

# Μάντεψε τον Αριθμό

## Φύλλο Εργασίας

Ένα από τα πρώτα προγράμματα που συνηθίζεται να φτιάχνουν οι μαθητευόμενοι προγραμματιστές είναι ένα παιχνίδι στο οποίο ο παίκτης προσπαθεί να μαντέψει τον μυστικό αριθμό που έχει “σκεφτεί” ο υπολογιστής (ή το αντίστροφο).

Υπάρχουν πολλοί καλοί λόγοι που αυτό το παιχνίδι είναι μια τόσο δημοφιλής επιλογή για τους αρχάριους: το πρόγραμμα δεν είναι ιδιαίτερα περίπλοκο, αν και συνδυάζει όλες τις βασικές αλγοριθμικές έννοιες, ενώ το παιχνίδι καθαυτό είναι πολύ διασκεδαστικό.

Με αυτό το φύλλο εργασίας θα γνωρίσουμε τα βασικά είδη εντολών που επιτρέπουν στα προγράμματά μας να αλληλεπιδρούν με το χρήστη, δηλαδή να του εμφανίζουν μηνύματα στην οθόνη και να του ζητούν να εισάγει τιμές από το πληκτρολόγιο.

Θα εξετάσουμε επίσης πως μπορούμε να κάνουμε τα προγράμματά μας να επιλέγουν τον τρόπο που θα συμπεριφερθούν, εξετάζοντας τις συνθήκες που επικρατούν κατά την εκτέλεσή τους. Θα δούμε ακόμα πως μπορούμε να κάνουμε τα προγράμματά μας να εκτελούν επαναληπτικά, δηλαδή ξανά και ξανά, τις ίδιες εντολές.

## Οδηγίες

Αρχικά, το πρόγραμμά μας θα δίνει κάποιες οδηγίες στον παίκτη σχετικά με το παιχνίδι.

1. Ξεκινήστε το πρόγραμμα με την εντολή:

```
print("Θα σκεφτώ έναν αριθμό από το 1 μέχρι το 32.")
```

Εκτελέστε το πρόγραμμα και δείτε τι συμβαίνει.

2. Βασιστείτε στην εντολή του προηγούμενου βήματος και συμπληρώστε το πρόγραμμα με μια ακόμα εντολή, έτσι ώστε να εμφανίζεται στην οθόνη το μήνυμα:

Εσύ θα προσπαθήσεις να τον μαντέψεις.

Εκτελέστε ξανά το πρόγραμμα. Τα καταφέρατε;

.....



22 Ιουλίου 2016  
13:19

Στη διεύθυνση [pythonies.mysch.gr](http://pythonies.mysch.gr) μπορείτε να βρείτε κι άλλα φύλλα εργασίας σαν και αυτό, καθώς και το συνοδευτικό τους υλικό.



3. Το πρόγραμμά θα ορίζει τώρα τον μυστικό αριθμό. Έστω ότι επιλέγει το τυχερό 13. Προσθέστε την παρακάτω εντολή στο πρόγραμμα:

```
secret = 13
```

Η `secret` είναι μια *μεταβλητή* στην οποία αποθηκεύεται η τιμή του μυστικού αριθμού που επιλέγει το πρόγραμμά μας. Τώρα μπορούμε να αναφερόμαστε στον μυστικό αριθμό με το όνομα `secret`, χωρίς να έχει σημασία ποια είναι η τιμή του.

Εκτελέστε το πρόγραμμα. Παρατηρείτε κάποια διαφορά στην εκτέλεση μετά την προσθήκη της παραπάνω εντολής;

.....



### Γνωριμίες

Το πρόγραμμά μας θα ζητά από τον παίκτη το όνομά του και θα τον καλωσορίζει στο παιχνίδι.

4. Προσθέστε τις παρακάτω εντολές οπουδήποτε στο πρόγραμμα.

```
print("Πώς σε λένε;")
player = input()
```

Η `input()` επιστρέφει το κείμενο που πληκτρολόγησε ο χρήστης, επιστρέφει δηλαδή μια *αλφαριθμητική τιμή*. Εδώ χρησιμοποιούμε την `input()` για να διαβάσουμε την απάντηση του χρήστη, η οποία αποθηκεύεται στη μεταβλητή `player`.

