Text

Description automatically generated

Example of given example

Text

Description automatically generated with medium confidence

Example with a sequence of 15. Maintains spacing.

Text

Description automatically generatedExample with an odd sequence (non-factor of 5) that display how format remains even.

using System;

int Sequence, Fibonacci, F\_n1, F\_n2;

Fibonacci = 0;

F\_n1 = 0;

F\_n2 = 1;

String[] Fib\_Array = new String[32]; //Note Im not using [0] and [31] , Im aware both exist.

Fib\_Array[0] = "0";

Fib\_Array[1] = "1";

Console.Write("Enter the sequence of Fibonacci: ");

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

Console.Write("\t");

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

{

Console.Write(Fib\_Array[i] + "\t");

Fibonacci = F\_n1 + F\_n2;

Fib\_Array[i+1] = Fibonacci.ToString();

F\_n1 = F\_n2;

F\_n2 = Fibonacci;

if (i % 5 == 0)

{

Console.WriteLine("");

Console.Write("\t");

}

}

Console.WriteLine("");

Graphical user interface, text, application, chat or text message

Description automatically generatedGraphical user interface, text, application

Description automatically generatedGraphical user interface, text

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace HW\_Part\_2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void Lista\_Box\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void Main\_Btn\_Click(object sender, EventArgs e)

{

int User\_Input;

String User\_Input\_Validation = User\_Box.Text;

bool validation = int.TryParse(User\_Input\_Validation, out User\_Input);

if (validation)

{

if (User\_Input == 0)

{

Lista\_Box.Items.Add("The factorial of 0! = 0");

}

else if (User\_Input < 0)

{

User\_Input \*= -1;

MessageBox.Show("Remember that the final result you get in your list will negative!");

}

int[] n\_factorial = new int[User\_Input];

long factorial = 1;

int count = 0;

Lista\_Box.Items.Add("Let's break down the factorial of " + User\_Input + "! :");

while (count < User\_Input)

{

n\_factorial[count] = count + 1;

factorial \*= n\_factorial[count];

Lista\_Box.Items.Add("The factorial of " + n\_factorial[count] + "! is = " + factorial);

count++;

}

}

else

{

MessageBox.Show("Only integers");

}

}

private void Second\_Btn\_Click(object sender, EventArgs e)

{

int User\_Input;

String User\_Input\_Validation = User\_Box.Text;

bool validation = int.TryParse(User\_Input\_Validation, out User\_Input);

if (validation)

{

if (User\_Input <= 0)

{

MessageBox.Show("Incorrect input, only positive integers");

}

else

{

int[] e\_factorial = new int[User\_Input];

int count = 0;

double factorial, e\_value;

factorial = 1;

e\_value = 1;

while (count < User\_Input)

{

e\_factorial[count] = count + 1;

factorial \*= e\_factorial[count];

e\_value += Convert.ToDouble((1 / (factorial)));

Lista\_Box.Items.Add("The approx value of e to the factorial " + e\_factorial[count] + " is = " + Math.Round(e\_value, 6));

count++;

}

}

}

else

{

MessageBox.Show("Only integers");

}

}

private void Clear\_Btn\_Click(object sender, EventArgs e)

{

Lista\_Box.Items.Clear();

}

private void Exit\_Btn\_Click(object sender, EventArgs e)

{

Close();

}

}

}