

Το Παιχνίδι της Αφαίρεσης

Φύλλο Εργασίας

Σε αυτό το φύλλο εργασίας θα φτιάξουμε ένα παιχνίδι για δύο παίκτες που ονομάζεται NIM. Είναι πολύ παλιό και πιθανότατα προέρχεται από την Κίνα. Εδώ θ' ασχοληθούμε με μια από τις πολλές παραλλαγές του, μια απλή εκδοχή που ονομάζεται *το παιχνίδι της αφαίρεσης*. Στο τέλος, το πρόγραμμά μας θα συμμετέχει στο παιχνίδι, παίζοντας το ρόλο ενός από τους δύο παίκτες.

Έννοιες: δομή επιλογής, δομή επανάληψης, υποπρογράμματα

Για ν' ακολουθήσετε αυτό το φύλλο εργασίας, θα πρέπει *ήδη* να μπορείτε να φτιάχνετε προγράμματα που εμφανίζουν μηνύματα, διαβάζουν τιμές, επιλέγουν τη συμπεριφορά τους ανάλογα με τις συνθήκες που επικρατούν κατά την εκτέλεσή τους και εκτελούν επαναλαμβανόμενα εντολές. Θα πρέπει επίσης να έχετε εξοικειωθεί με το "σπάσιμο" λειτουργιών του προγράμματος σε *υποπρογράμματα*. Διαφορετικά, θα πρέπει πρώτα ν' ανατρέξετε στο *εισαγωγικό υλικό*.

Σε αυτό το φύλλο θα εξασκηθούμε στην επαναληπτική εκτέλεση εντολών και θα εξετάσουμε σε μεγαλύτερο βάθος την ανάπτυξη υποπρογραμμάτων. Έχετε ήδη έρθει σ' επαφή με αυτές τις έννοιες στο εισαγωγικό υλικό κι αυτό το φύλλο είναι μια ευκαιρία για εμβάθυνση.

Το Στήσιμο

Ένα πλήθος από αντικείμενα (π.χ. σπίρτα, ξυλάκια) τοποθετούνται στη σειρά και ο κάθε ένας από τους δύο παίκτες αφαιρεί με τη σειρά του από ένα μέχρι και τρία αντικείμενα, μέχρι να μη μείνει κανένα. Ο παίκτης που θα πάρει το τελευταίο αντικείμενο *χάνει* το παιχνίδι.

1. Ας υποθέσουμε ότι τα αντικείμενα που χρησιμοποιούν οι παίκτες είναι σπίρτα. Οι κανόνες του παιχνιδιού δεν προσδιορίζουν το αρχικό πλήθος σπίρτων, μπορούμε λοιπόν να το ορίσουμε μόνοι μας.

Γίνεται το παιχνίδι να ξεκινάει κάθε φορά μ' ένα διαφορετικό αριθμό σπίρτων; Αν ναι, ποιος είναι ο τρόπος;

.....
.....



10 Σεπτεμβρίου 2016
10:00

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)
[guess.pdf](http://pythonies.mysch.gr/chapters/guess.pdf)
[guess-worksheet.pdf](http://pythonies.mysch.gr/chapters/guess-worksheet.pdf)

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



Ας ονομάσουμε το πλήθος των σπύρτων `matches`. Μπορούμε να αρχικοποιήσουμε τη μεταβλητή `matches` με μια τυχαία τιμή (ας πούμε από το 7 μέχρι και το 21).

Προσθέστε τις κατάλληλες εντολές στο πρόγραμμά, ώστε να εισάγει τη βιβλιοθήκη `gandom` και να παράγει μια τυχαία τιμή ανάμεσα στο 7 και στο 21, την οποία και θ' αποθηκεύει στη μεταβλητή `matches`.

2. Προσθέστε την κατάλληλη εντολή στο πρόγραμμά, προκειμένου να εμφανίζει τον αρχικό αριθμό σπύρτων στον παίκτη. Για παράδειγμα:

Αρχικό πλήθος σπύρτων: 13

Εκτελέστε μερικές φορές το πρόγραμμά σας. Εμφανίζεται διαφορετικός αριθμός σπύρτων σε κάποιες εκτελέσεις;

Σημείωση: Δεν αποκλείεται κάποιες από τις τυχαίες τιμές να είναι οι ίδιες.

.....