Εκτελέστε το πρόγραμμα. Η μεταβλητή `player` παίρνει την τιμή που πληκτρολογείτε (αναλαμβάνοντας το ρόλο του χρήστη πλέον).

5. Συμπληρώστε το πρόγραμμα με την παρακάτω εντολή.

```
print("Γεια σου", player)
```

Εκτελέστε το πρόγραμμα σας 2-3 φορές, παίζοντας το ρόλο του χρήστη, πληκτρολογήστε κάθε φορά διαφορετικά ονόματα και παρατηρήστε τι εμφανίζει.

Ποια πιστεύετε ότι θα ήταν η διαφορά αν είχατε πληκτρολογήσει `print("Γεια σου player");`

.....

.....

Ποια θα ήταν η διαφορά αν αντί της εντολής `player = input()` χρησιμοποιούσαμε την εντολή `player = "Μυρσίνη";`

.....

.....



Ποια θα ήταν η διαφορά αν στο βήμα 3 γράφαμε `secret = input()`, αντί για `secret = 13`;

.....  
 .....



## Φρέναρε λίγο

Το πρόγραμμά μας εμφανίζει όλα τα μηνύματα μαζεμένα. Θα δώσουμε μια ανάσα στον παίκτη, ώστε να προλαβαίνει να τα διαβάξει έχοντας λίγο χρόνο στη διάθεσή του, πριν του εμφανίσουμε το επόμενο μήνυμα.

Στο σημείο αυτό θα χρειαστούμε μια λειτουργία που δεν προσφέρεται από τις βασικές εντολές της Python. Για το σκοπό αυτό θα χρειαστεί να εισάγουμε στο πρόγραμμά μας μια βιβλιοθήκη και συγκεκριμένα τη βιβλιοθήκη `time`, η οποία παρέχει τη λειτουργικότητα που μας χρειάζεται. Οι βιβλιοθήκες είναι συλλογές από μικρά προγράμματα που μπορούμε να χρησιμοποιήσουμε στα προγράμματά μας.

6. Προσθέστε στην αρχή του προγράμματος την εντολή που ακολουθεί, για να εισάγετε τη βιβλιοθήκη `time`:

```
import time
```

7. Εισάγετε, ανάμεσα στις δύο πρώτες `print`, την παρακάτω εντολή:

```
time.sleep(1)
```

Με την εντολή αυτή χρησιμοποιείται η *συνάρτηση* `sleep` από τη βιβλιοθήκη `time`. Τι αποτέλεσμα έχει η προσθήκη αυτής της εντολής;

.....  
 Τί ρόλο παίζει η *παράμετρος* 1; Διερευνήστε τι θα συμβεί αν χρησιμοποιήσουμε άλλον αριθμό στη θέση του 1.

.....



## Ας παίξουμε

Έρθε η ώρα να ξεκινήσει το πραγματικό παιχνίδι. Το πρόγραμμά θα ζητά από τον παίκτη έναν αριθμό και στη συνέχεια θα εμφανίζει μήνυμα αν βρήκε τον μυστικό αριθμό ή όχι.

8. Προσθέστε τις παρακάτω εντολές στο πρόγραμμα:

```
print("Δώσε αριθμό:")  
number = int(input())
```

Τι διαφορά παρατηρείτε σε σχέση με τις εντολές που χρησιμοποιήσατε για να ζητήσετε το όνομα του παίκτη στο βήμα 4;

.....  
 .....

Ποια διαφορά πιστεύετε ότι έχει η τιμή του ονόματος του παίκτη από την τιμή του `number`;

.....  
 .....

9. Συμπληρώστε το πρόγραμμα, ώστε να εμφανίζει στον παίκτη τον αριθμό που πληκτρολόγησε, για παράδειγμα:

Έδωσες τον αριθμό 20

Αν δυσκολευτείτε, ανατρέξτε στο βήμα 5.

