

## Print Pattern

	Question Description
<pre> 9      public class Program //DO NOT change the class 10     { 11         public static void Main(string[] args) // 12         {   int num1=1;           595 13             int num2=1; 14             //Implement your code here 15             //taking one variable run till 2 itera 16             //reseting the second var when 2 17             do 18             {           595 19                 Console.WriteLine(num1+" "+num2); 20                 num2++; 21                 if (num2&gt;2) 22                 { 23                     num1++; 24                     num2=1;           595 25                 } 26             } 27             while(num1&lt;=5); 28         } 29     } </pre>	<p><b>Sample Output:</b></p> <pre> 1 1 1 2 2 1 2 2 3 1 3 2 4 1 4 2 5 1 </pre>

## Palindrome

//palandrome	
<pre> int r,sum=0,temp; temp=a; 95 while(a&gt;0) {      r=a%10;     sum=(sum*10)+r;     a=a/10; }           595 if(temp==sum) {     Console.WriteLine("Palindrome"); } </pre>	

## Validate string

```

//Implement your code here
Console.WriteLine("Enter the names:");
name=Console.ReadLine();
string[] namecoll=name.Split(",");

for(i=0;i<namecoll.Length;i++)
{
    name=namecoll[i].Replace(" ", "");
    if(char.IsLetter(namecoll[i],7))
    {
        Console.WriteLine(name+" => Veri
    }
    else
    {
        Console.WriteLine(name+" => Inva

```

#### Sample Output 1:

```

LiamSmith => Verified
NoahJohnson => Verified
OliviaGarcia => Verified
OliverWilliams => Verified
AvaDavis => Verified
ElijahJones => Verified
EmmaBro3n => Invalid det

```

Change odd place.

```

string tempchar="";
bool flag=false;
//Implement your code here
Console.WriteLine("Enter the string:");
input=Console.ReadLine();
for(int i=0;i<input.Length;i++)
{
    char temp=input[i];
    if(!char.IsLetter(temp))
    {
        flag=true;
        Console.WriteLine("Invalid input");
        break;
    }
    else if((i+1)%2!=0)
    {
        if(temp=='z')
        {
            tempchar+='a';
        }
        else if(temp=='Z')
        {
            tempchar+='A';
        }
        else
        {
            temp++;
            tempchar+=temp;
        }
    }
}

```

#### Question Description

Enter the string.

#### Curiosity

#### Sample Output 1:

```

CuriosityDusipsjtz

```

#### Sample Input 2:

Enter the string:

```

H3llo

```

#### Sample Output 2:

Invalid input

Replace in string

```

Console.WriteLine("Enter the character to be searched:");
char srch = Console.ReadLine()[0];
//replace first occurrence
Console.WriteLine("Enter the character to replace:");
char replace = Console.ReadLine()[0];

int flag = 0;
int len = str.Length;
char[] words = new char[len];
words = str.ToCharArray();
for (int i = 0; i < len; i++)
{
    char r = str[i];
    if (r == srch)
    {
        words[i] = replace;
        flag=1;
        break;
    }
}
if (flag == 0)
{
    Console.WriteLine("character not found");
}
else
{
    foreach (char i in words)
    {
        Console.Write(i);
    }
}

```

#### Question

Enter the  
o

#### Sample c

ootnet pr

#### Sample i

Enter the  
**Apple**  
Enter the  
d

Enter the  
o

#### Sample c

```

string[] applicantName=new string[num];
Console.WriteLine("Enter the applicant names:");
for(i=0;i<num;i++)
{
    applicantName[i]=Console.ReadLine();
}

for(i=0;i<num;i++)
{
    int length=applicantName[i].Length;
    string tempname=applicantName[i];
    flagnotverify=0;
    for(int j=0;j<length;j++)
    {
        char temp=tempname[j];
        if(!char.IsLetter(temp) & !char.IsWhiteSpace(temp))
        {
            flagnotverify++;
            break;
        }
    }
    if(flagnotverify>0)
    {
        notverified++;
    }
    else
    {
        verified++;
    }
}

Console.WriteLine("Verified = {0}\n\nNot verified = {1}",verified,notverified);

```

Question Description

Enter the applicant r

Liam Smith

Noah Johns\n

Olivia Garcia

Oliver William\$

Ava Davis

Sample Output 1:

Verified = 3

Not verified = 2

### Digit multiply

```

public static void Main(string[] args) //DO NOT change the method signature
{
    int Num1,Num2;
    //Implement your code here
    Console.WriteLine("Enter the input 1:");
    Num1=int.Parse(Console.ReadLine());
    Console.WriteLine("ENter the input 2:");
    Num2=int.Parse(Console.ReadLine());

    if(Num1.ToString().Length==Num2.ToString().Length)
    {
        int result=0;
        while(Num1>0)
        {
            result+=(Num1%10) * (Num2%10);
            Num1 /=10;
            Num2 /=10;
        }
        Console.WriteLine(result);
    }
    else
    {
        Console.WriteLine("Invalid inputs");
    }
}

