

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.

Έννοιες: δομή επιλογής, δομή επανάληψης, υποπρογράμματα, αναζήτηση.

Για ν' ακολουθήσετε αυτό το φύλλο εργασίας, θα πρέπει ήδη να μπορείτε να φτιάχνετε προγράμματα που εμφανίζουν μηνύματα, διαβάζουν τιμές και επιλέγουν τη συμπεριφορά τους ανάλογα με τις συνθήκες που επικρατούν κατά την εκτέλεσή τους. Διαφορετικά, θα πρέπει πρώτα ν' ανατρέξετε στο εισαγωγικό υλικό.

Σε αυτό το φύλλο θα δούμε πως μπορούμε να κάνουμε τα προγράμματά μας να εκτελούν επαναληπτικά τις ίδιες εντολές. Θα δούμε επίσης πως μπορούμε να φτιάχνουμε υποπρογράμματα, δηλαδή ομάδες εντολών που υλοποιούν μια συγκεκριμένη λειτουργία και καλούνται στα σημεία όπου θέλουμε να εκτελεστούν. Πιθανώς να έχετε ήδη έρθει σε επαφή με αυτές τις έννοιες στο εισαγωγικό υλικό, οπότε αυτό το φύλλο θα είναι μια ευκαιρία να εξασκηθείτε και να εμβαθύνετε.

Μάντεψε!

1. Το πρόγραμμα θα ξεκινά επιλέγοντας τον μυστικό αριθμό. Η τιμή που επιλέγει το πρόγραμμά μας θα αποθηκεύεται σε μια μεταβλητή με το όνομα `secret`.

Έστω ότι το πρόγραμμα επιλέγει το τυχερό 13. Προσθέστε στο πρόγραμμα μια εντολή η οποία ορίζει ότι η τιμή της `secret` είναι 13.

Τώρα μπορούμε ν' αναφερόμαστε στο μυστικό αριθμό με το όνομα `secret`, χωρίς να έχει σημασία ποια είναι η τιμή του.

3

22 Ιουλίου 2016

13:21

Εισαγωγικό υλικό:
pythonies.mysch.gr/chapters/answer.pdf
[answer-worksheet.pdf](http://pythonies.mysch.gr/chapters/answer-worksheet.pdf)
[craps.pdf](http://pythonies.mysch.gr/chapters/craps.pdf)
[craps-worksheet.pdf](http://pythonies.mysch.gr/chapters/craps-worksheet.pdf)

Διαβάστε το αντίστοιχο κεφάλαιο:
pythonies.mysch.gr/chapters/guess.pdf

2. Συμπληρώστε το πρόγραμμα έτσι ώστε να ζητά από το χρήστη να μαντέψει τον μυστικό αριθμό, εμφανίζοντας κατάλληλη προτροπή. Για παράδειγμα:

Μάντεψε τον αριθμό:

24

Αποθηκεύστε την τιμή που πληκτρολογεί ο χρήστης σε μια μεταβλητή `number`.

Εκτελέστε το πρόγραμμα. Τα καταφέρατε;

.....

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

Έδωσες τον αριθμό 24

4. Συμπληρώστε το πρόγραμμά σας με μια **if-else**, έτσι ώστε να ελέγχει αν ο παίκτης βρήκε τον μυστικό αριθμό ή όχι και να του εμφανίζει ανάλογο μήνυμα. Για παράδειγμα, τώρα που ο μυστικός αριθμός είναι ο 13, υπάρχουν δύο πιθανές περιπτώσεις:

Μάντεψε τον αριθμό:

13

Σωστά!

Μάντεψε τον αριθμό:

24

Λάθος...

Στη συνθήκη της **if** χρησιμοποιήσατε την μεταβλητή `secret` ή συγκρίνατε την τιμή της `number` απευθείας με το 13;

.....

Στη δεύτερη περίπτωση, *τροποποιήστε* τη συνθήκη της **if**, έτσι ώστε η μεταβλητή `number` να συγκρίνεται με την `secret` και όχι με το 13.

Ποιο πλεονέκτημα πιστεύετε ότι έχει η χρήση της μεταβλητής `secret`, αντί της σταθεράς 13;

.....

.....

