Array assignments

When creating the program code, you must apply the following basic principles:

- create a separate project for each assignment;
- use name ‘assignment1’, ‘assignment2’, etcetera for the projects;
- create one solution for each week containing the projects for that week;
- make sure the output of your programs is the same as the given screenshots;

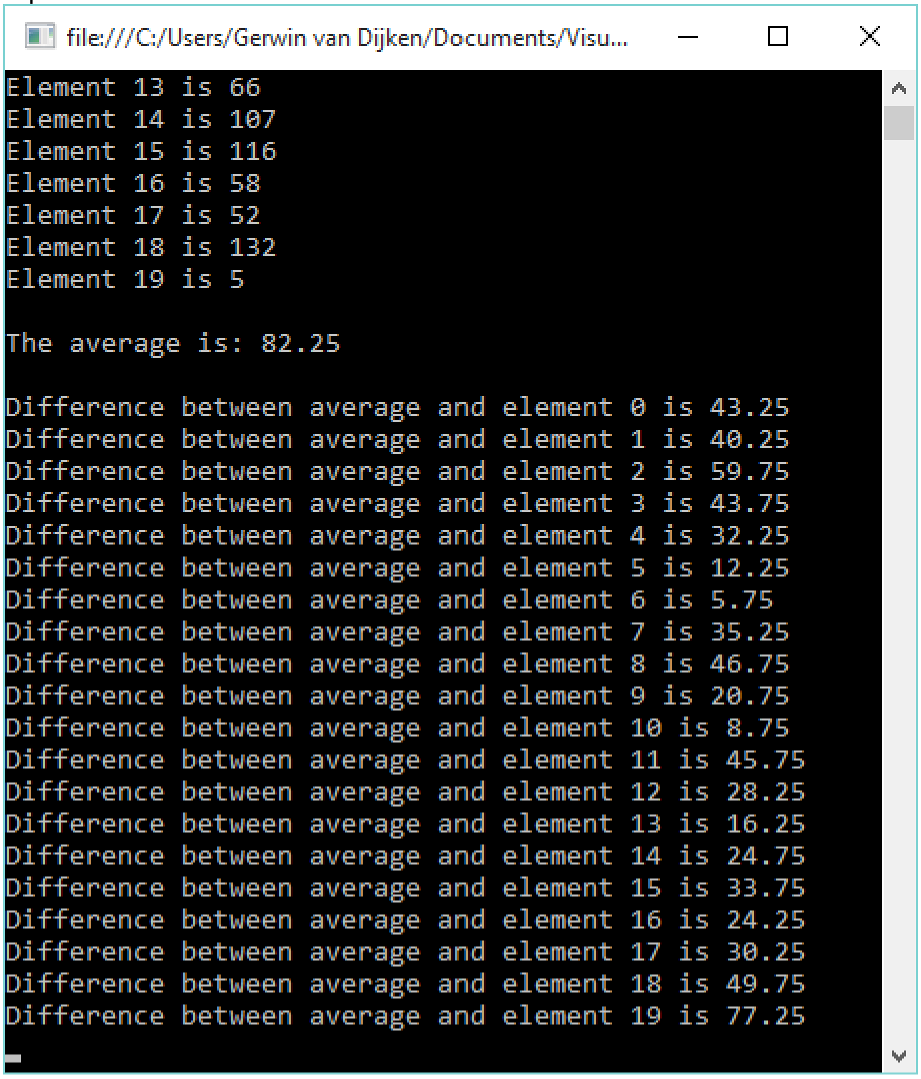
Note: for assignment 1 and assignment 3, your output must contain a dot (.) as a decimal separator, and not a comma (,), see screenshots of these assignments. To make sure your program uses a dot, add the following code to your program (2 using-statements and the code at the start of your Main-method):

```
using System;
using System.Globalization;
using System.Threading;

static void Main(string[] args)
{
    // set culture of program
    CultureInfo ci = new CultureInfo("en-US");
    Thread.CurrentThread.CurrentUICulture = ci;
    Thread.CurrentThread.CurrentCulture = ci;

    // your code here...
}
```

<u>Assignment 1 (Console App)</u>

The computer inserts 20 random numbers (with values between 0 and 200) in an array of 20 elements. After the array has been completely filled, all elements are shown and the average of the numbers in the array is determined and shown. Lastly, the difference between the numbers in the array with the average is shown.