```

Question Description

521      350 => 2

521      350 => 1

Sample Input 2:

Enter the input 1:

51

Enter the input 2:

250

Sample Output 2

Invalid inputs

### Sum Positive

```

int[] arr = new int[s];
Console.WriteLine("Enter the elements:");
for (int i = 0; i < s; i++)
{
    arr[i] = Convert.ToInt32(Console.ReadLine());
}
int sum = 0;

for (int i = 0; i < s; i++)
{
    if(arr[i] < 0)
    {
        continue;
    }
    else if (arr[i] >= 0 & arr[i] <= 10)
    {
        int f=1;
        for(int j = 1; j <= arr[i]; j++)
        {
            f *= j;
        }

        sum+=f;
    }
}

if (sum == 0)
{
    Console.WriteLine("No positive and single digit n

```

#### Question D

Example if  
be **122**

#### Sample Input

Enter the s

**5**

Enter the e

**2**

**-56**

**-13**

**6**

**56**

Property Get;set;

```

double balance;

//Declare the Properties and methods here
string AccountNumber{get; set;}
string AccountName{get; set;}
public double Balance
{
    set{balance=value;}
    get{return balance;}
}

public string RegisterAccount(string accountNumber,string accountName,double balance)
{
    this.AccountNumber=accountNumber;
    this.AccountName=accountName;
    this.Balance=balance;
    return "Account registered";
}

public double DepositCash(Program obj,double bal)
{
    obj.Balance+=bal;
    return obj.Balance;
}

public double WithDrawCash(Program obj,double cash)
{
    if(cash<=this.balance)
    this.balance =this.balance - cash;
    return this.balance;
}

```

```

public class Account
{
    int id;
    string accountType;
    double balance;
    //setting value using setter
    public int Id
    {
        set{id=value;}
        get{return id;}
    }

    public string AccountType
    {
        set{accountType =value;}
        get{return accountType;}
    }

    public double Balance
    {
        set{balance=value;}
        get{return balance;}
    }

    public Account()
    {
    }
}

```

```

public Account(int id,string accountType,double balance)
{
    this.id=id;
    this.accountType=accountType;
    this.balance=balance;
}

public bool WithDraw(double amount)
{
    if (amount<=this.balance)
    {
        return true;
    }
    else
    {
        return false;
    }
}

```

```

public string GetDetails()
{
    Account acc= new Account();
    if (acc.WithDraw(this.balance)==true)
    {
        string a=("Account Id: "+this.id + "\nAccount Type: "+this.accountType+ "\nBalance: "+this.balance);
        return a;
    }
    else
    {
        return "balance error";
    }
}

}

public class Program
{
    public static void Main()
    {
        int accountId;
        string accountType;
        double balance;
        Console.WriteLine("Enter account id");
        accountId=Convert.ToInt32(Console.ReadLine());
        Console.WriteLine("Enter account type");
        accountType=Console.ReadLine();
        Console.WriteLine("Enter account balance");
        balance=Convert.ToDouble(Console.ReadLine());
        Account obj =new Account(accountId,accountType,balance);
        Console.WriteLine(obj.GetDetails());
    }
}

```

## Inheritance

```

public static void Main(string[] args) //DO NOT change the method
{
    //Implement your code here
    double height;
    double width;
    Console.WriteLine("Enter the Parameter1");
    height=Double.Parse(Console.ReadLine());
    Console.WriteLine("Enter the Parameter2");
    width= Double.Parse(Console.ReadLine());
    Triangle t=new Triangle();
    t.Para(height,width);
    Console.WriteLine("Area of triangle: "+t.AreaOfTriangle());
    Console.WriteLine("Area of rectangle: "+t.AreaOfRectangle());
}

```

```

public class Parameters //DO NOT change
{
    public double Parameter1{get;set;}
    public double Parameter2{get;set;}

    public void Para(double P1,double P2)
    {
        //Implement your code here
        this.Parameter1=P1;
        this.Parameter2=P2;
    }
}

class Triangle : Parameters
{
    public double AreaOfTriangle()
    {
        //Implement your code here
        double area=(Parameter1*Parameter2)/2;
        return area;
    }

    public double AreaOfRectangle()
    {
        double area=(Parameter1*Parameter2);
        return area;
    }
}

```

## Abstract class



```

public static void Main(string[] args) //DO NOT change
{
    //Implement your code here
    int choice=int.Parse(Console.ReadLine());
    double amount=Double.Parse(Console.ReadLine());
    //abstract class
    Restaurant customer;

    if(choice ==1)
    {
        customer=new StandardCustomer();
    }
    else if(choice==2)
    {
        customer=new VipCustomer();
    }
    else
    {
        Console.WriteLine("Invalid choice");
        return;
    }
    double billAmount=customer.Billing(amount);
    Console.WriteLine(billAmount);
}