5. Εκτελέστε το πρόγραμμα δύο φορές και παίξτε το ρόλο του χρήστη. Την πρώτη φορά δώστε σωστά τον μυστικό αριθμό, ενώ την επόμενη δώστε έναν διαφορετικό.

Μήπως το πρόγραμμά σας εμφανίζει το μήνυμα "**Λάθος...**", ακόμα κι όταν ο χρήστης μαντέψει σωστά τον μυστικό αριθμό; Σε αυτή την περίπτωση, δοκιμάστε να κάνετε την παρακάτω τροποποίηση:

```
number = int( input() )
```



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

Με το `!=` ελέγχεται αν δύο τιμές είναι διαφορετικές.



Γιατί είναι απαραίτητη η χρήση της `int`; Τί πιστεύετε ότι συμβαίνει όταν δεν χρησιμοποιείται η `int` και το πρόγραμμα δεν λειτουργεί;

.....



6. Προσθέστε στο πρόγραμμα μια εντολή που θα εμφανίζει τον μυστικό αριθμό στην περίπτωση που ο παίκτης απαντήσει λάθος, όπως παρακάτω:

Ο μυστικός αριθμός ήταν ο 13.

Βεβαιωθείτε κι εδώ ότι αποκαλύπτετε στο χρήστη την τιμή της `secret`, και όχι την σταθερά 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`.

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

.....



Τυχειότητα

Το παιχνίδι μας δεν αξίζει να το παίξεις πάνω από μία-δύο φορές, αν ο μυστικός αριθμός είναι πάντα το 13. Αυτό που χρειαζόμαστε είναι να επιλέγεται κάθε φορά ένας διαφορετικός αριθμός. Για τον σκοπό αυτό, θα χρειαστούμε τη βιβλιοθήκη `random`.

7. Στην αρχή του προγράμματος προσθέστε την εντολή εισαγωγής της βιβλιοθήκης `random`.

```
import random
```

Τροποποιήστε την εντολή του βήματος 1, 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.

.....

Πως θα χρησιμοποιούσατε τη `randint` για να προσομοιώσετε τη ρίψη ενός ζαριού, τη ρίψη ενός κέρματος ή την επιλογή ενός τυχαίου χαρτιού από μια τράπουλα;

ρίψη ζαριού: `dice = random.randint(... , ...)`

ρίψη κέρματος: `coin = random.randint(... , ...)`

επιλογή χαρτιού: `card = random.randint(... , ...)`

Μπορείτε να σκεφτείτε άλλες περιπτώσεις στις οποίες θα χρειαζόσασταν τη `randint`;

.....

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

.....

.....

Γύρω-Γύρω Όλοι

Θα θέλαμε να επεκτείνουμε το παιχνίδι έτσι ώστε ο παίκτης να έχει επαναλαμβανόμενες ευκαιρίες να βρει τον μυστικό αριθμό, ενώ ο μυστικός αριθμός παραμένει ο ίδιος.

9. Προσθέστε τη γραμμή που ακολουθεί αμέσως μετά από την εντολή του βήματος 7, που δίνει μια τυχαία τιμή στη μεταβλητή `secret`.

```
while True:
```

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

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

.....

.....



Μπορείτε να διακόψετε την εκτέλεση του προγράμματός σας με τον συνδυασμό πλήκτρων `Ctrl + C`.

Η εντολή του βήματος 7, 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 6 εμφανίζει τον μυστικό αριθμό, στην περίπτωση που ο χρήστης δεν τον μαντέψει. Τώρα, αυτή η εντολή βρίσκεται μέσα στη `while` κι εκτελείται σε κάθε αποτυχημένη προσπάθεια του χρήστη. Γιατί αυτό είναι πρόβλημα;

.....



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;

.....



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

Πρέπει να ελέγχετε αν τοποθετήσατε την `break` στο σωστό σημείο. Εκτελέστε το πρόγραμμα και παίξτε το παιχνίδι δοκιμάζοντας αριθμούς. Διακόπτεται η επανάληψη όταν μαντέψετε τον μυστικό αριθμό;

.....



11. 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.

12. Τί 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.

