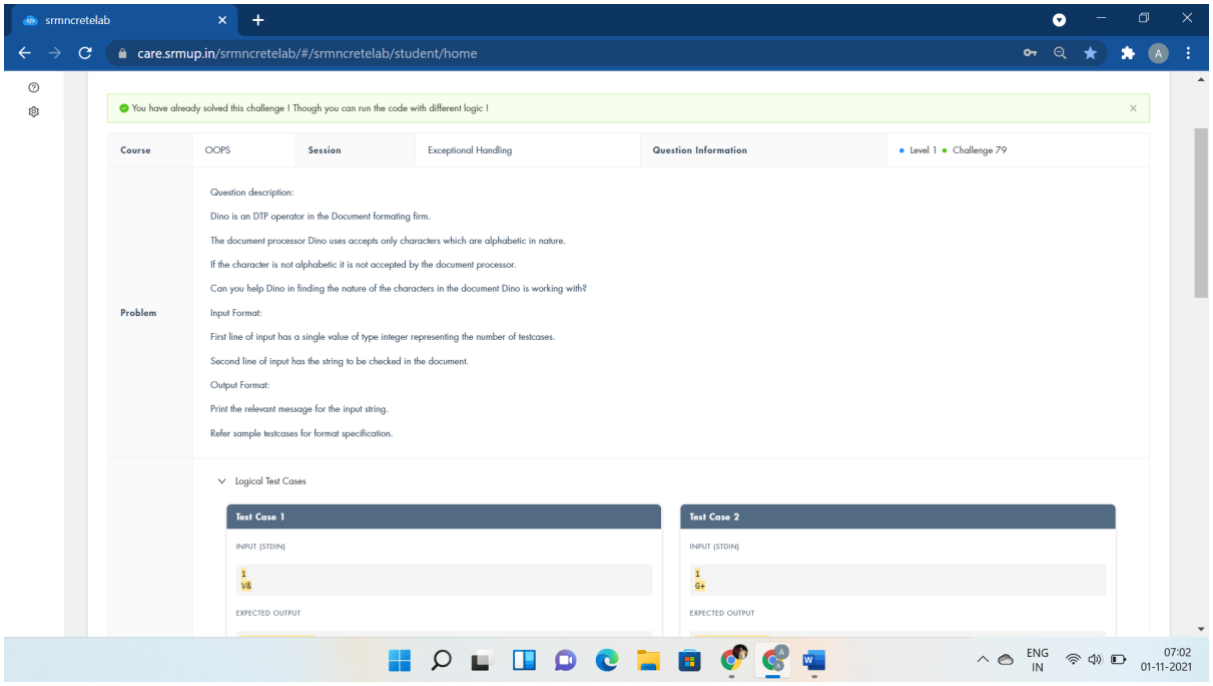

```
#include <iostream>
#include <math.h>
using namespace std;
int main()
{
   int a;
   try {
      cin>>a;
      if (a>0 && a<=100)
      cout<<"Valid Mark";
      else
      throw "e";
   }
   catch(const char* t){
      cout<<"Invalid Mark";
   }
}
```



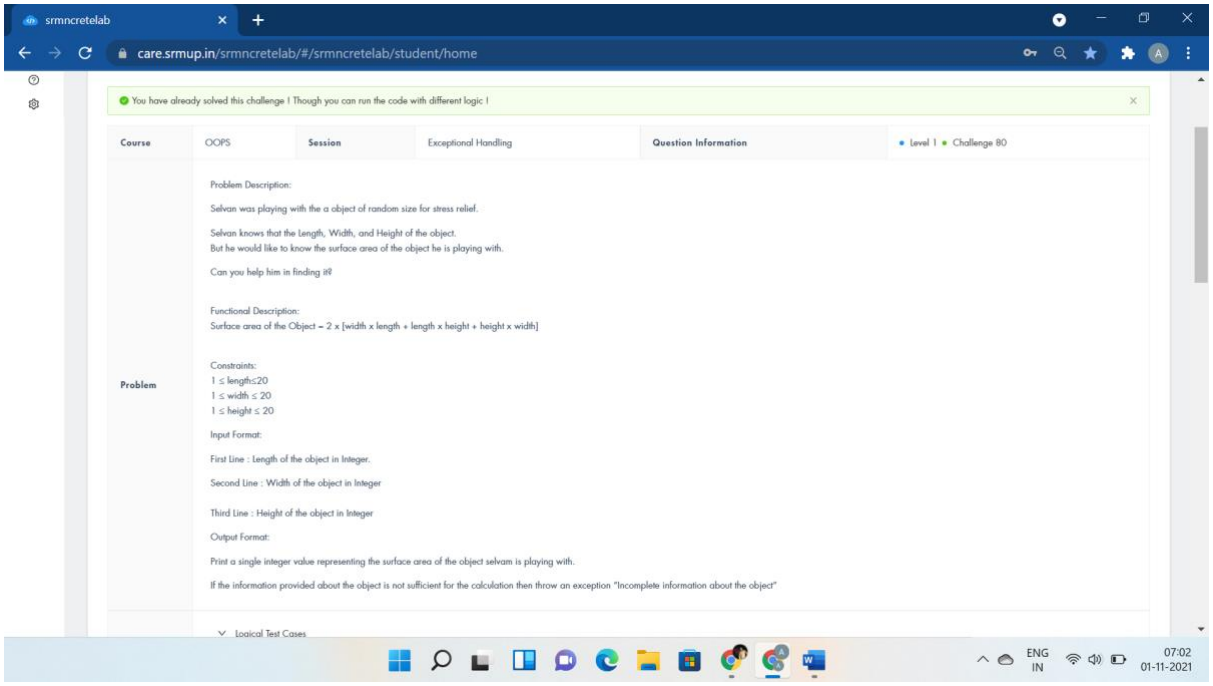

```
#include <bits/stdc++.h>

using namespace std;

int main()

{

   int unitconsumed,costperunit;

   try{

      cin>>unitconsumed;

      cin>>costperunit;

      long int res;

      res=pow(unitconsumed,costperunit);

      if(cin){

         cout<<res;

      }

      else

      throw 0;

   }

   catch(int unit){

      cout<<"Incomplete Data";

   }

            return 0;

}
```




```
#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;

}
```

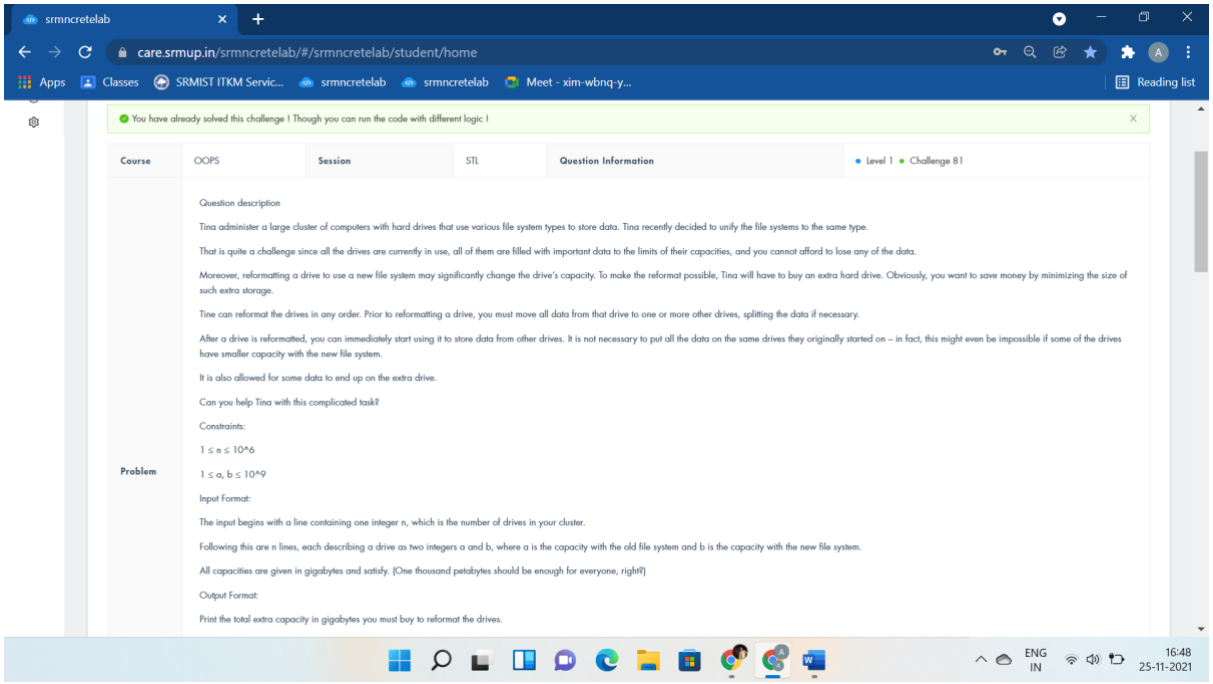



```
#include<bits/stdc++.h>

#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;

         }

      }

   }

}
```