10. Προσθέστε στο πρόγραμμά σας τις εντολές:

```
print("Σωστά!")
print("Λάθος...")
```

Τι πιστεύετε ότι θα εμφανίσει το πρόγραμμα αν το εκτελέσουμε τώρα;

.....

Εκτελέστε το πρόγραμμα. Επιβεβαιώνεται η απάντησή σας;

.....

Σε ποια περίπτωση είναι επιθυμητό να εμφανίζεται το "Σωστά!"; Αντίστοιχα, σε ποια περίπτωση είναι επιθυμητό να εμφανίζεται το "Λάθος...";

.....

.....

11. Θα τροποποιήσουμε το πρόγραμμά μας, έτσι ώστε να ελέγχει αν ο παίκτης μάντεψε τον μυστικό αριθμό και στη συνέχεια να εμφανίζει μόνο το κατάλληλο μήνυμα.

Προσθέστε τις παρακάτω εντολές:

```
if number == secret:
    print("Σωστά!")
else:
    print("Λάθος...")
```

Εκτελέστε το πρόγραμμα. Παρατηρήστε το *σφάλμα* που εμφανίζει.

Στις περισσότερες γλώσσες προγραμματισμού δεν θα αντιμετωπίζατε κάποιο πρόβλημα, όμως η Python έχει μια "ευαισθησία": Οι



Με το `==` ελέγχεται αν δύο τιμές είναι ίσες. Μην το συγχέετε με το `=` που χρησιμοποιείται για να δώσουμε τιμή σε μια μεταβλητή.



εντολές που εκτελούνται στη μία ή στην άλλη περίπτωση πρέπει να ξεχωρίζουν, πρέπει με κάποιον τρόπο να επισημανθεί ότι οι εντολές αυτές ανήκουν αντίστοιχα στην **if** και την **else**.

Στην Python, αυτό επιτυγχάνεται με τις εσοχές.

12. Προσθέστε 4 κενά πριν από τις δύο **print** και εκτελέστε το πρόγραμμα.

```
if number == secret:
    print("Σωστά!")
else:
    print("Λάθος...")
```

Εκτελέστε το πρόγραμμα σας δύο φορές. Στην πρώτη δώστε σωστά τον μυστικό αριθμό, ενώ στην επόμενη δώστε έναν διαφορετικό.

Εμφανίζεται το κατάλληλο μήνυμα σε κάθε περίπτωση;

.....

Σε ποια περίπτωση εκτελούνται οι εντολές της else;

.....

.....

13. Προσθέστε στο πρόγραμμα μια **print**, η οποία θα εμφανίζει τον μυστικό αριθμό στην περίπτωση που ο παίκτης απαντήσει λάθος, όπως παρακάτω:

Ο μυστικός αριθμός ήταν ο 13.

Εκτελέστε το πρόγραμμα δύο φορές. Στην πρώτη δώστε σωστά τον μυστικό αριθμό, ενώ στην επόμενη δώστε έναν διαφορετικό.

Εμφανίζει το μυστικό αριθμό μόνο στην περίπτωση που ο παίκτης δεν τον μαντέψει; Αν όχι, γιατί πιστεύετε ότι συμβαίνει αυτό;

.....

.....

Αν ο μυστικός αριθμός εμφανίζεται είτε ο χρήστης τον μαντέψει, είτε όχι, τότε πιθανότατα δεν τοποθετήσατε τη νέα εντολή μέσα στην **else**, αλλά μετά από αυτή. Προσθέστε 4 κενά πριν την εντολή, για να υποδηλώσετε ότι κι αυτή ανήκει στην **else**.

14. Στο μήνυμα που εμφανίσατε προηγουμένως χρησιμοποιήσατε την μεταβλητή **secret**;

.....

Αν απαντήσατε όχι, τροποποιήστε κατάλληλα την εντολή **print**, ώστε να χρησιμοποιεί τη **secret** και όχι το 13.