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

```

if secret == number:
    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;

.....

.....

.....



14. Αναδιατάξτε τις περιπτώσεις της **if** όπως φαίνεται παρακάτω. Και πάλι, θα πρέπει να συμπληρώσετε μια από τις συνθήκες που ελέγχονται. Είναι καλή εξάσκηση και θα σας κάνει να σκεφτείτε για την σειρά με την οποία ελέγχονται οι συνθήκες.

```

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

Εκτελέστε το πρόγραμμα και βεβαιωθείτε ότι λειτουργεί σωστά.

Τί 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.

15. Αμέσως μετά τη **while**, δηλαδή στην αρχή της επανάληψης, προσθέστε την εντολή που ακολουθεί:

```
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
```

16. Δώστε στην `tries` την αρχική τιμή 4. Αυτό θα είναι το πλήθος των προσπαθειών που διαθέτει ο χρήστης όταν ξεκινά το παιχνίδι.

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

.....

Εκτελέστε το πρόγραμμα. Τί παρατηρείτε;

.....

17. Η μεταβλητή `tries` θα πρέπει να μειώνεται σε κάθε γύρο του παιχνιδιού, δηλαδή εντός της επανάληψης. Μετά την `print` του βήματος 15 που εμφανίζει το πλήθος των προσπαθειών, 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 16 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.

18. Προς το παρόν, προσθέστε τις εντολές που ακολουθούν στο σημείο που κρίνετε κατάλληλο ώστε η επανάληψη να τερματίζεται όταν εξαντληθούν οι προσπάθειες του παίκτη.

```
if tries == 0:
    break
```

Εκτελέστε το πρόγραμμά σας. Φροντίστε, ως παίκτης, να εξαντλήσετε τις προσπάθειές σας χωρίς να μαντέψετε τον αριθμό.

Ποιες τιμές επιλέξατε να δοκιμάσετε, για να είστε βέβαιοι ότι δεν θα πετύχετε τον μυστικό αριθμό και θα εξαντλήσετε τις προσπάθειες;

.....

Τερματίζεται το πρόγραμμα όταν εξαντληθούν οι προσπάθειες του παίκτη;

.....

Είστε βέβαιοι ότι το πρόγραμμα επιτρέπει στον παίκτη να χρησιμοποιήσει όλες του τις προσπάθειες; Αν προσθέσατε τις εντολές αμέσως μετά την εντολή `tries = tries - 1`, τότε το πρόγραμμά σας θα στερεί από τον παίκτη μια προσπάθεια!

19. Διαγράψτε τις εντολές που προσθέσατε στο προηγούμενο βήμα. Αντί για την `break`, θα διερευνήσουμε έναν εναλλακτικό (και συχνά προτιμότερο) τρόπο για να τερματίζουμε την επανάληψη όταν εξαντληθούν οι προσπάθειες.

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;

.....

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




```
print("Ζντονκ!", tries)
```

Εκτελέστε το πρόγραμμα μέχρι να εξαντληθούν οι προσπάθειες. Εμφανίζεται στο τέλος το μήνυμα "Ζντονκ! 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.

- 21. Αν ο παίκτης εξαντλήσει τις προσπάθειές του και δεν καταφέρει 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.

Θα χρειαστεί, μεταξύ άλλων, να επανεισάγετε την εντολή που αφαιρέσατε στο βήμα 9, η οποία εμφάνιζε τον μυστικό αριθμό.

Ο μυστικός αριθμός ήταν ο 13.

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

Τοποθετήσατε τις νέες εντολές μέσα στην επανάληψη ή μετά από αυτή; Για ποιο λόγο κάνατε αυτή την επιλογή;

.....

.....

Τι διαφορά θα υπήρχε αν είχατε κάνει την αντίθετη επιλογή;

.....

.....

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 μαντεύει 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

22. 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;

.....

23. 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
.....
.....
.....

Θεωρείτε "δίκαιο" να δίνονται στον παίκτη τέσσερις προσπάθειες και γιατί;

.....

.....