Κάνε Ένα Γύρο

3. Συμπληρώστε το πρόγραμμα ώστε να ζητά από τον παίκτη τον αριθμό των σπύρτων που θα αφαιρέσει, εμφανίζοντας κατάλληλη προτροπή. Για παράδειγμα:

Πόσα σπύρτα θέλεις;

Αποθηκεύστε την απάντηση του παίκτη στη μεταβλητή `removed`.

4. Προσθέστε την παρακάτω εντολή προκειμένου να μειωθεί ο αριθμός των σπύρτων `matches` ανάλογα με το πλήθος των σπύρτων `removed` που ζήτησε ο παίκτης να αφαιρέσει.

```
matches = matches - removed
```

Εκτελέστε το πρόγραμμά σας. Λειτουργεί σωστά ή εμφανίζει κάποιο μήνυμα σφάλματος;

.....

Σε περίπτωση που σας εμφανίζει το παρακάτω μήνυμα σφάλματος:

`TypeError: unsupported operand type(s) for -`

βεβαιωθείτε ότι έχετε χρησιμοποιήσει την `int()` όταν ζητάτε τον αριθμό των σπύρτων που θα αφαιρέσει ο παίκτης, προκειμένου να μετατρέψετε την τιμή αυτή σε ακέραιο αριθμό.

5. Συμπληρώστε το πρόγραμμα με την κατάλληλη εντολή ώστε να εμφανίζεται στους παίκτες ο αριθμός των σπύρτων που απομένουν. Έτσι θα μπορούν ν' αποφασίζουν για το πλήθος των σπύρτων που θα αφαιρέσουν στον επόμενο γύρο. Για παράδειγμα:

Σπύρτα που απομένουν: 15



Εκτελέστε το πρόγραμμά σας. Εμφανίζει σωστά τον αριθμό των σπάρτων που απομένουν;



6. Η αφαίρεση σπάρτων από τους παίκτες δεν γίνεται μόνο μια φορά, αλλά πολλές. Θα χρησιμοποιήσουμε τη **while** προκειμένου οι εντολές να εκτελούνται επαναληπτικά.

Το συγκεκριμένο παιχνίδι συνεχίζεται όσο απομένει τουλάχιστον ένα σπάρτο. Τι τιμές πρέπει να έχει η μεταβλητή `matches`, προκειμένου να συνεχίζεται η επανάληψη;



*Προσθέστε την εντολή **while** μαζί με την κατάλληλη συνθήκη που εξετάζει την τιμή της `matches`, όπως την περιγράψατε προηγουμένως. Εμφωλεύστε τις εντολές που πιστεύετε ότι χρειάζεται να εκτελούνται επαναληπτικά, προσθέτοντας τις κατάλληλες εσοχές.*

Εκτελέστε το πρόγραμμά σας. Τερματίζει η επανάληψη και, αν ναι, τότε γίνεται αυτό;



Αν η επανάληψη δεν σταματά τότε ελέγξτε ότι έχετε τοποθετήσει την εντολή που μειώνει τον αριθμό των σπάρτων μέσα στην **while** χρησιμοποιώντας τις κατάλληλες εσοχές και εκτελέστε ξανά το πρόγραμμα.



Τι τιμή παίρνει η συνθήκη που εξετάζει η **while**, ώστε να σταματήσει η επανάληψη;



7. Γιατί πιστεύετε ότι χρησιμοποιήσαμε τη μεταβλητή `matches` στο βήμα 4 για να αποθηκεύσουμε εκ νέου το αποτέλεσμα; Τι διαφορά θα είχε αν είχαμε χρησιμοποιήσει ένα άλλο όνομα στη θέση της;



Τροποποιήστε την εντολή του βήματος 4 που αφαιρεί τα σπάρτα που ζήτησε ο παίκτης, ώστε να μην αποθηκεύει το αποτέλεσμα της ξανά στη μεταβλητή `matches`, αλλά σε μια νέα μεταβλητή, όπως παρακάτω:

```
remaining = matches - removed
```

*Τροποποιήστε και την **print** που εμφανίζει τον αριθμό των σπάρτων που απομένουν, ώστε να εμφανίζει πλέον τη μεταβλητή `remaining`.*

Εκτελέστε το πρόγραμμα δίνοντας διάφορες τιμές στον αριθμό των σπάρτων που αφαιρεί ο παίκτης. Τι παρατηρείτε;



.....

Γιατί πιστεύετε ότι συμβαίνει αυτό;



.....

Επαναφέρετε το πρόγραμμα, ώστε να χρησιμοποιεί τη μεταβλητή matches αντί για τη μεταβλητή gameing, όπως ήταν αρχικά.

