1.

using System;

class LeapYearChecker

{

    static void Main()

    {

        DateTime currentTime = DateTime.Now;

        string dt = $"{currentTime.ToString("dd/MM/yyyy h:mm tt")}";

        Console.WriteLine(dt);

        Console.WriteLine("Name: Uday Sheth");

        Console.WriteLine("Roll No : 62");

        Console.WriteLine("Enter a year: ");

        int year = Convert.ToInt32(Console.ReadLine());

        bool isLeapYear = IsLeapYear(year);

        if (isLeapYear)

        {

            Console.WriteLine("leap year.");

        }

        else

        {

            Console.WriteLine("not a leap year.");

        }

    }

    static bool IsLeapYear(int year)

    {

        if (year % 4 == 0)

        {

            if (year % 100 == 0)

            {

                return year % 400 == 0;

            }

            else

            {

                return true;

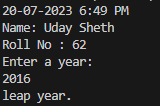
            }

        }

        return false;

    }

}



2.

using System;

class TriangleTypeChecker

{

    static void Main()

    {

        DateTime currentTime = DateTime.Now;

        string dt = $"{currentTime.ToString("dd/MM/yyyy h:mm tt")}";

        Console.WriteLine(dt);

        Console.WriteLine("Name: Uday Sheth");

        Console.WriteLine("Roll No : 62");

        Console.WriteLine("Enter the lengths of the triangle's sides:");

        Console.Write("Side 1: ");

        int side1 = Convert.ToInt32(Console.ReadLine());

        Console.Write("Side 2: ");

        int side2 = Convert.ToInt32(Console.ReadLine());

        Console.Write("Side 3: ");

        int side3 = Convert.ToInt32(Console.ReadLine());

        string triangleType = GetTriangleType(side1, side2, side3);

        Console.WriteLine("The triangle is {0}.", triangleType);

    }

    static string GetTriangleType(int side1, int side2, int side3)

    {

        if (side1 == side2 && side2 == side3)

        {

            return "Equilateral";

        }

        else if (side1 == side2 || side1 == side3 || side2 == side3)

        {

            return "Isosceles";

        }

        else

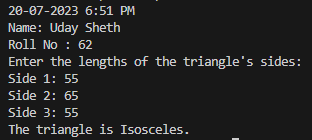
        {

            return "Scalene";

        }

    }

}



3.

using System;

class ElectricityBillCalculator

{

    static void Main()

    {

        DateTime currentTime = DateTime.Now;

        string dt = $"{currentTime.ToString("dd/MM/yyyy h:mm tt")}";

        Console.WriteLine(dt);

        Console.WriteLine("Name: Uday Sheth");

        Console.WriteLine("Roll No : 62");

        Console.Write("Enter customer ID: ");

        int customerId = Convert.ToInt32(Console.ReadLine());

        Console.Write("Enter customer name: ");

        string customerName = Console.ReadLine();

        Console.Write("Enter units consumed: ");

        int unitsConsumed = Convert.ToInt32(Console.ReadLine());

        double unitCharge = GetUnitCharge(unitsConsumed);

        double totalAmountDue = unitsConsumed \* unitCharge;

        if (totalAmountDue > 400)

        {

            totalAmountDue += totalAmountDue \* 0.15;

        }

        if (totalAmountDue < 100)

        {

            totalAmountDue = 100;

        }

        Console.WriteLine("\nElectricity Bill");

        Console.WriteLine("Customer ID: {0}", customerId);

        Console.WriteLine("Customer Name: {0}", customerName);

        Console.WriteLine("Units Consumed: {0}", unitsConsumed);

        Console.WriteLine("Total Amount Due: Rs. {0}", totalAmountDue);

    }

    static double GetUnitCharge(int unitsConsumed)

    {

        if (unitsConsumed <= 199)

        {

            return 1.20;

        }

        else if (unitsConsumed >= 200 && unitsConsumed < 400)

        {

            return 1.50;

        }

        else if (unitsConsumed >= 400 && unitsConsumed < 600)

        {

            return 1.80;

        }

        else

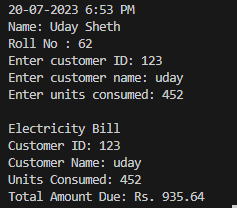
        {

            return 2.00;

        }

    }

}



4.

using System;

class ReverseOrderLetters

{

    static void Main()

    {

        DateTime currentTime = DateTime.Now;

        string dt = $"{currentTime.ToString("dd/MM/yyyy h:mm tt")}";

        Console.WriteLine(dt);

        Console.WriteLine("Name: Uday Sheth");

        Console.WriteLine("Roll No : 62");

        Console.WriteLine("Enter three letters:");

        Console.Write("Enter letter: ");

        string letter1 = Console.ReadLine();

        Console.Write("Enter letter: ");

        string letter2 = Console.ReadLine();

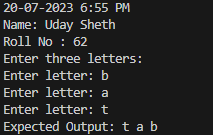
        Console.Write("Enter letter: ");

        string letter3 = Console.ReadLine();

        Console.WriteLine("Expected Output: {0} {1} {2}", letter3, letter2, letter1);

    }

}



5.

using System;

namespace CompareStrings

{

    class Program

    {

        static void Main(string[] args)

        {

            DateTime currentTime = DateTime.Now;

            string dt = $"{currentTime.ToString("dd/MM/yyyy h:mm tt")}";

            Console.WriteLine(dt);

            Console.WriteLine("Name: Uday Sheth");

            Console.WriteLine("Roll No : 62");

            string str1, str2;

            Console.WriteLine("Input the 1st string: ");

            str1 = Console.ReadLine();

            Console.WriteLine("Input the 2nd string: ");

            str2 = Console.ReadLine();

            bool areEqual = str1.Equals(str2);

            if (areEqual)

            {

                Console.WriteLine("The strings are equal.");

            }

            else

            {

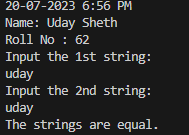
                Console.WriteLine("The strings are not equal.");

            }

        }

    }

}



6.

using System;

namespace CheckUsernameAndPassword

{

    class Program

    {

        static void Main(string[] args)

        {

            string username, password;

            bool isLoggedIn = false;

            do

            {

                DateTime currentTime = DateTime.Now;

                string dt = $"{currentTime.ToString("dd/MM/yyyy h:mm tt")}";

                Console.WriteLine(dt);

                Console.WriteLine("Name: Uday Sheth");

                Console.WriteLine("Roll No : 62");

                Console.WriteLine("Input a username: ");

                username = Console.ReadLine();

                Console.WriteLine("Input a password: ");

                password = Console.ReadLine();

                if (username == "abcd" && password == "1234")

                {

                    isLoggedIn = true;

                    break;

                }

                else

                {

                    Console.WriteLine("Invalid username or password. Please try again.");

                }

            } while (!isLoggedIn);

            if (isLoggedIn)

            {

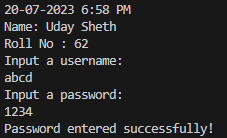
                Console.WriteLine("Password entered successfully!");

            }

        }

    }

}



7.

using System;

namespace LoginAttempts

{

    class Program

    {

        static void Main(string[] args)

        {

            string userId, password;

            int attempts = 0;

            do

            {

                DateTime currentTime = DateTime.Now;

                string dt = $"{currentTime.ToString("dd/MM/yyyy h:mm tt")}";

                Console.WriteLine(dt);

                Console.WriteLine("Name: Uday Sheth");

                Console.WriteLine("Roll No : 62");

                Console.WriteLine("Input user id: ");

                userId = Console.ReadLine();

                Console.WriteLine("Input password: ");

                password = Console.ReadLine();

                if (userId == "admin" && password == "123")

                {

                    Console.WriteLine("Login successful!");

                    break;

                }

                else

                {

                    attempts++;

                    Console.WriteLine("Invalid user id or password. Please try again.");

                }

            } while (attempts < 3);

            if (attempts == 3)

            {

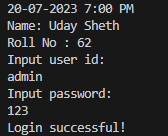
                Console.WriteLine("Login attempt exceeded. Please try later.");

            }

        }

    }

}



8.

using System;

class MatrixOperations

{

    static int[,] matrix;

    static int[,] newMatrix;

    static int rows, columns;

    static void Main()

    {

        DateTime currentTime = DateTime.Now;

        string dt = $"{currentTime.ToString("dd/MM/yyyy h:mm tt")}";

        Console.WriteLine(dt);

        Console.WriteLine("Name: Uday Sheth");

        Console.WriteLine("Roll No : 62");

        int choice;

        Console.WriteLine("Enter the number of rows for the matrix:");