24. 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
```

- 25. 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 και παρατηρήστε τις τιμές των `low` και `high`, καθώς προσπαθείτε να μαντέψετε τον μυστικό αριθμό. Υπάρχει κάτι που σας ενοχλεί και φαίνεται να μη δουλεύει σωστά;



.....

- 26. 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 23, 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 23.

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`.

- 27. 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;



.....

28. 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.

Gimme a Break

Η χρήση της **break** είναι μια πρακτική που δεν ακολουθείται από όλους. Ορισμένοι θεωρούν ότι ο κώδικας είναι πιο κατανοητός όταν υπάρχει ένα μοναδικό σημείο εξόδου από την επανάληψη: η συνθήκη συνέχειας.

29. Μπορούμε να χρησιμοποιήσουμε μια λογική μεταβλητή, την οποία θα ονομάσουμε found, για να “θυμάται” το πρόγραμμά μας αν ο χρήστης βρήκε τον μυστικό αριθμό.

Προσθέστε στην αρχή του προγράμματός σας την εντολή:

```
found = False # η Found είναι ψευδής
```

Για διαγνωστικούς λόγους, προσθέστε προσωρινά στο τέλος της επανάληψης (μέσα σε αυτή, όχι μετά) την εντολή:

```
print("Ζητοκ!", found)
```

Εκτελέστε το πρόγραμμά σας. Θα πρέπει σε κάθε επανάληψη να βλέπετε το μήνυμα "Ζητοκ! False"

30. Προσθέστε την εντολή που ακολουθεί στο τμήμα του προγράμματος που εκτελείται μόνο όταν ο χρήστης εντοπίσει τον μυστικό αριθμό.

```
found = True # η Found γίνεται αληθής
```

Εκτελέστε το πρόγραμμα. Πότε εμφανίζεται το "Ζητοκ! True" και γιατί;

.....

31. Αφαιρέστε την **break** από το πρόγραμμα. Ποιο αποτέλεσμα πιστεύετε ότι θα έχει αυτό;

.....

Προσθέστε στη συνθήκη της **while** μια επιπλέον συνθήκη που θα ελέγχει την τιμή της found για να διαπιστώνει αν η επανάληψη θα πρέπει να συνεχιστεί.



Υπάρχουν μόνο δύο λογικές τιμές: True και False.



Για να συνδυάσετε αυτή την συνθήκη με την ήδη υπάρχουσα συνθήκη `tries > 0` χρησιμοποιήστε ανάμεσά τους το **and** που έχει σαν αποτέλεσμα την σύζευξη των συνθηκών.

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

.....

Πότε εμφανίζεται το **"Ζντονκ! True"** και γιατί;

.....

Αφαιρέστε το διαγνωστικό μήνυμα από την επανάληψη.

32. Αντικαταστήστε την συνθήκη `number != secret` στο τέλος του παιχνιδιού με μια συνθήκη που ελέγχει την τιμή της `found`.

33. Προηγουμένως, το παιχνίδι έληγε όταν ο παίκτης έβρισκε τον αριθμό με τη χρήση της **break**. Τώρα η **break** αφαιρέθηκε και ο τερματισμός της επανάληψης γίνεται αποκλειστικά όταν η συνθήκη συνέχειας της **while** είναι ψευδής. Ποια πιστεύετε ότι είναι η διαφορά ανάμεσα στις δύο περιπτώσεις;

.....

.....

.....

Εξαρτήματα

34. Και τώρα, ας ξεχάσουμε για λίγο το πρόγραμμα που έχουμε αναπτύξει. Διαβάστε προσεκτικά τον κώδικα που ακολουθεί. Μην τον πληκτρολογήσετε, απλά εξετάστε τον.

```
print("Μάντεψε τον αριθμό.")
print("Δοκίμασε ανάμεσα στο", a, "και το", b)
c = int(input())
print("Έδωσες τον αριθμό", c)
```

Ποιες μεταβλητές χρειάζεται να έχουν ήδη τιμή για να λειτουργήσει αυτό το τμήμα κώδικα;

.....

Σε ποιες μεταβλητές αποδίδεται τιμή σε αυτό το τμήμα κώδικα;

.....

Περιγράψτε, όσο καλύτερα μπορείτε, τη λειτουργία αυτού του μικρού τμήματος κώδικα. Τί θα λέγατε ότι κάνει;

.....



Υπάρχει κάποιο τμήμα κώδικα στο πρόγραμμά σας που να εκτελεί αυτή τη συγκεκριμένη λειτουργία;

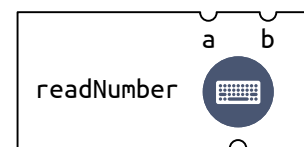


35. Θα διαπιστώσατε ότι το πρόγραμμα περιλαμβάνει ένα τέτοιο τμήμα κώδικα, το οποίο χρησιμοποιείται για να εισαχθεί από το χρήστη μια πιθανή τιμή για τον μυστικό αριθμό.

Θα περικλείσουμε αυτόν τον κώδικα σε μια *συνάρτηση*, ένα υποπρόγραμμα που επιτελεί μια συγκεκριμένη λειτουργία.

Προσθέστε στην αρχή του προγράμματος τα εξής:

```
def readNumber(a,b):
    print("Μάντεψε τον αριθμό.")
    print("Δοκίμασε ανάμεσα στο", a, "και το", b)
    c = int(input())
    print("Έδωσες τον αριθμό", c)
    return c
```



Εικόνα: Η συνάρτηση `readNumber` είναι ένα ανεξάρτητο τμήμα κώδικα, ένα αυτόνομο “εξάρτημα” που επιτελεί μια συγκεκριμένη λειτουργία.

Με τον τρόπο αυτό *ορίζεται* η συνάρτηση `readNumber`, η οποία λειτουργεί ως εξής: Δέχεται δύο παραμέτρους, δύο τιμές που τις ονομάζει `a` και `b` για να μπορεί να αναφέρεται σε αυτές. Προτρέπει το χρήστη να πληκτρολογήσει έναν αριθμό μεταξύ των `a` και `b` και *επιστρέφει* με την εντολή `return` την ακέραια τιμή που πληκτρολογεί ο χρήστης.

Οι μεταβλητές `a`, `b` και `c` έχουν *τοπική εμβέλεια*, δηλαδή δεν είναι “ορατές” από το υπόλοιπο πρόγραμμα. Δημιουργούνται εκ νέου κάθε φορά που εκτελείται η συνάρτηση `readNumber` και μετά παύουν να υφίστανται.

Γιατί φτιάξαμε αυτό το μικρό “εξάρτημα”; Πρώτον, για να το χρησιμοποιήσουμε στο πρόγραμμά μας. Δεύτερον, για να μπορέσουμε αργότερα να το αντικαταστήσουμε με παρόμοια εξαρτήματα που κάνουν την ίδια δουλειά (επιστρέφουν μια πιθανή τιμή για τον μυστικό αριθμό) με διαφορετικό τρόπο.

36. Κατασκευάσαμε ένα “εξάρτημα” που ονομάζεται `readNumber` και επιτελεί μια συγκεκριμένη λειτουργία, ωστόσο δεν το έχουμε ακόμα *χρησιμοποιήσει*.

Εντοπίστε στο πρόγραμμά σας τις παρακάτω εντολές:

```
print("Μάντεψε τον αριθμό.")
print("Δοκίμασε ανάμεσα στο", low, "και το", high)
number = int(input())
print("Έδωσες τον αριθμό", number)
```

Διαγράψτε τις και αντικαταστήστε με την παρακάτω γραμμή:

```
number = readNumber(low, high)
```

Κατά την εκτέλεση του προγράμματος, αυτή η γραμμή θα καλέσει την `readNumber`, θα ενεργοποιήσει την εκτέλεση των εντολών της, παρέχοντας ως παραμέτρους τις τιμές των `low` και `high`. Την τιμή που επιστρέφεται από την συνάρτηση την ονομάζουμε `number`.

Εκτελέστε το πρόγραμμα. Παρατηρείτε κάποια διαφορά στον τρόπο που λειτουργεί;

.....



37. Προσθέστε στην αρχή του προγράμματός σας τον κώδικα που ακολουθεί. Συμπληρώστε κατάλληλα τις γραμμές που συνοδεύονται από σχόλιο, έτσι ώστε η συνάρτηση `randNumber` που ορίζεται εδώ να επιστρέφει έναν τυχαίο αριθμό ανάμεσα στα `a` και `b`.