```

```

public class VipCustomer:Restaurant //DO NOT change
{
    //Implement your code here
    public override double Billing(double amount)
    {
        return amount*0.7;
    }
}

```

```

public class StandardCustomer:Restaurant //DO NOT
{
    //Implement your code here
    public override double Billing(double amount)
    {
        return amount*0.9;
    }
}

```

```

//-----
public abstract class Restaurant
{
    public abstract double Billing(double amount);
}

```



## Interface Example

```
public class Program //DO NOT change the clas
{
    public static void Main(string[] args) /
    {
        //Implement your code here
        Polygon p=new Polygon();
        Hexagon h=new Hexagon();
        Console.WriteLine(p.CreateShape());
        Console.WriteLine(h.CreateShape());
    }
}

class Polygon:Shapes
{
    public string CreateShape()
    {
        return "Polygon created";
    }
}

//Implement your code here
interface Shapes
{
    string CreateShape();
}
```

openableInterface

```
public interface IOpenable
{
    string OpenSesame();
}
public class TreasureBox:IOpenable
{
    public string OpenSesame() 595
    {
        string str="Congratulations , Here is your lucky win";
        return str;
    }
}
public class Parachute:IOpenable
{
    public string OpenSesame()
    {
        string str="Have a thrilling experience flying in air";
        return str;
    }
}
public class Program 595 595
{
    public static void Main()
    {
        string input =Console.ReadLine();
        IOpenable fortune;
        if(input=="T")
        {
            fortune=new TreasureBox();
        }
        else if(input=="P")
        {
            fortune = new Parachute(); 595
        }
        else
        {
            Console.WriteLine("Invalid input.Please enter T or P");
            return;
        }
        Console.WriteLine(fortune.OpenSesame());
    }
}
```

File read write

```
public class Program //DO NOT change the class name
{
    public void WriteOnFile(string fileName,string text)
    {
        using (StreamWriter writer = new StreamWriter(fileName))
        {
            writer.Write(text); 595
        }
    }

    public string[] ReadFile(string fileName)
    {
        string[] value=File.ReadAllLines(fileName);
        return value; 595
    }

    public static void Main(string[] args) //DO NOT change the
    {
        //Implement your code here
        int choice;
        string fileName; 595
        Console.WriteLine("1. Write on file");
        Console.WriteLine("2. Read file");
        choice=Convert.ToInt32(Console.ReadLine());

        Program obj=new Program(); 595
        if (choice ==1)
        {
            Console.WriteLine("Enter the filename");
            fileName=Console.ReadLine();
            Console.WriteLine("Enter the text to write");
            string text=Console.ReadLine();
            obj.WriteOnFile(fileName,text); 595
        }
        else if(choice ==2)
        {
            Console.WriteLine("Enter the file to read");
            fileName=Console.ReadLine();
            string [] fileContent=obj.ReadFile(fileName);
            foreach (string line in fileContent) 595
            {
                Console.WriteLine(line);
            }
        }
    }
}
```

## Append file

```
public class Program //DO NOT change the class name
{
    public static void Main(string[] args) //DO NOT chang
    {
        //Implement your code here
        Console.WriteLine("Enter the file name");
        string fileName=Console.ReadLine();
        Console.WriteLine("Enter the text to append");
        string text=Console.ReadLine();
        //append
        Program obj=new Program();
        obj.AppendFile(fileName,text);
    }

    public void AppendFile(string fileName,string text)
    {
        using(StreamWriter ap=File.AppendText(fileName))
        {
            ap.WriteLine(text);
        }
    }
}
```

## Write file

```
Console.WriteLine("Enter the Insurance number : ");
insNum=int.Parse(Console.ReadLine());
Console.WriteLine("ENter the insurance holder name:");
name=Console.ReadLine();
policyName=Console.ReadLine();
amount=Double.Parse(Console.ReadLine());

using (StreamWriter rt =new StreamWriter("insurance.txt"))
{
    rt.WriteLine(insNum+"-"+name+"-"+policyName+"-"+amount);
}
Console.WriteLine(File.ReadAllText("insurance.txt"));
}
```

