

## Lesson 5. Complex Data Types

### Exercises I – Arrays

**Exercise 1.** Write a C program that creates a 10-element integer array and reads its values from the keyboard. Print the values of the array on the screen.

**Exercise 2.** Write a C program that reads integer values until value 0 is introduced or 20 values are read. The values must be stored in an array in reverse order (i.e., the first value to be read must be the last value of the array). Print the values of the resulting array.

**Exercise 3.** Write a C program that reads from the keyboard the elements of an integer 2-dimension array of 4 rows and 3 columns. Print on the screen the largest value resulting from the sum of the values of each column.

**Exercise 4.** Write a C program that reads the values of two arrays of integers (size 5 and 7, respectively –these size values must be declared as constants). Create a third array that stores the result of joining the two previous arrays. Print on the screen the values of the three arrays.

**Exercise 5.** Write a C program that declares a 2-dimension array of NxN size –N value must be declared as a constant. Assign the value 1 to all the elements of the array. Print the array. Assign the value 1 to the main diagonal of the array. Print the array.

**Exercise 6.** Write a C program that declares a 2-dimension array of NxM size –N, M values must be declared as constants. Read the values of the array from the keyboard and print them. Calculate the transpose of the matrix, store the result in a second 2-dimension array, and print the values.

**Exercise 7.** Write a C program to obtain the binary representation of a positive integer value read from the keyboard. The binary representation must be stored in a proper array. (NOTE: Fix an upper threshold to limit the maximum value that can be transformed and stored in the array.)

**Exercise 8.** Write a C program to calculate the weighted mean qualification of a student in a semester.

The program must read from the keyboard the marks of the student –5 courses, 6 exams per lecture (at most). The program must calculate the average qualification by applying the weights: 1.25 if the course is passed in the first exam, 1 if the course is passed in the second exam, 0.9 for the third exam, 0.75 for the fourth exam, 0.6 for the fifth exam, 0.5 for the sixth exam.

(NOTE: Store the marks in a 2-dimension array of size 5x6; store the weights in a 1-dimension array of size 6.)

