1)**AIM:**

To print the C# program to print all elements in the Two-Dimensional array.

PROGRAM:

```
        using System;

class Program
{
  static void Main()
  {
    int[,] array = { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 } };

    bool isFirstElement = true;

    for (int I = 0; I < array.GetLength(0); i++)
    {
      for (int j = 0; j < array.GetLength(1); j++)
      {
        if (!isFirstElement)
        {
          Console.Write(“ “);
        }

        Console.Write(array[I, j]);

        isFirstElement = false;
      }
    }

    Console.WriteLine();
  }
}
```

OUTPUT:

```
1 2 3 4 5 6 7 8 9
```

2)AIM:

To create C# program for check the eligibility.

PROGRAM:

```
using System;
class EligibilityCheck
{
static void Main()
  {
      Console.Write("Enter your age: ");
      if (!int.TryParse(Console.ReadLine(), out int age))
      {
        Console.WriteLine("Invalid input for age. Please enter a valid number.");
        return;
      }
      Console.Write("Enter your pass percentage: ");
      if (!double.TryParse(Console.ReadLine(), out double passPercentage))
      {
        Console.WriteLine("Invalid input for pass percentage. Please enter a valid number.");
        return;
      }
      if (age > 18 && age <= 30 && passPercentage >= 65)
      {
        Console.WriteLine("You are eligible.");
      }
      else
      {
        Console.WriteLine("You are not eligible.");
      }
  }}
```

OUTPUT:

```
Enter your age: 20

Enter your pass percentage: 67

You are eligible.
```

3)AIM:

To create a C# program to check the mobile number is validate or not.

PROGRAM:

```
using System;

using System.Text.RegularExpressions;


class Validation

{

   static void Main()

   {

      Console.Write("Enter mobile number: ");

      string mobileNumber = Console.ReadLine();


      if (Regex.IsMatch(mobileNumber, @"^[98]\d{3}-\d{6}$"))

      {

         Console.WriteLine("Valid mobile number");

      }

      else

      {

         Console.WriteLine("Invalid mobile number");

      }

   }

}
```

OUTPUT:

```
Enter mobile number: 9344-702259

Valid mobile number
```

4)AIM:

To create a C# program to find missing snippets and the statements.

PROGRAM:

```
using System;
class Person
{
   public string Name { get; set; }
   public int Age { get; set; }
   public double Weight { get; set; }

   public void PrintPerson()
   {
      Console.WriteLine($"Name: {Name}, Age: {Age}, Weight: {Weight}");
   }
}

class PersonData
{
   static void Main(string[] args)
   {
      Person person = new Person
      {
         Name = "Kannan",
         Age = 19,
         Weight = 65.5
      };

      person.PrintPerson();
   }
}
```

OUTPUT:

```
Name: Kannan, Age: 19, Weight: 65.5
```

5)AIM:

To create a C# program to create a console application to inpatient.

PROGRAM:

```
using System;

class Patient

{

   public string Name { get; set; }

   public DateTime DateOfAdmission { get; set; }

   public int Age { get; set; }

   public string Disease { get; set; }

   public DateTime DateOfDischarge { get; set; }

   public double TotalBillsPaid { get; set; }


   public void PrintDetails()

   {

      Console.WriteLine($"Name: {Name}, Age: {Age}, Disease: {Disease}, Total Bills Paid: {TotalBillsPaid:C}");

      Console.WriteLine($"Admission Date: {DateOfAdmission:MMMM d, yyyy}, Discharge Date: 
{DateOfDischarge:MMMM d, yyyy}");

   }

}


class Hospital

{

   static void Main()

   {

      Console.Write("Enter patient's name: ");

      string name = Console.ReadLine();


      Console.Write("Enter patient's age: ");

      int age = int.Parse(Console.ReadLine());


      Console.Write("Enter the disease: ");

      string disease = Console.ReadLine();
```

```
        Console.Write("Enter date of admission (yyyy-mm-dd): ");
        DateTime dateOfAdmission = DateTime.Parse(Console.ReadLine());

        Console.Write("Enter date of discharge (yyyy-mm-dd): ");
        DateTime dateOfDischarge = DateTime.Parse(Console.ReadLine());

        Console.Write("Enter total bills paid: ");
        double totalBillsPaid = double.Parse(Console.ReadLine());

        Patient patient = new Patient
        {
            Name = name,
            Age = age,
            Disease = disease,
            DateOfAdmission = dateOfAdmission,
            DateOfDischarge = dateOfDischarge,
            TotalBillsPaid = totalBillsPaid
        };

        patient.PrintDetails();
    }
}
```

OUTPUT:

```
Enter patient's name: Dharsan
Enter patient's age: 22
Enter the disease: Dengue
Enter date of admission (yyyy-mm-dd): 2024-10-09
Enter date of discharge (yyyy-mm-dd): 2024-10-12
Enter total bills paid: 5000
Name: Dharsan, Age: 22, Disease: Dengue, Total Bills Paid: ?5,000.00
Admission Date: October 9, 2024, Discharge Date: October 12, 2024
```

6)AIM:

To create a C# program add two vectors using operator overloading.