Ποιός Παίζει;

Σε κάθε κύκλο του παιχνιδιού, το πρόγραμμα ρωτάει πόσα σπέρτα θα αφαιρεθούν. Ωστόσο, δεν καταγράφει ποιος από τους δύο παίκτες είναι που αφαιρεί κάθε φορά τα σπέρτα κι έτσι δεν είναι σε θέση να υπολογίσει ποιος είναι ο νικητής όταν τα σπέρτα τελειώσουν.

- Ένας τρόπος για να λύσουμε το πρόβλημα είναι να χρησιμοποιήσουμε μια μεταβλητή `player` που να δείχνει τον *αριθμό* του παίκτη που παίζει σε κάθε γύρο.

Τι τιμές θα παίρνει η μεταβλητή αυτή; Πώς θ' αλλάζει σε κάθε γύρο;



.....

Μια απλή προσέγγιση είναι να χρησιμοποιήσουμε μια μεταβλητή `player`, η οποία σε κάθε γύρο θα παίρνει *εναλλάξ* την τιμή 1 ή 2, υποδεικνύοντας έτσι ποιος παίκτης έχει σειρά να παίζει.

Αρχικά, πριν ξεκινήσει η διαδικασία του παιχνιδιού, ορίζουμε ποιος παίκτης ξεκινάει πρώτος: αυτός θα είναι πάντα ο παίκτης με αριθμό 1.

Προσθέστε την κατάλληλη εντολή, ώστε η μεταβλητή `player` να παίρνει σαν αρχική τιμή το 1.

Τοποθετήσατε αυτή την εντολή πριν από τη **while** ή μέσα σε αυτή; Ποια πιστεύετε ότι είναι η διαφορά;



.....

- Τροποποιήστε* την εντολή του βήματος 3 που εμφανίζει το μήνυμα προτροπής, ώστε πλέον να απευθύνεται σε συγκεκριμένο παίκτη. Χρησιμοποιήστε κατάλληλα τη μεταβλητή `player`. Για παράδειγμα:

Παίκτη 1 πόσα σπέρτα θέλεις;



Εκτελέστε το πρόγραμμά σας. Εμφανίζει τον αριθμό του παίκτη που παίζει σε κάθε γύρο;



10. Στο τέλος κάθε κύκλου της επανάληψης, η τιμή της μεταβλητής `player` θα πρέπει να τροποποιείται, ώστε να υποδεικνύει τον επόμενο παίκτη που έχει σειρά να παίζει.

Προσθέστε μέσα στη `while` τις κατάλληλες εντολές που θα δίνουν στην `player` την τιμή 2 όταν αυτή έχει την τιμή 1 και αντιστρόφως. Χρησιμοποιήστε την `if-else` για το σκοπό αυτό.

Εκτελέστε το πρόγραμμά. Εναλλάσσει πλέον σε κάθε γύρο τον αριθμό του παίκτη που παίζει;



Εμφανίζεται σωστά ο αριθμός του παίκτη, δηλαδή την πρώτη φορά ο αριθμός 1 και στον επόμενο γύρο ο αριθμός 2;



Αν εμφανίζεται στον πρώτο γύρο ο αριθμός 2 και στον επόμενο γύρο ο αριθμός 1, που πιστεύετε ότι μπορεί να οφείλεται αυτό το σφάλμα;

Να απαντήσετε ακόμα κι αν η συμπεριφορά του δικού σας προγράμματος είναι ορθή.



Σε περίπτωση που χρησιμοποιήσατε την `elif` και όχι την `else` στο πρόγραμμά σας, αντικαταστήστε την με μια `else` διαγράφοντας τη συνθήκη και εκτελέστε το πρόγραμμα. Θα διαπιστώσετε ότι λειτουργεί και πάλι σωστά.

Είτε τη χρησιμοποιήσατε, είτε όχι, γιατί είναι περιττή η `elif` στο παράδειγμά μας;



11. Όταν η επανάληψη τελειώσει, η μεταβλητή `player` δείχνει ποιος παίκτης είχε σειρά να παίζει όταν τελείωσαν τα σπίρτα. Συνεπώς, μπορούμε να χρησιμοποιήσουμε τη μεταβλητή `player` για να διαπιστώσουμε ποιος παίκτης κέρδισε.

Προσθέστε την κατάλληλη εντολή μετά την επανάληψη, ώστε να εμφανίζει τον αριθμό του παίκτη που κέρδισε. Για παράδειγμα:

Παίκτη 2 κέρδισες!