A possible result is shown below:



```
Element 13 is 66
Element 14 is 107
Element 15 is 116
Element 16 is 58
Element 17 is 52
Element 18 is 132
Element 19 is 5

The average is: 82.25

Difference between average and element 0 is 43.25
Difference between average and element 1 is 40.25
Difference between average and element 2 is 59.75
Difference between average and element 3 is 43.75
Difference between average and element 4 is 32.25
Difference between average and element 5 is 12.25
Difference between average and element 6 is 5.75
Difference between average and element 7 is 35.25
Difference between average and element 8 is 46.75
Difference between average and element 9 is 20.75
Difference between average and element 10 is 8.75
Difference between average and element 11 is 45.75
Difference between average and element 12 is 28.25
Difference between average and element 13 is 16.25
Difference between average and element 14 is 24.75
Difference between average and element 15 is 33.75
Difference between average and element 16 is 24.25
Difference between average and element 17 is 30.25
Difference between average and element 18 is 49.75
Difference between average and element 19 is 77.25
```

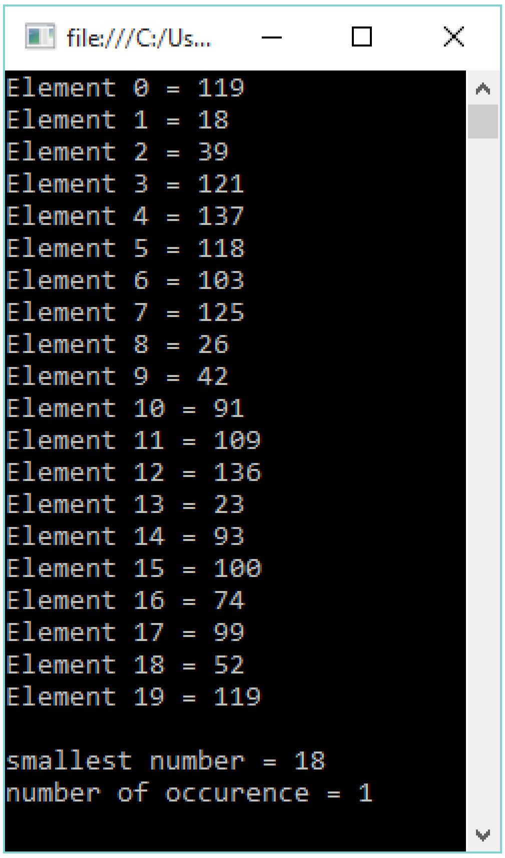
NB:

- try to obtain a similar output
- the numbers from the possible result obviously differ because they are generated randomly
- use `Math.Abs(...)` to obtain an absolute number (without a minus sign)

Tip: Carry out this assignment in parts: first fill in the array, then print the elements, then determine the average and lastly, the difference.

Assignment 2 (Console App)

Twenty random numbers (random values between 0 and 150) are inserted in an array. After that, the elements are shown, the smallest number is determined and how often it occurs. The smallest number and the number of times it occurs are printed.

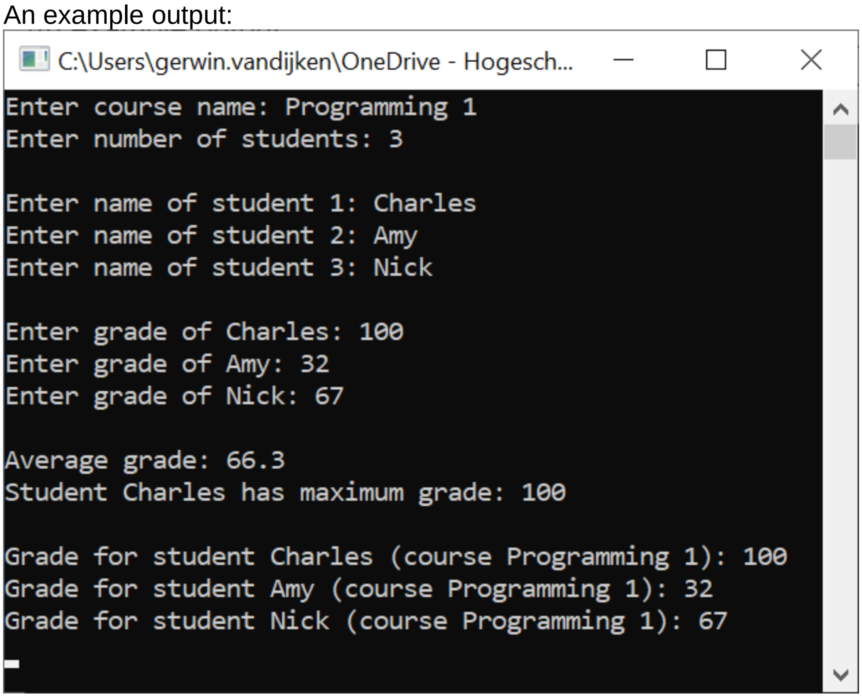


```
Element 0 = 119
Element 1 = 18
Element 2 = 39
Element 3 = 121
Element 4 = 137
Element 5 = 118
Element 6 = 103
Element 7 = 125
Element 8 = 26
Element 9 = 42
Element 10 = 91
Element 11 = 109
Element 12 = 136
Element 13 = 23
Element 14 = 93
Element 15 = 100
Element 16 = 74
Element 17 = 99
Element 18 = 52
Element 19 = 119

smallest number = 18
number of occurence = 1
```

Assignment 3 (Console App)
Ask for the name of a course. Then ask how many students have taken a test. After that, ask for the students' names, one by one. Lastly, ask for each student's grade.

After entering all the data, assign the average grade and the highest grade stating the name of the person who has been awarded the highest grade (if several students have the highest grade, only one grade should be shown). Lastly, provide a complete overview of all students and their grades.

An example output:



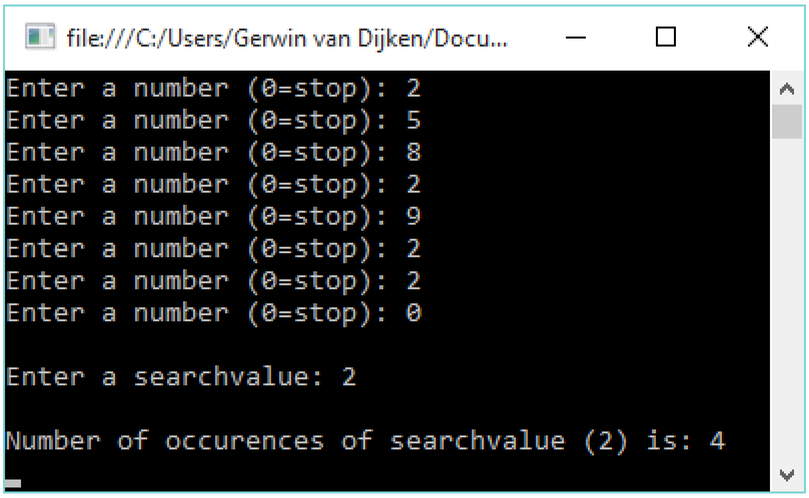
```
Enter course name: Programming 1
Enter number of students: 3

Enter name of student 1: Charles
Enter name of student 2: Amy
Enter name of student 3: Nick

Enter grade of Charles: 100
Enter grade of Amy: 32
Enter grade of Nick: 67

Average grade: 66.3
Student Charles has maximum grade: 100

Grade for student Charles (course Programming 1): 100
Grade for student Amy (course Programming 1): 32
Grade for student Nick (course Programming 1): 67
```

Tips:

- use two arrays: one for the names and one for the grades;
- the sizes of the two arrays are equal to the number of students entered.

Assignment 4 (Console App)

Numbers are entered and inserted in an array, the input ends with 0. A maximum of 20 numbers can be inserted in the array. If more numbers are provided, all numbers after the 20th number will not be processed. A new number will then be requested and entered - this is a search value. Determine whether this searchvalue appears in the array and how often.



```
Enter a number (0=stop): 2
Enter a number (0=stop): 5
Enter a number (0=stop): 8
Enter a number (0=stop): 2
Enter a number (0=stop): 9
Enter a number (0=stop): 2
Enter a number (0=stop): 2
Enter a number (0=stop): 0

Enter a searchvalue: 2

Number of occurences of searchvalue (2) is: 4
```

Assignment 5 (Console App) – this assignment is optional

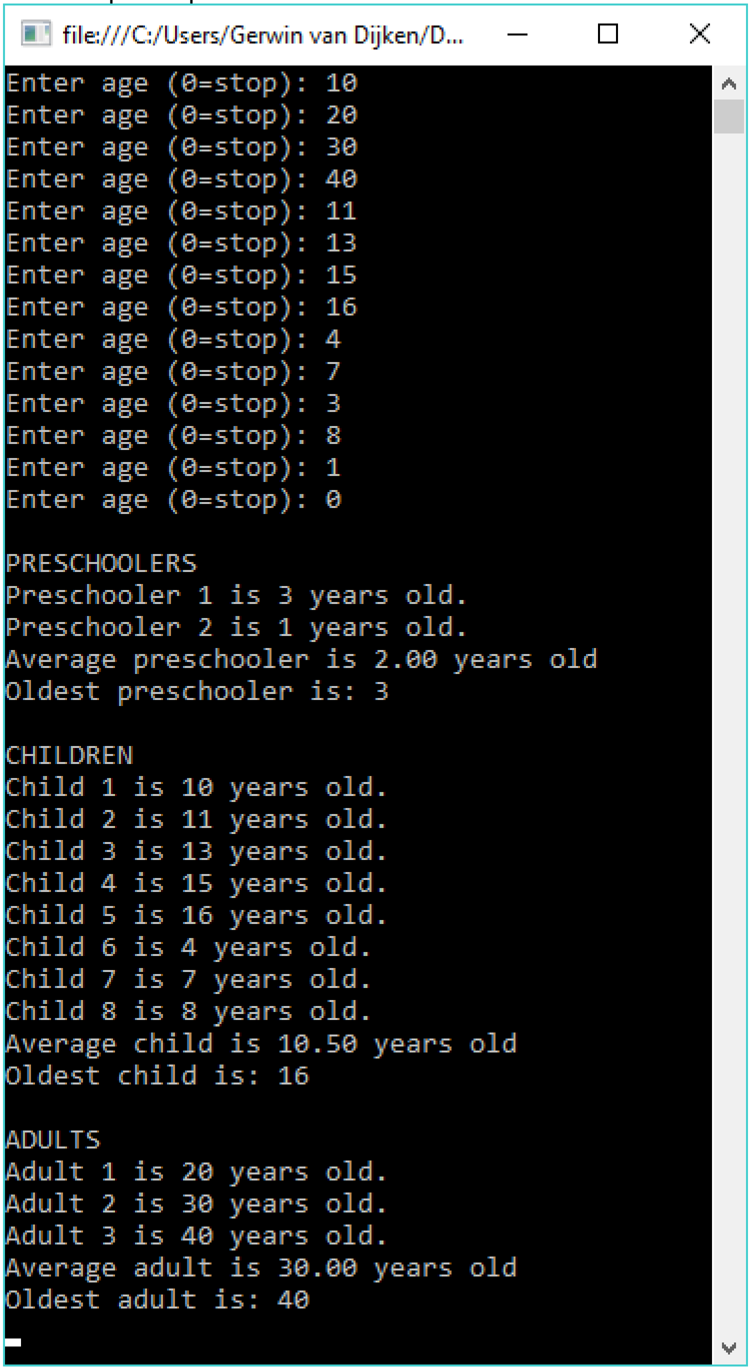
We have identified three different age categories in a population:

PRESCHOOLERS:	1 - 3 years
CHILDREN	4 - 17 years
ADULTS:	18 years and older

Enter an unknown number of ages (the input ends with 0). Insert each age entered in the correct category (each category is an array with 10 elements). There is no need to check whether more than 10 ages are in a category. Each category shows the following: the name of the category, the ages entered, the average age and the oldest age.

Have a look at what happens when more than 10 ages are entered in one specific category. Explain what happens.

An example output:



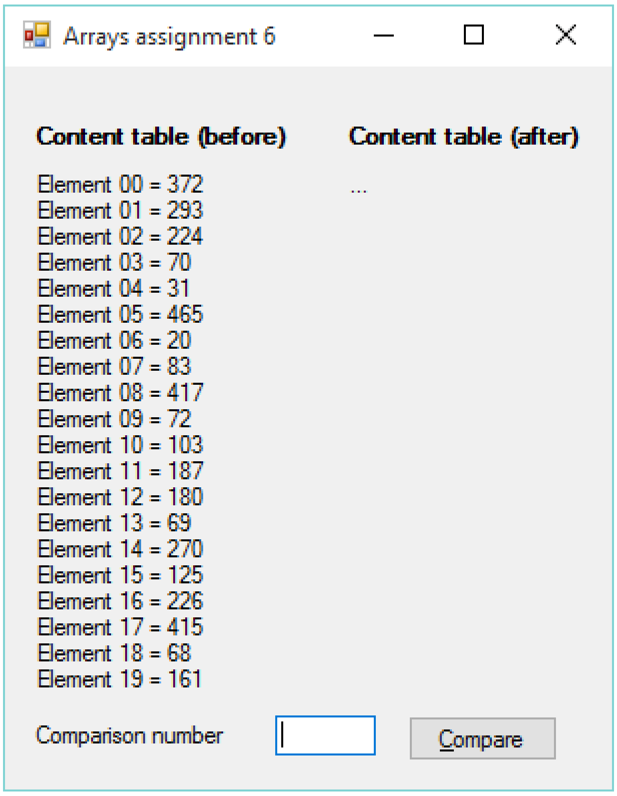
```
Enter age (0=stop): 10
Enter age (0=stop): 20
Enter age (0=stop): 30
Enter age (0=stop): 40
Enter age (0=stop): 11
Enter age (0=stop): 13
Enter age (0=stop): 15
Enter age (0=stop): 16
Enter age (0=stop): 4
Enter age (0=stop): 7
Enter age (0=stop): 3
Enter age (0=stop): 8
Enter age (0=stop): 1
Enter age (0=stop): 0

PRESCHOOLERS
Preschooler 1 is 3 years old.
Preschooler 2 is 1 years old.
Average preschooler is 2.00 years old
Oldest preschooler is: 3

CHILDREN
Child 1 is 10 years old.
Child 2 is 11 years old.
Child 3 is 13 years old.
Child 4 is 15 years old.
Child 5 is 16 years old.
Child 6 is 4 years old.
Child 7 is 7 years old.
Child 8 is 8 years old.
Average child is 10.50 years old
Oldest child is: 16

ADULTS
Adult 1 is 20 years old.
Adult 2 is 30 years old.
Adult 3 is 40 years old.
Average adult is 30.00 years old
Oldest adult is: 40
```

<u>Assignment 6 (Windows App)</u>

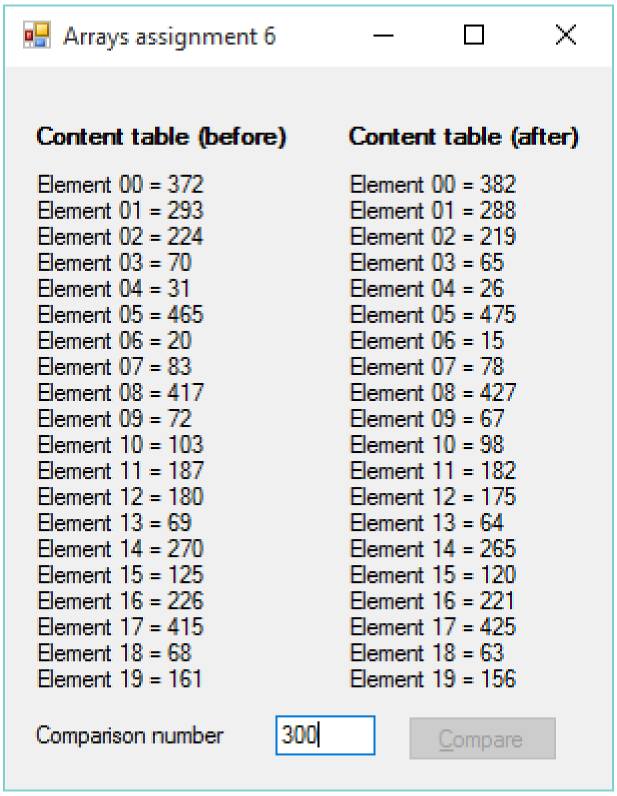
Twenty random numbers (random values between 0 and 500) are inserted in an array. These numbers are also printed. All numbers in the array are compared with one number entered (comparison number). If the value in the table is larger than or equal to that number, the value in the array is increased by 10. If the value in the table is smaller than that number, the value in the array is decreased by 5. All new numbers are also printed.

Comments:

- the array is filled when loading the form *(double click the form to create Form_Load event)*
- the array elements are compared after pressing the command button
- after the command button has been pressed, it should be disabled



```
Content table (before)     Content table (after)

Element 00 = 372           ...
Element 01 = 293
Element 02 = 224
Element 03 = 70
Element 04 = 31
Element 05 = 465
Element 06 = 20
Element 07 = 83
Element 08 = 417
Element 09 = 72
Element 10 = 103
Element 11 = 187
Element 12 = 180
Element 13 = 69
Element 14 = 270
Element 15 = 125
Element 16 = 226
Element 17 = 415
Element 18 = 68
Element 19 = 161

Comparison number
```



```
Content table (before)     Content table (after)

Element 00 = 372           Element 00 = 382
Element 01 = 293           Element 01 = 288
Element 02 = 224           Element 02 = 219
Element 03 = 70            Element 03 = 65
Element 04 = 31            Element 04 = 26
Element 05 = 465           Element 05 = 475
Element 06 = 20            Element 06 = 15
Element 07 = 83            Element 07 = 78
Element 08 = 417           Element 08 = 427
Element 09 = 72            Element 09 = 67
Element 10 = 103           Element 10 = 98
Element 11 = 187           Element 11 = 182
Element 12 = 180           Element 12 = 175
Element 13 = 69            Element 13 = 64
Element 14 = 270           Element 14 = 265
Element 15 = 125           Element 15 = 120
Element 16 = 226           Element 16 = 221
Element 17 = 415           Element 17 = 425
Element 18 = 68            Element 18 = 63
Element 19 = 161           Element 19 = 156

Comparison number    300
```

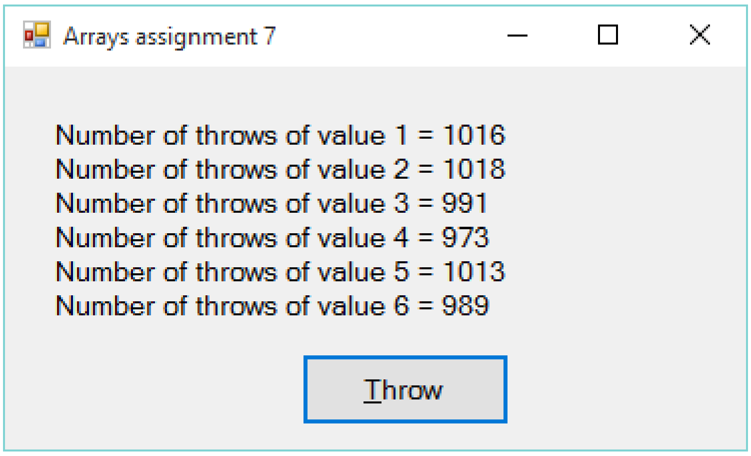
Assignment 7 (Windows App)
Create an array with 6 elements of the type integer named 'dice'. Fill this array with the value zero (so 6 times). Generate a random number with a value between 1 and 6 (just like a dice) in a repetition 6000 times.

If the value is 1, increase the element 0 in the array by 1, the same applies to the values 2, 3, 4, 5 and 6. The dice[0] element indicates the number of times value 1 has occurred. Or in general: dice[x-1] indicates the number of times that x has been thrown.

At the end of the repetition, show the contents of the array on your screen. If the dice is 100% random, what output would you expect?

Comments:

- only use the clicked event under the command button 'Throw'
- to generate a value between 1 and 6, you must use min. 1 and max. 7 for 'Random.Next(…)'



```
Number of throws of value 1 = 1016
Number of throws of value 2 = 1018
Number of throws of value 3 = 991
Number of throws of value 4 = 973
Number of throws of value 5 = 1013
Number of throws of value 6 = 989
```