

# 1: Sensor with not valid data check

Consider the problem already assigned at the end of Lecture 1.

A sensor collects temperature measurements  $T$ . Write a program that prints different messages on the screen:

- Normal:  $T \leq 24\text{ C}$
- Warning:  $24\text{ C} < T \leq 30\text{ C}$
- Critical:  $T > 30\text{ C}$

And the solution discussed during Lecture 2:

```
class Program
{
    static void Main(string[] args)
    {
        int temperature;

        Console.WriteLine("Insert measurement:");
        temperature = Convert.ToInt32(Console.ReadLine());

        if (temperature > 30)
        {
            Console.WriteLine("Critical");
        }
        else if (temperature > 24)
        {
            Console.WriteLine("Warning");
        }
        else
        {
            Console.WriteLine("Normal");
        }
    }
}
```

## a) Reuse the above provided solution and modify the program to deal with non-valid measurements (with Tutor)

A measurement is not valid if it is greater than 100 or if it is less than 0. A message should be printed on the screen accordingly.

## b) Modify the code so that after a temperature value is read, and a corresponding message is printed, more values can be provided as input by the user (with Tutor)

Create a loop that reads and processes multiple temperature values. The program also has to keep track of the number of Critical, Warning and Not Valid measurements. The loop terminates when 3 invalid measurements are detected and prints on the screen the message "The sensor is broken". It should also print a report with the number of Critical, Warning and Not Valid measurements previously detected.

c) Modify the program by using a switch case instead of an if-else-if (independent work)

Refer to the example discussed and presented during Lecture 2.

## 2: Write a program that simulates the throw of two dice (with Tutor)

The program should continuously throw the two dice (at the same time) and print the results on the screen. It should stop when the outcome for both dice is equal to 6.

*Example*

*Dice 1: 4, Dice 2: 3*

*Dice 1: 6, Dice 2: 4*

*Dice 1: 6, Dice 2: 5*

*Dice 1: 1, Dice 2: 6*

*Dice 1: 4, Dice 2: 5*

*Dice 1: 6, Dice 2: 6*

To simulate a throw, generate a random integer value using the following instructions:

```
Random rand = new Random();
```

```
int n = rand.Next(1, 7);
```

Hint:

```
class Program
{
    static void Main(string[] args)
    {
        Random r = new Random();

        int dice1 = 1, dice2 = 1;

        while ( ) // complete with a proper logical expression
        {
            // random numbers from 1 to 6
            dice1 = r.Next(1, 7);
            dice2 = r.Next(1, 7);

            Console.WriteLine( );
        }
    }
}
```

## 3: Write a program that calculates the factorial of a number (independent work)

In mathematics, the factorial of a non-negative integer  $n$ , denoted by  $n!$ , is the product of all positive integers less than or equal to  $n$ :

$$n! = n \times (n-1) \times (n-2) \times (n-3) \times \dots \times 3 \times 2 \times 1$$

For example,  $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$

The number  $n$  should be provided as input from the keyboard.

Consider the maximum value of  $n$  that can be calculated using an `int` variable for its factorial. Can any other variable type be used?

Hint:

```
class Program
{
    static void Main(string[] args)
    {
        // read n as input from the keyboard

        int factorial = 1;

        // count used to represent the factors in the loop
        int count = n;
        while (count > 1)
        {
            // properly update factorial and count

        }

        // print result
    }
}
```