Τι θα συμβεί αν τοποθετήσουμε την παραπάνω εντολή μέσα στην **while**;

.....

Εκτελέστε το πρόγραμμά σας. Εμφανίζει σωστά τον αριθμό του παίκτη που κέρδισε;

.....



12. Με τις εντολές που προσθέσαμε στο βήμα 10, η μεταβλητή `player` εναλλάσσεται σε κάθε γύρο μεταξύ των τιμών 1 και 2.

Τώρα θα φτιάξουμε το πρώτο μας υποπρόγραμμα, το οποίο θα συντελεί στην υλοποίηση αυτής της λειτουργίας και θα μας βοηθήσει να “συμμαζέψουμε” λίγο τον κώδικα του κυρίως προγράμματος.

Το υποπρόγραμμα θα επιστρέφει τον αριθμό του παίκτη που παίζει στον επόμενο γύρο.

Προσθέστε την παρακάτω εντολή που ορίζει τη συνάρτηση **next**, με παράμετρο τον αριθμό `p` ενός παίκτη.

```
def next(p):
```

Προσθέστε στη συνάρτηση τις κατάλληλες εντολές έτσι ώστε να ελέγχει τον αριθμό `p` του παίκτη και να επιστρέφει τον αριθμό του παίκτη που έχει σειρά να παίζει μετά από αυτόν.

Στο πρόγραμμά σας υπάρχουν ήδη οι εντολές που επιτελούν την παραπάνω λειτουργία, χρησιμοποιήστε ανάλογη προσέγγιση και στη συνάρτηση. Θυμηθείτε να υπάρχουν οι κατάλληλες εσοχές στις εντολές της συνάρτησης και να χρησιμοποιήσετε την εντολή **return**, προκειμένου να επιστρέψετε το αποτέλεσμα στο πρόγραμμα.

13. Μέσα στη **while**, αντικαταστήστε τις εντολές του προγράμματος που εναλλάσσουν τον αριθμό του παίκτη με την εντολή που ακολουθεί:

```
player = next(player)
```

Η τιμή της μεταβλητής `player` είναι η τιμή που επιστρέφει η συνάρτηση **next**, δηλαδή ο αριθμός του επόμενου παίκτη.

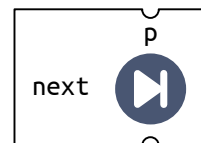
Εκτελέστε το πρόγραμμα. Εμφανίζει σωστά τον αριθμό του παίκτη που παίζει σε κάθε γύρο;

.....

Παρατηρείτε κάποια διαφορά στη λειτουργία του προγράμματος;

.....

14. Υπάρχουν κι άλλοι τρόποι να επιτευχθεί το ίδιο αποτέλεσμα, όπως φαίνεται στις εναλλακτικές υλοποιήσεις που ακολουθούν. Ωστόσο, ο τρόπος που λειτουργεί η συνάρτηση δεν έχει σημασία για εκείνους που την καλούν.



Η συνάρτηση **next** δέχεται σαν παράμετρο τον αριθμό `p` ενός παίκτη κι επιστρέφει τον αριθμό του παίκτη που παίζει μετά τον `p`.



```
def next(p):
    return (p % 2) + 1
```

```
def next(p):
    return 3 - p
```

Αντικαταστήστε τις εντολές της συνάρτησης `next` με όποια από τις παραπάνω εναλλακτικές υλοποιήσεις θέλετε.

Εκτελέστε ξανά το πρόγραμμα. Παρατηρείτε κάποια διαφορά στη λειτουργία του σε σχέση με πριν;

.....

Για να κατανοήσετε καλύτερα τη λειτουργία των παραπάνω εκδοχών της συνάρτησης `next` συμπληρώστε τον πίνακα που ακολουθεί για τις τιμές 1 και 2 της παραμέτρου `p`.

πιθανές τιμές της παραμέτρου <code>p</code>	τιμές της έκφρασης $(p \% 2) + 1$	τιμές της έκφρασης $3 - p$
1
2

Ποια από τις δύο υλοποιήσεις της συνάρτησης `next` επιλέξατε να χρησιμοποιήσετε στο πρόγραμμά σας; Τι σας ώθησε να κάνετε την επιλογή αυτή;

.....

.....

Εντοπίζετε κάποιο πλεονέκτημα στις εναλλακτικές υλοποιήσεις σε σχέση με την αρχική;

.....

.....