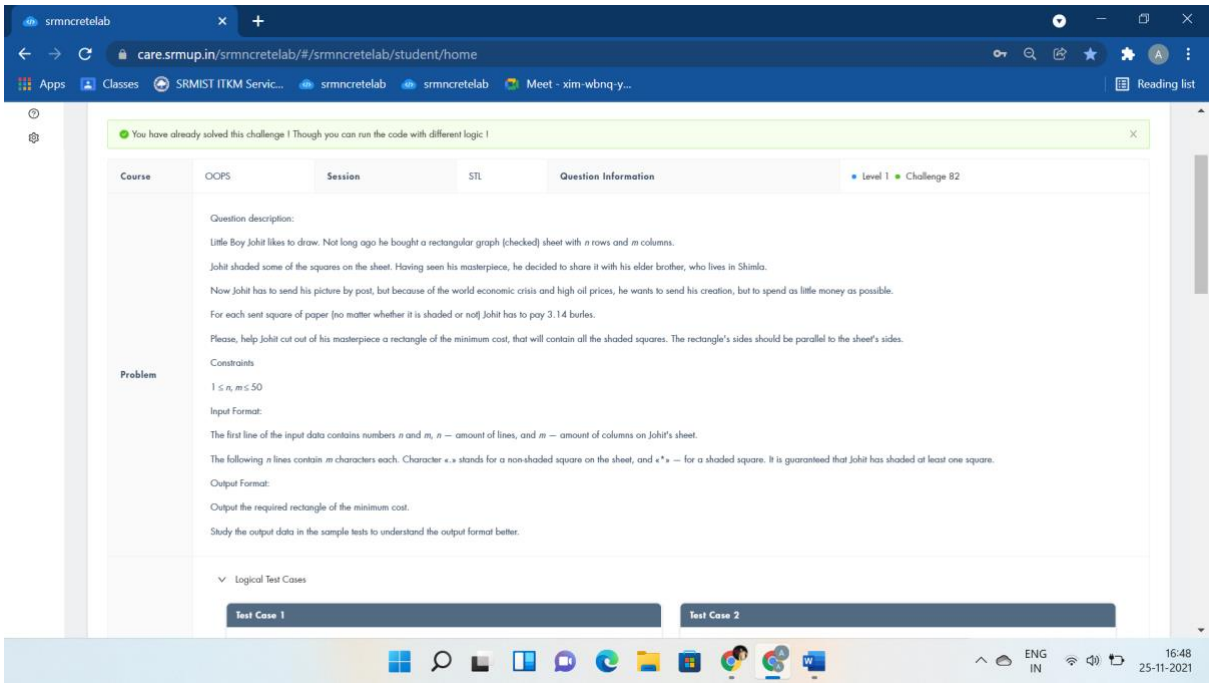

```
#include <iostream>

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:-



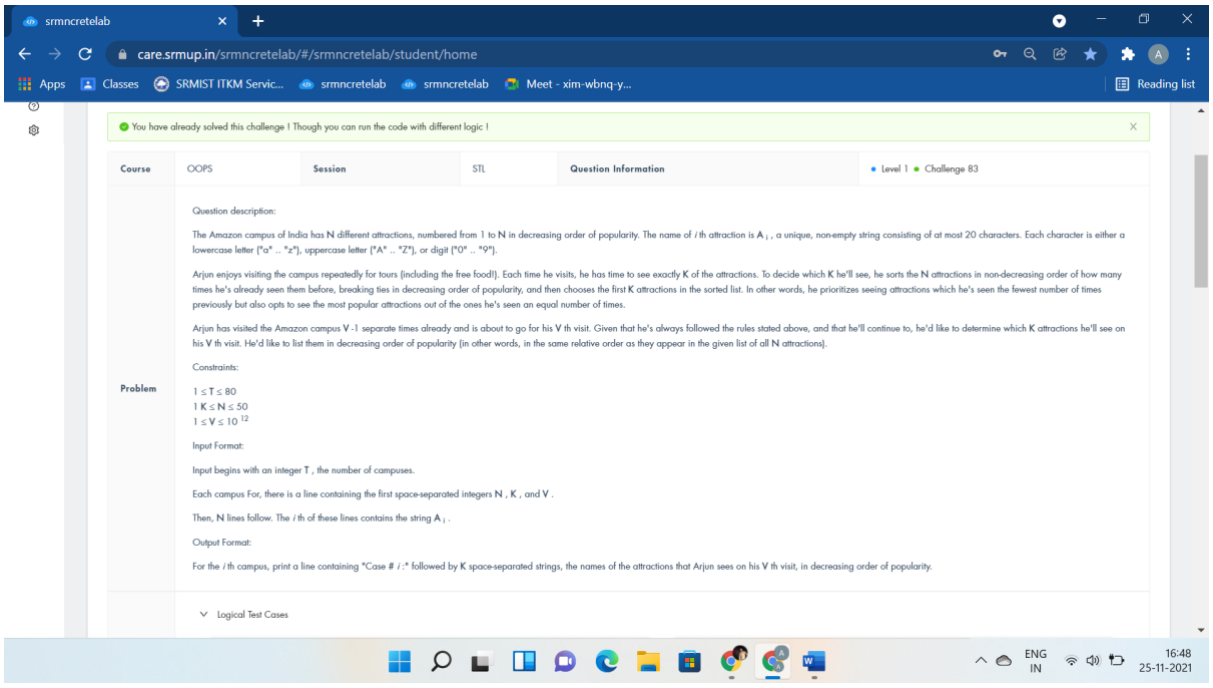

```
#include <algorithm>
#include <iostream>
#include <vector>
using namespace std;

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;
 }
}
```



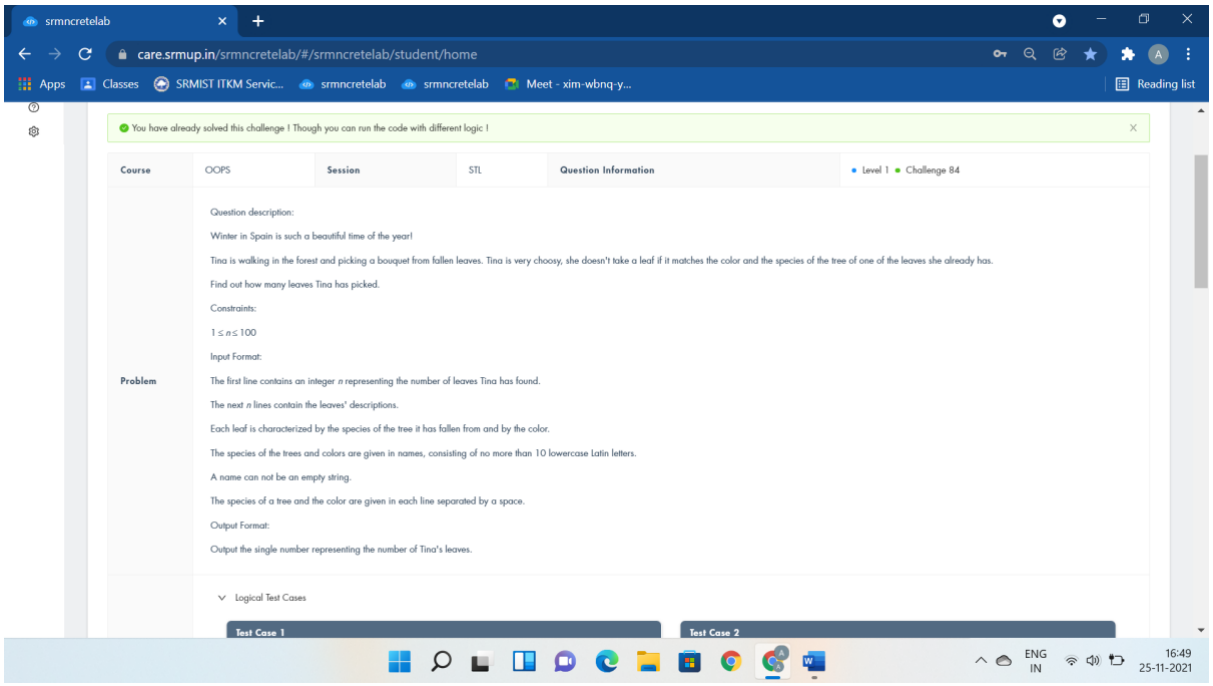