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

        Console.WriteLine("Enter the number of columns for the matrix:");

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

        matrix = new int[rows, columns];

        while (true)

        {

            Console.WriteLine("\nMenu:");

            Console.WriteLine("1. Square of created matrix");

            Console.WriteLine("2. Create another matrix and perform addition");

            Console.WriteLine("3. Perform multiplication of the new matrix with the created matrix");

            Console.WriteLine("4. Print transpose of the created matrix");

            Console.WriteLine("5. Exit");

            Console.WriteLine("Enter your choice (1-5):");

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

            switch (choice)

            {

                case 1:

                    CreateMatrix();

                    SquareMatrix();

                    break;

                case 2:

                    CreateNewMatrix();

                    AddMatrices();

                    break;

                case 3:

                    MultiplyMatrices();

                    break;

                case 4:

                    TransposeMatrix();

                    break;

                case 5:

                    Console.WriteLine("Exiting program...");

                    return;

                default:

                    Console.WriteLine("Invalid choice. Please try again.");

                    break;

            }

        }

    }

    static void CreateMatrix()

    {

        Console.WriteLine("Enter the elements of the matrix:");

        for (int i = 0; i < rows; i++)

        {

            for (int j = 0; j < columns; j++)

            {

                Console.Write($"Enter element at position ({i + 1},{j + 1}): ");

                matrix[i, j] = int.Parse(Console.ReadLine());

            }

        }

    }

    static void CreateNewMatrix()

    {

        Console.WriteLine("Enter the elements of the new matrix:");

        newMatrix = new int[rows, columns];

        for (int i = 0; i < rows; i++)

        {

            for (int j = 0; j < columns; j++)

            {

                Console.Write($"Enter element at position ({i + 1},{j + 1}): ");

                newMatrix[i, j] = int.Parse(Console.ReadLine());

            }

        }

    }

    static void SquareMatrix()

    {

        Console.WriteLine("Square of the created matrix:");

        for (int i = 0; i < rows; i++)

        {

            for (int j = 0; j < columns; j++)

            {

                Console.Write(matrix[i, j] \* matrix[i, j] + " ");

            }

            Console.WriteLine();

        }

    }

    static void AddMatrices()

    {

        Console.WriteLine("Addition of the created matrix and new matrix:");

        for (int i = 0; i < rows; i++)

        {

            for (int j = 0; j < columns; j++)

            {

                Console.Write(matrix[i, j] + newMatrix[i, j] + " ");

            }

            Console.WriteLine();

        }

    }

    static void MultiplyMatrices()

    {

        if (newMatrix == null)

        {

            Console.WriteLine("Please create a new matrix first.");

            return;

        }

        if (columns != rows)

        {

            Console.WriteLine("Matrix multiplication is not possible. The number of columns in the first matrix should be equal to the number of rows in the second matrix.");

            return;

        }

        int[,] resultMatrix = new int[rows, columns];

        for (int i = 0; i < rows; i++)

        {

            for (int j = 0; j < columns; j++)

            {

                for (int k = 0; k < columns; k++)

                {

                    resultMatrix[i, j] += matrix[i, k] \* newMatrix[k, j];

                }

            }

        }

        Console.WriteLine("Multiplication of the created matrix and new matrix:");

        for (int i = 0; i < rows; i++)

        {

            for (int j = 0; j < columns; j++)

            {

                Console.Write(resultMatrix[i, j] + " ");

            }

            Console.WriteLine();

        }

    }

    static void TransposeMatrix()

    {

        Console.WriteLine("Transpose of the created matrix:");

        for (int j = 0; j < columns; j++)

        {

            for (int i = 0; i < rows; i++)

            {

                Console.Write(matrix[i, j] + " ");

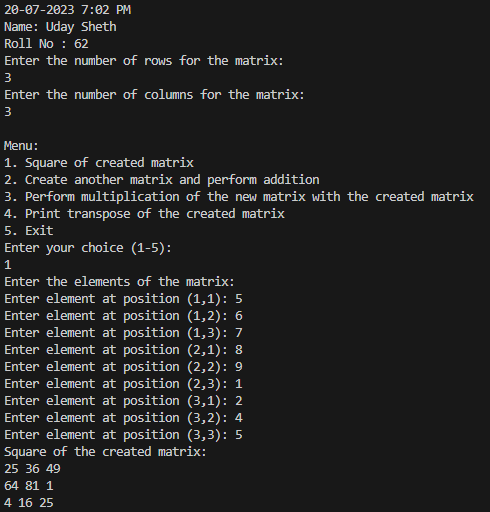
            }

            Console.WriteLine();