Ποια ή ποιες από τις υλοποιήσεις της `next`, συμπεριλαμβανομένης της αρχικής, θεωρείτε ότι μπορεί να γενικευτεί εύκολα, ώστε η `next` να είναι επαναχρησιμοποιήσιμη σε παιχνίδια όπου συμμετέχουν περισσότεροι από δύο παίκτες;

.....

Μη Λέμε κι Ό,τι Θέλουμε

Το πρόγραμμά μας επιτρέπει στον παίκτη που παίζει κάθε φορά ν' αφαιρέσει όσα σπύρτα θέλει. Θα επεκτείνουμε το πρόγραμμα, έτσι ώστε να ελέγχει τον αριθμό των σπύρτων που ζητά ν' αφαιρέσει ο παίκτης και να μην επιτρέπει κινήσεις που παραβιάζουν τους κανόνες του παιχνιδιού.

15. Για το σκοπό αυτό θα υλοποιήσουμε αρχικά μια συνάρτηση που θα δέχεται σαν παράμετρο το πλήθος `n` των σπύρτων που έχουν απο-

Η έκφραση `p%2` υπολογίζει το υπόλοιπο της ακέραιας διαίρεσης του `p` με το 2.



μείνει και θα επιστρέφει το μέγιστο πλήθος σπέρτων που επιτρέπεται ν' αφαιρεθούν.

Σύμφωνα με τους κανόνες του παιχνιδιού, ένας παίκτης επιτρέπεται να αφαιρέσει από ένα μέχρι και τρία σπέρτα κάθε φορά, αρκεί τα σπέρτα που έχουν απομείνει στο τραπέζι να είναι περισσότερα από τρία. Σε διαφορετική περίπτωση, θα πρέπει πάρει το πολύ όσα σπέρτα βρίσκονται στο τραπέζι.

Προσθέστε την παρακάτω εντολή στο πρόγραμμά σας για να δηλώσετε τη συνάρτηση `maxMatches`.

```
def maxMatches(m):
```

Χρησιμοποιήστε κατάλληλα την `if-else`, ώστε η συνάρτηση να επιστρέφει το μέγιστο αριθμό σπέρτων που μπορούν να αφαιρεθούν ανάλογα με τον αριθμό των σπέρτων `m` που βρίσκονται στο τραπέζι.

16. Προσθέστε τις παρακάτω εντολές αμέσως μετά την εντολή του βήματος 9 που διαβάζει τον αριθμό των σπέρτων που θέλει ν' αφαιρέσει ο παίκτης.

```
print("Παίκτη", player, "πόσα σπέρτα θέλεις;")
removed = int(input())
```

```
limit = maxMatches(matches)
if removed < 1 or removed > limit:
    print("Πάρε από 1 μέχρι και", limit, "σπέρτα")
```

Τι αποτέλεσμα πιστεύετε ότι θα έχει η προσθήκη των παραπάνω εντολών;

.....

Εκτελέστε το πρόγραμμα δίνοντας μια σειρά από λανθασμένες τιμές στον αριθμό των σπέρτων που αφαιρούνται. Εμφανίζεται το μήνυμα για τον μη-έγκυρο αριθμό σπέρτων;

.....

Σε περίπτωση που ο παίκτης δώσει μη-έγκυρη τιμή το πρόγραμμα την ξαναζητά ή συνεχίζει κανονικά τη λειτουργία του;

.....

Γιατί πιστεύετε ότι συμβαίνει αυτό;

.....

Τι θα προτείνατε για να διορθωθεί η λειτουργία του προγράμματος;

.....



17. Προσθέστε εκ νέου και μέσα στην **if** τις εντολές του βήματος 9 που προτρέπουν τον παίκτη να δώσει αριθμό σπύρων και ζητούν την απάντησή του, ώστε όταν δίνει μη-έγκυρη απάντηση το πρόγραμμα να ξαναζητά τα σπύρα που θέλει ο παίκτης να αφαιρέσει.

Εκτελέστε το πρόγραμμα δίνοντας διαδοχικά μια μη-έγκυρη τιμή και στη συνέχεια μια έγκυρη. Λειτουργεί σωστά;

.....

Εκτελέστε ξανά το πρόγραμμα δίνοντας διαδοχικά δύο μη-έγκυρες τιμές. Τι παρατηρείτε;

.....

.....

18. Παρόλο που στην πρώτη μη-έγκυρη τιμή το πρόγραμμα εμφανίζει μήνυμα σφάλματος στον παίκτη και του ζητάει να δώσει ξανά την τιμή των σπύρων, στην επόμενη του επιτρέπει να συνεχίσει με τον αριθμό σπύρων που διάλεξε. Θα χρησιμοποιήσουμε τη **while**, προκειμένου το πρόγραμμα να ξαναζητά τον αριθμό των σπύρων που θέλει ο παίκτης όσο η απάντηση του παραμένει εκτός ορίων.

