PROGRAM – 1

AIM: Create a program that prompts the user for their name and age and prints a personalized message.

SOURCE CODE:

```
using System;
class PersonalizedMessage
{
   static void Main()
   {
      Console.WriteLine("Name:
      Roshni"); Console.WriteLine("Roll
      No: 66");
      Console.WriteLine("Class: IT-B\n");

      Console.Write("Enter your name:
      "); string name =
      Console.ReadLine();

      Console.Write("Enter your age: ");
      int age = Convert.ToInt32(Console.ReadLine());

      Console.WriteLine("Hello, " + name + "! You are " + age + " years old.");
   }
}
```

Output

```
Name: Roshni
Roll No: 66
Class: IT-B

Enter your name: Roshni
Enter your age: 21
Hello, Roshni! You are 21 years old.

=== Code Execution Successful ===
```

PROGRAM – 2

AIM: Create a program that prompts the user for their age and tells them if they can vote in the next election.

SOURCE CODE:

```
using System;
class Program
{
   static void Main()
   {
      Console.WriteLine("Name:
      Roshni");
      Console.WriteLine("Class: IT-B");
      Console.WriteLine("Roll No: 66");
      Console.WriteLine();
      Console.Write("Enter your age: ");
      int age =
      Convert.ToInt32(Console.ReadLine()); if (age
       >= 18)
      {
         Console.WriteLine("You are eligible to vote in the next election.");
      }
      else
      {
         Console.WriteLine("You are not eligible to vote in the next election.");
      }
   }
}
```

Output

```
Name: Roshni
Class: IT-B
Roll No: 66

Enter your age: 21
You are eligible to vote in the next election.

=== Code Execution Successful ===
```

PROGRAM – 3

AIM: Create a program that calculates the factorial of a number entered by the user using a loop.

SOURCE CODE:

```
using System;
class Program
{
   static void Main()
   {
      Console.WriteLine("Name: Roshni");
      Console.WriteLine("Class: IT-B");
      Console.WriteLine("Roll No: 66");

      Console.Write("Enter a number to find its factorial:
      "); int num = Convert.ToInt32(Console.ReadLine());

      long factorial = 1;
      for (int i = 1; i <= num; i++)
      {
         factorial *= i;
      }
      Console.WriteLine("Factorial of " + num + " is: " + factorial);
   }
}
```

Output

```
Name: Roshni
Class: IT-B
Roll No: 66
Enter a number to find its factorial: 5
Factorial of 5 is: 120

=== Code Execution Successful ===
```

PROGRAM - 4

AIM: Create a program that prompts the user for a list of numbers and then sorts them in ascending order.

SOURCE CODE:

```
using System;
using
System.Collections.Generic;
class Program
{
   static void Main()
   {
      Console.WriteLine("Name: Roshni");
      Console.WriteLine("Class: IT-B");
      Console.WriteLine("Roll No: 66\n");

      Console.Write("Enter numbers separated by
      spaces: "); string input = Console.ReadLine();

      string[] parts = input.Split(' ', StringSplitOptions.RemoveEmptyEntries);
      List<int> numbers = new List<int>();
      foreach (string part in parts)
      {
         numbers.Add(Convert.ToInt32(part));
      }
      numbers.Sort();
      Console.WriteLine("Numbers in ascending
      order:"); foreach (int num in numbers)
      {
         Console.Write(num + " ");
      }
      Console.WriteLine();
   }
}
```

Output

```
Name: Roshni
Class: IT-B
Roll No: 66

Enter numbers separated by spaces: 2 4 1 5
Numbers in ascending order:
1 2 4 5

=== Code Execution Successful ===
```

PROGRAM - 5

AIM: Create a program that prompts the user for a string and then prints out the string reversed.

SOURCE CODE:

```
using System;


class Program
{
   static void Main()
   {
      Console.WriteLine("Name: Roshni");
      Console.WriteLine("Class: IT-B");
      Console.WriteLine("Roll No: 66\n");

      Console.Write("Enter a string: ");
      string input = Console.ReadLine();

      char[] charArray = input.ToCharArray();
      Array.Reverse(charArray);
      string reversed = new string(charArray);

      Console.WriteLine("Reversed string: " + reversed);
   }
}
```

Output

```
Name: Roshni
Class: IT-B
Roll No: 66

Enter a string: hello
Reversed string: olleh

=== Code Execution Successful ===
```

PROGRAM - 6

AIM: Create a program that defines a function to calculate the area of a circle based on the radius entered by the user.

SOURCE CODE:

```
using System;
class Program
{
   static double CalculateCircleArea(double radius)
   {
      return Math.PI * radius * radius;
   }
   static void Main()
   {
      Console.WriteLine("Name: Roshni");
      Console.WriteLine("Class: IT-B");
      Console.WriteLine("Roll No: 66\n");

      Console.Write("Enter the radius of the circle: ");
      double radius =
      Convert.ToDouble(Console.ReadLine()); double area =
      CalculateCircleArea(radius);

      Console.WriteLine("Area of the circle: " + area);
   }
}
```

Output

```
Name: Roshni
Class: IT-B
Roll No: 66

Enter the radius of the circle: 4
Area of the circle: 50.2654824574367

=== Code Execution Successful ===
```

PROGRAM - 7

AIM: Create a program that defines a class to represent a car and then creates an object of that class with specific attributes.

SOURCE CODE:

```
using System;
class Car
{
   public          string
   Brand;         public
   string         Model;
   public int Year;
   public void DisplayDetails()
   {
      Console.WriteLine("Car Brand: " + Brand);
      Console.WriteLine("Car Model: " + Model);
      Console.WriteLine("Year of Manufacture: " +
      Year);
   }
}
class Program
{
   static void Main()
   {
      Console.WriteLine("Name:
      Roshni");
      Console.WriteLine("Class: IT-B");
      Console.WriteLine("Roll No:
      66\n"); Car myCar = new Car();
      myCar.Brand = "Toyota";
      myCar.Model = "Corolla";
      myCar.Year = 2022;
      myCar.DisplayDetails();
   }
}
```

Output

```
Name: Roshni
Class: IT-B
Roll No: 66

Car Brand: Toyota
Car Model: Corolla
Year of Manufacture: 2022

=== Code Execution Successful ===
```

PROGRAM - 8

AIM: Create a program that reads data from a file and writes it to another file in a different format.

SOURCE CODE:

```
using System;
using System.IO;
class Program7
{
    static void Main()
    {
        string inputFile = "input.txt";
        string outputFile = "output.txt";
        try
        {
            string[] lines = File.ReadAllLines(inputFile);

            using (StreamWriter writer = new StreamWriter(outputFile))
            {
                foreach (string line in lines)
                {
                    writer.WriteLine(line.ToUpper());
                }
            }

            Console.WriteLine($"Data from {inputFile} has been written to {outputFile} in uppercase.");
        }
        catch (Exception e)
        {
            Console.WriteLine("Error: " + e.Message);
        }
    }
}
```

Input.txt

```
Roshni Kumari
hey how are you
```

Output.txt

```
ROSHNI KUMARI
HEY HOW ARE YOU
```

PROGRAM - 9

AIM: Create a program that prompts the user for two numbers and then divides them, handling any exceptions that may arise.

SOURCE CODE:

```
using System;
class Program9
{
   static void Main()
   {
      Console.WriteLine("Name:
      Roshni");
      Console.WriteLine("Class: IT-B");
      Console.WriteLine("Roll No: 66\n");
      try
      {
         Console.Write("Enter num 1: ");
         int num1 = Convert.ToInt32(Console.ReadLine());
         Console.Write("Enter num 2: ");
         int num2 =
         Convert.ToInt32(Console.ReadLine()); int result
         = num1 / num2; Console.WriteLine($"\nResult:
         {result}");
      }
      catch (DivideByZeroException)
      {
         Console.WriteLine("Error: Division by zero is not allowed.");
      }
      catch (FormatException)
      {
         Console.WriteLine("Error: Please enter valid numbers.");
      }
      catch (Exception e)
      {
         Console.WriteLine("Unexpected error: " + e.Message);
```

```
    }
  }
}
```

Output

```
Name: Roshni
Class: IT-B
Roll No: 66

Enter num 1: 10
Enter num 2:
ERROR!
Error: Division by zero is not allowed.

=== Code Execution Successful ===
```

