

Ασκήσεις

Υπάρχουν πολλά παιχνίδια στα οποία κάποιος σας ζητά να σκεφτείτε έναν μυστικό αριθμό και, μετά από μερικές ερωτήσεις και υπολογισμούς, «μαντεύει» τον αριθμό που είχατε σκεφτεί. Ένα πολύ παλιό παράδειγμα βασίζεται σ' ένα θεώρημα του κινέζου Sun Tzu, που έζησε κάπου ανάμεσα στον 3 και τον 5 αιώνα. Ας υποθέσουμε ότι ο άγνωστος αριθμός είναι ο x . Αν ονομάσουμε α , β και γ τα υπόλοιπα της διαίρεσης του x με το 3, το 5 και το 7 αντίστοιχα, τότε για να βρούμε τον άγνωστο αριθμό θα πρέπει να υπολογίσουμε την τιμή της παράστασης $70\alpha + 21\beta + 15\gamma$ και στη συνέχεια να διαιρέσουμε με το 105. Το υπόλοιπο της διαίρεσης θα είναι ο μυστικός αριθμός. Να αναπτύξετε ένα πρόγραμμα το οποίο ζητάει από το χρήστη να σκεφτεί έναν μυστικό αριθμό από το 1 μέχρι και το 100. Στη συνέχεια, το πρόγραμμα ρωτάει το χρήστη ποιο είναι το υπόλοιπο της διαίρεσης αυτού του αριθμού με το 3, το 5 και το 7 και ανακοινώνει τον μυστικό αριθμό, αφήνοντας το χρήστη έκπληκτο με τις υπερφυσικές του ικανότητες.

Υπόδειξη

```
# εισαγωγή της βιβλιοθήκης time (για την εισαγωγή χρονοκαθυστερήσεων μεταξύ των
# μηνυμάτων που θα εμφανίζονται)
.....
# εμφάνιση μηνυμάτων προς τον χρήστη που θα τον προτρέπει να επιλέξει έναν αριθμό από
# το 1 έως το 100
.....
# εμφάνιση μηνυμάτων όπου ο χρήστης θα απαντά με τα υπόλοιπα των διαιρέσεων με το 3 το
# 5 και το 7 αντίστοιχα
.....
# υπολογισμός της αριθμητικής παράστασης ( $70*\alpha + 21*\beta + 15*\gamma$ ) και του υπολοίπου της
# διαίρεσης με το 105
.....
```

Η παραπάνω υλοποίηση εκτελεί την αναζήτηση του μυστικού αριθμού μια φορά. Προσθέτοντας κατάλληλα την δομή επανάληψης `while` μπορεί ο χρήστης να εκτελεί ξανά και ξανά το πρόγραμμα.

Λίγα λόγια για την `print`

Η `end` χρησιμοποιείται ώστε να μην γίνει αλλαγή γραμμής μεταξύ δυο `print` (`end = ""`)

Εναλλακτικά μπορεί να πάρει και άλλους χαρακτήρες εκτός από το κενό. Εάν ως τιμή δοθεί το `"\n\n"` τότε εκτυπώνεται μια κενή γραμμή μεταξύ των δυο `print`

Πειραματιστείτε με το παρακάτω κομμάτι κώδικα αλλάζοντας την τιμή της παραμέτρου `end`.

```
# μετά το μήνυμα τυπώνεται το περιεχόμενο της end
# (εδώ, ένα κενό ή τα αποσιωπητικά)
# και το επόμενο μήνυμα τυπώνεται στην ίδια γραμμή
print("Hello.", end = " ")
print("World", end = "... ")
print(("!!!"))
```

Να κατασκευάσετε ένα πρόγραμμα το οποίο θα παίζει επαναληπτικά το γνωστό παιχνίδι «Πέτρα-Ψαλίδι-Χαρτί» με το χρήστη. Σε κάθε γύρο, το πρόγραμμα θα ζητά από το χρήστη την επιλογή του (1 για την Πέτρα, 2 για το Ψαλίδι ή 3 για το Χαρτί) και στη συνέχεια θα εμφανίζει τη δική του επιλογή και θα ανακοινώνει το νικητή του γύρου. Αν ο χρήστης πληκτρολογήσει οτιδήποτε εκτός από τις τρεις έγκυρες επιλογές τότε θεωρείται ότι επιθυμεί να τερματίσει το παιχνίδι.