Τροποποιήστε την **if** και στη θέση της βάλτε την εντολή **while** ώστε να κάνει τον έλεγχο για την εγκυρότητα του αριθμού των σπύρων, όπως παρακάτω:

```
while removed < 1 or removed > limit:
```

Εκτελέστε το πρόγραμμα και δώστε μια σειρά από λανθασμένες και έγκυρες τιμές. Λειτουργεί σωστά;

.....

19. Με τις εντολές που προσθέσαμε στα βήματα 16, 17 και 18, το πρόγραμμά μας δε διαβάζει απλά από τον παίκτη το πλήθος των σπύρων που επιθυμεί ν' αφαιρέσει, αλλά επίσης ελέγχει την τιμή που δίνει ο παίκτης και εξασφαλίζει ότι αυτή δεν παραβιάζει τους κανόνες του παιχνιδιού. Τώρα θα κατασκευάσουμε ένα υποπρόγραμμα το οποίο υλοποιεί αυτή ακριβώς τη λειτουργία.

Προσθέστε την εντολή που ακολουθεί για να ορίσετε τη συνάρτηση `readMatches`, η οποία δέχεται ως παραμέτρους τον αριθμό `p` του παίκτη που έχει σειρά να παίξει και το πλήθος `m` των σπύρων που απομένουν.

```
def readMatches(p,m):
```

Προσθέστε στη συνάρτηση τις κατάλληλες εντολές έτσι ώστε να εμφανίζει μήνυμα προτροπής προς τον παίκτη με αριθμό `p`, να ζητά από τον παίκτη τον αριθμό των σπύρων που επιθυμεί να αφαιρέσει και, αφού εξασφαλίσει με τους απαραίτητους ελέγχους ότι ο αριθμός των σπύρων είναι έγκυρος με βάση τα σπύρα `m` που απομένουν,



να επιστρέφει τον αριθμό αυτό.

Στην ουσία ο κώδικας που χρειάζεστε υπάρχει έτοιμος στο κύριο πρόγραμμα. Ακολουθήστε ανάλογη προσέγγιση και στη συνάρτηση, χρησιμοποιώντας τις “τοπικές” μεταβλητές *p* και *m*. Σε περίπτωση που δυσκολευτείτε μπορείτε να ανατρέξετε στα βήματα 16, 17 και 18.

Θυμηθείτε να υπάρχουν οι κατάλληλες εσοχές στις εντολές της συνάρτησης και να χρησιμοποιήσετε την εντολή **return**, προκειμένου να επιστρέψετε το αποτέλεσμα στο πρόγραμμα.

20. Με τις εντολές που προσθέσαμε στα βήματα 16, 17 και 18, το πρόγραμμά μας εξασφαλίζει ότι η τιμή που θα πάρει η `removed` από τον παίκτη θα είναι συμβατή με τους κανόνες του παιχνιδιού. Τώρα διαθέτουμε για τον σκοπό αυτό την συνάρτηση `readMatches`.

Αντικαταστήστε όποιες εντολές του προγράμματος κρίνετε απαραίτητο με την κλήση της συνάρτησης, όπως φαίνεται παρακάτω.

```
removed = readMatches(player, matches)
```

Εκτελέστε το πρόγραμμα. Λειτουργεί σωστά; Παρατηρείτε κάποια διαφορά στη λειτουργία του;

.....

Παρατηρήστε την έκταση και την πολυπλοκότητα που έχει αυτή την στιγμή το κύριο πρόγραμμα. Πιστεύετε ότι η χρήση συναρτήσεων έχει βελτιώσει την αναγνωσιμότητα του προγράμματος;

Ένα πρόγραμμα είναι αναγνώσιμο όταν το διαβάζει κανείς και καταλαβαίνει πως λειτουργεί χωρίς να καταβάλει μεγάλη προσπάθεια.

.....

Χαζό Μηχάνημα

Θα τροποποιήσουμε τώρα το πρόγραμμα, ώστε να αναλαμβάνει το ρόλο ενός από τους δύο παίκτες. Δεν είναι ανάγκη να καταλήξουμε από την αρχή σε κάποια ευφυή στρατηγική για το πρόγραμμά μας.

