

1. Γράψτε ένα κώδικα σε Python ο οποίος να παίρνει την διάσταση ενός τετραγώνου και θα φτιάχνει μέσα από λίστες τον αντίστοιχο πίνακα. Στην συνέχεια θα βρίσκει το πλήθος των θέσεων και θα γεμίζει στην τύχη τις μισές με μονάδες (στρογγυλοποίηση προς τα πάνω). Σκοπός είναι να μετρήσετε πόσες τετράδες από μονάδες υπάρχουν οριζόντια, κάθετα, και διαγώνια. Το πρόγραμμα επαναλαμβάνεται 100 φορές (για την ίδια διάσταση) και επιστρέφει τον μέσο όρο των τετράδων.
2. Γράψτε ένα κώδικα σε Python ο οποίος ελέγχει αν ο n όρος της ακολουθίας Fibonacci είναι πρώτος ή όχι. Για να απαντήσετε αν ένας αριθμός p είναι πρώτος θα πρέπει για 20 τυχαίες επιλογές του a να ισχύει ότι $a^p \equiv a \pmod{p}$. Ο κώδικάς σας παίρνει τον όρο της ακολουθίας Fibonacci από το χρήστη.
Για παράδειγμα:
5ος όρος είναι ο 5 και είναι πρώτος.
11ος όρος είναι ο 89 και είναι πρώτος.
15ος όρος είναι ο 610 και δεν είναι πρώτος
3. Χρησιμοποιήστε το API του ΟΠΑΠ (<https://www.opap.gr/web-services>) από την Python για να εμφανίσετε τα στατιστικά των αριθμών για πρώτη κλήρωση της κάθε ημέρας για το ΚΙΝΟ τον τρέχον μήνα (Οι αριθμοί που κληρώθηκαν). (Δείτε τις βιβλιοθήκες `date`, `datetime`)
4. Γράψτε ένα κώδικα σε Python ο οποίος να παίρνει σαν είσοδο ένα αρχείο ASCII κειμένου και να το κόβει σε συνεχόμενες τριάδες λέξεων (όλες τις δυνατές). Στην συνέχεια, διαλέγει τυχαία μια τριάδα και προσπαθεί να συντάξει ένα τυχαίο κείμενο από αυτό, χρησιμοποιώντας τις δυο τελευταίες λέξεις και επιλέγοντας μια τριάδα που να ξεκινάει από αυτές τις δυο. Το πρόγραμμα ολοκληρώνεται, όταν γράψει 200 λέξεις ή δεν μπορεί να επιλεγεί άλλη τριάδα λέξεων.
Για το κείμενο "A tale of two cities" που είναι στην περιοχή έγγραφα, οι πρώτες 33 λέξεις για seed 5 είναι οι:
her speculating thousands of times a come to life was to be sure of it eminence in those two of the blaze the stone faces on the road plied his dusty labour and
5. Γράψτε ένα κώδικα σε Python ο οποίος να παίρνει τις διαστάσεις ενός ορθογωνίου και θα φτιάχνει μέσα από λίστες τον αντίστοιχο πίνακα. Στην συνέχεια θα βρίσκει το πλήθος των θέσεων και γεμίζει στην τύχη τις μισές με 5 και τις μισές με 0 (στρογγυλοποίηση προς τα πάνω). Σκοπός είναι να μετρήσετε πόσες φορές εμφανίζεται το 5 οριζόντια, κάθετα, και διαγώνια. Το πρόγραμμα επαναλαμβάνεται 100 φορές (για τις ίδιες διαστάσεις) και επιστρέφει τον μέσο όρο των τριάδων 505.
6. Γράψτε ένα κώδικα σε Python ο οποίος συνδέεται στο twitter (χρησιμοποιήστε το `tweepy`) και επιλέξτε τα 10 τελευταία tweets του χρήστη που θα σας δηλώσει ο χρήστης. Εμφανίστε τις 5 μεγαλύτερες λέξεις και τις 5 μικρότερες λέξεις του.
7. Χρησιμοποιήστε το API του ΟΠΑΠ (<https://www.opap.gr/web-services>) από την Python για να βρείτε τον αριθμό που εμφανίζεται συχνότερα στο ΚΙΝΟ κάθε μέρα του τρέχοντα μήνα.
8. Γράψτε ένα κώδικα σε Python ο οποίος να παίρνει σαν είσοδο ένα λεξικό από αρχείο το οποίο περιέχει τα κρυπτονομίσματα (όνομα) που έχει ένας χρήστης και πόσα από αυτά. Χρησιμοποιήστε το API <https://min-api.cryptocompare.com> για να βρείτε σε τι ποσό σε ευρώ αντιστοιχούν.
{`"BTC":1.345,"ETH":8.2,"LTC":1.234`}
9. Γράψτε ένα κώδικα σε Python ο οποίος να παίρνει σαν είσοδο ένα αρχείο ASCII κειμένου και μετατρέπει τον κάθε χαρακτήρα στον αντίστοιχο αριθμό ASCII και κρατάει τους μονούς. Εμφανίστε τα στατιστικά εμφάνισης του κάθε γράμματος με "μπάρες" χρησιμοποιώντας το χαρακτήρα *, όπου κάθε * αντιστοιχεί σε 1%. Η στρογγυλοποίηση θα γίνει προς τα πάνω.

