

1. Γράψτε ένα πρόγραμμα το οποίο κατασκευάζει μια λίστα με τους διαιρέτες ενός φυσικού αριθμού.
2. Γράψτε ένα πρόγραμμα το οποίο ελέγχει αν δυο λίστες έχουν τουλάχιστον ένα κοινό στοιχείο.
3. Γράψτε ένα πρόγραμμα το οποίο αποθηκεύει σε μια λίστα τις λέξεις που εισάγει ο χρήστης και στη συνέχεια εμφανίζει μόνο τις λέξεις με περισσότερα από 3 γράμματα. Η είσοδος των λέξεων θα πρέπει να τερματίζει όταν ο χρήστης δώσει τη λέξη 'end'.
4. Γράψτε ένα πρόγραμμα το οποίο αποθηκεύει στη λίστα L τις τιμές της συνάρτησης f στα σημεία ενός ομοιόμορφου διαμερισμού του διαστήματος $[0, 1]$ με n εσωτερικά σημεία.
5. Γράψτε μια συνάρτηση η οποία με είσοδο δύο λίστες τα στοιχεία των οποίων είναι σε αύξουσα σειρά επιστρέφει μια λίστα με όλα τα στοιχεία και από τις δύο λίστες, πάλι σε αύξουσα σειρά.
6. Γράψτε μια συνάρτηση η οποία υπολογίζει τη συχνότητα εμφάνισης όλων των στοιχείων μιας λίστας.
7. Γράψτε ένα πρόγραμμα το οποίο αφαιρεί από μια λίστα τις πολλαπλές εμφανίσεις των στοιχείων της. Για παράδειγμα, η λίστα $[10, 12, 13, 13, 10, 9, 8, 7, 8, 10, 10, 8, 13]$ θα πρέπει να γίνει $[10, 12, 13, 9, 8, 7]$
8. Μπορούμε να βρούμε το μικρότερο στοιχείο μιας λίστας χρησιμοποιώντας τη συνάρτηση `min`. Πως μπορούμε να βρούμε το δεύτερο μικρότερο στοιχείο μιας λίστας; Η άσκηση είναι δύσκολη – αν κάποιος ενδιαφέρεται για μια κομψή λύση.
9. Ο απλούστερος (και ο πιο αργός) τρόπος ταξινόμησης είναι ο λεγόμενος αλγόριθμος της φουσαλίδας (bubble sort), μια υλοποίηση του οποίου θα μπορούσε να είναι η ακόλουθη:

```
def bubbleSort(L):  
    for n in range(len(L)-1,0,-1):  
        for i in range(n):  
            if L[i] > L[i+1]:  
                tmp = L[i]; L[i] = L[i+1]; L[i+1] = tmp  
  
L = [54, 26, 93, 17, 77, 31, 44, 55, 20]  
bubbleSort(L)  
print(L)
```

Προσπαθήστε να καταλάβετε την ιδέα του αλγορίθμου τυπώνοντας τα περιεχόμενα της λίστας L μετά από κάθε ανακύκλωση. Μετρήστε επίσης τον αριθμό των συγκρίσεων που κάνει η μέθοδος. Μπορείτε να ανακαλύψετε κάποια σχέση με το μήκος της λίστας;