1.    Write a code that by given name prints on the console "**Hello, <name>!"** (for example: "**Hello, Peter!**").

using System;

namespace Detyra1

{

class Program

{

static void ReturnName(string name)

{

Console.WriteLine("Hello {0}", name);

}

static void Main(string[] args)

{

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

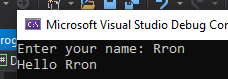
ReturnName(Console.ReadLine());

Console.ReadKey();

}

}

}



2.    Create a method **GetMax()** with two integer (**int**) parameters, that returns **maximal** of the two numbers. Write a program that reads three numbers from the console and prints the biggest of them. Use the **GetMax()**method you just created. Write a test program that validates that the methods works correctly.

using System;

namespace Detyra2

{

class Program

{

static int value;

static void GetMax(int num1, int num2)

{

if (num1 > num2) value = num1;

else value = num2;

}

static void Main(string[] args)

{

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

int number1 = int.Parse(Console.ReadLine());

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

int number2 = int.Parse(Console.ReadLine());

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

int number3 = int.Parse(Console.ReadLine());

GetMax(number1, number2);

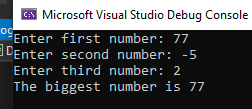
GetMax(number1, number3);

Console.WriteLine("The biggest number is {0}", value);

}

}

}



3.    Write a method that returns the **English name of the last digit** of a given number. Example: for **512** prints "**two**"; for **1024** à "**four**".

using System;

namespace Detyra3

{

class Program

{

static void Main(string[] args)

{

Console.Write("Enter number: ");

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

LastNumber(num);

}

public static void LastNumber(int num)

{

int lastNum = num % 10;

switch (lastNum)

{

case 0:

Console.WriteLine("Zero");

break;

case 1:

Console.WriteLine("One");

break;

case 2:

Console.WriteLine("Two");

break;

case 3:

Console.WriteLine("Three");

break;

case 4:

Console.WriteLine("Four");

break;

case 5:

Console.WriteLine("Five");

break;

case 6:

Console.WriteLine("Six");

break;

case 7:

Console.WriteLine("Seven");

break;

case 8:

Console.WriteLine("Eight");

break;

case 9:

Console.WriteLine("Nine");

break;

default:

Console.WriteLine("Wrong Input");

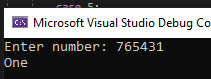
break;

}

}

}

}



4.    Write a method that finds **how many times certain number can be found in a given array**. Write a program to test that the method works correctly.

using System;

namespace Detyra4

{

class Program

{

static void Main(string[] args)

{

Console.Write("Enter the length of array: ");

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

FillArray(length);

}

static void FillArray(int length)

{

int[] arr = new int[length];

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

{

Console.Write("Fill array[{0}]: ", i + 1);

arr[i] = int.Parse(Console.ReadLine());

}

Console.Write("Number to find: ");

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

NumberAppears(arr, number);

}

public static void NumberAppears(int[] arr, int number)

{

int count = 0;

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

{

if (arr[i] == number)

{

count++;

}

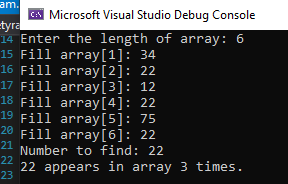
}

Console.WriteLine("{0} appears in array {1} times.", number, count);

}

}

}



5.    Write a method that checks whether an element, from a certain position in an array is **greater than its two neighbors**. Test whether the method works correctly.

6.    Write a method that returns the position of **the first occurrence** of an element from an array, such that it is greater than its two neighbors simultaneously. Otherwise the result must be **-1**.

using System;

namespace Detyra6

{

class Program

{

static void Main(string[] args)

{

Console.Write("Enter the length of array: ");

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

FillArray(length);

}

static void FillArray(int length)

{

int[] arr = new int[length];

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

{

Console.Write("Fill array[{0}]: ", i);

arr[i] = int.Parse(Console.ReadLine());

}

FindPosition(arr);

}

static void FindPosition(int[] arr)

{

int result = -1;

for (int i = 0; i < arr.Length - 1; i++)

{

if ((i == 0) || (i == arr.Length - 1))

{

continue;

}

else

{

if ((arr[i - 1] < arr[i]) && (arr[i] > arr[i + 1]))

{

result = i;

break;

}

}

}

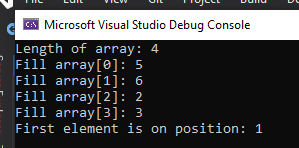
Console.WriteLine(result == -1 ? "-1\tDoen\'t have such element." :

"First element is on position: " + result);

}

}

}



7.    Write a method that prints the digits of a given decimal number in a reversed order. For example **256**, must be printed as **652**.

using System;

namespace Detyra7

{

class Program

{

static void Main(string[] args)

{

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

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

Reverse(number);

}

static void Reverse(int number)

{

int reversed = 0;

int c = 0;

while (number != 0)

{

c = number % 10;

reversed = (10 \* reversed) + c;

number /= 10;

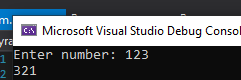
}

Console.WriteLine(reversed);

