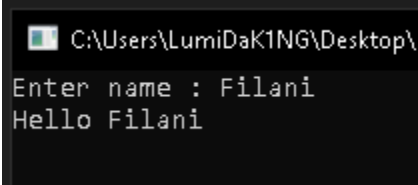


# Chapter 9. Methods

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

```
using System;

namespace ex1
{
    class Program
    {
        static void Hello(string name)
        {
            Console.WriteLine("Hello {0}", name);
        }
        static void Main(string[] args)
        {
            Console.Write("Enter name : ");
            Hello(Convert.ToString(Console.ReadLine()));
            Console.ReadKey();
        }
    }
}
```



```
C:\Users\LumiDaK1NG\Desktop\
Enter name : Filani
Hello Filani
```

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;

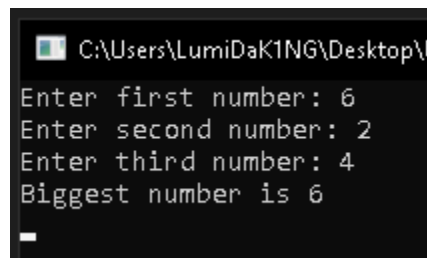
namespace ex2
{
    class Program
    {
        static int a;

        static void GetMax(int first, int second)
        {
            if (first > second) a = first;
            else a = second;
        }

        static void Main(string[] args)
        {
            Console.Write("Enter first number: ");
            a = Int32.Parse(Console.ReadLine());
            Console.Write("Enter second number: ");
            int b = Int32.Parse(Console.ReadLine());
            Console.Write("Enter third number: ");
            int c = Int32.Parse(Console.ReadLine());

            GetMax(a, b);
            GetMax(a, c);

            Console.WriteLine("Biggest number is {0}", a);
            Console.ReadKey();
        }
    }
}
```



```
C:\Users\LumiDaK1NG\Desktop\L
Enter first number: 6
Enter second number: 2
Enter third number: 4
Biggest number is 6
```

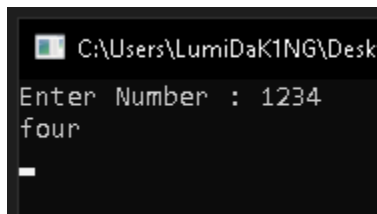
3. Write a method that returns the English name of the last digit of a given number. Example: for 512 prints "two"; for 1024 à "four".

```
using System;

namespace ex3
{
    class Program
    {
        static void GetLast(string number)
        {
            switch (number[number.Length - 1]) {
                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;
            }
        }

        static void Main(string[] args)
        {
            Console.Write("Enter Number : ");
            string num = Console.ReadLine();

            GetLast(num);
            Console.ReadKey();
        }
    }
}
```



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 ex4
{
    class Program
    {
        static int GetCount(int num, int[] arr)
        {
            int count = 0;

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

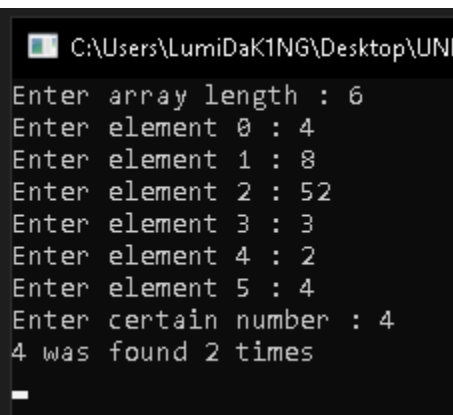
            return count;
        }
        static void Main(string[] args)
        {
            Console.Write("Enter array length : ");
            int length = Int32.Parse(Console.ReadLine());

            int[] arr = new int[length];

            for(int i = 0; i < arr.Length; i++)
            {
                Console.Write("Enter element {0} : ", i);
                arr[i] = Int32.Parse(Console.ReadLine());
            }

            Console.Write("Enter certain number : ");
            int num = Int32.Parse(Console.ReadLine());

            Console.WriteLine("{0} was found {1} times", num, GetCount(num, arr));
            Console.ReadKey();
        }
    }
}
```



```
C:\Users\LumiDaK1NG\Desktop\UNI
Enter array length : 6
Enter element 0 : 4
Enter element 1 : 8
Enter element 2 : 52
Enter element 3 : 3
Enter element 4 : 2
Enter element 5 : 4
Enter certain number : 4
4 was found 2 times
```