Τι πλεονέκτημα πιστεύετε ότι έχει η χρήση της μεταβλητής `secret` στο μήνυμα, αντί της απευθείας εμφάνισης του αριθμού 13;

.....



Δοκιμάστε να παίξετε 2–3 φορές ακόμα. Υπάρχει κάτι που σας ενοχλεί στον τρόπο που λειτουργεί το πρόγραμμα;

.....



## Τυχαιότητα

Το παιχνίδι μας δεν αξίζει να το παίξεις πάνω από μία-δύο φορές, αν ο μυστικός αριθμός είναι πάντα το 13. Αυτό που πραγματικά χρειαζόμαστε είναι να επιλέγεται κάθε φορά ένας διαφορετικός αριθμός.

Για τον σκοπό αυτό, θα χρειαστούμε τη βιβλιοθήκη `random`, όπως προηγουμένως είχαμε χρησιμοποιήσει τη βιβλιοθήκη `time`.

15. Προσθέστε στο πρόγραμμα την εντολή εισαγωγής της βιβλιοθήκης `random`.

```
import random
```

Τροποποιήστε την εντολή του βήματος 3, όπου ορίζεται η τιμή της μεταβλητής `secret`, ως εξής:

```
secret = random.randint(1,32)
```

Εκτελέστε το πρόγραμμα αρκετές φορές. Τί είδους τιμές παρατηρείτε ότι παίρνει η μεταβλητή `secret`;

.....



Ποιος πιστεύετε ότι είναι ο ρόλος των παραμέτρων 1 και 32; Τί συμβαίνει αν δοκιμάσετε άλλες τιμές στη θέση τους, για παράδειγμα 33 και 64; Αν χρειαστεί εκτελέστε πάλι το πρόγραμμα αρκετές φορές προκειμένου να απαντήσετε στην ερώτηση.

.....



16. Τί θα αλλάζατε στο πρόγραμμα όπως έχει μέχρι στιγμής; Πώς πιστεύετε ότι πρέπει να επεκταθεί για να γίνει πιο ενδιαφέρον;

.....



.....

## Γύρω-Γύρω Όλοι

17. Προσθέστε τη γραμμή που ακολουθεί αμέσως πριν από τις εντολές που προσθέσατε στο βήμα 8, όπου ζητείται από το χρήστη να μαντέψει τον μυστικό αριθμό.

**while True:**

Προσθέστε τέσσερα κενά μπροστά από όλες τις εντολές που ακολουθούν τη **while**, σηματοδοτώντας έτσι ότι αυτές οι εντολές εμφωλεύονται στη **while**, δηλαδή περιέχονται σε αυτήν.

Εκτελέστε το πρόγραμμα. Ποια αλλαγή παρατηρείτε ότι επιφέρει η χρήση της **while**;

.....  
 .....

Η εντολή του βήματος 15, που δίνει μια τυχαία τιμή στην μεταβλητή **secret**, βρίσκεται πριν από τη **while**. Ποια πιστεύετε ότι θα ήταν η διαφορά αν βρισκόταν μέσα στη **while**;

.....  
 .....

Η εντολή του βήματος 13 που εμφανίζει τον μυστικό αριθμό στο χρήστη τώρα δημιουργεί πρόβλημα. Γιατί;

.....  
 .....

Διαγράψτε την εντολή που εμφανίζει τον μυστικό αριθμό.

Παίξτε το παιχνίδι μέχρι να μαντέψετε τον μυστικό αριθμό. Υπάρχει κάτι που σας ενοχλεί; Κάτι που φαίνεται να μη δουλεύει σωστά;

.....  
 .....

- 18. Η εντολή **break** διακόπτει την επανάληψη μέσα στην οποία βρίσκεται αμέσως μόλις εκτελεστεί. Προσθέστε την **break** στο σημείο που θεωρείτε κατάλληλο, έτσι ώστε το παιχνίδι να τερματίζεται όταν ο παίκτης μαντέψει τον αριθμό.

Παίξτε και πάλι το παιχνίδι μέχρι να μαντέψετε τον μυστικό αριθμό, για να επιβεβαιώσετε ότι τοποθετήσατε την **break** σε σωστό σημείο.

- 19. Για δοκιμαστικούς λόγους, κάτω από την **break** προσθέστε την εντολή:

**print("Ζντογκ!")**

Εκτελέστε πάλι το πρόγραμμα μέχρι να μαντέψετε τον αριθμό. Εμφανίζεται το μήνυμα "Ζντογκ!"; Γιατί πιστεύετε ότι συμβαίνει αυτό;

.....  
 .....



Αφαιρέστε τώρα την εντολή που προσθέσατε.

20. Τι θα αλλάζατε στο πρόγραμμα όπως έχει μέχρι στιγμής; Πώς πιστεύετε ότι πρέπει να επεκταθεί για να γίνει πιο ενδιαφέρον;

.....  
 .....



## Επιλογές, Επιλογές

Θα επεκτείνουμε το πρόγραμμα έτσι ώστε να δίνει στο χρήστη περισσότερη πληροφορία: αντί να τον ενημερώνει απλά αν βρήκε τον μυστικό αριθμό ή όχι, θα τον κατευθύνει αν πρέπει να τον αναζητήσει ψηλότερα ή χαμηλότερα.

Για την επέκταση αυτή δεν αρκεί πια η απλή **if-else**, η οποία μπορεί να διακρίνει μόνο ανάμεσα σε δύο περιπτώσεις.

21. Τροποποιήστε την **if** που ελέγχει αν ο χρήστης βρήκε τον μυστικό αριθμό. Συμπληρώστε την συνθήκη που λείπει:

```
if number == secret:
    print("Σωστά!")
    break
elif συνθήκη: # συμπληρώστε την συνθήκη
    print("Ο μυστικός αριθμός είναι μεγαλύτερος.")
else:
    print("Ο μυστικός αριθμός είναι μικρότερος.")
```

Εκτελέστε το πρόγραμμα και βεβαιωθείτε ότι λειτουργεί σωστά.

Σε ποια περίπτωση εκτελούνται οι εντολές της **else**; Γιατί πιστεύετε ότι δεν ελέγχουμε καμία συνθήκη σε αυτή την τρίτη περίπτωση;

.....  
 .....



22. Αναδιατάξτε τις περιπτώσεις της **if** όπως φαίνεται παρακάτω. Και πάλι, θα πρέπει να συμπληρώσετε τις συνθήκες που ελέγχονται σε κάθε περίπτωση.

```
if συνθήκη: # συμπληρώστε την συνθήκη
    print("Ο μυστικός αριθμός είναι μικρότερος.")
elif συνθήκη: # συμπληρώστε την συνθήκη
    print("Ο μυστικός αριθμός είναι μεγαλύτερος.")
else:
    print("Σωστά!")
    break
```



23. Τι θα αλλάζατε στο πρόγραμμα όπως έχει μέχρι στιγμής; Πώς πιστεύετε ότι πρέπει να επεκταθεί για να γίνει πιο ενδιαφέρον;

.....  
 .....



### Μέτρηση

Ο αριθμός των προσπαθειών που διαθέτει ο παίκτης δεν θα έπρεπε να είναι απεριόριστος. Θα επεκτείνουμε το παιχνίδι ώστε να τερματίζεται όταν εξαντληθούν οι προσπάθειες του παίκτη.

24. Προσθέστε την εντολή που ακολουθεί αμέσως μετά τη **while**, δηλαδή στην αρχή της επανάληψης.

```
print("Απομένουν", tries, "προσπάθειες.")
```

Είναι εμφανές ότι η τιμή της μεταβλητής `tries` θα αντιστοιχεί στο πλήθος των προσπαθειών που απομένουν στον παίκτη.

Αν εκτελέσετε το πρόγραμμα όπως έχει θα εμφανιστεί μήνυμα λάθους, αφού επιχειρούμε να εμφανίσουμε την τιμή της `tries`, χωρίς πουθενά προηγουμένως να της έχουμε αποδώσει μια τιμή.

NameError: name 'tries' is not defined



25. Δώστε στην `tries` μια αρχική τιμή, ίση με το πλήθος των προσπαθειών που διαθέτει ο χρήστης όταν ξεκινά το παιχνίδι. Αποφασίστε εσείς μια τιμή που να σας φαίνεται “δίκαιη”.

Τοποθετήσατε τις νέες εντολές πριν την επανάληψη ή μέσα σε αυτή; Για ποιο λόγο κάνατε αυτή την επιλογή;

.....  
 .....

Ποιο είναι το αρχικό πλήθος προσπαθειών που δώσατε στον παίκτη και γιατί θεωρήσατε αυτή την τιμή “δίκαιη”.

.....  
 .....

Εκτελέστε το πρόγραμμα. Τι παρατηρείτε;

.....



26. Η μεταβλητή `tries` θα πρέπει να μειώνεται σε κάθε γύρο του παιχνιδιού, δηλαδή εντός της επανάληψης. Μετά την `print` του βήματος 24 που εμφανίζει την τιμή της `tries`, προσθέστε τη γραμμή:

```
tries = tries - 1
```

Εκτελέστε το πρόγραμμα. Μειώνεται το πλήθος των προσπαθειών που απομένουν στον παίκτη; Αν το πλήθος των προσπαθειών δεν μειώνεται, βεβαιωθείτε ότι η εντολή που προσθέσατε στο βήμα 25 και δίνει αρχική τιμή στην `tries` βρίσκεται πριν από την επανάληψη και όχι μέσα σε αυτή.



Περιγράψτε πως ακριβώς πιστεύετε ότι λειτουργεί η εντολή που προσθέσατε για να μειώνεται η `tries`.



.....

.....

.....

Υπάρχει κάτι που σας ενοχλεί και φαίνεται να μη δουλεύει σωστά;



.....

## Τερματισμός

Το παιχνίδι δεν πρέπει να τερματίζεται μόνο όταν ο παίκτης μαντέψει τον αριθμό, αλλά και όταν τελειώσουν οι προσπάθειές του.

27. Η `while` συνοδεύεται από μια συνθήκη. Στην αρχή κάθε κύκλου, η συνθήκη αυτή ελέγχεται εκ νέου. Αν η συνθήκη είναι αληθής τότε η επανάληψη συνεχίζεται για άλλον έναν κύκλο.

Εμείς χρησιμοποιήσαμε μέχρι τώρα την τετριμμένη συνθήκη `True`, η οποία είναι πάντα αληθής, γι' αυτό και η επανάληψη δεν διακόπτονταν λόγω της συνθήκης.

Αντικαταστήστε την συνθήκη `True` με την συνθήκη `tries > 0`, που είναι αληθής μόνο όταν απομένουν κι άλλες προσπάθειες στον παίκτη. Σε περίπτωση που αυτό δεν ισχύει, η επανάληψη θα διακοπεί.

`while tries > 0 :`

Εκτελέστε και πάλι το πρόγραμμα και διερευνήστε την συμπεριφορά του. Λειτουργεί σωστά ή εντοπίζετε προβλήματα;



.....

28. Για δοκιμαστικούς λόγους, προσθέστε αμέσως κάτω από την εντολή `tries = tries - 1` τη γραμμή:

`print("Ζντογκ!", tries)`

Εκτελέστε το πρόγραμμα μέχρι να εξαντληθούν οι προσπάθειες. Εμφανίζεται στο τέλος το μήνυμα `"Ζντογκ! 0"`;



.....

Ενώ η συνθήκη της `while` είναι `tries > 0`, από το μήνυμα φαίνεται ότι η εκτέλεση των εντολών της επανάληψης δεν διακόπτεται άμεσα

όταν μηδενιστεί η μεταβλητή `tries` και η συνθήκη πάψει να ισχύει. Άρα η συνθήκη `tries > 0` της `while` δεν ελέγχεται συνεχώς αλλά μόνο στην αρχή κάθε νέου κύκλου της επανάληψης.