        }

    }

}



9.

using System;

class MatrixOperations

{

    private int[,] matrix;

    private int size;

    public MatrixOperations(int size)

    {

        this.size = size;

        matrix = new int[size, size];

    }

    public void CreateMatrix()

    {

        DateTime currentTime = DateTime.Now;

        string dt = $"{currentTime.ToString("dd/MM/yyyy h:mm tt")}";

        Console.WriteLine(dt);

        Console.WriteLine("Name: Uday Sheth");

        Console.WriteLine("Roll No : 62");

        Console.WriteLine("Enter matrix elements:");

        for (int i = 0; i < size; i++)

        {

            for (int j = 0; j < size; j++)

            {

                Console.Write("Enter element at position [{i}, {j}]: ");

                matrix[i, j] = Convert.ToInt32(Console.ReadLine());

            }

        }

    }

    public void PrintDiagonals()

    {

        Console.WriteLine("Diagonals of the matrix:");

        for (int i = 0; i < size; i++)

        {

            Console.Write(matrix[i, i] + " ");

        }

        Console.WriteLine();

        Console.WriteLine("Reverse Diagonal of the matrix:");

        for (int i = 0; i < size; i++)

        {

            Console.Write(matrix[i, size - i - 1] + " ");

        }

        Console.WriteLine();

    }

    public void PrintUpperTriangular()

    {

        Console.WriteLine("Upper Triangular Matrix:");

        for (int i = 0; i < size; i++)

        {

            for (int j = 0; j < size; j++)

            {

                if (j >= i)

                {

                    Console.Write(matrix[i, j] + " ");

                }

                else

                {

                    Console.Write("0 ");

                }

            }

            Console.WriteLine();

        }

    }

    public void PrintLowerTriangular()

    {

        Console.WriteLine("Lower Triangular Matrix:");

        for (int i = 0; i < size; i++)

        {

            for (int j = 0; j < size; j++)

            {

                if (j <= i)

                {

                    Console.Write(matrix[i, j] + " ");

                }

                else

                {

                    Console.Write("0 ");

                }

            }

            Console.WriteLine();

        }

    }

    public int CalculateDeterminant()

    {

        if (size != 3)

        {

            Console.WriteLine("Determinant calculation is supported only for 3x3 matrices.");

            return 0;

        }

        int determinant = matrix[0, 0] \* (matrix[1, 1] \* matrix[2, 2] - matrix[1, 2] \* matrix[2, 1])

                        - matrix[0, 1] \* (matrix[1, 0] \* matrix[2, 2] - matrix[1, 2] \* matrix[2, 0])

                        + matrix[0, 2] \* (matrix[1, 0] \* matrix[2, 1] - matrix[1, 1] \* matrix[2, 0]);

        return determinant;

    }

}

class MenuDrivenMatrixApp

{

    static void Main()

    {

        Console.Write("Enter the size of the square matrix: ");

        int size = Convert.ToInt32(Console.ReadLine());

        MatrixOperations matrixOp = new MatrixOperations(size);

        int choice;

        do

        {

            Console.WriteLine("\nMatrix Operations Menu");

            Console.WriteLine("1. Print diagonals of matrix");

            Console.WriteLine("2. Print upper triangular matrix");

            Console.WriteLine("3. Print lower triangular matrix");

            Console.WriteLine("4. Calculate determinant of matrix");

            Console.WriteLine("0. Exit");

            Console.Write("Enter your choice: ");

            choice = Convert.ToInt32(Console.ReadLine());

            switch (choice)

            {

                case 1:

                    matrixOp.CreateMatrix();

                    matrixOp.PrintDiagonals();

                    break;

                case 2:

                    matrixOp.CreateMatrix();

                    matrixOp.PrintUpperTriangular();

                    break;

                case 3:

                    matrixOp.CreateMatrix();

                    matrixOp.PrintLowerTriangular();

                    break;

                case 4:

                    matrixOp.CreateMatrix();

                    int determinant = matrixOp.CalculateDeterminant();

                    Console.WriteLine("Determinant of the matrix: " + determinant);

                    break;

                case 0:

                    Console.WriteLine("Exiting the program...");

                    break;

                default:

                    Console.WriteLine("Invalid choice! Please try again.");

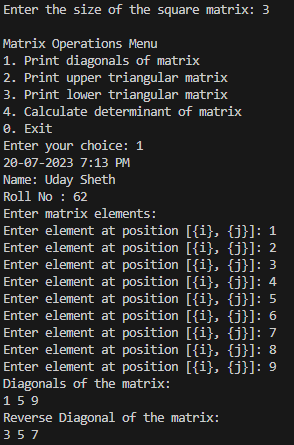
                    break;

            }

        } while (choice != 0);

    }

}



10.

using System;

public class Program

{

    public static void Main()

    {

        DateTime currentTime = DateTime.Now;

        string dt = $"{currentTime.ToString("dd/MM/yyyy h:mm tt")}";

        Console.WriteLine(dt);

        Console.WriteLine("Name: Uday Sheth");

        Console.WriteLine("Roll No : 62");

        int[] arr = { 1, 2, 3, 1, 2, 5, 6, 7, 8, 9 };

        int count = 0;

        for (int i = 0; i < arr.Length; i++)

        {

            for (int j = i + 1; j < arr.Length; j++)

            {

                if (arr[i] == arr[j])

                {

                    count++;

                    break;

                }

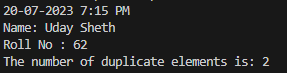
            }

        }

        Console.WriteLine("The number of duplicate elements is: {0}", count);

    }

}



11.

using System;

class OddEvenSeparator

{

    static void Main()

    {

        DateTime currentTime = DateTime.Now;

        string dt = $"{currentTime.ToString("dd/MM/yyyy h:mm tt")}";

        Console.WriteLine(dt);

        Console.WriteLine("Name: Uday Sheth");

        Console.WriteLine("Roll No : 62");

        Console.Write("Input the number of elements to be stored in the array: ");

        int numElements = Convert.ToInt32(Console.ReadLine());

        int[] inputArray = new int[numElements];

        int[] oddArray;

        int[] evenArray;

        Console.WriteLine("Input {0} elements in the array:", numElements);

        for (int i = 0; i < numElements; i++)

        {

            Console.Write("element - {0}: ", i);

            inputArray[i] = Convert.ToInt32(Console.ReadLine());

        }

        int oddCount = 0;

        int evenCount = 0;

        for (int i = 0; i < numElements; i++)

        {

            if (inputArray[i] % 2 == 0)

            {

                evenCount++;

            }

            else

            {

                oddCount++;

            }

        }

        oddArray = new int[oddCount];

        evenArray = new int[evenCount];

        int oddIndex = 0;

        int evenIndex = 0;

        for (int i = 0; i < numElements; i++)

        {

            if (inputArray[i] % 2 == 0)

            {

                evenArray[evenIndex] = inputArray[i];

                evenIndex++;

            }

            else

            {

                oddArray[oddIndex] = inputArray[i];

                oddIndex++;

            }

        }

        Console.WriteLine("\nOdd numbers:");

        for (int i = 0; i < oddCount; i++)

        {

            Console.WriteLine(oddArray[i]);

        }

        Console.WriteLine("\nEven numbers:");

        for (int i = 0; i < evenCount; i++)

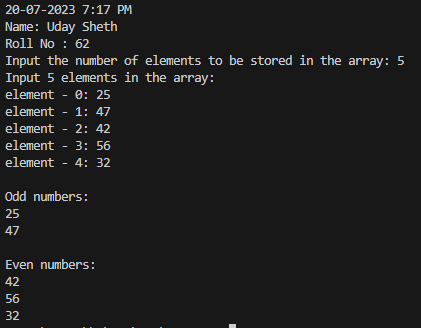
        {

            Console.WriteLine(evenArray[i]);

        }

    }

}



12.

using System;

class MenuDrivenApplication

{

    static void Main()

    {

        int choice;

        bool exit = false;

        do

        {

            DateTime currentTime = DateTime.Now;

            string dt = $"{currentTime.ToString("dd/MM/yyyy h:mm tt")}";

            Console.WriteLine(dt);

            Console.WriteLine("Name: Uday Sheth");

            Console.WriteLine("Roll No : 62");

            Console.WriteLine("Menu:");

            Console.WriteLine("1. Sum of Even Natural Numbers");

            Console.WriteLine("2. Multiplication Table");

            Console.WriteLine("3. Perfect Number Check");