```
#include<bits/stdc++.h>
using namespace std;
int n,m,sx=99999,sy=99999,x,y;
char a[55][55];
int main(){
        cin>>n>>m;
        for(int i=1;i<=n;i++){
                for(int j=1;j<=m;j++){
                        cin>>a[i][j];
                        if(a[i][j]=='*'){
                                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]";
}
```



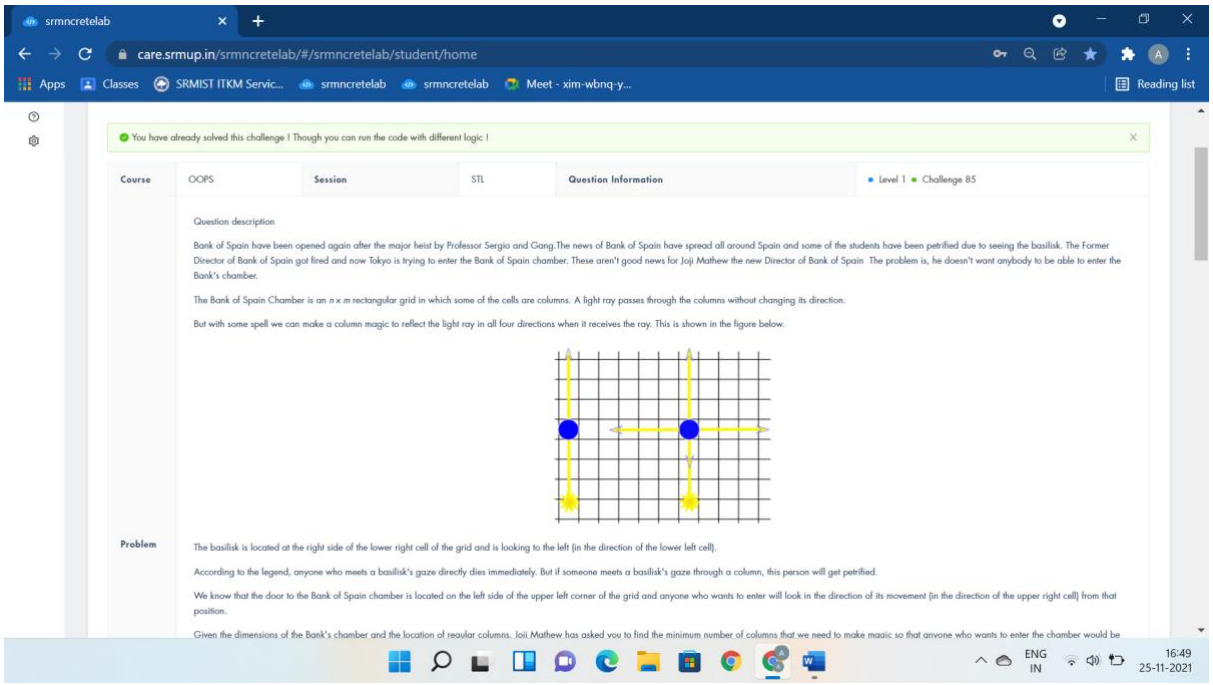

```
#include <bits/stdc++.h>
using namespace std;
typedef long long LL;
const int N=55;
LL n, k, v, idx;
string name[N];
int main(){
  LL t; cin>>t; while(t--){
    cin>>n>>k>>v;
    for(int i=0; i<n; i++)
      cin>>name[i];
    LL st=((v-1)*k)%n;
```

```
  //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());";
}
```



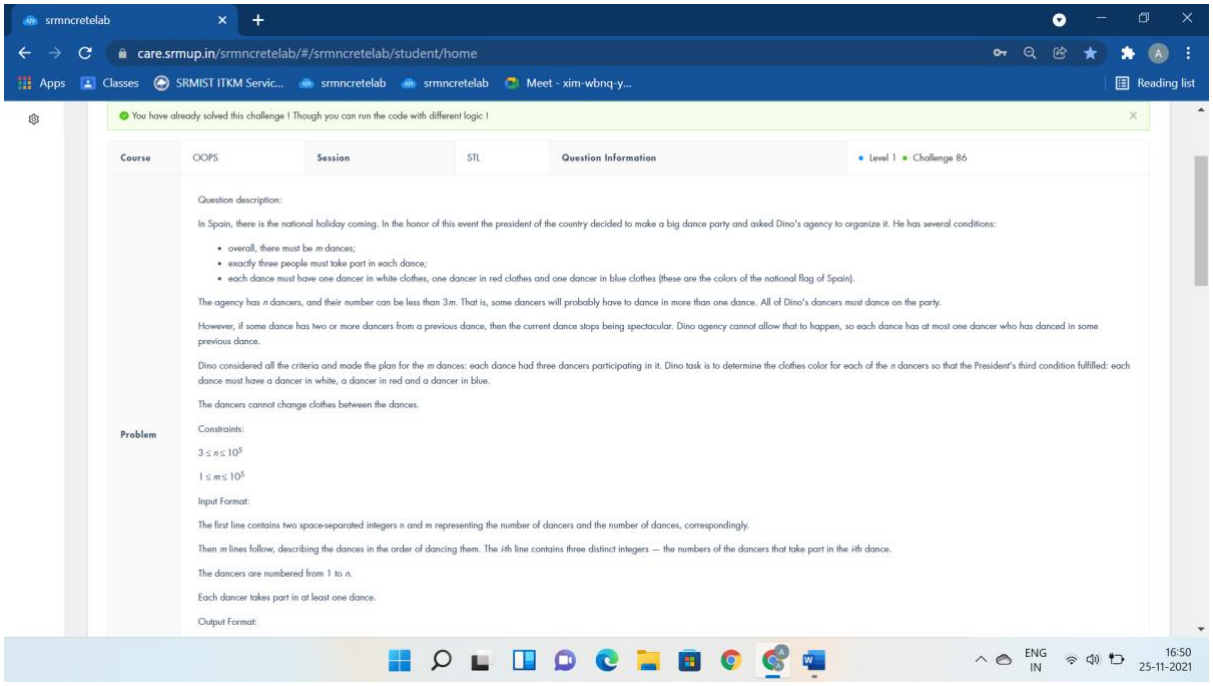

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
   int n;
   cin>>n;
   set<pair<string,string>>Descriptionofleaves;
   string species,color;
   while(n--){
```

```
    cin>>species>>color;
  Descriptionofleaves.insert(make_pair(species,color));
  }
        cout<<Descriptionofleaves.size();
        return 0;
}
```



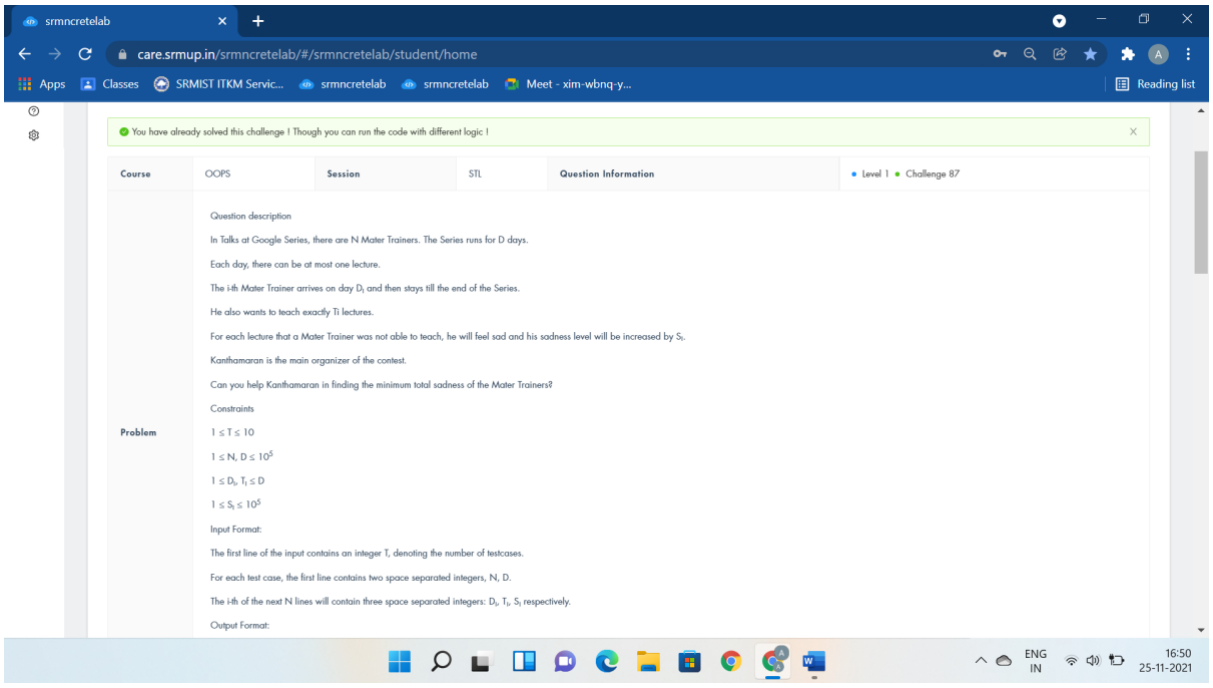

```
#include <bits/stdc++.h>
using namespace std;
void sum(){}
int n,m;
vector <int> use[2020];
int cost[2020];
string g[1010];
int main()
{
        cin>>n>>m;
        for(int i=0;i<n;i++)
        {
                cin>>g[i];
                for(int j=0;j<m;j++)
                {
```

```
                        if(g[i][j]=='#')
                        {
                                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);";
}
```