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.

```
using System;

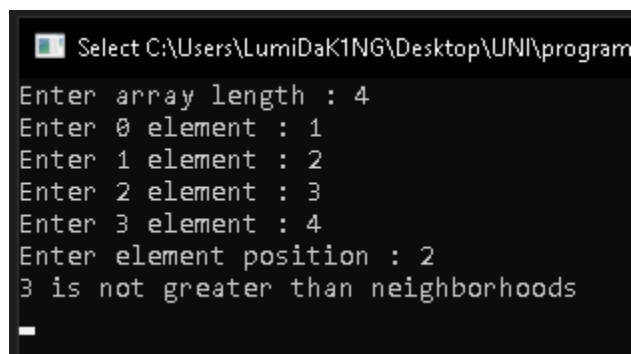
namespace ex5
{
    class Program
    {
        static void Compare(int pos, int[] arr)
        {
            if (arr[pos] > arr[pos + 1] & arr[pos] > arr[pos - 1])
                Console.WriteLine("{0} is greater than neighborhoods", arr[pos]);
            else Console.WriteLine("{0} is not greater than neighborhoods", 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 element position : ");
            int pos = Int32.Parse(Console.ReadLine());

            Compare(pos, arr);
            Console.ReadKey();
        }
    }
}
```



```
Select C:\Users\LumiDaK1NG\Desktop\UNI\program
Enter array length : 4
Enter 0 element : 1
Enter 1 element : 2
Enter 2 element : 3
Enter 3 element : 4
Enter element position : 2
3 is not greater than neighborhoods
```

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 ex6
```

```
{
```

```
    class Program
```

```
    {
```

```
        static void GetGreater(int[] arr)
```

```
        {
```

```
            int greater = 0;
```

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

```
            {
```

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

```
                {
```

```
                    greater = arr[i];
```

```
                    break;
```

```
                }
```

```
            }
```

```
            Console.WriteLine("{0} is greater than is neighborhoods", greater);
```

```
        }
```

```
        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());
```

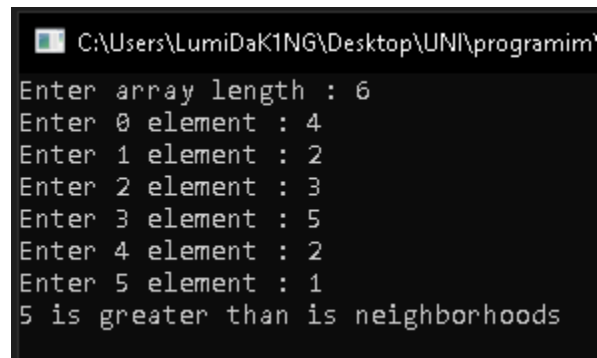
```
            }
```

```
            GetGreater(arr);
```

```
            Console.ReadKey();
```

```
        }
```

```
    } }
```



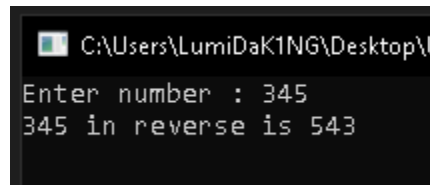
```
C:\Users\LumiDaK1NG\Desktop\UNI\programim>
Enter array length : 6
Enter 0 element : 4
Enter 1 element : 2
Enter 2 element : 3
Enter 3 element : 5
Enter 4 element : 2
Enter 5 element : 1
5 is greater than is neighborhoods
```

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 ex7
{
    class Program
    {
        static string Reverse(string num)
        {
            char[] numSplit = num.ToCharArray();
            Array.Reverse(numSplit);
            return new string(numSplit);
        }
        static void Main(string[] args)
        {
            Console.Write("Enter number : ");
            string num = Console.ReadLine();

            Console.WriteLine("{0} in reverse is {1}", num, Reverse(num));
            Console.ReadKey();
        }
    }
}
```



```
C:\Users\LumiDaK1NG\Desktop\L
Enter number : 345
345 in reverse is 543
```

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 ex8
{
    class Program
    {
        static void Main(string[] args)
        {
            int length;
            int[] Array1 = new int[10000];
            int[] Array2 = new int[10000];

            Console.WriteLine("Enter first number: ");
            String s1 = Console.ReadLine();
            length = s1.Length;

            for (int i = 0; i < s1.Length; i++)
            {
                Array1[i] = Convert.ToInt32(s1.Substring(s1.Length - 1 - i, 1));
            }