            Console.WriteLine("4. Strong Number Check");

            Console.WriteLine("5. Exit");

            Console.Write("Enter your choice (1-5): ");

            choice = Convert.ToInt32(Console.ReadLine());

            switch (choice)

            {

                case 1:

                    SumOfEvenNaturalNumbers();

                    break;

                case 2:

                    PrintMultiplicationTable();

                    break;

                case 3:

                    CheckPerfectNumber();

                    break;

                case 4:

                    CheckStrongNumber();

                    break;

                case 5:

                    exit = true;

                    break;

                default:

                    Console.WriteLine("Invalid choice. Please try again.");

                    break;

            }

            Console.WriteLine();

        } while (!exit);

    }

    static void SumOfEvenNaturalNumbers()

    {

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

        int n = Convert.ToInt32(Console.ReadLine());

        int sum = 0;

        for (int i = 1; i <= n; i++)

        {

            if (i % 2 == 0)

            {

                sum += i;

            }

        }

        Console.WriteLine("Sum of even natural numbers up to {0}: {1}", n, sum);

    }

    static void PrintMultiplicationTable()

    {

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

        int n = Convert.ToInt32(Console.ReadLine());

        for (int i = 1; i <= 10; i++)

        {

            int result = n \* i;

            Console.WriteLine("{0} X {1} = {2}", n, i, result);

        }

    }

    static void CheckPerfectNumber()

    {

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

        int n = Convert.ToInt32(Console.ReadLine());

        int sum = 0;

        for (int i = 1; i < n; i++)

        {

            if (n % i == 0)

            {

                sum += i;

            }

        }

        if (sum == n)

        {

            Console.WriteLine("{0} is a perfect number.", n);

        }

        else

        {

            Console.WriteLine("{0} is not a perfect number.", n);

        }

    }

    static void CheckStrongNumber()

    {

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

        int n = Convert.ToInt32(Console.ReadLine());

        int temp = n;

        int sum = 0;

        while (temp != 0)

        {

            int digit = temp % 10;

            int fact = Factorial(digit);

            sum += fact;

            temp /= 10;

        }

        if (sum == n)

        {

            Console.WriteLine("{0} is a strong number.", n);

        }

        else

        {

            Console.WriteLine("{0} is not a strong number.", n);

        }

    }

    static int Factorial(int n)

    {

        if (n == 0 || n == 1)

        {

            return 1;

        }

        else

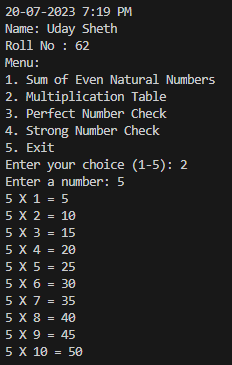
        {

            return n \* Factorial(n - 1);

        }

    }

}



13.

using System;

struct Employee

{

    public string Name;

    public int DayOfBirth;

    public int MonthOfBirth;

    public int YearOfBirth;

}

class EmployeeData

{

    static void Main()

    {

        Employee employee;

        DateTime currentTime = DateTime.Now;

        string dt = $"{currentTime.ToString("dd/MM/yyyy h:mm tt")}";

        Console.WriteLine(dt);

        Console.WriteLine("Name: Uday Sheth");

        Console.WriteLine("Roll No : 62");

        Console.Write("Name of the employee: ");

        employee.Name = Console.ReadLine();

        Console.Write("Input day of birth: ");

        employee.DayOfBirth = Convert.ToInt32(Console.ReadLine());

        Console.Write("Input month of birth: ");

        employee.MonthOfBirth = Convert.ToInt32(Console.ReadLine());

        Console.Write("Input year for birth: ");

        employee.YearOfBirth = Convert.ToInt32(Console.ReadLine());

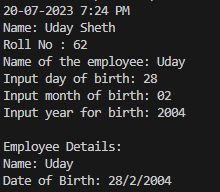
        Console.WriteLine("\nEmployee Details:");

        Console.WriteLine("Name: {0}", employee.Name);

        Console.WriteLine("Date of Birth: {0}/{1}/{2}", employee.DayOfBirth, employee.MonthOfBirth, employee.YearOfBirth);

    }

}



14.

using System;

struct Book

{

    public string Name;

    public string Author;

}

class BookInformation

{

    static void Main()

    {

        Book book1, book2;