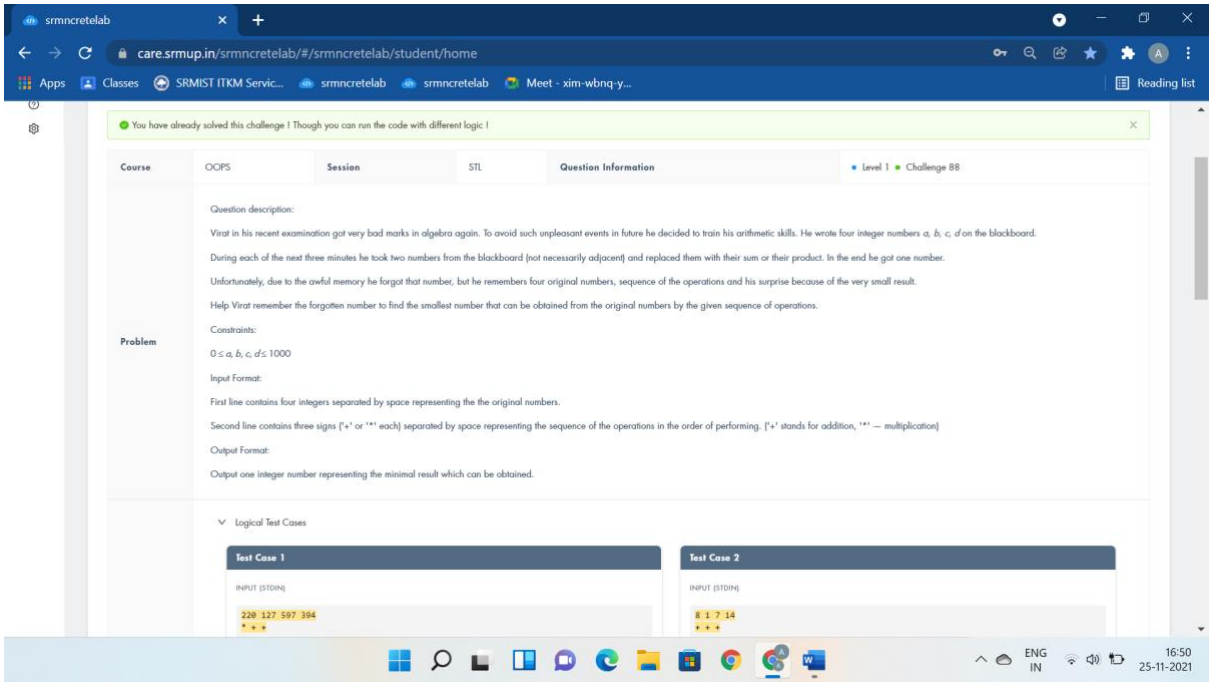

```
#include<bits/stdc++.h>
using namespace std;
typedef long long int ll;
ll a[100006],c[3];
int main()
{
   ll n,m,i,j,k,l,sum=0;
   cin>>n>>m;
   for(i=0;i<m;i++)
   {
      sum=0;
      for(j=0;j<3;j++)
      {
         cin>>c[j];
         sum=sum+a[c[j]];
      }
      l=1;
      for(k=0;k<3;k++)
      {
         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;";}
```



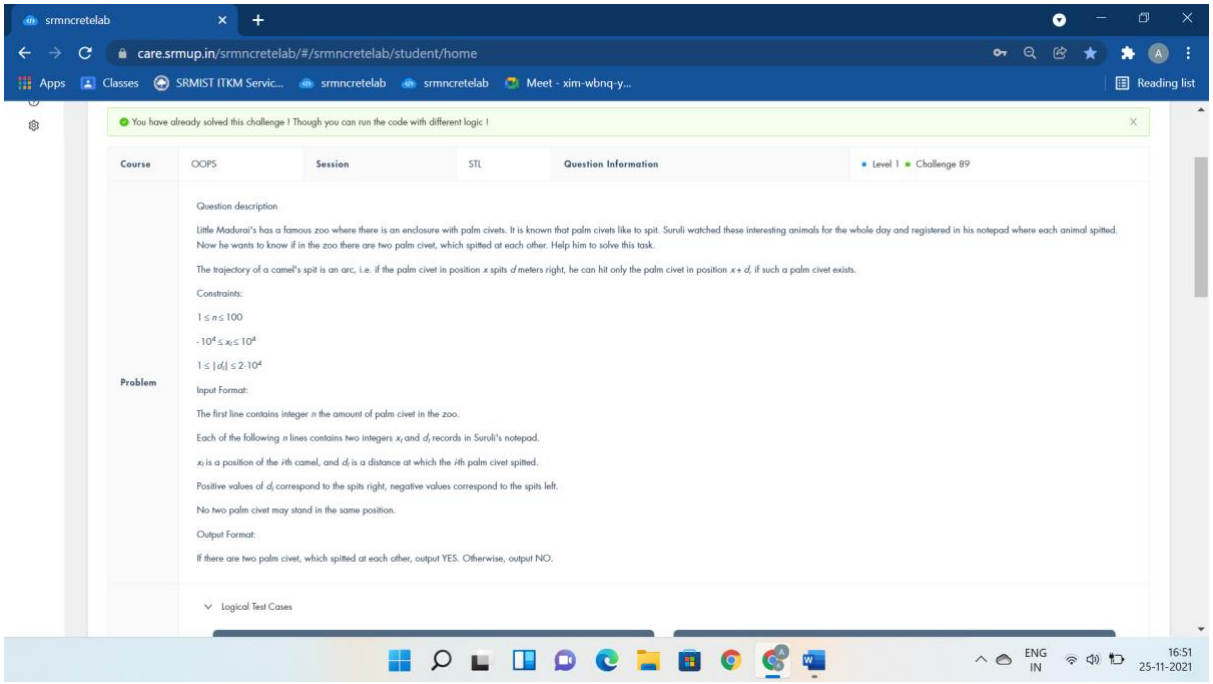

```
#include <bits/stdc++.h>
#define ll long long
using namespace std;
int main(){
    int t;
    cin >> t;
    while (t--) {
        int n, d;
        cin >> n >> d;
        map<ll, vector<pair<long,long>>>TGS;
        for (int i = 0; i < n; i++){
            ll day, lec, sad;
```

```
        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 ";}
```



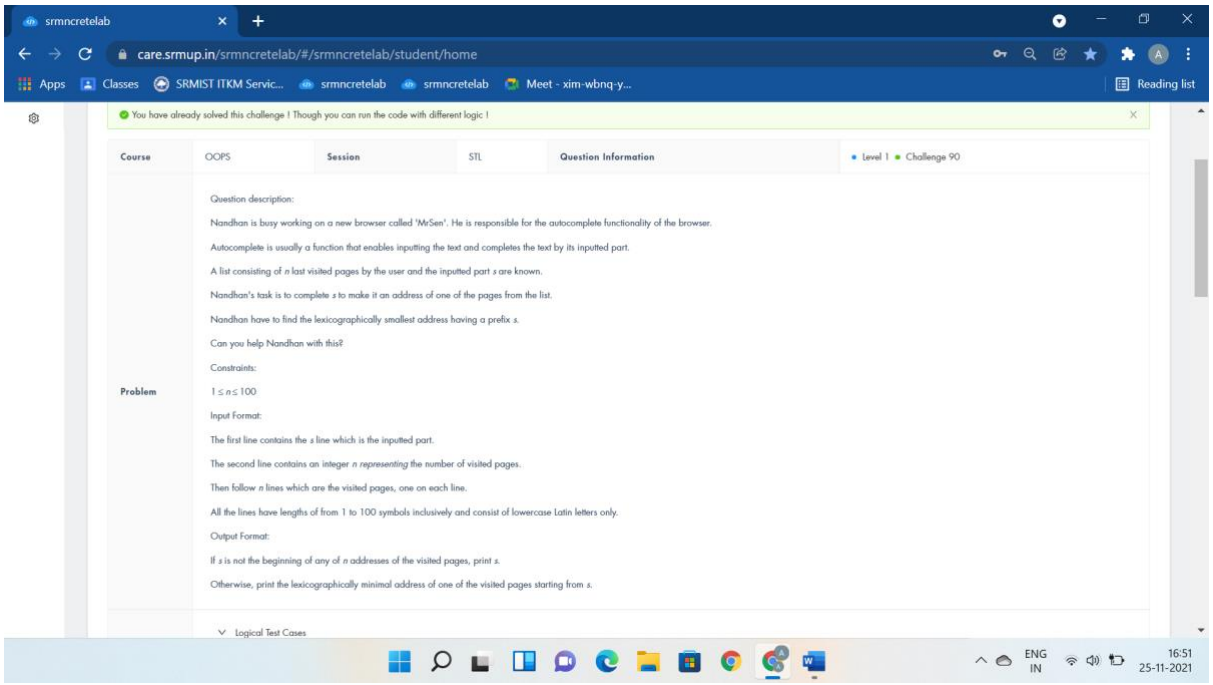

```
#include <bits/stdc++.h>
using namespace std;
long long ans=1e15;
deque<char>Operations(20);
void solve(vector<long long> a,int id){
	if((int)a.size()==1){
		ans=min(ans,a[0]);
		return;
	}
	for(int i=0;i<(int)a.size();i++){
		for(int j=0;j<i;j++){
			vector<long long> b;
			if(Operations[id]=='+') 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;
}
```




```
#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;
                        }
```