Αφαιρέστε τώρα την εντολή που προσθέσατε.

29. Αν ο παίκτης εξαντλήσει τις προσπάθειές του χωρίς να βρει τον αριθμό τότε χάνει και το παιχνίδι σταματά. Προσθέστε τις κατάλληλες εντολές στο πρόγραμμα έτσι ώστε, στην περίπτωση αυτή, να εμφανίζει στον παίκτη τον αριθμό που αναζητούσε, για παράδειγμα:

Ο μυστικός αριθμός ήταν ο 13.

Ουσιαστικά θα επανεισάγετε την εντολή που αφαιρέσατε στο βήμα 17, η οποία εμφανίζει τον μυστικό αριθμό. Θα πρέπει να την τοποθετήσετε στο κατάλληλο σημείο του προγράμματος.

Φροντίστε να εμφανίζεται το μήνυμα μόνο όταν είναι απαραίτητο: αν ο παίκτης βρει τον μυστικό αριθμό, τότε το μήνυμα δε χρειάζεται.

Εκτελέστε το πρόγραμμα. Λειτουργεί σωστά στην περίπτωση που ο παίκτης μαντέψει τον αριθμό; Μήπως στο τέλος του παιχνιδιού του εμφανίζει τον μυστικό αριθμό, παρόλο που τον έχει βρει;

.....

Σε αυτή την περίπτωση, διορθώστε το πρόγραμμα.

Το πρόγραμμα λειτουργεί σωστά στην περίπτωση που ο παίκτης μαντέψει τον αριθμό στην τελευταία του προσπάθεια; Μήπως του εμφανίζει και πάλι τον μυστικό αριθμό, παρόλο που τον έχει βρει;

.....

Αν η απάντηση ήταν καταφατική, μάλλον προσπαθείτε να διαπιστώσετε αν ο παίκτης έχασε ελέγχοντας την συνθήκη `tries == 0`. Σκεφτείτε όμως: αν ο παίκτης μαντέψει τον αριθμό στην τελευταία του προσπάθεια τότε θα ισχύει ότι `tries == 0` όμως ο παίκτης δεν θα έχει χάσει. Χρειάζεται να διορθώσετε το πρόγραμμα ελέγχοντας με διαφορετική συνθήκη αν ο παίκτης απέτυχε να μαντέψει τον αριθμό.

## Περισσότερη Βοήθεια

30. Εκτελέστε το πρόγραμμα. Στην πρώτη σας προσπάθεια δοκιμάστε τον αριθμό 13. Ο μυστικός αριθμός είναι μικρότερος ή μεγαλύτερος;

Αν ο μυστικός αριθμός είναι το 13, απλά εκτελέστε και πάλι το πρόγραμμα.

.....

Σε ποιο διάστημα θ' αναζητήσετε τώρα τον μυστικό αριθμό, δηλαδή ποια είναι η ελάχιστη και ποια η μέγιστη δυνατή τιμή που γνωρίζετε τώρα ότι μπορεί να έχει ο μυστικός αριθμός;

.....



31. Στη δεύτερη προσπάθεια, δοκιμάστε έναν αριθμό που ανήκει στο διάστημα που απαντήσατε προηγουμένως. Συνεχίστε μέχρι να τελειώσει το παιχνίδι, συμπληρώνοντας τον πίνακα που ακολουθεί. Σημειώστε σε κάθε βήμα τον αριθμό που δοκιμάσατε, την απάντηση του προγράμματος και το διάστημα μέσα στο οποίο “εγκλωβίσατε” κάθε φορά τον μυστικό αριθμό. Το διάστημα αυτό ορίζεται από την ελάχιστη (low) και τη μέγιστη (high) δυνατή τιμή που έχει νόημα να δοκιμάσετε μετά από κάθε προσπάθεια.

αριθμός number	ο μυστικός είναι (μικρότερος / μεγαλύτερος)	ελάχιστη low	μέγιστη high
13	.....	.....	.....
.....	.....	.....	.....
.....	.....	.....	.....
.....	.....	.....	.....