PROGRAM:

```
using System;

class Vector
{
    public int X { get; set; }
    public int Y { get; set; }

    public Vector(int x, int y)
    {
        X = x;
        Y = y;
    }

    public static Vector operator +(Vector a, Vector b)
    {
        return new Vector(a.X + b.X, a.Y + b.Y);
    }

    public void Display()
    {
        Console.WriteLine($"Vector: ({X}, {Y})");
    }
}

class Program
{
    static void Main(string[] args)
    {
        Vector v1 = new Vector(1, 2);
        Vector v2 = new Vector(3, 4);
        Vector v3 = v1 + v2;

        v3.Display();

    }

}
```

OUTPUT:

```
Vector: (4, 6)
```

7)AIM:

To create a C# program to a student basic details.

PROGRAM:

```
using System;
class Student
{
   public string Name { get; set; }
   public int Age { get; set; }
   public string Address { get; set; }
   public string MobileNumber { get; set; }

   public override string ToString()
   {
      return $"Name: {Name}, Age: {Age}, Address: {Address}, Mobile Number: {MobileNumber}";
   }
}
class StudentMark : Student
{
   public int Marks { get; set; }
   public override string ToString()
   {
      char grade = CalculateGrade();
      return $"{base.ToString()}, Marks: {Marks}, Grade: {grade}";
   }
   public char CalculateGrade()
   {
      if (Marks >= 90) return 'A';
      if (Marks >= 75) return 'B';
      if (Marks >= 50) return 'C';
      return 'F';
   }
}
class Program
```

```
{
    static void Main(string[] args)
    {
        StudentMark student = new StudentMark();

        Console.Write("Enter student's name: ");
        student.Name = Console.ReadLine();

        Console.Write("Enter student's age: ");
        student.Age = int.Parse(Console.ReadLine());

        Console.Write("Enter student's address: ");
        student.Address = Console.ReadLine();

        Console.Write("Enter student's mobile number: ");
        student.MobileNumber = Console.ReadLine();

        Console.Write("Enter student's marks: ");
        student.Marks = int.Parse(Console.ReadLine());

        Console.WriteLine("\nStudent Information:");
        Console.WriteLine(student.ToString());
    }
}
```

OUTPUT:

```
Enter student's name: Anjali
Enter student's age: 24
Enter student's address: 4/155 mettur,salem
Enter student's mobile number: 6732456781
Enter student's marks: 95

Student Information:
Name: Anjali, Age: 24, Address: 4/155 mettur,salem, Mobile Number: 6732456781, Marks: 95, Grade: A
```

8)AIM:

To create a C# program for employee salary.

PROGRAM:

```
using System;

class Employee
{
   public int JobCategory { get; set; }
   public int EmpNo { get; set; }
   public string EmpName { get; set; }
   public double BasicSalary { get; set; }
   public double HRA { get; set; }
   public double DA { get; set; }
   public double Loan { get; set; }
   public double PF { get; set; }

   public Employee(int jobCategory)
   {
      JobCategory = jobCategory;
      SetSalaryComponents();
   }

   private void SetSalaryComponents()
   {
      if (JobCategory == 1)
      {
         BasicSalary = 8000;
         HRA = 0.1 * BasicSalary;
         DA = 0.2 * BasicSalary;
         Loan = 300;
         PF = 500;
      }
      else if (JobCategory == 2)
      {
        BasicSalary = 15000;
        HRA = 0.2 * BasicSalary;
        DA = 0.3 * BasicSalary;
        Loan = 600;
        PF = 1000;
    }
    else
    {
        throw new ArgumentException("Invalid job category. Must be 1 or 2.");
    }
}

public void InputDetails(int empNo, string empName)
{
    EmpNo = empNo;
    EmpName = empName;
}

public double CalculateSalary()
{
    return BasicSalary + HRA + DA - (Loan + PF);
}

public void Display()
{
    Console.WriteLine($"Employee No: {EmpNo}");
    Console.WriteLine($"Employee Name: {EmpName}");
    Console.WriteLine($"Job Category: {JobCategory}");
    Console.WriteLine($"Basic Salary: {BasicSalary:C}");
    Console.WriteLine($"HRA: {HRA:C}");
    Console.WriteLine($"DA: {DA:C}");
    Console.WriteLine($"Loan Deduction: {Loan:C}");
    Console.WriteLine($"PF Deduction: {PF:C}");
    Console.WriteLine($"Net Salary: {CalculateSalary():C}");
    }
}

class Program
{
    static void Main(string[] args)
    {
        int jobCategory;
        while (true)
        {
            Console.Write("Enter Job Category (1 or 2): ");
            if (int.TryParse(Console.ReadLine(), out jobCategory) && (jobCategory == 1 || jobCategory == 2))
            {
                break;
            }
            Console.WriteLine("Invalid input. Please enter 1 or 2.");
        }

        Employee emp = new Employee(jobCategory);

        int empNo;
        while (true)
        {
            Console.Write("Enter Employee Number: ");
            if (int.TryParse(Console.ReadLine(), out empNo))
            {
                break;
            }
            Console.WriteLine("Invalid input. Please enter a valid integer.");
        }

        Console.Write("Enter Employee Name: ");
        string empName = Console.ReadLine();
        emp.InputDetails(empNo, empName);
        emp.Display();
    }
}
```

OUTPUT:

```
Enter Job Category (1 or 2): 2
Enter Employee Number: 55
Enter Employee Name: karan
Employee No: 55
Employee Name: karan
Job Category: 2
Basic Salary: 15,000.00
HRA: 3,000.00
DA: 4,500.00
Loan Deduction: 600.00
PF Deduction: 1,000.00
Net Salary: 20,900.00
```