PROGRAM - 10

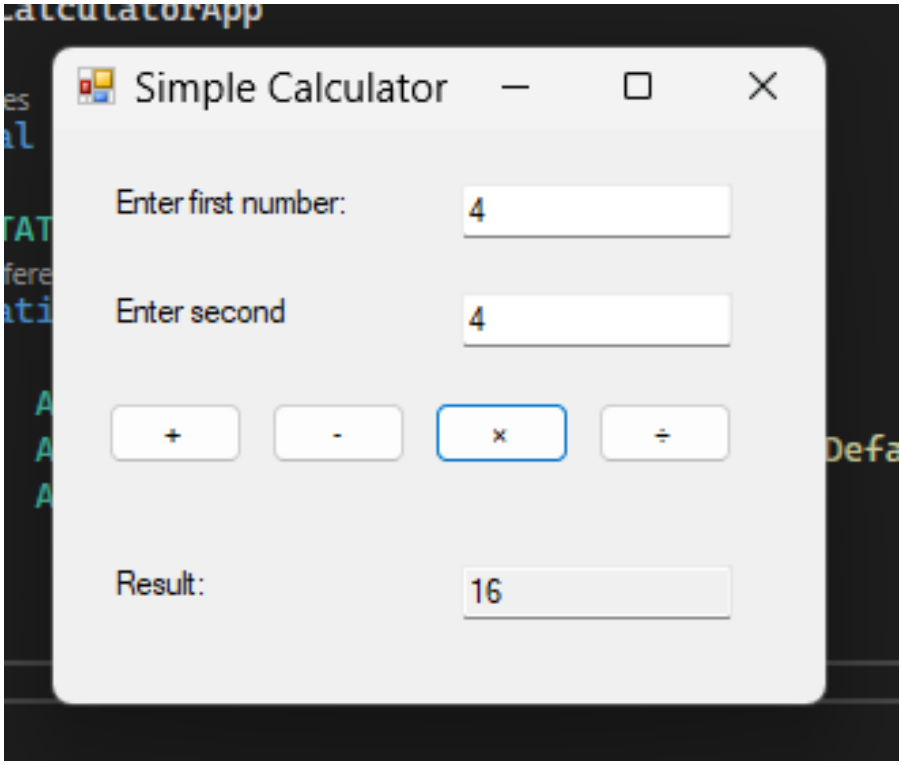
AIM: Create a program that uses a graphical user interface to allow the user to perform simple calculations.

SOURCE CODE:

```
using System;
using
System.Windows.Forms;
namespace CalculatorApp
{
   public partial class Form1 : Form
   {
      TextBox input1, input2, resultBox;
      Button addBtn, subBtn, mulBtn, divBtn;

      public Form1()
      {
         InitializeComponent();

         this.Text = "Simple Calculator";
         this.Width = 300;
         this.Height = 250;

         Label label1 = new Label() { Text = "Enter first number:", Top = 20, Left = 20 };
         Label label2 = new Label() { Text = "Enter second number:", Top = 60, Left = 20 };
         input1 = new TextBox() { Top = 20, Left = 150, Width = 100 };
         input2 = new TextBox() { Top = 60, Left = 150, Width = 100 };
         resultBox = new TextBox() { Top = 160, Left = 150, Width = 100, ReadOnly = true };

         addBtn = new Button() { Text = "+", Top = 100, Left = 20, Width = 50 };
         subBtn = new Button() { Text = "-", Top = 100, Left = 80, Width = 50 };
         mulBtn = new Button() { Text = "×", Top = 100, Left = 140, Width = 50 };
         divBtn = new Button() { Text = "÷", Top = 100, Left = 200, Width = 50 };
```

```
    Label resultLabel = new Label() { Text = "Result:", Top = 160, Left = 20 };

    addBtn.Click += (s, e) =>
    Calculate("+"); subBtn.Click += (s, e)
    => Calculate("-"); mulBtn.Click += (s, e)
    => Calculate("*"); divBtn.Click += (s, e)
    => Calculate("/");

    this.Controls.Add(label1);
    this.Controls.Add(label2);
    this.Controls.Add(input1);
    this.Controls.Add(input2);
    this.Controls.Add(resultBox);
    this.Controls.Add(addBtn);
    this.Controls.Add(subBtn);
    this.Controls.Add(mulBtn);
    this.Controls.Add(divBtn);
    this.Controls.Add(resultLabel);
}

void Calculate(string op)
{
    try
    {
        double num1 = Convert.ToDouble(input1.Text); double num2 = Convert.ToDouble(input2.Text);
        double result = 0;

        switch (op)
        {
            case "+": result = num1 + num2; break;
            case "-": result = num1 - num2; break;
            case "*": result = num1 * num2; break;
            case "/": result = num1 / num2; break;
        }
```

```
            resultBox.Text = result.ToString();
        }
        catch (Exception)
        {
            MessageBox.Show("Please enter valid numbers.");
        }
    }
}
```