21. Μπορείτε να προτείνετε έναν απλό – “χαζό” τρόπο για να επιλέγει το πρόγραμμα το πλήθος των σπύρτων που θα αφαιρεί όταν έχει σειρά να παίξει;

.....

Θα κατασκευάσουμε μια απλή συνάρτηση η οποία θα επιλέγει τυχαία το πλήθος των σπύρτων που θα αφαιρεί το πρόγραμμα, φροντίζοντας αυτό το πλήθος να μην υπερβαίνει το μέγιστο πλήθος σπύρτων που επιτρέπεται να αφαιρεθούν.

Πώς θα υπολογίζει η συνάρτηση τον μέγιστο πλήθος σπύρτων που μπορούν να αφαιρεθούν;

.....



Προσθέστε στο πρόγραμμα την παρακάτω εντολή που ορίζει τη συνάρτηση `randomMatches`, με παράμετρο το πλήθος `m` των σπίρτων που απομένουν.

```
def randomMatches(m):
```

Προσθέστε στη συνάρτηση την κατάλληλη εντολή ή εντολές, προκειμένου να επιστρέφει στο πρόγραμμα έναν τυχαίο ακέραιο αριθμό από το 1 μέχρι και το μέγιστο επιτρεπτό αριθμό σπίρτων που μπορεί να αφαιρέσει ο παίκτης.

22. Ας εστιάσουμε τώρα στο κύριο πρόγραμμα. Θα χρησιμοποιήσουμε μια μεταβλητή `computer`, η οποία στην αρχή του παιχνιδιού θα παίρνει τυχαία την τιμή 1 ή 2. Η τιμή της `computer` υποδεικνύει τον αριθμό του παίκτη που παίζει αυτοματοποιημένα, δηλαδή το ρόλο του τον έχει αναλάβει το ίδιο το πρόγραμμα.

Προσθέστε στο πρόγραμμα την κατάλληλη εντολή που θα επιλέγει τυχαία είτε την τιμή 1 είτε την τιμή 2 και θα την αποθηκεύει στη μεταβλητή `computer`.

23. Θυμηθείτε ότι στο τέλος κάθε γύρου, η μεταβλητή `player` εναλλάσσεται μεταξύ των τιμών 1 και 2, υποδεικνύοντας ποιος παίκτης έχει σειρά να παίξει στον επόμενο γύρο. Τώρα λοιπόν θα πρέπει να ελέγχουμε αν ο επόμενος παίκτης είναι ο άνθρωπος ή το πρόγραμμά μας, ώστε να καλέσουμε σε κάθε περίπτωση την ανάλογη συνάρτηση που θα μας επιστρέψει το πλήθος των σπίρτων που θα αφαιρεθούν.

Χρησιμοποιήστε την **if-else**, ώστε η μεταβλητή `removed` να παίρνει την τιμή που επιστρέφει η συνάρτηση `randomMatches` όταν ο παίκτης που παίζει είναι ο υπολογιστής, και την τιμή που επιστρέφει η συνάρτηση `readMatches` σε διαφορετική περίπτωση. Για να ελέγξετε με την **if** ποιος έχει σειρά να παίξει, συγκρίνετε τις τιμές των μεταβλητών `player` και `computer`.

24. Προσθέστε την κατάλληλη εντολή, ώστε όταν παίζει ο υπολογιστής να εμφανίζεται το πλήθος των σπίρτων που αφαιρεί. Για παράδειγμα:

Ο υπολογιστής παίρνει 2 σπίρτα.



25. Χρειάζεται να φροντίσουμε μια ακόμα σημαντική λεπτομέρεια. Μέχρι στιγμής, στο τέλος του παιχνιδιού ανακοινώνεται ο αριθμός του παίκτη που έχασε. Τώρα που μόνο ο ένας παίκτης είναι άνθρωπος, είναι προτιμότερο να εμφανίζουμε διαφοροποιημένα μηνύματα.

Προσθέστε τις κατάλληλες εντολές στο πρόγραμμα, ώστε ο νικητής ν' ανακοινώνεται ως εξής: αν κερδίσει ο παίκτης, να ανακοινώνεται ο αριθμός του, όπως ακριβώς συνέβαινε μέχρι στιγμής. Αν κερδίσει το πρόγραμμα, να εμφανίζεται το μήνυμα:

Ο υπολογιστής κέρδισε.



Εκτελέστε το πρόγραμμά σας δύο φορές. Φροντίστε τη μία φορά να κερδίσετε εσείς ως παίκτης, ενώ την άλλη ο υπολογιστής. Εμφανίζει το κατάλληλο μήνυμα σε κάθε περίπτωση;



.....