        DateTime currentTime = DateTime.Now;

        string dt = $"{currentTime.ToString("dd/MM/yyyy h:mm tt")}";

        Console.WriteLine(dt);

        Console.WriteLine("Name: Uday Sheth");

        Console.WriteLine("Roll No : 62");

        Console.WriteLine("Insert the information of two books:");

        Console.WriteLine("Information of book 1");

        Console.Write("Input name of the book: ");

        book1.Name = Console.ReadLine();

        Console.Write("Input the author: ");

        book1.Author = Console.ReadLine();

        Console.WriteLine();

        Console.WriteLine("Information of book 2");

        Console.Write("Input name of the book: ");

        book2.Name = Console.ReadLine();

        Console.Write("Input the author: ");

        book2.Author = Console.ReadLine();

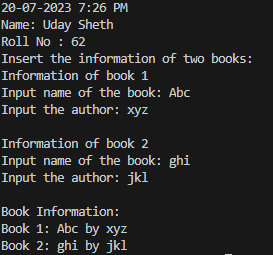
        Console.WriteLine("\nBook Information:");

        Console.WriteLine("Book 1: {0} by {1}", book1.Name, book1.Author);

        Console.WriteLine("Book 2: {0} by {1}", book2.Name, book2.Author);

    }

}



15.

using System;

struct Square

{

    public double Length;

    public double Breadth;

}

class SquareImplementation

{

    static void Main()

    {

        Square square;

        DateTime currentTime = DateTime.Now;

        string dt = $"{currentTime.ToString("dd/MM/yyyy h:mm tt")}";

        Console.WriteLine(dt);

        Console.WriteLine("Name: Uday Sheth");

        Console.WriteLine("Roll No : 62");

        Console.WriteLine("Input the dimensions of the Square (equal length and breadth):");

        Console.Write("Length: ");

        square.Length = Convert.ToDouble(Console.ReadLine());

        Console.Write("Breadth: ");

        square.Breadth = Convert.ToDouble(Console.ReadLine());

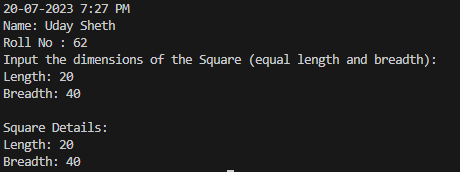
        Console.WriteLine("\nSquare Details:");

        Console.WriteLine("Length: {0}", square.Length);

        Console.WriteLine("Breadth: {0}", square.Breadth);

    }

}



16.

using System;

enum Months

{

    January,

    February,

    March,

    April,

    May,

    June,

    July,

    August,

    September,

    October,

    November,

    December

}

class MonthEnumeration

{

    static void Main()

    {

        DateTime currentTime = DateTime.Now;

        string dt = $"{currentTime.ToString("dd/MM/yyyy h:mm tt")}";

        Console.WriteLine(dt);

        Console.WriteLine("Name: Uday Sheth");

        Console.WriteLine("Roll No : 62");

        Console.WriteLine("Printing Months using foreach loop:");

        foreach (string month in Enum.GetNames(typeof(Months)))

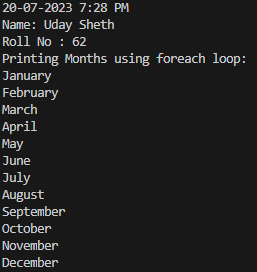
        {

            Console.WriteLine(month);

        }

    }

}



17.

using System;

enum Status

{

    Open,

    InProgress,

    Resolved,

    Closed,

    Reopened

}

class StatusEnumeration

{

    static void Main()

    {

        DateTime currentTime = DateTime.Now;

        string dt = $"{currentTime.ToString("dd/MM/yyyy h:mm tt")}";

        Console.WriteLine(dt);

        Console.WriteLine("Name: Uday Sheth");

        Console.WriteLine("Roll No : 62");

        Console.WriteLine("Status using switch case:");

        Status status = Status.Open;

        switch (status)

        {

            case Status.Open:

                Console.WriteLine("Status: Open");

                break;

            case Status.InProgress:

                Console.WriteLine("Status: In Progress");

                break;

            case Status.Resolved:

                Console.WriteLine("Status: Resolved");

                break;

            case Status.Closed:

                Console.WriteLine("Status: Closed");

                break;

            case Status.Reopened:

                Console.WriteLine("Status: Reopened");

                break;

            default:

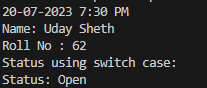
                Console.WriteLine("Invalid Status");

                break;

        }

    }

}



18.

using System;

class NumberValidator

{

    static void Main()

    {

        try

        {

            DateTime currentTime = DateTime.Now;

            string dt = $"{currentTime.ToString("dd/MM/yyyy h:mm tt")}";

            Console.WriteLine(dt);

            Console.WriteLine("Name: Uday Sheth");

            Console.WriteLine("Roll No : 62");

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

            int number = Convert.ToInt32(Console.ReadLine());

            ValidateNumber(number);

            Console.WriteLine("Number is valid.");

        }

        catch (ArgumentException ex)

        {

            Console.WriteLine("Exception caught: {0}", ex.Message);

        }

    }

    static void ValidateNumber(int number)

    {

        if (number < 0)

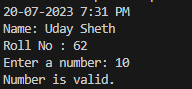
        {

            throw new ArgumentException("Number cannot be negative.");

        }

    }

}



19.

using System;

class DivisionCalculator

{

    static void Main()

    {

        try

        {

            DateTime currentTime = DateTime.Now;

            string dt = $"{currentTime.ToString("dd/MM/yyyy h:mm tt")}";

            Console.WriteLine(dt);

            Console.WriteLine("Name: Uday Sheth");

            Console.WriteLine("Roll No : 62");

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

            int numerator = Convert.ToInt32(Console.ReadLine());

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

            int denominator = Convert.ToInt32(Console.ReadLine());

            int result = DivideNumbers(numerator, denominator);

            Console.WriteLine("Result: {0}", result);

        }

        catch (DivideByZeroException ex)

        {

            Console.WriteLine("Error: {0}", ex.Message);

        }

    }

    static int DivideNumbers(int numerator, int denominator)

    {

        if (denominator == 0)

        {

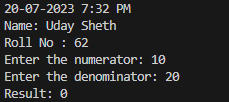
            throw new DivideByZeroException("Cannot divide by zero.");

        }

        return numerator / denominator;

    }

}



20.

using System;

namespace MenuDrivenApplication

{

    class Program

    {

        static void Main(string[] args)

        {

            int choice;

            do

            {

                DateTime currentTime = DateTime.Now;

                string dt = $"{currentTime.ToString("dd/MM/yyyy h:mm tt")}";

                Console.WriteLine(dt);

                Console.WriteLine("Name: Uday Sheth");

                Console.WriteLine("Roll No : 62");

                Console.WriteLine("Select a menu:");

                Console.WriteLine("1. Star pattern 1");

                Console.WriteLine("2. Star pattern 2");

                Console.WriteLine("3. Star pattern 3");

                Console.WriteLine("4. Star pattern 4");

                Console.WriteLine("5. Exit");

                Console.WriteLine("Enter your choice: ");

                choice = Convert.ToInt32(Console.ReadLine());

                switch (choice)

                {

                    case 1:

                        PrintStarPattern1();

                        break;

                    case 2:

                        PrintStarPattern2();

                        break;

                    case 3:

                        PrintStarPattern3();

                        break;

                    case 4:

                        PrintStarPattern4();

                        break;

                    case 5:

                        break;

                    default:

                        Console.WriteLine("Invalid choice!");

                        break;

                }

            } while (choice != 5);

        }

        static void PrintStarPattern1()

        {

            for (int i = 1; i <= 4; i++)

            {

                for (int j = 0; j < i; j++)

                {

                    Console.Write("\*");

                }

                Console.WriteLine();

            }

        }

        static void PrintStarPattern2()

        {

            for (int i = 0; i < 5; i++)

            {

                for (int j = 0; j <= i; j++)

                {

                    Console.Write("\*");

                }

                Console.WriteLine();

            }

        }

        static void PrintStarPattern3()

        {

            for (int i = 0; i < 5; i++)

            {

                for (int j = 4; j >= i; j--)

                {

                    Console.Write("\*");

                }

                Console.WriteLine();

            }

        }

        static void PrintStarPattern4()

        {

            for (int i = 0; i < 5; i++)

            {

                for (int j = 0; j < 5 - i; j++)

                {

                    Console.Write(" ");

                }

                for (int j = 0; j <= i; j++)

                {

                    Console.Write("\*");

                }

                Console.WriteLine();

            }

        }

    }

}