            Console.WriteLine("Enter second number: ");
            String s2 = Console.ReadLine();

            if (s2.Length > length)
            {
                length = s2.Length;
            }

            for (int i = 0; i < s2.Length; i++)
            {
                Array2[i] = Convert.ToInt32(s2.Substring(s2.Length - 1 - i, 1));
            }

            for (int i = 0; i < length; i++)
            {
                Array1[i] += Array2[i];

                if (Array1[i] >= 10)
                {
                    Array1[i] -= 10;
                    Array1[i + 1]++;
                }
            }

            if (Array1[length] == 1)
```



```

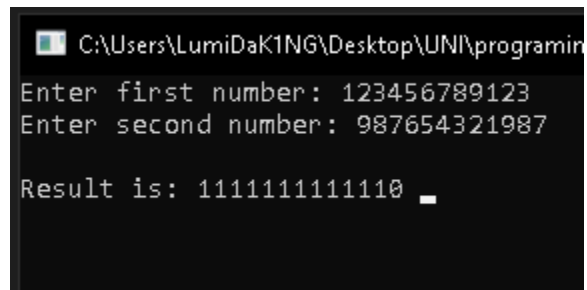
    {
        length++;
    }

    Console.WriteLine();
    Console.Write("Result is: ");

    for (int i = 0; i < length; i++)
    {
        Console.Write(Array1[length - 1 - i]);
    }

    Console.ReadLine();
    Console.ReadKey();
}
}
}

```



```

C:\Users\LumiDaK1NG\Desktop\UNI\programin
Enter first number: 123456789123
Enter second number: 987654321987

Result is: 1111111111110 _

```

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

```
using System;

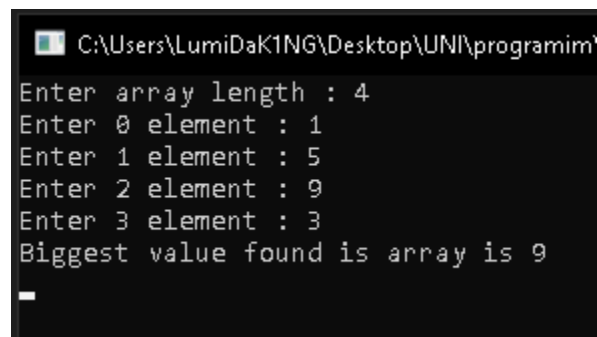
namespace ex9
{
    class Program
    {
        static void GetMax(int[] arr)
        {
            int maxValue = 0;

            for(int i = 0; i < arr.Length - 1; i++)
            {
                if (arr[i] > arr[i + 1]) maxValue = arr[ i ];
                else maxValue = arr[i + 1];
            }

            Console.WriteLine("Biggest value found is array is {0}", maxValue);
        }
        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());
            }
            GetMax(arr);
            Console.ReadKey();
        }
    }
}
```



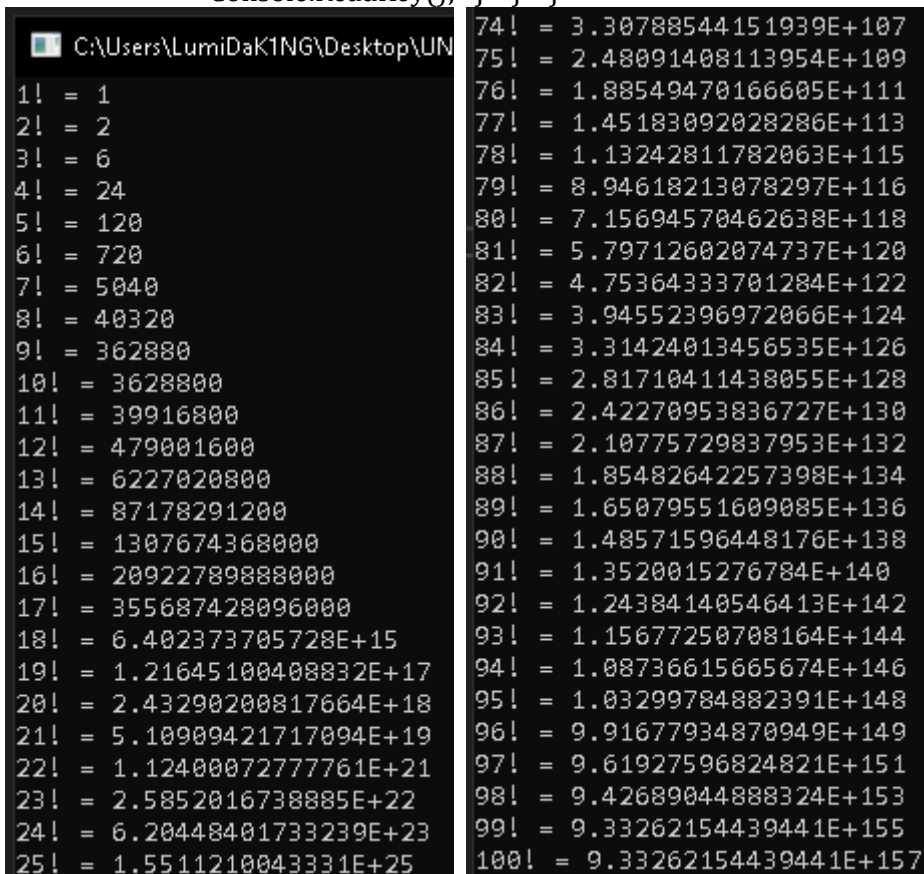
```
C:\Users\LumiDaK1NG\Desktop\UNI\programim
Enter array length : 4
Enter 0 element : 1
Enter 1 element : 5
Enter 2 element : 9
Enter 3 element : 3
Biggest value found is array is 9
```

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