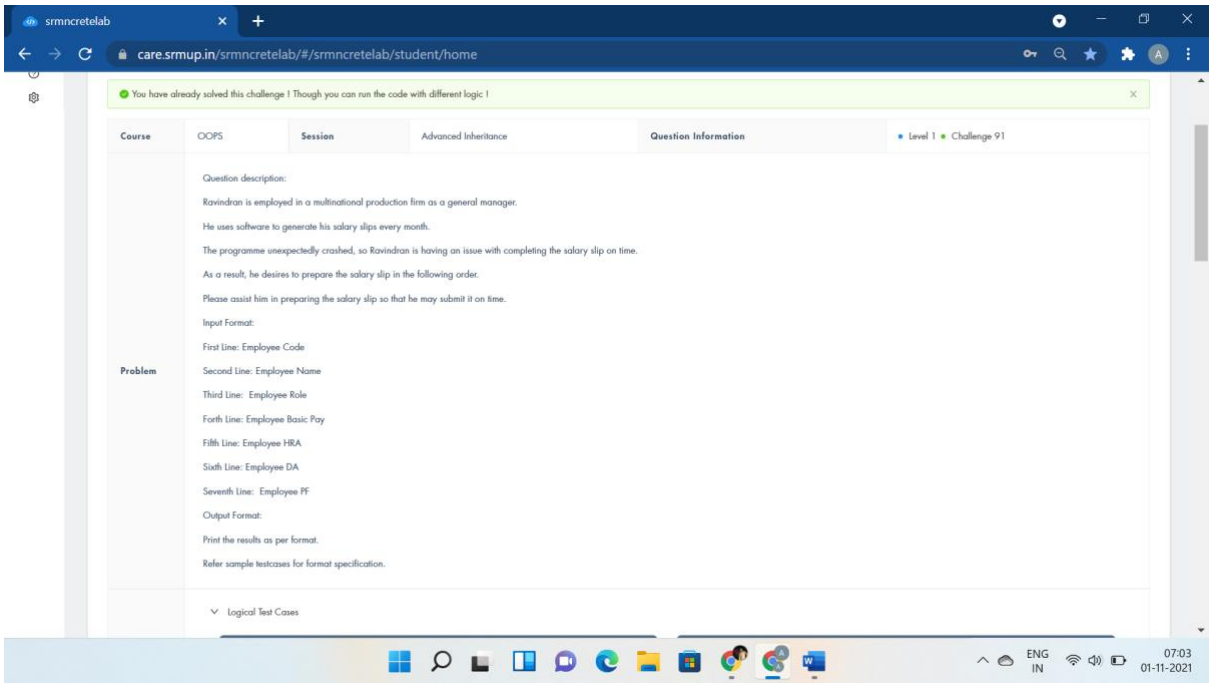
```
            }
        }
        cout << "NO";
        return 0;
        cout<<"map<long long,long long>palm; ";
}
```




```
#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;
```

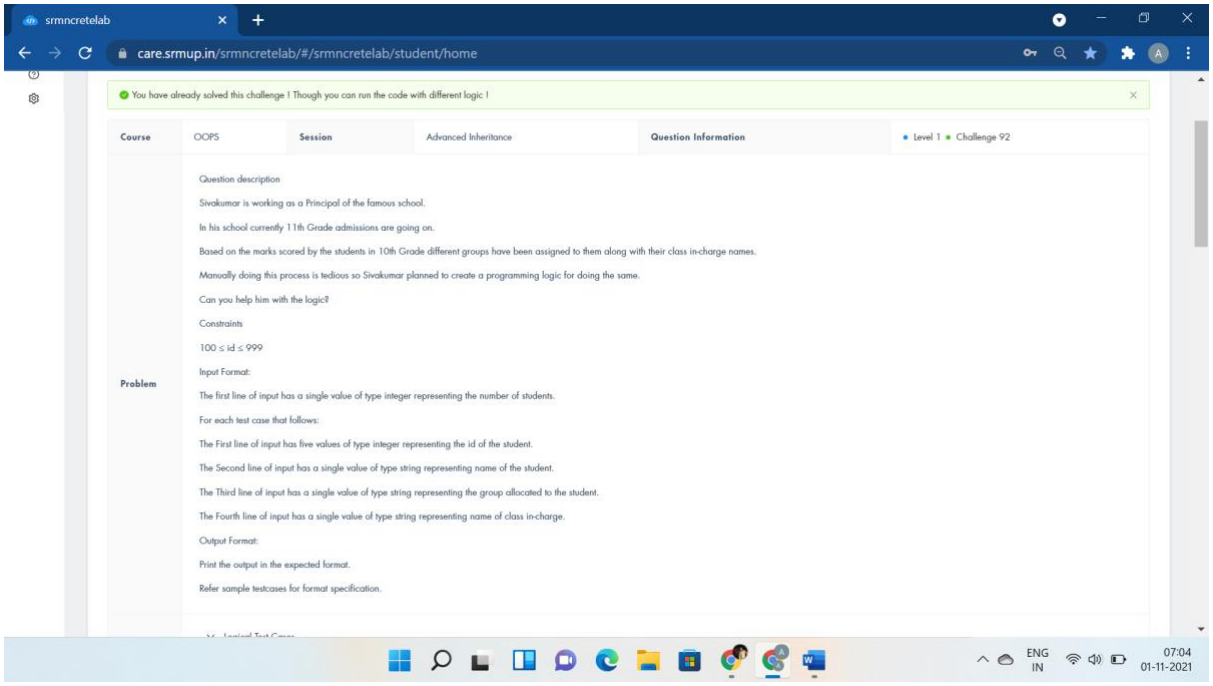
```
        }
        if(u.empty())cout<<s;
        else cout<<u;
        return 0;
        cout<<"unordered_map<string,string>website; map<string,bool>searchlist; cin>>n;";
}
```

Advanced Inheritance:-



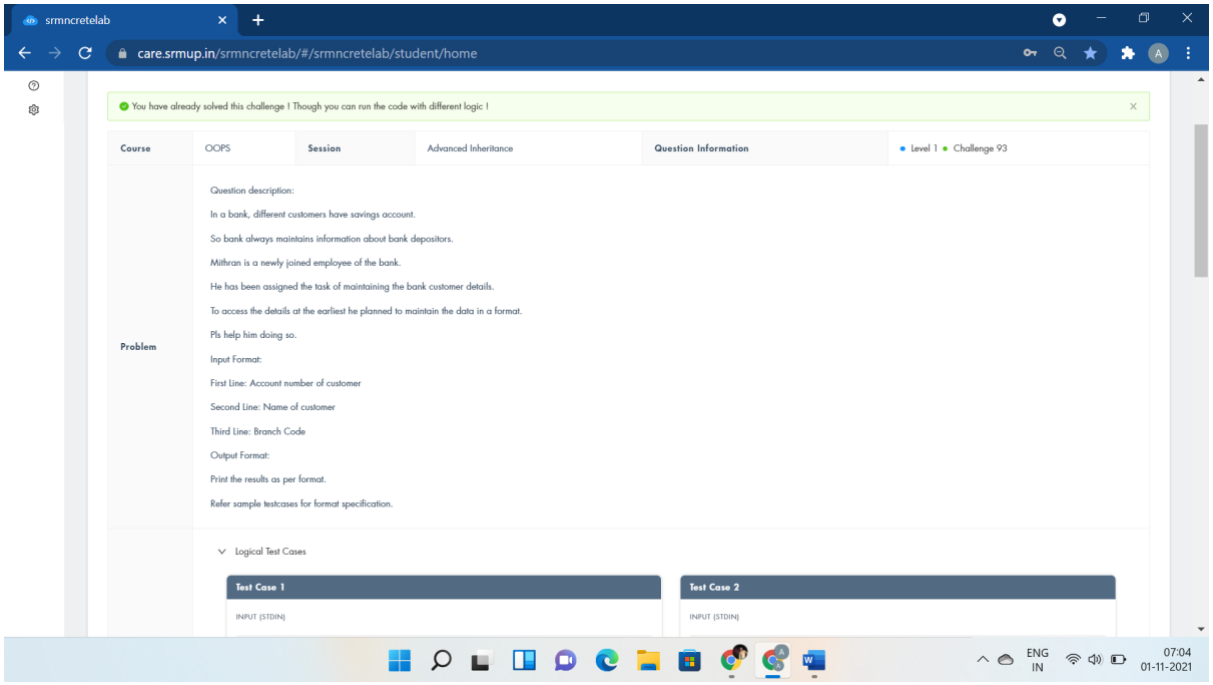

```
#include <iostream>
using namespace std;
class Employee{
   public:
};
class Salary : public Employee{
   public:
   int code,basic,hra,da,pf,total;
   string name,position;
   void getEmpDetails(){
```

```
    cin>>code>>name>>position;
  }
  void getPayDetails(){
    cin>>basic>>hra>>da>>pf;
  }
  void calculate(){
    total=basic+hra+da-pf;
  }
  void display(){
    cout<<"Employee Number:"<<code<<endl;
    cout<<"Employee Name:"<<name<<endl;
    cout<<"Employee Role:"<<position<<endl;
    cout<<"Employee Net Pay:"<<total<<endl;
  }
};
int main()
{
  Salary s;
  s.getEmpDetails();
  s.getPayDetails();
  s.calculate();
  s.display();
        return 0;
}
```