Υπόδειξη

Ως γνωστόν: Η πέτρα νικάει το ψαλίδι και χάνει από το χαρτί, το ψαλίδι νικάει το χαρτί και χάνει από την πέτρα, το χαρτί νικάει την πέτρα και χάνει από το ψαλίδι

Για να χρησιμοποιήσουμε μια βιβλιοθήκη θα πρέπει πρώτα να την εισάγουμε (`import`). Εδώ θα εισάγουμε τη βιβλιοθήκη **random** και θα χρησιμοποιήσουμε την **randint()**, που παράγει τυχαίους ακέραιους εντός συγκεκριμένων ορίων που καθορίζει ο προγραμματιστής.

```
# παράδειγμα για την λειτουργία της randint
import random
print random.randint(0,10) # πιθανά αποτελέσματα 0,1,2,...,10
```

Θα χρειαστεί (προαιρετικά αλλά θα εμφανιστεί καλύτερα η έξοδος του προγράμματος) να εισάγουμε και τη βιβλιοθήκη **time** για χρήση της **sleep**.

```
import random
import time
```

Δηλώστε δυο μεταβλητές που θα λειτουργούν ως μετρητές για τον υπολογισμό των νικών του παίχτη και των συνολικών παιχνιδιών που παίχθηκαν

```
# μετρητής των συνολικών παιχνιδιών
.....
# μετρητής των παιχνιδιών που κέρδισε ο παίκτης
.....
```

Ανακοινώστε με ένα μήνυμα την έναρξη του παιχνιδιού

```
print(".....")
```

Ο παίκτης θα έχει την δυνατότητα να παίζει όσα παιχνίδια θέλει επιλέγοντας 1 για πέτρα, 2 για ψαλίδι και 3 για χαρτί . Όταν δώσει μια διαφορετική επιλογή (π.χ 0, 4, 20 κτλ) το παιχνίδι θα τερματίζει. Θα χρειαστούμε μια δομή επανάληψης **while** που θα περιέχει έναν έλεγχο (if) της τιμής που έδωσε ο παίκτης.

```
while True:
    .....
    .....
    # αν ο παίκτης δώσει κάτι διαφορετικό εκτός από 1, 2 ή 3 τότε τερματίζουμε άμεσα την
    επανάληψη με break
    if .....
        .....
        break
```

Η **break** διακόπτει και τερματίζει άμεσα τον κύκλο της επανάληψης, χωρίς να ελεγχθεί η συνθήκη συνέχειας. Οι εντολές μετά την **break** αγνοούνται και η εκτέλεση συνεχίζεται από την πρώτη εντολή που

ακολουθεί την επανάληψη. Για να κατανοήσετε καλύτερα τι σημαίνει η άμεση διακοπή της επανάληψης όταν εκτελεστεί η `break`, δοκιμάστε να τοποθετήσετε κάποια εντολή `print` μετά την `break`. Θα διαπιστώσετε ότι τα μηνύματα δεν θα εμφανιστούν ποτέ.

Στην συνέχεια θα συμπληρώσετε τον κώδικα που θα εκτελεστεί όταν δεν ενεργοποιείται η **break**

```
# η επιλογή του προγράμματος με τη χρήση της random.randint
.....
# δομή επιλογής για την εμφάνιση της τυχαίας επιλογής του προγράμματος
if .....
    .....
elif .....
    .....
else:
    .....
```

Το **elif** σημαίνει **else if**. Μετά από μια **if** μπορούμε να χρησιμοποιήσουμε όσες **elif** είναι απαραίτητες. Οι εντολές που ακολουθούν τις **if** και **elif** είναι στοιχισμένες δεξιότερα. Η στοίχιση υποδηλώνει ότι αυτές οι εντολές θα εκτελεστούν μόνο αν η αντίστοιχη συνθήκη είναι αληθής και καμία από τις προηγούμενες συνθήκες δεν είναι αληθείς. Οι εντολές που ακολουθούν την `else` είναι επίσης στοιχισμένες δεξιότερα και θα εκτελεστούν μόνο εφόσον οι προηγούμενες συνθήκες είναι ψευδείς.

```
# σύγκριση των επιλογών του παίκτη και του υπολογιστή και εμφάνιση κατάλληλου μηνύματος
if .....
    print("Ισοπαλία.")
elif .....
    print("Κέρδισες!")
    ..... # εδώ αυξάνουμε την μεταβλητή που μετράει το πλήθος των νικών του παίκτη
else:
    print("Κέρδισε ο υπολογιστής :P")
    ..... # εδώ αυξάνουμε την μεταβλητή που μετράει το πλήθος των παιχνιδιών
# εκτύπωσε 20 φορές τον χαρακτήρα "-"
print(20 * "-")
time.sleep(2)
# εμφάνιση μηνύματος για την συνολική επίδοση (πόσες νίκες σε πόσα παιχνίδια)
# η print θα εκτελεστεί μία φορά μετά την break
print(".....")
```

Ιδέες για συνέχεια

Τροποποιήστε κατάλληλα τον κώδικα ώστε να δημιουργήσετε το παιχνίδι «Πέτρα – Μολύβι – Ψαλίδι – Χαρτί»

Μπορείτε να προτείνετε μια λύση χωρίς τη χρήση της `break` ;