

## PROBLEMS - Lesson 3c: Python Iterative control structures

**Problems in yellow** → Assessable problems (evaluable)

The rest of the problems are of intermediate difficulty. If you have any difficulty moving from one recommended problem to the next one, do some of the problems that appear in between for more progressive learning.

**Problems in blue** → Optional problems (not assessable but available at VPL for autoevaluation).

### Problem 1

Make a program that calculates the summation of 10 (the sum of the first 10 natural numbers) using a repetitive structure while. The program will display the value of the result without any other message.

### Problem 2

Make a program to calculate the summation of an integer N given by the user. The program will check that the input number is positive and greater than zero. If the number is lower than or equal to zero, the program should display the message: "Error: the value must be greater than zero" and must ask for an input number again. This process must be repeated until the user enters a correct value. The program then calculates the sum of the first N integers and displays the result with a message with the format:

"The sum of N is: XXX" where N is the number entered by the user and XXX is the result of the summation.

Note: Do the problem with a while structure.

### Problem 3

Make a program that calculates the sum of 10 (the sum of the first 10 natural numbers) using a repetitive structure for. The program will display the value of the result without any other message.

### Problem 4

Make a program to calculate the summation of an integer N given by the user. The program will check that the input number is  $\leq 0$ , otherwise, it will display the message: "Error: the value must be greater than zero" and must ask for an input number again. This process must be repeated until the user enters a correct value. The program then calculates the sum of the first N integers and displays the result with a message with the format: "The sum of N is: XXX" where N is the number entered by the user and XXX is the result of the summation.

Note: Do the problem with a for structure.

### **Problems with the WHILE repetitive structure**

#### **Problem 5**

Make a program that asks an integer and displays it on the screen with the message "Entered value: NN", where NN is the integer entered by the user. The program must repeat these two actions meanwhile the user enters positive values.

#### **Problem 6**

Make a program that asks the user to enter an integer between 0 and 100 and then displays it with the message "Entered value: NN", where NN is the integer entered by the user. The program must repeat these two actions meanwhile the user enters values within the range [0,100] (both included).

#### **Problem 7**

Make a program that adds a sequence of integers that are entered by keyboard. The sequence ends when a 0 is entered.

#### **Problem 8**

Make a program that calculates the average of all the numbers of a sequence of integers entered by keyboard. The sequence ends when a 0 is entered. Display the result with a message in the format: "Average: XX" where XX is the average of the sequence values.

Note that:

- 0 is not part of the sequence. It should not be considered when computing the mean.
- it is necessary to check if the sequence is empty (the first entered number is 0) and display a message "Error: The sequence is empty. The average cannot be calculated."

#### **Problem 9**

Write a program that, given a sequence of integers which end with the number 0, sum all the positive numbers in the sequence.

#### **Problem 10**

Make a program that, given a sequence of integers which end with the number 0, checks if all numbers in the sequence are odd. If all are odd, the program will write the message, "There are no even numbers in the sequence." Otherwise, the program will output "Some number in the sequence is even."

Note: If only a 0 is entered, output "There are no even numbers in the sequence."

#### **Problem 11**

Make a program that calculates the maximum value of a sequence of positive integers ending in 0. The program must ensure that all numbers in the sequence are positive. If a negative number is entered, an error message should be displayed, then, it will ignore this number, and continue reading the script.

#### **Problem 12**

Make a program that reads integers until a negative number is entered or until 10 values have been read. The program will calculate the minimum and maximum of all values (do not consider the negative value that indicates the end of the sequence of numbers) and display a message in the format "Maximum: XX - Minimum: YY" where XX and YY are the maximum and minimum values of the sequence respectively. If the first number entered is negative the program will output "There are no numbers in the sequence."

### Problem 13

Make a program that asks the user to enter a leap year. The program should check if it is indeed a leap year, otherwise it will display the message "This year is not a leap year" and prompt the user to enter a year again. This process must be repeated until the user enters a year that is actually a leap year.

### Problem 14

Modify the program that increases by one second the time entered by the user through in HH:MM:SS format and make the time update second by second as in a real clock. To do this, the program must wait 1 second before changing the time and repeat the process. The program should end after updating the time 5 times (i.e. after 5 seconds).

To make the program wait a second, you can use the function `sleep(SS)` where SS is the seconds we want the program to wait (in our case `sleep(1)`). To use this feature, the first line of your program must be:

*from time import sleep*

Note: Show each hour on a different line.

### Problem 15

Implement a program that allows you to invoice purchases in a store. The program must allow more than one product to be included in the invoice. Therefore, the program should ask the user the product price and the number of units sold for every product until the user enters a negative price.

A 7% VAT must be applied to the total amount of the purchase. If the final amount, once applied VAT is above 500 euros, a 5% discount will be made. If it is above 1000 euros, a 10% discount will be made.