A: *****
C: ***
E: *****
G: **

10. Γράψτε ένα κώδικα σε Python ο οποίος να παίρνει σαν είσοδο ένα λεξικό από αρχείο. Το λεξικό μπορεί να περιέχει και άλλα λεξικά, λίστες, κτλ. Για κάθε εμφωλευμένο λεξικό, λίστα, το βάθος αυξάνεται κατά 1. Εμφανίστε στο χρήστη το μεγαλύτερο βάθος του λεξικού.

```
{ } Βάθος: 0  
{"x":5} Βάθος: 1  
{"x":5,"y":[1,2,3]} Βάθος: 2  
{"x":5,"y":{"a":2,"b":67}} Βάθος: 2  
{"x":5,"y":[ [ 1 , 2 ] , [ 3 , 4 ] ]} Βάθος: 3
```

11. Γράψτε ένα κώδικα σε Python ο οποίος να παίρνει σαν είσοδο ένα λεξικό από αρχείο. Το λεξικό μπορεί να περιέχει και άλλα λεξικά, λίστες, κτλ. Εμφανίστε στο χρήστη τα κλειδιά που εμφανίζονται τις περισσότερες φορές, ελέγχοντας και τα εμφωλευμένα λεξικά.

```
{"results":[{"x":5}, {"x":5,"y":[1,2,3]}, {"x":5,"y":{"a":2,"b":67}}]}
```

Δημοφιλέστερο κλειδί το x

12. Γράψτε ένα κώδικα σε Python ο οποίος να παίρνει σαν είσοδο ένα αρχείο ASCII κειμένου και μετατρέπει τον κάθε χαρακτήρα του στον “κατοπτρικό” του χαρακτήρα ASCII. Κατοπτρικοί χαρακτήρες είναι αυτοί των οποίων το άθροισμα είναι 128. Εμφανίστε το κατοπτρικό κείμενο στο χρήστη με ανάποδη σειρά χαρακτήρων.

13. Γράψτε ένα κώδικα σε Python ο οποίος να παίρνει σαν είσοδο ένα αρχείο ASCII κειμένου, το χωρίζει σε λέξεις και εμφανίζει τα ζευγάρια λέξεων όπου το συνολικό **τους μήκος χαρακτήρων είναι ακριβώς 20**. Κάθε ζευγάρι φεύγει από το σύνολο και το πρόγραμμα τελειώνει όταν εξαντληθούν τα ζευγάρια. Για το κείμενο "Tale of two cities" που υπάρχει στα έγγραφα, δείγμα λέξεων που μένουν ανά μήκος: '12': [], '10': [], '15': [], '13': [], '11': [], '14': [], '16': [], '18': [], '19': [], '17': [], '20': ['carriagedoortenderly', 'thingnothingstartles']