32. Τώρα θα επεκτείνουμε το πρόγραμμα έτσι ώστε να βοηθάει το χρήστη ακόμα περισσότερο. Θα χρησιμοποιήσουμε δύο μεταβλητές low και high, οι οποίες αντιστοιχούν στην ελάχιστη και τη μέγιστη δυνατή τιμή που γνωρίζουμε ότι μπορεί να έχει ο μυστικός αριθμός.

Στην αρχή του προγράμματος, αποδώστε αρχικές τιμές σε αυτές τις μεταβλητές:

```
low = 1
high = 32
```

33. Αμέσως πριν από την `input()` με την οποία το πρόγραμμα διαβάζει από το χρήστη έναν αριθμό, προσθέστε μια εντολή που εμφανίζει στο χρήστη τα low και high, για να τον βοηθά στην επιλογή του. Για παράδειγμα, αν τα low και high είναι αντίστοιχα 14 και 23, τότε να εμφανίζει:

Δοκίμασε ανάμεσα στο 14 και το 23.

Εκτελέστε το πρόγραμμα. Υπάρχει κάτι που σας ενοχλεί και φαίνεται να μη δουλεύει σωστά;

.....



34. Στο πρόγραμμα υπάρχει ήδη μια `if` που ελέγχει αν ο μυστικός αριθμός είναι μικρότερος από τον αριθμό του χρήστη:

```
if secret < number:
    print("Ο μυστικός αριθμός είναι μικρότερος.")
```

Στην περίπτωση αυτή, όπως φαίνεται κι από τον πίνακα που συμπληρώσατε στο βήμα 31, μόνο μία από τις μεταβλητές low και high χρειάζεται να αλλάξει τιμή. Ποιά από τις δύο;

.....



Να προσθέσετε σε αυτή την περίπτωση της **if** μια εντολή που μεταβάλλει κατάλληλα την τιμή της `low` ή της `high`. Αν δυσκολευτείτε, ανατρέξτε στον πίνακα που συμπληρώσατε στο βήμα 31.

Στο πρόγραμμα υπάρχει επίσης μια **if** που ελέγχει αν ο μυστικός αριθμός είναι μεγαλύτερος από τον αριθμό του χρήστη:

```
elif secret > number:
    print("Ο μυστικός αριθμός είναι μεγαλύτερος.")
```

Να προσθέσετε και σε αυτή την περίπτωση την εντολή που μεταβάλλει κατάλληλα την τιμή της `low` ή της `high`.

- 35. Εκτελέστε το πρόγραμμά σας αρκετές φορές και ελέγξτε το διεξοδικά, για να διαπιστώσετε αν οι τιμές των μεταβλητών `low` και `high` μεταβάλλονται σωστά.

Λειτουργεί σωστά το πρόγραμμα; Υπάρχει κάποια περίπτωση στην οποία η συμπεριφορά του είναι προβληματική;

.....  
 .....



- 36. **Επέκταση:** Το βήμα αυτό είναι προαιρετικό.

Εκτελέστε ακόμα μερικές φορές το πρόγραμμα. Δοκιμάστε, ως παίκτης, να δίνετε τιμές που είναι εκτός των ορίων που προτείνει το πρόγραμμα. Τί παρατηρείτε; Δημιουργείται πρόβλημα;

.....  
 .....



Αν υπάρχει πρόβλημα όταν ο χρήστης δίνει τιμές εκτός του διαστήματος που ορίζουν οι `low` και `high`, προσθέστε τους κατάλληλους ελέγχους στο πρόγραμμα ώστε να διορθωθούν τυχόν σφάλματα.

## Δραστηριότητες για Εξάσκηση

Για περισσότερη εξάσκηση στις έννοιες που γνωρίσατε σ' αυτό το φύλλο εργασίας, μπορείτε ν' ανατρέξετε στις ασκήσεις των Κεφαλαίων "Μπαρμπούτι" και "Μάντεψε τον Αριθμό".

[pythonies.mysch.gr/complete](https://pythonies.mysch.gr/complete)