The program must show the final cost of the purchase with a message: "The final price is the purchase are XX euros." where XX will be a real number.

### Problem 16

Make a program that reads the grades of all students in a class and counts how many students have failed (D grade in [0,5)), passed (C grade in [5-7)), how many have a B grade (grade in [7-9)), an A grade (grade in [9-10)) and how many have A with honors (A+, grade equal to 10). The program will stop reading notes when an incorrect note value is entered. The output must be a message in the format:

"D: X1 - C: X2 - B: X3 - A: X4 - A+: X5"

where X1, X2, ... are the number of failed, passed, etc.

### Problem 17

Make a program that implements a basic calculator of operations with real numbers. The program should ask two real numbers and then show a menu with five options (addition, subtraction, multiplication, division and exit):

MENU

- 1.- Summation
- 2.- Subtraction
- 3.- Multiplication
- 4.- Division
- 5.- Exit

Select one of the options:

The user will select the operation via keyboard with the corresponding number and must perform the corresponding operation. Then you must display the result of the operation with a format as in the following example: "6 + 8 = 14". Use the symbols: +, -, \*, : for operations.

After each operation, the menu must appear again until the user selects the "Exit" option. When the user selects "Exit", the program must ask:

"Do you want to operate with other operands? (S / -)".

If the user writes 'S', the process will start again. Otherwise (the user writes a character different than 'S'), then the program ends.

In case of division, it is necessary to control that a division by zero does not take place. If the second operand is a zero, it will output the message: "Error: Division by zero".

If the user enters an option that is not in the menu, the program should display an error message: "Error: Option not valid" and show the menu again.

### Problem 18

Make a program to guess a number between 1 and 10, where the player has 5 attempts to guess it. Initially, the program must generate a random number between 1 and 10, and then the game begins.

At each attempt, the player will enter a number between 1 and 10 and the program must tell whether the guessed number is lower or greater than the number entered by the user.

In the end, if the player has not guessed it, the program must write the guessed number. If the user has guessed it, then it must say how many attempts the user has needed.

In either case, when the game is over the user must be asked if he/she wants to play again and if the answer is affirmative, the game starts again.

To generate random numbers, use the *randint(a,b)* function that returns a value between a and b (both inclusive). To use this feature, the first line of your program must be:

```
from random import randint
```

### Problem 19

Make a program to check if a number  $n$  is prime. If the number is prime, the message "Prime" will be displayed, otherwise the message "NOT Prime" will be displayed.

### Problem 20

Given the series defined by:  $a_0 = 0, a_1 = 1, a_n = 3 \cdot a_{n-1} + 2 \cdot a_{n-2}$

- Make a program that says which is the value and position within the series of the first element greater than or equal to 1000.
- Modify the previous program so that it tells the value and position of the first element greater than or equal to a number entered by keyboard.

## **Problems with the repetitive structure FOR**

### **Problem 21**

Motivation and a positive mindset are very important in learning programming. Make a program that asks you to enter a positive integer N and write the message N times on the screen: "Programming in Python is fun and I will pass this subject."

### **Problem 22**

Make a program that:

1. Prints all numbers from 1 to 10 (both included) separated by a blank space.
2. In the bottom line, the program should print all the numbers from 25 to 35 (both inclusive), also separated by a blank space between them.
3. In the next line, the program must print all even numbers between 50 and 80 (both inclusive) separated by a blank space.
4. Finally, in the next line, the program should print all numbers between 50 and 20 (both inclusive) counting down 3 by 3 (e.g. 50 47 ...).

### **Problem 23**

Make a program that asks the user to enter a positive integer and prints on the screen all multiples of 3 lower than the number entered by the user. If the number entered by the user is not positive, display the message "Error: the number must be positive" and ask the user to enter the number again.

### **Problem 24**

Write a program to compute and print all the powers of two smaller or equal than  $2^N$ , where N is a number (greater than 0) entered by keyboard. Each output value must be separated from the next value by a blank space.

### **Problem 25**

Make a program that reads 5 numbers via keyboard and counts how many positive numbers and how many negative numbers we been entered. Zero will not count as positive nor negative. The result will be displayed with a message with the format: "Positives: XX - Negatives: YY" where XX and YY are the number of positives and negatives that have been entered.

### **Problem 26**

Make a program that reads 10 integers via keyboard and outputs the sum of the even numbers, how many even numbers there are, and the average of odd numbers.

### **Problem 27**

Make a program that computes the sum, the product and the average of the even numbers ranging between two limits that are read via keyboard. The result will be displayed with the formatted message; "Sum: XX - Product: YY - Average: ZZ" where XX, YY and ZZ are the values of the sum, the product, and the average, respectively.

### **Problem 28**

Make a program that asks the user to enter a number by keyboard and calculate the factorial of the input number.