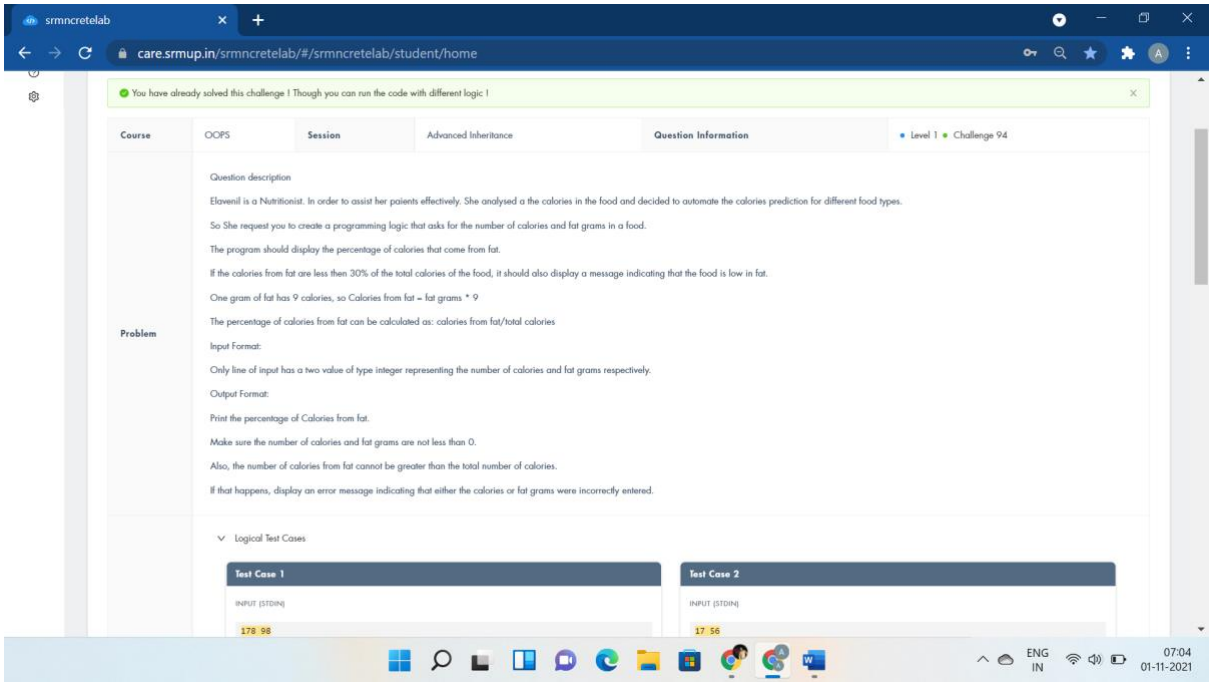

```
#include <iostream>
using namespace std;
class Person{
};
class Teaching : public Person{
};
class Instructor : public Teaching{
   public:
   int id;
   string name,group,staff;
   void accept_instructor_details(){
      cin>>id>>name>>group>>staff;
   }
   void display_instructor_details(){
      cout<<"Id:"<<id<<endl;
      cout<<"Name:"<<name<<endl;
      cout<<"Group:"<<group<<endl;
      cout<<"Staff:"<<staff<<endl;
   }
};
int main()
{
   int n;
```

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



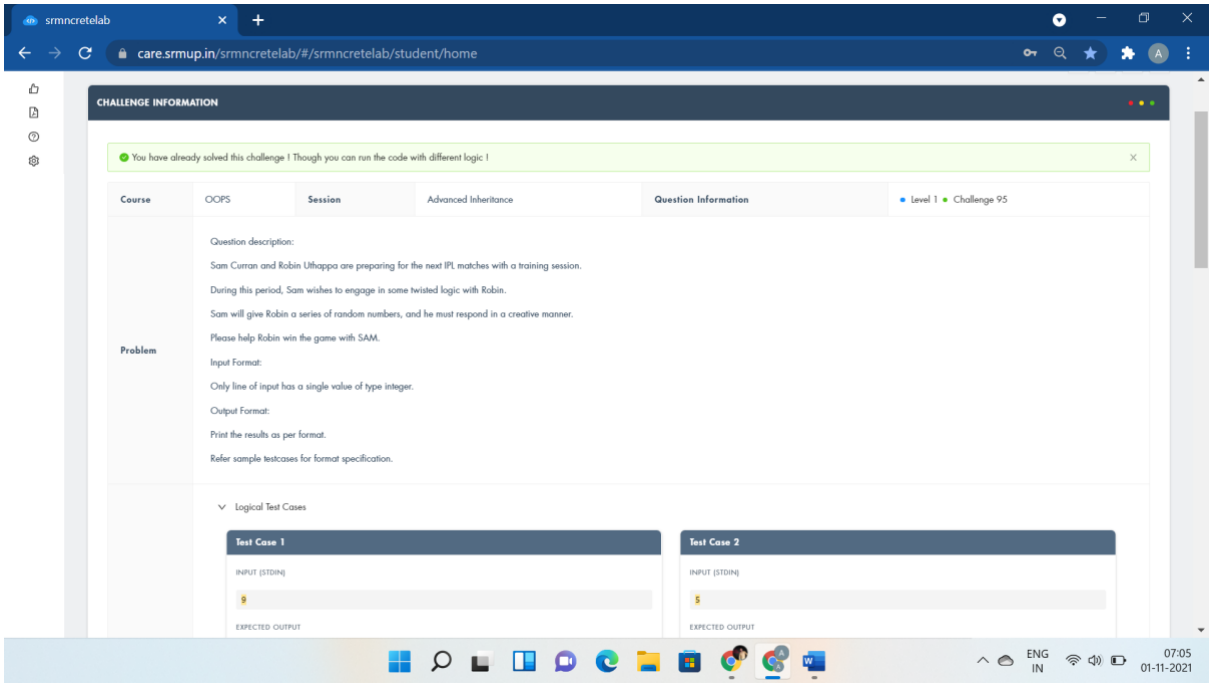

```
#include <iostream>
using namespace std;
class acc{
    public:
    int no;
    void getacc(){
        cin>>no;
    }
};
class branch:public acc{
    public:
    string name;
```

```
    int code;
    void getbranch(){
        cin>>name>>code;
    }
    void display(){
        cout<<"Acc No:"<<no<<endl;
        cout<<"Name:"<<name<<endl;
        cout<<"Branch Code:"<<code<<endl;
    }
};
int main()
{
    branch b;
    b.getacc();
    b.getbranch();
    b.display();
            return 0;
}
```



