1)simple arithmetic calculator

// Implementing a calculator in

// C# using switch statement.

using System;

using System.Text;

using System.Threading.Tasks;

namespace calculator\_c\_sharp

{

class Program

{

static void Main(string[] args)

{

string value;

do

{

int res;

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

int num1 = Convert.ToInt32(Console.ReadLine());

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

int num2 = Convert.ToInt32(Console.ReadLine());

Console.Write("Enter symbol(/,+,-,\*):");

string symbol = Console.ReadLine();

switch (symbol)

{

case "+":

res = num1 + num2;

Console.WriteLine("Addition:" + res);

break;

case "-":

res = num1 - num2;

Console.WriteLine("Subtraction:" + res);

break;

case "\*":

res = num1 \* num2;

Console.WriteLine("Multiplication:" + res);

break;

case "/":

res = num1 / num2;

Console.WriteLine("Division:" + res);

break;

default:

Console.WriteLine("Wrong input");

break;

}

Console.ReadLine();

Console.Write("Do you want to continue(y/n):");

value = Console.ReadLine();

}

while (value=="y" || value=="Y");

}

}

}

2)armstrong

/\*

\* C# Program to Print all the Armstrong Numbers from 1 to 1000

\*/

using System;

class Program

{

static void Main()

{

int a, b, c, d;

for (int i = 1; i <= 1000; i++)

{

a = i / 100;

b = (i - a \* 100) / 10;

c = (i - a \* 100 - b \* 10);

d = a \* a \* a + b \* b \* b + c \* c \* c;

if (i == d)

{

System.Console.WriteLine("{0}", i);

}

}

Console.Read();

}

}

3)substring of string

using System;

namespace LogicalPrograms

{

class Program

{

static void Main(string[] args)

{

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

string inputString = Console.ReadLine();

int len = inputString.Length;

Console.WriteLine("All substrings for given string are : ");

//This loop maintains the starting character

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

{

//This loop adds the next character every iteration for the substring and then print

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

{

Console.Write (inputString.Substring(i, j + 1) + " ");

}

}

Console.ReadKey();

}

}

}

4)division by zero

C# PROGRAM-4

Develop a C# program to demonstrate Division by Zero and Index Out of Range exceptions.

using System;

class Program

{

static void Main()

{

// Division by Zero Exception

try

{

int numerator = 10;

int denominator = 0;

int result = numerator / denominator; // Attempting to divide by zero

Console.WriteLine("Result of Division: " + result);

}

catch (DivideByZeroException ex)

{

Console.WriteLine("Division by Zero Exception: " + ex.Message);

}

Console.WriteLine();

// Index Out of Range Exception

try

{

int[] numbers = { 1, 2, 3, 4, 5 };

int index = 10; // Attempting to access an index that is out of range

int value = numbers[index];

Console.WriteLine("Value at index " + index + ": " + value);

}

catch (IndexOutOfRangeException ex)

{

Console.WriteLine("Index Out of Range Exception: " + ex.Message);

}

}

}

5)pascals

C# program 5

Develop a C# program to generate and printPascal Triangle using Two Dimensional arrays.

using System;

class Program

{

static void Main()

{

Console.Write("Enter the number of rows for Pascal's Triangle: ");

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

int[,] pascalTriangle = GeneratePascalsTriangle(numRows);

Console.WriteLine("\nPascal's Triangle:");

PrintPascalsTriangle(pascalTriangle);

}

static int[,] GeneratePascalsTriangle(int numRows)

{

int[,] triangle = new int[numRows, numRows];

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

{

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

{

if (j == 0 || j == i)

{

triangle[i, j] = 1; // First and last element in each row is 1

}

else

{

triangle[i, j] = triangle[i - 1, j - 1] + triangle[i - 1, j];

}

}

}

return triangle;

}

static void PrintPascalsTriangle(int[,] triangle)

{

for (int i = 0; i < triangle.GetLength(0); i++)

{

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

{

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

}

Console.WriteLine();

}

}

}

6)Floyds

using System;

class Program

{

static void Main()

{

Console.Write("Enter the number of rows for Floyd's Triangle: ");

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

int[][] floydsTriangle = GenerateFloydsTriangle(numRows);

Console.WriteLine("\nFloyd's Triangle:");

PrintFloydsTriangle(floydsTriangle);

}

static int[][] GenerateFloydsTriangle(int numRows)

{

int[][] triangle = new int[numRows][];

int count = 1;

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

{

triangle[i] = new int[i + 1];

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

{

triangle[i][j] = count++;

}

}

return triangle;

}

static void PrintFloydsTriangle(int[][] triangle)

{

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

{

for (int j = 0; j < triangle[i].Length; j++)

{

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

}

Console.WriteLine();

}

}

}