Detyra 1.

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.

Kodi: static int a;

static void GetMax(int first, int second)

{

if (first > second) a = first;

else a = second;

}

static void Main(string[] args)

{

Console.Write("Shkruaj numrin e pare: ");

a = Int32.Parse(Console.ReadLine());

Console.Write("Shkruaj numrin e dyte: ");

int b = Int32.Parse(Console.ReadLine());

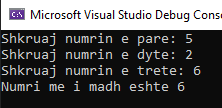
Console.Write("Shkruaj numrin e trete: ");

int c = Int32.Parse(Console.ReadLine());

GetMax(a, b);

GetMax(a, c);

Console.WriteLine("Numri me i madh eshte {0}", a);

REzultati:

Detyra 2.

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

Kodi: static void GetName(string number)

{

switch(number[number.Length - 1])

{

case '1': Console.WriteLine("Nje"); break;

case '2': Console.WriteLine("Dy"); break;

case '3': Console.WriteLine("Tre"); break;

case '4': Console.WriteLine("Kater"); break;

case '5': Console.WriteLine("Pese"); break;

case '6': Console.WriteLine("Gjashte"); break;

case '7': Console.WriteLine("Shtate"); break;

case '8': Console.WriteLine("Tete"); break;

case '9': Console.WriteLine("Nente"); break;

case '0': Console.WriteLine("Zero"); break;

}

}

static void Main(string[] args)

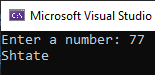
{

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

string number = Console.ReadLine();

GetName(number);

}

Rezultati:

Detyra 3.

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.

Kodi: static int CountNumber(int numri, int[] arr)

{

int counter = 0;

for (int i = 0; i < arr.Length; i++) if (numri == arr[i]) counter++;

return counter;

}

static void Main(string[] args)

{

Console.Write("Shkruaj gjatsine e array: ");

int gjatsia = Int32.Parse(Console.ReadLine());

int[] arr = new int[gjatsia];

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

{

Console.Write("Elementi i array{0}:", i);

arr[i] = Int32.Parse(Console.ReadLine());

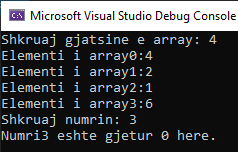
}

Console.Write("Shkruaj numrin: ");

int numri = Int32.Parse(Console.ReadLine());

Console.WriteLine("Numri{0} eshte gjetur {1} here.", numri, CountNumber(numri, arr));

}

Rezultati:

Detyra 4.

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.

Kodi:

static void CompareNumber(int pos, int[] arr)

{

if (pos == 0)

{

if (arr[0] < arr[1]) Console.WriteLine("{0} is smaller than it's right.", arr[0]);

else if (arr[0] > arr[1]) Console.WriteLine("{0} is bigger than it's right.", arr[0]);

else Console.WriteLine("{0} is equal to it's right.", arr[0]);

}

else if (pos == arr.Length - 1)

{

if (arr[arr.Length - 1] < arr[arr.Length - 2]) Console.WriteLine("{0} is smaller than it's left.", arr[pos]);

else if (arr[arr.Length - 1] > arr[arr.Length - 2]) Console.WriteLine("{0} is bigger than it's left.", arr[pos]);

else Console.WriteLine("{0} is equal to it's left.", arr[pos]);

}

else

{

if(arr[pos] < arr[pos - 1])

{

if (arr[pos] < arr[pos + 1]) Console.WriteLine("{0} is smaller than it's neighbours.", arr[pos]);

else if (arr[pos] == arr[pos + 1]) Console.WriteLine("{0} is smaller than it's left and equal to it's right.", arr[pos]);

else Console.WriteLine("{0} is smaller than it's left and bigger than it's right.", arr[pos]);

}

else if (arr[pos] == arr[pos - 1])

{

if (arr[pos] < arr[pos + 1]) Console.WriteLine("{0} is euqal to it's left and smaller than it's right.", arr[pos]);

else if (arr[pos] == arr[pos + 1]) Console.WriteLine("{0} is equal to it's neighbours.", arr[pos]);

else Console.WriteLine("{0} is equal to it's left and bigger than it's right.", arr[pos]);

}

else

{

if (arr[pos] < arr[pos + 1]) Console.WriteLine("{0} is bigger than it's left and smaller than it's right.", arr[pos]);

else if (arr[pos] == arr[pos + 1]) Console.WriteLine("{0} is bigger than it's left and equal to it's right.", arr[pos]);

else Console.WriteLine("{0} is bigger than it's neighbours.", arr[pos]);

}

}

}

static void Main(string[] args)

{

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

int length = Int32.Parse(Console.ReadLine());

int[] arr = new int[length];

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

{

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

arr[i] = Int32.Parse(Console.ReadLine());

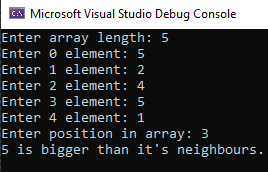
}

Console.Write("Enter position in array: ");

int pos = Int32.Parse(Console.ReadLine());

CompareNumber(pos, arr);

}

Rezultati:

Detyra 5.

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**.

Kodi:

static int number = int.MinValue;

static void CompareNumber(int[] arr)

{

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

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

{

number = arr[i];

break;

}

}

static void Main(string[] args)

{

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

int length = Int32.Parse(Console.ReadLine());

int[] arr = new int[length];

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

{

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

arr[i] = Int32.Parse(Console.ReadLine());

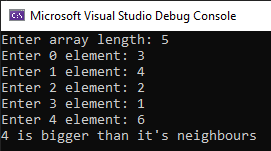
}

CompareNumber(arr);

if (number == int.MinValue) Console.WriteLine("-1");

else Console.WriteLine("{0} is bigger than it's neighbours", number);

}

Rezultati:

Detyra 6.

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

Kodi: public static string Reverse(string number)

{

char[] charArray = number.ToCharArray();

Array.Reverse(charArray);

return new string(charArray);

}

static void Main(string[] args)

{

Console.Write("Enter number: ");

string number = Console.ReadLine();

Console.WriteLine(Reverse(number));

}

Rezultati: