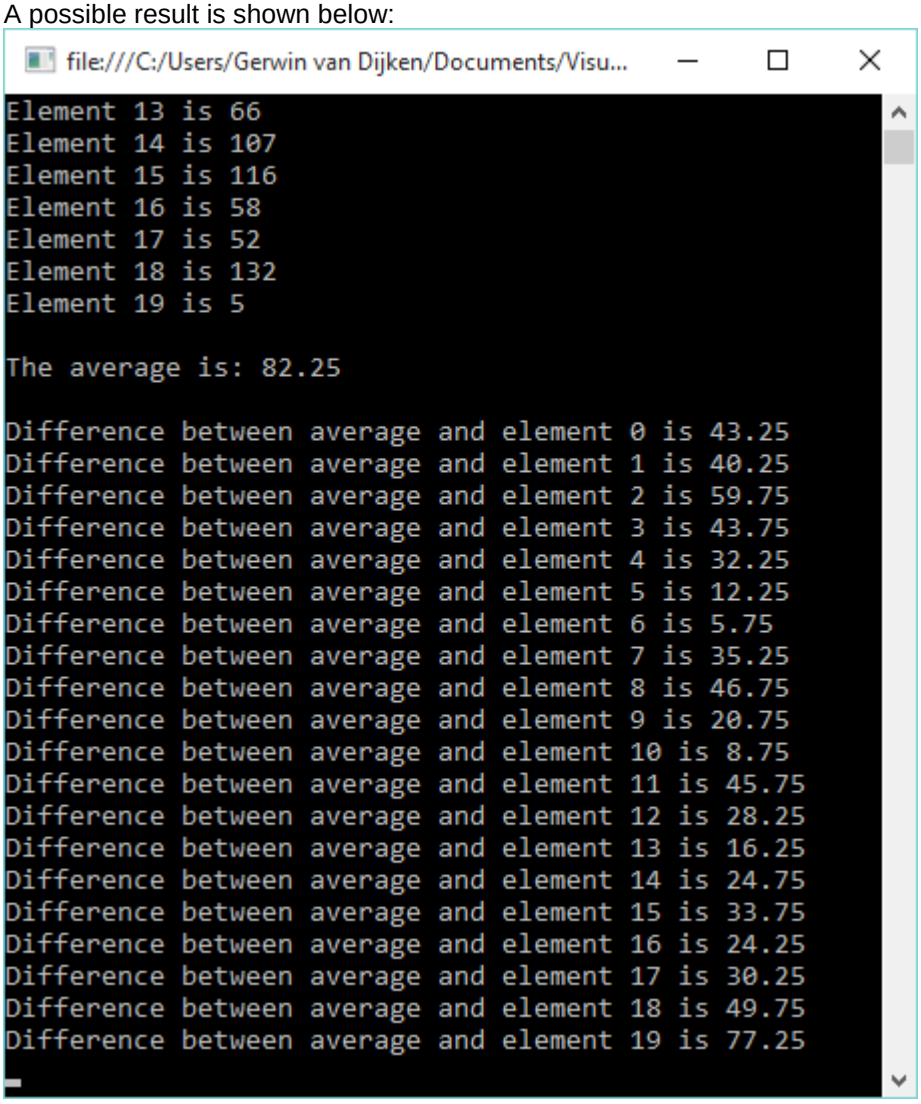
Array assignments

When creating the program code, you must apply the following basic principles:
- create a separate project for each assignment;
- create one solution for each week containing the projects for that week.

Assignment 1 (Console Application)

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:

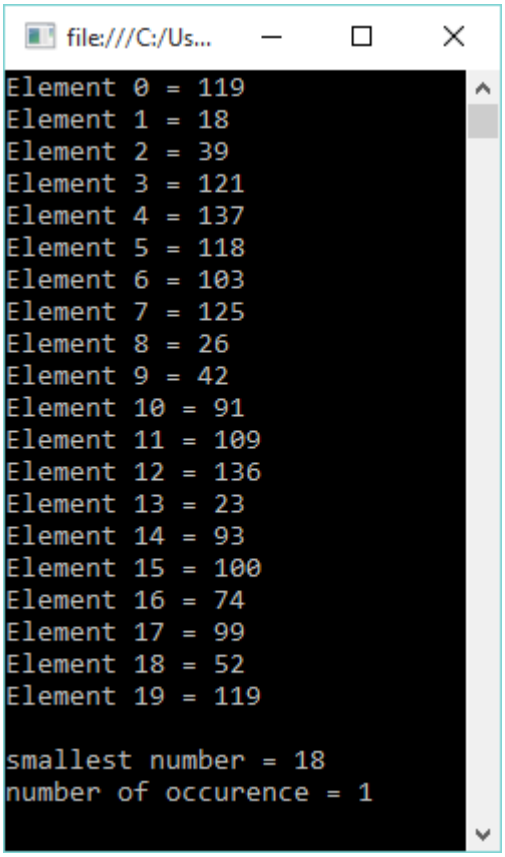


NB:
- try to obtain a similar output
- the numbers from the possible result obviously differ because they are generated randomly
- use `Math.Abs(...)` to obtain an absolute number (without a minus sign)

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 Application)

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.

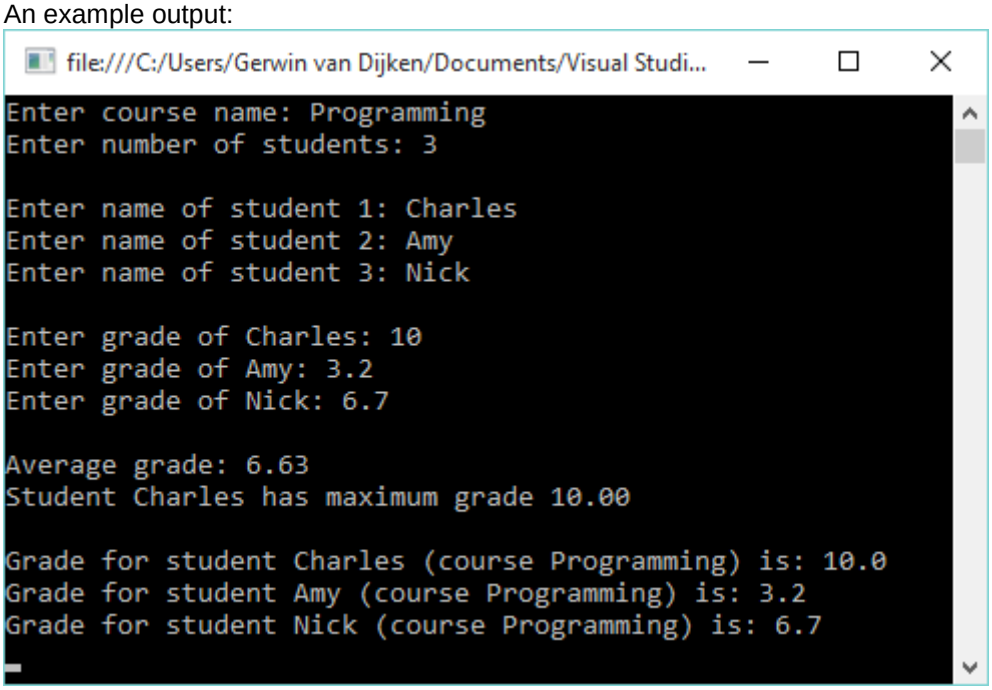


Assignment 3 (Console Application)

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:

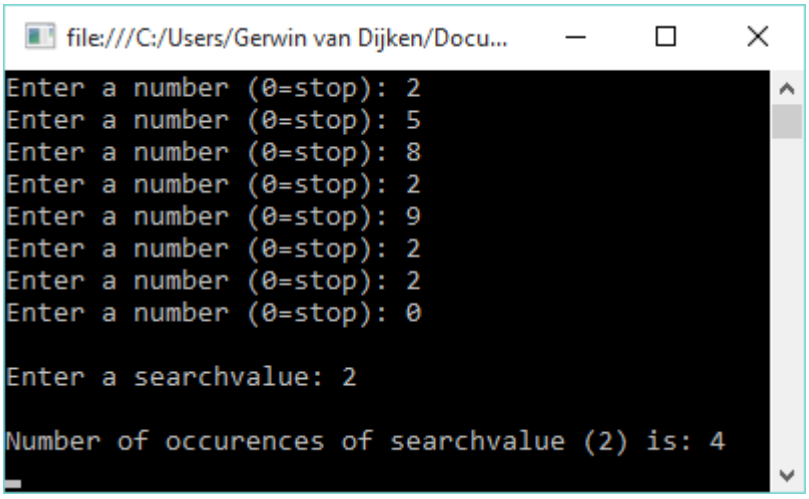


Tips:

- 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 Application)

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.



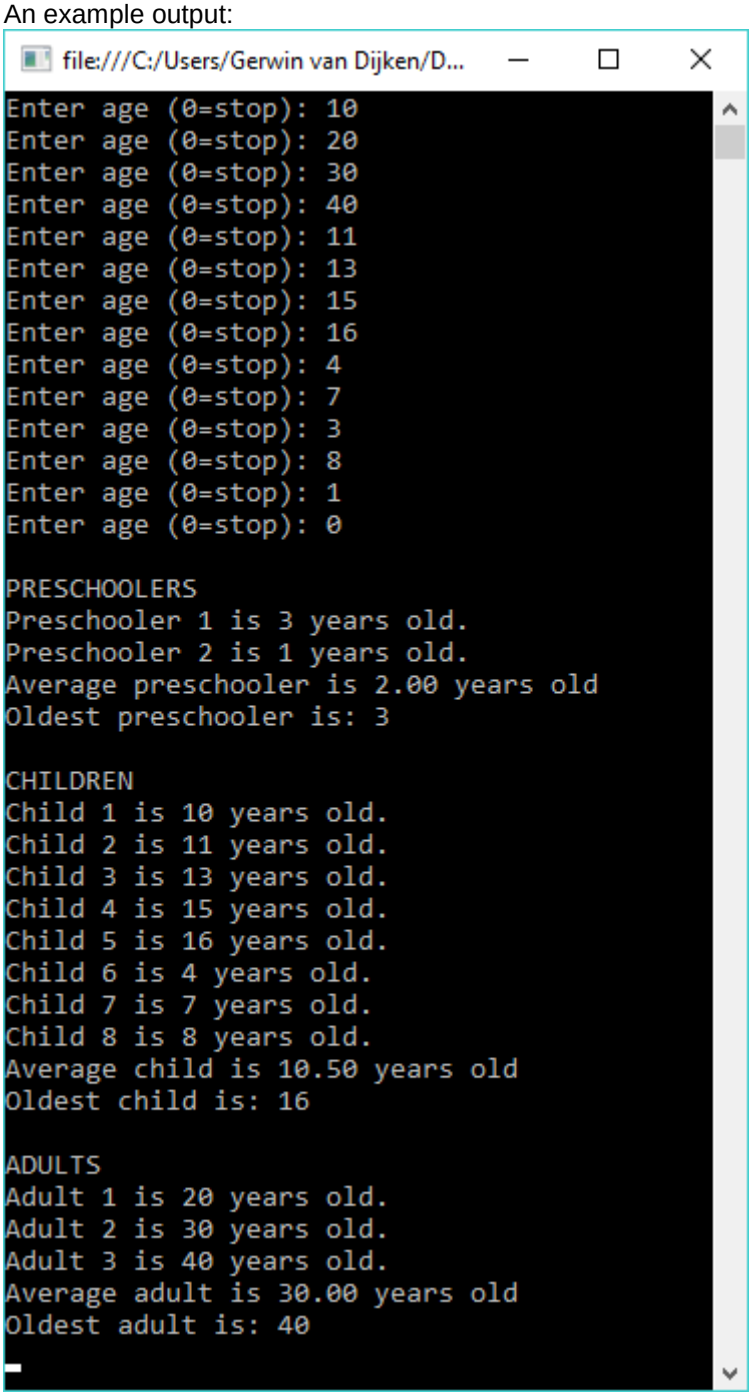
Assignment 5 (Console Application)
We have identified three different age categories in a population:

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:

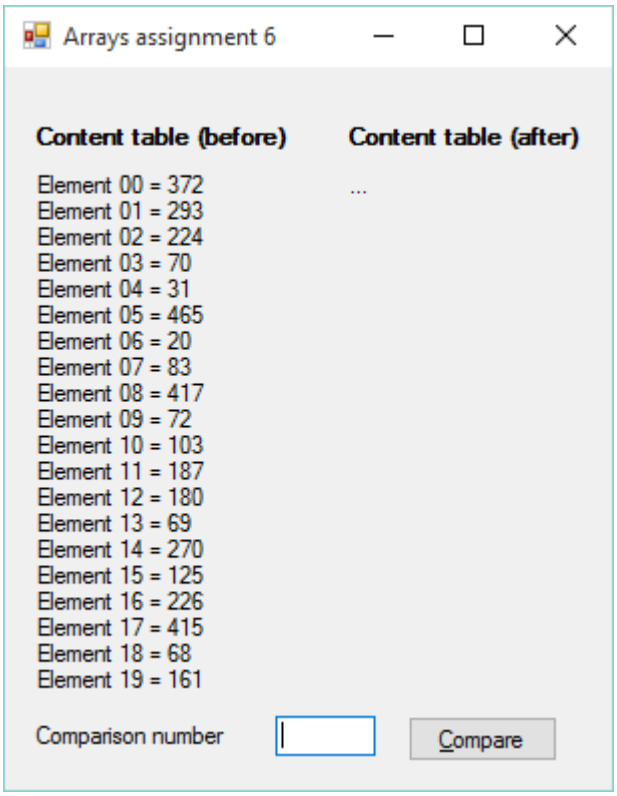


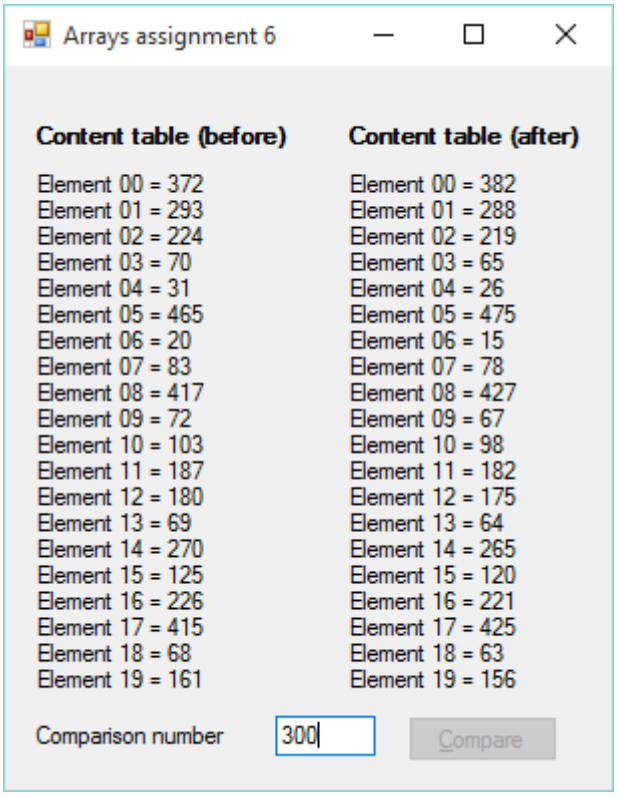
Assignment 6 (Windows Application)

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





Assignment 7 (Windows Application)

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(…)'