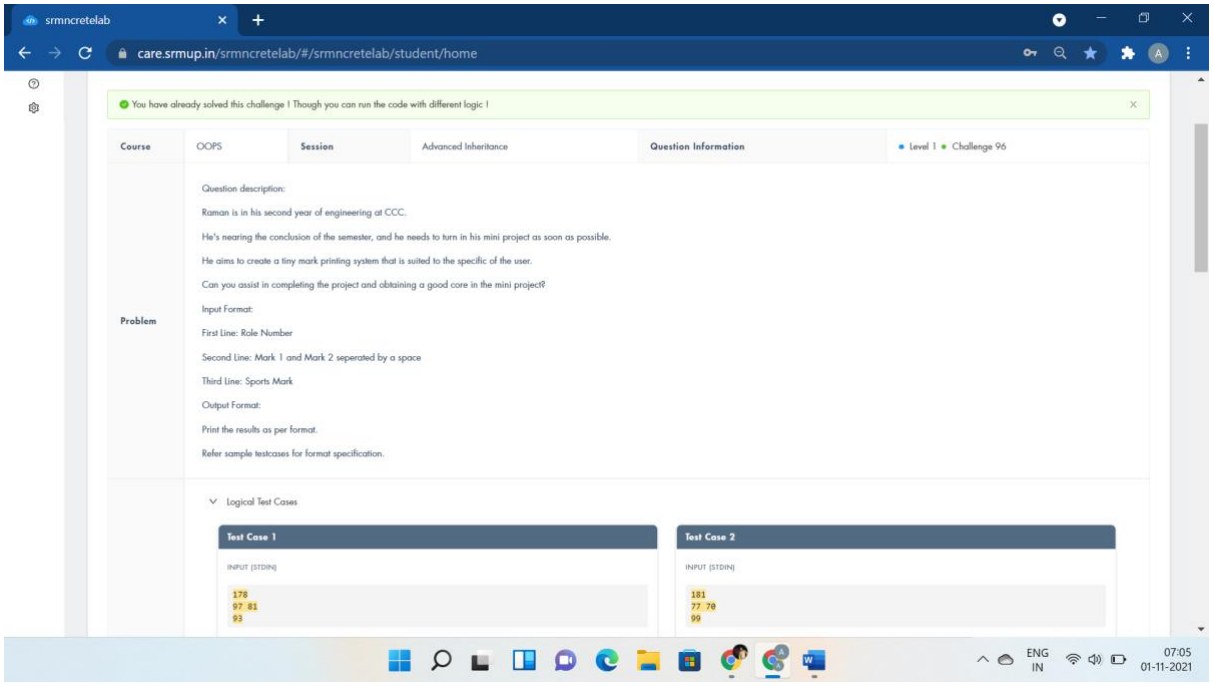

```
#include <iostream>
using namespace std;
class Food{
};
```

```
class Nutritionist:public Food{

};

class Patient:public Nutritionist{

  public:

  float cal,fat;

  void calorie(){

    cin>>cal>>fat;

  }

  void dplan(){

    if(cal<fat)

    cout<<"Fatgrams cannot be less than 0 or greater than calories"<<endl;

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

  }

};

int main()

{

  Patient p;

  p.calorie();

  p.dplan();

          return 0;

}
```



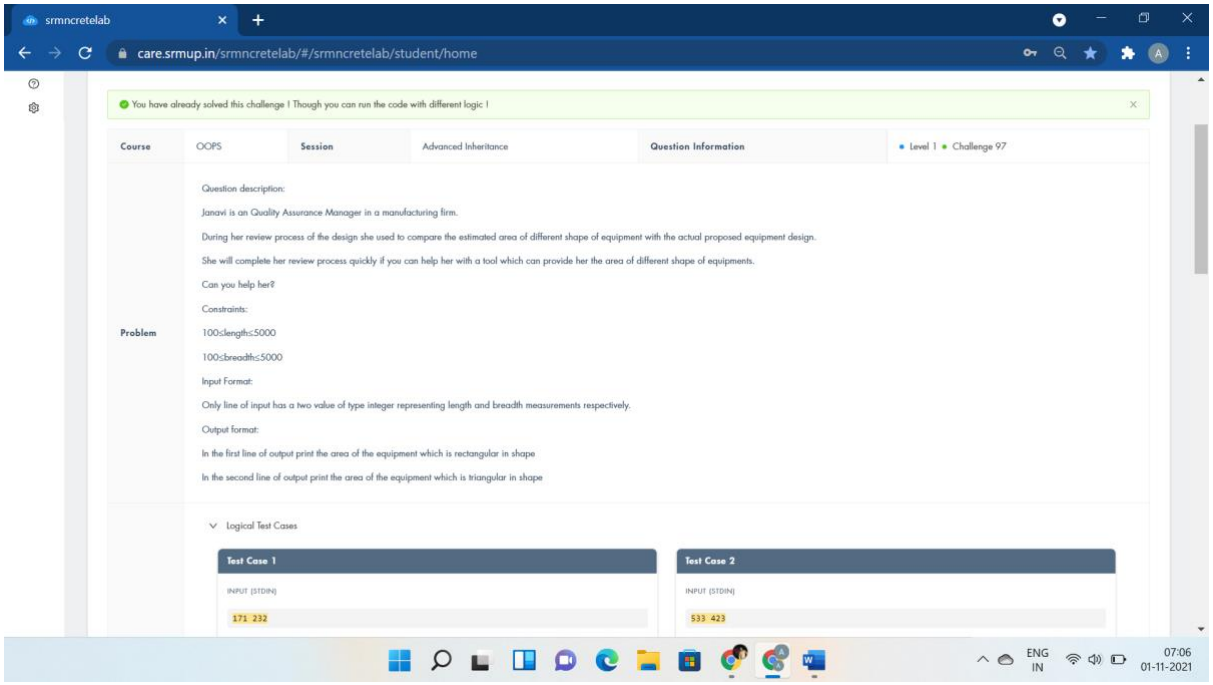

```
#include <iostream>
```

```
using namespace std;
class Sam{
};
class Robin:public Sam{
   public:
   int rows;
   void read(int y){
      rows=y;
   }
   void display(){
      for(int i=0;i<rows;i++){
         for(int j=0;j<rows;j++){
            cout<<"* ";
         }
         cout<<endl;
      }
   }
};
int main()
{
   Robin obj;
   int y;
   cin>>y;
   obj.read(y);
   obj.display();
               return 0;
}
```



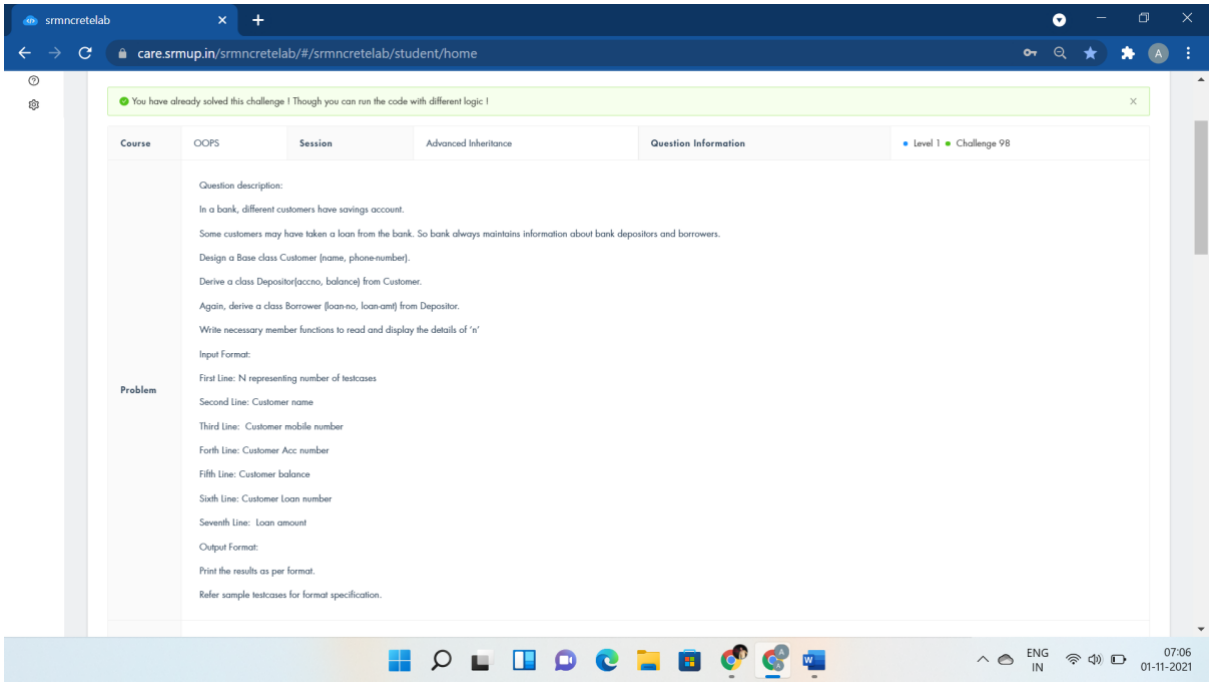

```
#include <iostream>
using namespace std;
class student{
  public:
  int roll,m1,m2;
  void get(){
    cin>>roll>>m1>>m2;
  }
};
class sports{
  public:
  int sp;
  void getsm(){
    cin>>sp;
  }
};
class statement : public student, public sports{
  public:
  void display(){
    cout<<"Roll No:"<<roll<<endl;
    cout<<"Total:"<<m1+m2+sp<<endl;
    cout<<"Average:"<<(m1+m2+sp)/3<<endl;
  }
```