```
using System;

namespace ex10
{
    class Program
    {
        static double Factorial(double number)
        {
            if (number <= 1)
                return 1;
            else
                return number * Factorial(number - 1);
        }

        static void Main(string[] args)
        {
            for (int i = 1; i < 101; i++)
            {
                Console.WriteLine("{0}! = {1}", i, Factorial(i));
            }
            Console.ReadLine();
            Console.ReadKey(); } } }
```



The screenshot shows a Windows command prompt window with the file path C:\Users\LumiDaK1NG\Desktop\UN. The output displays the factorial of numbers from 1 to 100. For numbers 1 through 20, the full integer value is shown. From 21 onwards, the values are displayed in scientific notation (E+). The output is as follows:

1! = 1	74! = 3.30788544151939E+107
2! = 2	75! = 2.48091408113954E+109
3! = 6	76! = 1.88549470166605E+111
4! = 24	77! = 1.45183092028286E+113
5! = 120	78! = 1.13242811782063E+115
6! = 720	79! = 8.94618213078297E+116
7! = 5040	80! = 7.15694570462638E+118
8! = 40320	81! = 5.79712602074737E+120
9! = 362880	82! = 4.75364333701284E+122
10! = 3628800	83! = 3.94552396972066E+124
11! = 39916800	84! = 3.31424013456535E+126
12! = 479001600	85! = 2.81710411438055E+128
13! = 6227020800	86! = 2.42270953836727E+130
14! = 87178291200	87! = 2.10775729837953E+132
15! = 1307674368000	88! = 1.85482642257398E+134
16! = 20922789888000	89! = 1.65079551609085E+136
17! = 355687428096000	90! = 1.48571596448176E+138
18! = 6.402373705728E+15	91! = 1.3520015276784E+140
19! = 1.21645100408832E+17	92! = 1.24384140546413E+142
20! = 2.43290200817664E+18	93! = 1.15677250708164E+144
21! = 5.10909421717094E+19	94! = 1.08736615665674E+146
22! = 1.12400072777761E+21	95! = 1.03299784882391E+148
23! = 2.5852016738885E+22	96! = 9.91677934870949E+149
24! = 6.20448401733239E+23	97! = 9.61927596824821E+151
25! = 1.5511210043331E+25	98! = 9.42689044888324E+153
	99! = 9.33262154439441E+155
	100! = 9.33262154439441E+157

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.