```
import time
def randNumber(a,b):
    print("Μάντεψε τον αριθμό.")
    print("Δοκίμασε ανάμεσα στο", a, "και το", b)
    c = επιλογή τυχαίου αριθμού # συμπληρώστε
    print("Το πρόγραμμα επιλέγει", c)
    time.sleep(3)
    return επιστροφή τιμής # συμπληρώστε
```

Στο κύριο πρόγραμμα, εντοπίστε την εντολή με την οποία η μεταβλητή `number` παίρνει τιμή από την συνάρτηση `readNumber` και προσθέστε μπροστά από την εντολή το σύμβολο `#`. Αυτό έχει σαν αποτέλεσμα η εντολή αυτή να θεωρείται σχόλιο και να αγνοείται κατά την εκτέλεση του προγράμματος.

Είναι συνηθισμένο να “σχολιάζουμε” μια εντολή όταν θέλουμε να παρακαμφθεί, χωρίς όμως να την αφαιρέσουμε πλήρως από το πρόγραμμα. Έτσι μπορούμε να την επαναφέρουμε αργότερα, αν αυτό είναι απαραίτητο.

Στο ίδιο σημείο, προσθέστε μια εντολή με την οποία η μεταβλητή `number` θα παίρνει τιμή από την συνάρτηση `randNumber`, με παραμέτρους τις τιμές των `low` και `high`. Αν δυσκολευτείτε, ανατρέξτε στο βήμα 37, όπου γίνεται κάτι ανάλογο.

Ουσιαστικά, εκεί που το πρόγραμμα χρησιμοποιούσε τη `readNumber` για να αποκτηθεί μια πιθανή τιμή για τον μυστικό αριθμό, τώρα αυτή αντικαταστάθηκε από την `randNumber`.

Χρησιμοποιείται πουθενά στο πρόγραμμα η `input`, για να παρέχει οποιαδήποτε είσοδο ο χρήστης; Αν όχι, ποιος θα επιλέγει τώρα αριθμούς, προσπαθώντας να μαντέψει τον τυχαίο αριθμό;

.....

.....

Εκτελέστε το πρόγραμμα. Τι αλλαγή επιφέρει η χρήση της συνάρτησης `randNumber`, αντί για τη `readNumber`;



.....



38. Όταν ξεκινά το πρόγραμμα και γνωρίζετε ότι ο μυστικός αριθμός βρίσκεται κάπου ανάμεσα στο 1 και το 32, ποια τιμή σας φαίνεται προτιμότερο να επιλέξετε;

.....
 Αν γνωρίζετε ότι ο μυστικός αριθμός βρίσκεται κάπου ανάμεσα στο 13 και το 23, ποια τιμή σας φαίνεται προτιμότερο να επιλέξετε;



.....
 Αν γνωρίζετε ότι ο μυστικός αριθμός βρίσκεται κάπου ανάμεσα στο low και το high, ποια τιμή σας φαίνεται προτιμότερο να επιλέξετε και γιατί;



.....



39. Κατασκευάστε μια συνάρτηση midNumber. Η συνάρτηση θα δέχεται σαν παραμέτρους δύο τιμές, που θ' αντιστοιχούν στα όρια του διαστήματος μέσα στο οποίο γνωρίζουμε ότι βρίσκεται ο μυστικός αριθμός, και θα επιστρέφει την τιμή που βρίσκεται στο μέσο αυτού του διαστήματος.

Χρησιμοποιήστε ως πρότυπα τις readNumber και randNumber.

Στο κύριο πρόγραμμα, εντοπίστε την εντολή με την οποία η μεταβλητή number παίρνει τιμή από την συνάρτηση randNumber και προσθέστε μπροστά από την εντολή το σύμβολο #.

Στο ίδιο σημείο, προσθέστε μια εντολή με την οποία η μεταβλητή number θα παίρνει τιμή από την συνάρτηση midNumber, με παραμέτρους τις τιμές των low και high.

Εκτελέστε το πρόγραμμα μερικές φορές. Καταφέρνει να εντοπίσει τον μυστικό αριθμό;

.....



Ο μυστικός αριθμός είναι ακέραιος, άρα η τιμή που επιλέγει η midNumber πρέπει να είναι επίσης ακέραια. Φροντίστε να χρησιμοποιήσετε τον τελεστή // της ακέραιας διαίρεσης για τον υπολογισμό του μέσου του διαστήματος.

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