Άνθρωπος Εναντίον Μηχανής

Όταν το πρόγραμμά μου παίζει τυχαία δεν έχει και μεγάλο ενδιαφέρον. Θα το κάνουμε λίγο πιο “έξυπνο”.

- 26. Θα πρέπει πρώτα εμείς να σχεδιάσουμε έναν καλύτερο τρόπο παιχνιδιού και μετά να τον περιγράψουμε με τις κατάλληλες εντολές. Είναι ευκολότερο ν’ αρχίσουμε μελετώντας ποιες είναι οι ενδεδειγμένες κινήσεις όταν απομένουν 2, 3 ή 4 σπέρτα: ο παίκτης που έχει σειρά να παίξει μπορεί να αφαιρέσει αντίστοιχα 1, 2 ή 3 σπέρτα και να κερδίσει άμεσα. Αντίθετα, όταν απομένουν 5 σπέρτα η κατάσταση είναι δύσκολη: όσα σπέρτα και να αφαιρέσει ο παίκτης που έχει σειρά να παίξει, η έκβαση είναι στα χέρια του αντιπάλου.

Ανάλογες δυσάρεστες καταστάσεις, που θα ονομάσουμε «ανεπιθύμητες νησίδες», αντιμετωπίζει ο παίκτης που έχει σειρά να παίξει όταν απομένουν 9, 13, 17, κ.ο.κ σπέρτα. Όσα σπέρτα κι αν αφαιρέσει, ο αντίπαλος μπορεί να τον στείλει στην επόμενη νησίδα.

Προκειμένου να κατανοήσετε καλύτερα τη στρατηγική που πρέπει να ακολουθήσει το πρόγραμμα (ή ο οποιοσδήποτε παίκτης) για να παίζει “έξυπνότερα” συμπληρώστε τον πίνακα που ακολουθεί.

σπέρτα	κίνηση	
	n	n%4 πόσα αφαιρούνται
6
7
8
9
10
11
12
13
14
15
16

Στην πρώτη στήλη δίνεται ο αριθμός των σπέρτων. Στη δεύτερη στήλη θα συμπληρώσετε το υπόλοιπο της διαίρεσης του αριθμού των σπέρτων με το 4, ενώ στην τρίτη στήλη τον αριθμό των σπέρτων που πρέπει να αφαιρεθούν, προκειμένου να οδηγηθεί ο αντίπαλος σε μια “ανεπιθύμητη” νησίδα, σύμφωνα με την περιγραφή που προηγήθηκε.

Παρατηρείτε κάποια σχέση ανάμεσα στον αριθμό της δεύτερης και της τρίτης στήλης;

.....



Στις περιπτώσεις που ο αριθμός των σπάρτων που βρίσκονται στο τραπέζι είναι τέτοιος που δεν μπορεί να οδηγήσει τον παίκτη σε νίκη (για παράδειγμα το 13), πώς θα μπορούσε να επιλέγει τα σπάρτα που θα αφαιρέσει;



-
27. Με βάση τα παραπάνω, μπορούμε να υλοποιήσουμε ένα υποπρόγραμμα που δέχεται σαν παράμετρο το πλήθος n των σπάρτων που απομένουν και επιστρέφει το πλήθος των σπάρτων που πρέπει να αφαιρεθούν ώστε ο αντίπαλος να οδηγηθεί σε μια ανεπιθύμητη νησίδα. Στην περίπτωση που ο παίκτης βρίσκεται ήδη σε ανεπιθύμητη νησίδα, επιστρέφεται ένας τυχαίος αριθμός σπάρτων, καλώντας την `randomMatches`.

```
def computeMatches(m):
    mod = m % 4
    if mod == 0:
        return 3
    elif mod == 1:
        return randomMatches(m)
    elif mod == 2:
        return 1
    else:
        return 2
```

28. Στο κύριο πρόγραμμα, αντικαταστήστε την κλήση της συνάρτησης `randomMatches` που επιλέγει ένα τυχαίο πλήθος σπάρτων, με μια κλήση στην `computeMatches`, ώστε πλέον το πρόγραμμα να επιλέγει το πλήθος των σπάρτων που αφαιρείται ακολουθώντας συγκεκριμένη στρατηγική.

Εκτελέστε το πρόγραμμά σας. Λειτουργεί σωστά;



Δραστηριότητες για Εξάσκηση

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

pythonies.mysch.gr/complete