```
using System;

namespace ex11
{
    class Program
    {

        public static string ReverseString(string s)
        {
            char[] arr = s.ToCharArray();
            Array.Reverse(arr);
            return new string(arr);
        }

        static void Reverse()
        {
            int numberReverse;
            string stringNumberReverse;
            do
            {
                Console.Clear();
                Console.Write("Enter a positive number : ");
                stringNumberReverse = Console.ReadLine();
                numberReverse = int.Parse(stringNumberReverse);
            } while (numberReverse < 0);

            Console.WriteLine("The number is inverted " +
                ReverseString(stringNumberReverse));
            Console.ReadLine();
        }

        static void Average()
        {
            int numberAverage = 0;
            int entries = -1;
            int temp;
            string numberAverageString;
```

```

    bool input;
    Console.Clear();

    do
    {
        Console.Write("Enter a number from the row. Enter a letter to
complete : ");
        numberAverageString = Console.ReadLine();
        input = Int32.TryParse(numberAverageString, out temp);
        numberAverage += temp;
        entries++;
    } while (input);

    Console.WriteLine("The arithmetic mean is {0}.", (float)numberAverage
/ entries);
    Console.ReadLine();
}

static void SolveEquation()
{
    int a = 0;

    do
    {
        Console.Clear();
        Console.Write("Enter a : ");
        a = int.Parse(Console.ReadLine());
    } while (a == 0);

    Console.Write("Enter b : ");
    int b = int.Parse(Console.ReadLine());

    Console.WriteLine("x = {0}", (float)-b / a);
    Console.ReadLine();
}

static void Main(string[] args)
{
    byte choice;

    do
    {
        Console.Clear();

        Console.WriteLine("#####
#####");
        Console.WriteLine("# M M EEEEEEE N N U U #");
        Console.WriteLine("# M M M M E N N N U U #");
        Console.WriteLine("# M M M EEEEE N N N U U #");
        Console.WriteLine("# M M E N N N U U #");
    }
}

```

```

        Console.WriteLine("# N N EEEEEEE N N UUUUU #");

Console.WriteLine("#####
#####" + System.Environment.NewLine);
    Console.WriteLine("1.Inverting the sequence of a number.");
    Console.WriteLine("2.Arithmetic mean of a number of numbers.");
    Console.WriteLine("3.Solve the equation: a * c + b = 0.");
    Console.WriteLine("4.Exit.");
    Console.Write("Enter a selection: ");

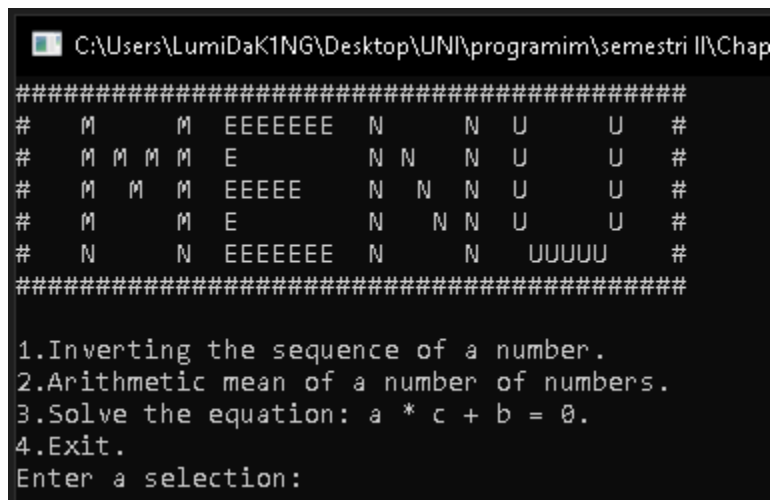
    choice = byte.Parse(Console.ReadLine());

    switch (choice)
    {
        case 1: Reverse(); break;
        case 2: Average(); break;
        case 3: SolveEquation(); break;
    }

} while (choice != 4);

Console.WriteLine(System.Environment.NewLine + "The end!");
Console.ReadLine();
}
}
}

```

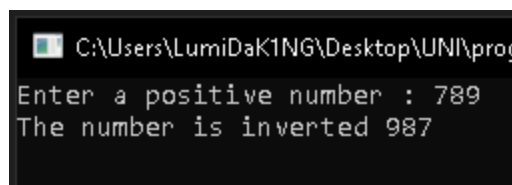


```

C:\Users\LumiDaK1NG\Desktop\UNI\programim\semestri II\Chap
#####
# M M EEEEEEE N N U U #
# M M M M E N N N U U #
# M M M EEEEE N N N U U #
# M M E N N N U U #
# N N EEEEEEE N N UUUUU #
#####

1.Inverting the sequence of a number.
2.Arithmetic mean of a number of numbers.
3.Solve the equation: a * c + b = 0.
4.Exit.
Enter a selection:

```



```

C:\Users\LumiDaK1NG\Desktop\UNI\prog
Enter a positive number : 789
The number is inverted 987

```