List

```
public class Lottery //DO NOT change the class name
{
    //Implement code here
    public int LotteryNumber{get;set;}
    public string LotteryType{get;set;}
    public int Cost{get;set;}
    public float PrizeAmount{get;set;}
    public string LotteryName{get;set;}

    public Lottery()
    {
    }

    public Lottery(int num,string type,int cost,float prize,string lname)
    {
        LotteryNumber=num;
        LotteryType=type;
        Cost=cost;
        PrizeAmount=prize;
        LotteryName=lname;
    }
}

public class Program //DO NOT change the class name
{
    public static void Main(string[] args)
    {
        //Implement code here
        string type=Console.ReadLine();
        LotteryManagement.Query(type);
        LotteryManagement.DisplayLotteryDetails(LotteryManagement.ResultList);
    }
}
```

```

public class LotteryManagement //DO NOT change the class name
{
    //Use the below List to query 'Lottery Type' based on the pattern
    public static List<Lottery> AvailableLottery = new List<Lottery>()
    {
        new Lottery(113,"Scratch",20,2600000,"JumboJackpot"),
        new Lottery(114,"Ticket",40,900000,"Lotto"),
        new Lottery(115,"Online",10,750000,"Powerball"),
        new Lottery(116,"Scratch",100,7000000,"Poker"),
        new Lottery(117,"Online",50,1200000,"BumperCards")
    };

    //Declare a action delegate 'Query' and add handler 'SearchByType'

    //Use the below List for adding filtered objects
    public static List<Lottery> ResultList = new List<Lottery>();

    //Implement Code Here
    public static Action<string>Query=SearchByType;
    public static void SearchByType(string pattern)
    {
        ResultList.Clear();
        foreach(var lottery in AvailableLottery)
        {
            if(lottery.LotteryType==pattern)
            {
                ResultList.Add(lottery);
            }
        }
    }
    public static void DisplayLotteryDetails(List<Lottery> lotterylist)
    {
        Console.WriteLine("\n Lottery Details:\n");

        foreach(var lottery in lotterylist)
        {
            Console.WriteLine("Lottery Name:"+lottery.LotteryName+"Lotter:
        }
    }
}

```

```
public class Program //DO NOT CHANGE THE CLASS NAME
{
    public delegate double MyDelegate();
    public static event MyDelegate DeleEvent = null;
    public static double595 amount = 0;
    static void Main(string[] args)
    {
        Console.WriteLine("Enter the Amount");
        amount = Convert.ToDouble(Console.ReadLine());
        CreateEvent();
        double amt = DeleEvent.Invoke();595
        Console.WriteLine("Discount Amount:" + amt);
    }

    public static void CreateEvent()595
    {
        DeleEvent = new MyDelegate(Calculation);
    }

    public static double Calculation()
    {
        if (amount < 75000)
        {
```



Try catch

```
public class Program //DO NOT change the class name
{
    public static void Main(string[] args) //DO NOT change the method signature
    {
        //Implement your code here
        float marks;
        try
        {
            marks=float.Parse(Console.ReadLine());
            Console.WriteLine("Result:"+marks+"%");
        }
        catch(Exception e)
        {
            Console.WriteLine(e);
        }
    }
}
```

## Custom exception

```
public class Program
{
    //Implement your code here
    public string CheckVotersAge(int age)
    {
        if(age>=18)
        {
            return "Voting is eligible";
        }
        else
        {
            throw new InvalidAgeException("Voting is not
        }

    public static void Main(string[] args)
    {
        Console.WriteLine("Enter the age:");
        try
        {
            Program obj =new Program();
            int age=int.Parse(Console.ReadLine());
            Console.WriteLine(obj.CheckVotersAge(age));
        }
        catch(InvalidAgeException e)
        {
            Console.WriteLine(e.Message);
        }
    }
}

{ //custom exception.
    //Implement your code here
    public class InvalidAgeException:Exception
    {
        public InvalidAgeException(string message):base(message)
        {
        }
    }
}
```

## Array List

```
namespace Array_List //DO NOT change the namespace name
{
    public class Program //DO NOT change the class name
    {
        public static ArrayList Attendance = new ArrayList()
        {
            "John",
            "Peter",
            "Jacob",
            "Archie",
            "Sophie",
            "Veronica",
            "Elizabeth",
            "Charles"
        };

        public bool RemoveStudent(string name)
        {
            int prelen=Program.Attendance.Count;
            Console.WriteLine(name);
            Program.Attendance.Remove(name);
            int afterlen=Program.Attendance.Count;
            if(prelen!=afterlen)
            {
                return true;
            }
            else
            {
                return false;
            }
        }
    }
}
```

```

public void SortTheAttendance()
{
    Program.Attendance.Sort();
    foreach(string element in (Program.Attendance))
    {
        Console.WriteLine(element);
    }
}

//Implement the methods here
public static void Main(string[] args) //DO NOT change th
{
    //Implement your code here
    int choice;
    choice=int.Parse(Console.ReadLine());
    Program obj=new Program();
    if(choice==1)
    {
        string name=Console.ReadLine();
        bool op=obj.RemoveStudent(name);
        if(op==true)
        {
            Console.WriteLine("Removed successfully");
        }
        else
        {
            Console.WriteLine("Not removed");
        }
    }
}

```