PROGRAM - 11

AIM: Create a program that uses multithreading to perform a time-consuming task in the background while the user can continue using the application.

SOURCE CODE:

```
using System;
using System.Threading;
class Program
{
   static void TimeConsumingTask()
   {
      for (int i = 1; i <= 5; i++)
      {
         Console.WriteLine($"Background task running... Step {i}");
         Thread.Sleep(2000);
      }
      Console.WriteLine("Background task completed!");
   }
   static void Main()
   {
      Console.WriteLine("Name: Roshni");
      Console.WriteLine("Class: IT-B");
      Console.WriteLine("Roll No: 66\n");

      Thread backgroundThread = new Thread(TimeConsumingTask);
      backgroundThread.Start();

      Console.WriteLine("You can continue using the app while the task runs in background!");
      Console.WriteLine("Type something below (the app remains responsive):");

      string input = Console.ReadLine();
      Console.WriteLine($"You typed: {input}");
```

```
        backgroundThread.Join();
        Console.WriteLine("Main thread finished.");
    }
}
```

Output

```
Name: Roshni
Class: IT-B
Roll No: 66

You can continue using the app while the task runs in background!
Type something below (the app remains responsive):
Background task running... Step 1
Background task running... Step 2
ii
You typed: ii
Background task running... Step 3
Background task running... Step 4
Background task running... Step 5
Background task completed!
Main thread finished.

=== Code Execution Successful ===
```

PROGRAM - 12

AIM: Create a program that uses LINQ to query and manipulate data from a database.

SOURCE CODE:

```
using System;
using
System.Collections.Generic;
using System.Linq;

class Program
{
   class Student
   {
      public int ID { get; set; }
      public string Name { get; set;
       } public int Marks { get; set; }
   }
   static void Main()
   {
      Console.WriteLine("Name: Roshni");
      Console.WriteLine("Class: IT-B");
      Console.WriteLine("Roll No: 66\n");

      List<Student> students = new List<Student>()
      {
         new Student { ID = 1, Name = "Aarav", Marks = 85 },
         new Student { ID = 2, Name = "Riya", Marks = 72 },
         new Student { ID = 3, Name = "Neha", Marks = 90 },
         new Student { ID = 4, Name = "Karan", Marks = 60 },
         new Student { ID = 5, Name = "Isha", Marks = 78 }
      };
      var highScorers = from s in students
                        where s.Marks > 75
```

```
                orderby s.Marks descending
                select s;

        Console.WriteLine("Students scoring above 75:\n");
        foreach (var s in highScorers)
        {
            Console.WriteLine($"{s.Name} - Marks: {s.Marks}");
        }

        students = students.Select(s => { s.Marks += 5; return s; }).ToList();

        Console.WriteLine("\nAfter adding 5 bonus marks:\n");
        foreach (var s in students)
        {
            Console.WriteLine($"{s.Name} - New Marks: {s.Marks}");
        }
    }
}
```

Output

```
Name: Roshni
Class: IT-B
Roll No: 66

Students scoring above 75:

Neha - Marks: 90
Aarav - Marks: 85
Isha - Marks: 78

After adding 5 bonus marks:

Aarav - New Marks: 90
Riya - New Marks: 77
Neha - New Marks: 95
Karan - New Marks: 65
Isha - New Marks: 83

=== Code Execution Successful ===
```