

## PYTHON CB13 – EXERCISES IV

1. Οι **πρώτοι αριθμοί** (prime numbers) είναι αυτοί που διαιρούνται ακριβώς μόνο με τον εαυτό τους και τη μονάδα. Το 1 δεν θεωρείται πρώτος. Οι πρώτοι αριθμοί μέχρι το 100 είναι οι ακόλουθοι : 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97.

Να γράψετε ένα Python module με το όνομα `primes.py`, το οποίο να ελέγχει αν κάποιος αριθμός  $x$  είναι πρώτος αριθμός, στο διάστημα των αριθμών από 2 έως και 100 και να τυπώνει μόνο τους πρώτους αριθμούς, όπως η παραπάνω ακολουθία. Να γίνει η άσκηση με χρήση συναρτήσεων.

2. Να υλοποιήσετε τον κώδικα του παιχνιδιού **ΚΡΕΜΑΛΑ (Hangman)**. Ο κώδικας του παιχνιδιού μπορεί να αποτελείται από τις παρακάτω συναρτήσεις:

A) Συνάρτηση print hangman: Ανάλογα με το πλήθος των αποτυχημένων προσπαθειών εμφανίζει το αντίστοιχο τμήμα του hangman, όπως αυτό παρακάτω:

$$\begin{array}{c} \bar{(\quad)} \\ \diagdown \quad | \quad \diagup \\ | \\ \diagup \quad \diagdown \end{array}$$

B) Συνάρτηση all\_letters: Λαμβάνει ως παραμέτρους την μουσική λέξη και τα γράμματα που βρέθηκαν και επιστρέφει True όταν όλα τα γράμματα της μουσικής λέξης έχουν βρεθεί.

Π.χ. `all_letters("abracadabra", "abrcd")` επιστρέφει `True`

Γ) Συνάρτηση print\_word, η οποία εμφανίζει το τμήμα της λέξης που έχει βρει ο παίκτης μέχρι εκείνη τη στιγμή. Στην θέση των γραμμάτων που δεν έχουν βρεθεί ακόμα θα εμφανίζεται ο χαρακτήρας \_.

Δ) Συνάρτηση `get_secret_word`, η οποία δημιουργεί μια λίστα από λέξεις, επιλέγει τυχαία μια λέξη με χρήση της `random` και την επιστρέφει.

Ε) Ενδεικτικός κώδικας του κυρίως προγράμματος είναι ο παρακάτω:

```
import random
secret_word = get_secret_word()
letters_found = ""
tries = 6
found = False
while tries > 0 and not found :
    letter = input( "Μάντεψε ένα γράμμα : " )
    print_word(secret_word, letters_found)
    if letter not in secret_word:
        tries = tries - 1
        print_hangman( tries )
    else :
        letters_found += letter
        found = all_letters(secret_word, letters_found)
    print_word(secret_word, letters_found)
    print()
if found :
    print( "Συγχαρητήρια κέρδισες" )
```

else:

```
print (" Δεν πειράζει θα ξαναπροσπαθήσεις")
```

**ΦΥΣΙΚΑ ΜΠΟΡΕΙΤΕ ΝΑ ΔΗΜΙΟΥΡΓΗΣΕΤΕ ΤΗ ΔΙΚΗ ΣΑΣ ΛΥΣΗ, ΧΩΡΙΣ ΝΑ ΛΑΒΕΤΕ ΥΠΟΨΗΝ ΤΗ ΒΟΗΘΕΙΑ ΤΩΝ ΠΑΡΑΠΑΝΩ ΥΠΟΔΕΙΞΕΩΝ.**

3. Να δημιουργήσετε ένα νέο Python Module με το όνομα **Equation.py**, που να δέχεται ως είσοδο ένα τριώνυμο:  $f(x) = ax^2 + bx + \gamma$ , το οποίο αντιστοιχεί στη λύση της δευτεροβάθμιας εξίσωσης:  $ax^2 + bx + \gamma = 0$ . Στην αρχή του προγράμματος να εισάγονται τιμές για το  $a$ ,  $b$ ,  $\gamma$ . Το πρόγραμμα θα πρέπει να αποφασίζει αν το τριώνυμο έχει, πόσες και ποιες πραγματικές ρίζες και να τυπώνει αντίστοιχο αποτέλεσμα μαζί με σχετικά μηνύματα. Πρέπει να προσέξετε ώστε να μην γίνεται διαίρεση με το 0 και οι πράξεις να τηρούν τους κανόνες προτεραιότητας πράξεων της Python. Μπορείτε να λάβετε βοήθεια από τον παρακάτω σχετικό πίνακα:

$\Delta = b^2 - 4a\gamma$	Η εξίσωση $ax^2 + bx + \gamma = 0, a \neq 0$
$\Delta > 0$	Έχει δύο ρίζες άνισες τις $x_{1,2} = \frac{-b \pm \sqrt{\Delta}}{2a}$
$\Delta = 0$	Έχει μια διπλή ρίζα τη $-\frac{b}{2a}$
$\Delta < 0$	Είναι αδύνατη στο $\mathbb{R}$ .

4. Write a Python program which accepts a sequence of comma-separated elements from the user. It generates a list of all the elements and a list of only the number elements. It also creates a tuple of only the number elements and finally it outputs the list of numbers, the tuple of numbers, the min and the max number of the tuple. Use split method for the user input.

For example the program asks the user:

Input some comma separated integers: 23,f,1,i,90,d,g,65

And at the output prints:

Only numbers : ['23', '1', '90', '65']

Tuple : ('23', '1', '90', '65')

Max number is 90

Min number is 1

5. Write a Python program to compute the future value of a specified principal amount, rate of interest, and a number of years. For example:

amt = 10000, int = 3.5, years = 7

Expected Output : 12722.79

6. Write a Python module to find the day of the week of a given date.
7. Write a Python module to ask the user for a string and print out whether this string is a palindrome or not. (A **palindrome** is a string that reads the same forwards and backwards. F.e. madam, radar etc.)

8. Να υλοποιηθεί το **flowchart** για το παιχνίδι Tic-Tac-Toe (είναι η τρίλιζα), με έναν παίκτη και τον υπολογιστή.
9. Perform a **Use Case flowchart** of an e-Library, based on the following description:

An **Online Public Access Catalog (OPAC)** is an **e-Library** website which is part a **Library Management System (LMS)**.

Patrons of the library can search library catalog online to locate various resources - books, periodicals, audio and visual materials, or other items under control of the library. Patrons may reserve or renew item, provide feedback, and manage their account.

10. The following board is 3x3 (like in tic tac toe). Obviously, there are many other sizes (8x8 for chess, 19x19 for Go, and many more). Make a Python module to ask the user what size game board they want to draw, and draw it for them to the screen using print statements.

```
--- --- ---
|   |   |   |
--- --- ---
|   |   |   |
--- --- ---
|   |   |   |
--- --- ---
```