}

}

}



8.    Write a method that calculates the **sum of two very long positive integer numbers**. The numbers are represented as **array digits** and the last digit (the ones) is stored in the array at index 0. Make the method work for all numbers with length up to 10,000 digits.

using System;

namespace Detyra8

{

class Program

{

static void Main(string[] args)

{

Console.Write("First number: ");

string num1 = Console.ReadLine();

Console.Write("Second number: ");

string num2 = Console.ReadLine();

ToArrays(num1, num2);

}

static void ToArrays(string num1, string num2)

{

int[] firstArr = new int[num1.Length];

int[] secondArr = new int[num2.Length];

firstArr[0] = Convert.ToInt32(num1[num1.Length - 1].ToString(), 10);

secondArr[0] = Convert.ToInt32(num2[num2.Length - 1].ToString(), 10);

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

{

firstArr[i] = Convert.ToInt32(num1[i - 1].ToString(), 10);

}

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

{

secondArr[i] = Convert.ToInt32(num2[i - 1].ToString(), 10);

}

PrintArrays(firstArr, secondArr);

}

static void PrintArrays(int[] first, int[] second)

{

Console.WriteLine("First:");

foreach (var item in first)

{

Console.Write(item + " ");

}

Console.WriteLine("\nSecond:");

foreach (var item in second)

{

Console.Write(item + " ");

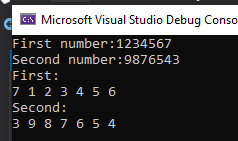
}

Console.WriteLine();

}

}

}



9.    Write a method that finds **the biggest element of an array**. Use that method to implement **sorting in descending order**.

using System;

namespace Detyra9

{

class Program

{

static void Main(string[] args)

{

Console.Write("Enter array length:");

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

Console.Write("Start at index:");

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

int[] arr = new int[length];

FillArray(arr);

FindMaxEl(arr, index);

Sort(arr);

}

static void FillArray(int[] arr)

{

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

{

Console.Write("Fill array[{0}]:", i + 1);

arr[i] = int.Parse(Console.ReadLine());

}

}

static void FindMaxEl(int[] arr, int index)

{

int max = int.MinValue;

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

{

if (arr[i] > max)

{

max = arr[i];

}

}

Console.WriteLine("Max element is:" + max);

}

static void Sort(int[] arr)

{

int choose = 0;

do

{

Console.WriteLine("1. Ascending");

Console.WriteLine("2. Descending");

Console.Write("Your choose:");

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

} while (choose < 1 || choose > 2);

if (choose == 1)

{

Array.Sort(arr);

}

else

{

Array.Sort(arr);

Array.Reverse(arr);

}

Console.WriteLine("Sorted array:");

foreach (var item in arr)

{

Console.Write(item + " ");

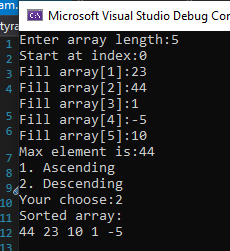
}

Console.WriteLine();

}

}

}



10.   Write a program that calculates and prints the **n!** for any **n** in the range **[1**…**100]**.

using System;

using System.Numerics;

namespace Detyra10

{

class Program

{

static void Main(string[] args)

{

for (int i = 1; i <= 100; i++)

{

Factorial(i);

}

Console.ReadKey();

}

static void Factorial(int number)

{

int num = number;

BigInteger factorial = 1;

do

{

factorial \*= number;

number--;

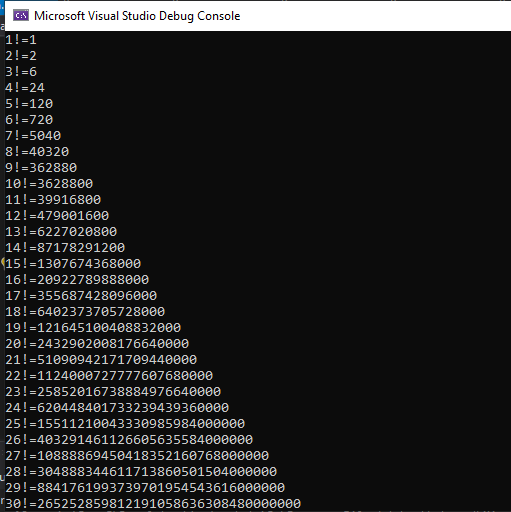
} while (number > 0);

Console.WriteLine("{0}!={1}", num, factorial);

}

}

}



11.   Write a program that solves the following tasks:

-     Put the digits from an integer number into a reversed order.

-     Calculate the average of given sequence of numbers.

-     Solve the linear equation **a \* x + b = 0**.

Create appropriate **methods** for each of the above tasks.

Make the program show a **text menu** to the user. By choosing an option of that menu, the user will be able to choose which task to be invoked.

Perform validation of the input data:

-     The integer number must be a positive in the range [1…50,000,000].

-     The sequence of numbers cannot be empty.

-     The coefficient **a** must be non-zero.

12.   Write a method that calculates the sum of two polynomials with integer coefficients, for example **(3x2 + x - 3) + (x - 1) = (3x2 + 2x - 4)**.

13.   \* Write a method that calculates the product of two polynomials with integer coefficients, for example **(3x2 + x - 3) \* (x - 1) = (3x3 - 2x2 - 4x + 3)**.