

Lesson 4. Control Flow

Exercises II – Loops

Exercise 1.

Write a program that prints the integer numbers from 1 to 100.

Write a program that prints the integer numbers from 1 to 100 counting two by two.

Write a program that prints the integer numbers from 100 to 1.

Write a program that prints the multiples of 3 from 1 to 100.

Exercise 2.

Write a program that prints the series: 5, 10, 15, 20, 25, 30... The program must read from the keyboard the number of terms of the series to print.

Exercise 3.

Write a program that reads a number a from the keyboard and prints on the screen the message `Input number <a>`. This process is repeated until the user introduces the value 0.

Extend the program to print the total number of a values introduced by the user (excluding value 0).

Extend the program to print the sum of the a values introduced by the user.

Exercise 4.

Write a program that reads a user name and a password value (as strings) and prints on the screen the message `Log in successful` if *name* is “root” and *password* is “1234”. If the user spends three unsuccessful tries, the program prints on the screen `No more tries available`.

NOTE: Character strings must have length 5 at least to store the characters “root” and “1234” and the termination character `\0`. Remember to use the `%s` modifier and no `&` in the `scanf` function. To compare two strings, use the `strcmp(string1, string2)` function of `<string.h>`, which returns 0 if *string1* is equal to *string2*.

Exercise 5.

Write a program that prints on the screen the ASCII table. The table must show the characters corresponding to the values from 33 to 126 (`%i`), their corresponding ASCII value (`%c`), and the hexadecimal encoding (`%x`).

Exercise 6.

Write a program that calculates the n -th power of a float value x (being n a positive integer value): a) with the `pow` function in `<math.h>`; b) without the `pow` function (use loop instructions).



Exercise 7.

Write a program that reads 10 positive numbers from the keyboard and prints the following values:

1. Addition of the numbers
2. Arithmetic mean of the numbers
3. Largest number
4. Smaller number

Exercise 8.

Write a program that reads integer numbers from the keyboard until the user introduces 10 numbers or the sum of the number is larger than 100. The program must print which condition was accomplished (or if both are true).

Exercise 9.

Write a program that plays the *guess the number* game. The program asks the user to introduce a number between 1 and 100. Next, this number is compared with the secret number, which is defined as a constant. If the user number is different from the secret number, the program must print a clue indicating if the user value is lesser or greater than the secret number. If the user number is equal to the secret number, the program prints the number of tries and the following message: if the user used less than 5 tries, it prints *Well done*; if the user used between 5 and 10 tries, it prints *Not bad*; if the user used more than 10 tries, it prints *Bad player*. After printing the corresponding message, the program finishes.

Exercise 10.

Extend the previous program to limit the number of tries to 5. Before finishing, the program only prints if the user won the game or not.

Exercise 11.

Write a program that reads integer numbers from the keyboard. If the number has 3 or more digits and it is even, the program prints *Large even number* and reads another value. If the read number is positive but it has less than 3 digits, the program directly asks for another value. If the read number is 0 or negative, the program finishes.

Exercise 12.

Write a program that reads the amount and price of various purchased items and calculates the final price of the shopping cart.

The program asks the user:

Enter number of products:

The user must enter a positive integer value. If the value is negative, the question is repeated until the user introduces a positive integer value.

Next, the program asks the user:



Enter value of product unit:

This number must be also a positive integer. If the value is negative, the question is repeated until the user introduces a positive integer value.

Introducing values for the shopping cart is finished when a 0 value is introduced as the answer of the first question. Next, the program prints out the number of lines and the total price of the cart. (The total price is the result of the sum of the prices of each line of the shopping cart; the price of a line is the result of the multiplication of the product price and the product amount.)