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



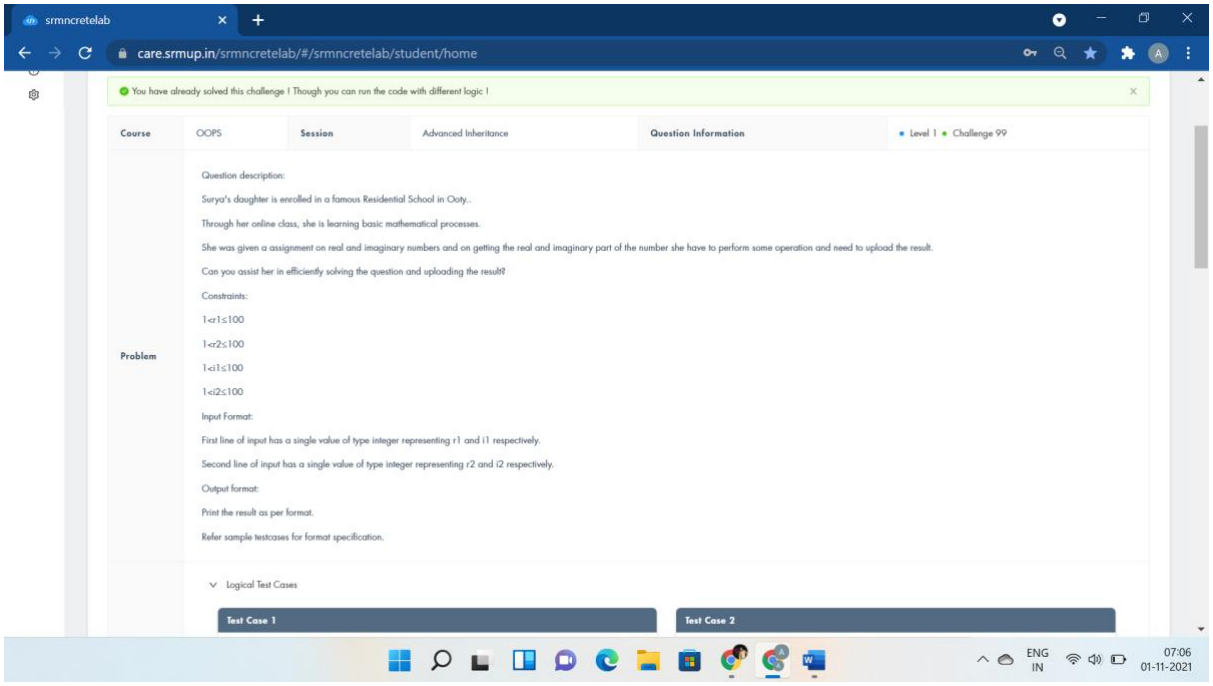

```
#include <iostream>
using namespace std;
class Shape{
   public:
   int len,wid;
   void input(int l,int b){
      len=l;
      wid=b;
   }
};
class Rectangle: public Shape{
   public:
```

```
   void output(){
      cout<<len*wid<<endl;
   }
};
class Triangle: public Shape{
   public:
   void output(){
      //if((len*wid)%2==0)
      cout<<0.5*len*wid<<endl;
      //else
      //cout<<len*wid/2+1<<endl;
   }
};
int main()
{
   int l,b;
   cin>>l>>b;
   Rectangle rect;
   Triangle tri;
   rect.input(l,b);
   tri.input(l,b);
   rect.output();
   tri.output();
            return 0;
}
```



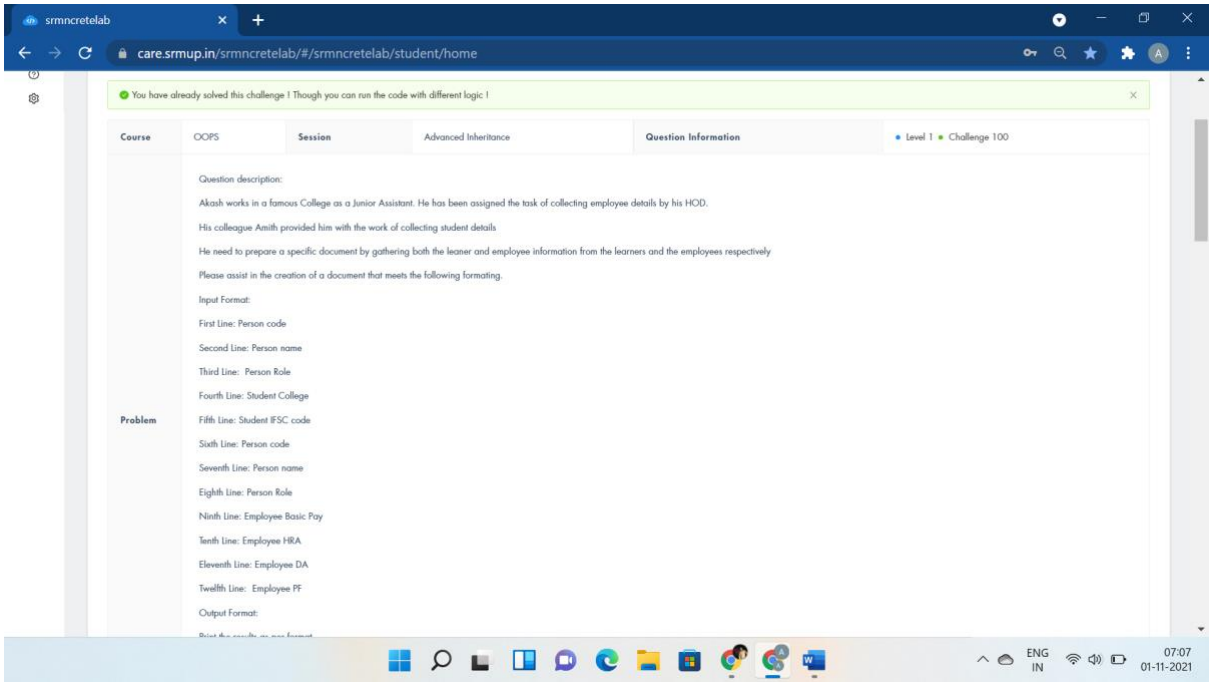

```
#include <iostream>
using namespace std;
class customer{
  public:
  int no;
  long long int mobile;
  string name;
  void acceptc(){
    cin>>name>>mobile>>no;
  }
};
class deposit:public customer{
  public:
  int bal;
  void acceptd(){
    cin>>bal;
  }
  void dispd(){
    cout<<"Customer Name:"<<name<<endl;
    cout<<"Customer Phone No:"<<mobile<<endl;
    cout<<"Customer A/c No:"<<no<<endl;
    cout<<"Balance:"<<bal<<endl;
  }
```

```
};
class borrow:public deposit{
   public:
   long long int loan_no,amt;
   void acceptb(){
      cin>>loan_no>>amt;
   }
   void dispb(){
      cout<<"Loan No:"<<loan_no<<endl;
      cout<<"Loan Amount:"<<amt<<endl;
   }
};
int main()
{
   int n;
   cin>>n;
   borrow b1[n];
   for(int i=0;i<n;i++){
      b1[i].acceptc();
      b1[i].acceptd();
      b1[i].acceptb();
      b1[i].dispd();
      b1[i].dispb();
   }
            return 0;
}
```




```
#include <iostream>
using namespace std;
class Receive{
   public:
   int r1,i1,r2,i2,r3,i3;
   void getdata(){
      cin>>r1>>i1>>r2>>i2;
   }
};
class Operate : public Receive{
   public:
   void add(){
      r3=r1+r2;
      i3=i1+i2;
   }
};
class Present :public Operate{
   public:
   void output(){
      cout<<r1<<"+"<<i1<<"i"<<endl;
      cout<<r2<<"+"<<i2<<"i"<<endl;
      cout<<r3<<"+"<<i3<<"i"<<endl;
   }
```

```
};
int main()
{
   Present calc;
   calc.getdata();
   calc.add();
   calc.output();
            return 0;
}
```




```
#include <iostream>
using namespace std;
class Person{
};
class Employee : private Person{
};
class Student : private Person{
   public:
   int n1,n2,basic,hra,da,pf;
   string name1,role1,col,ifsc,name2,role2;
   void getdetail(){
      cin>>n1>>name1>>role1>>col>>ifsc>>n2>>name2>>role2;
   }
```

```
    void getEmployeeDetails(){
        cin>>basic>>hra>>da>>pf;
    }
    void student_display(){
        cout<<"Person number:"<<n1<<endl;
        cout<<"Person name:"<<name1<<endl;
        cout<<"Person Role:"<<role1<<endl;
        cout<<"Student college Name:"<<col<<endl;
        cout<<"Student IFSC:"<<ifsc<<endl;
        cout<<"Person number:"<<n2<<endl;
        cout<<"Person name:"<<name2<<endl;
        cout<<"Person Role:"<<role2<<endl;
    }
    void employee_display(){
        cout<<"Employee Basic pay:"<<basic<<endl;
        cout<<"Employee HRA:"<<hra<<endl;
        cout<<"Employee DA:"<<da<<endl;
        cout<<"Employee PF:"<<pf<<endl;
        cout<<"Employee Net Pay:"<<basic+hra+da-pf<<endl;
    }
};
int main()
{
    Student e;
    e.getdetail();
    e.getEmployeeDetails();
    e.student_display();
    e.employee_display();
                return 0;
                cout<<"s.student_display();";
}
```