

Άσκηση 1 [6μ]

Ποιο θα είναι το αποτέλεσμα του παρακάτω προγράμματος και γιατί

```
def thing_one(x):
    y = 0
    if x == 1:
        y = x
        x = 2
    if x == 2:
        y = -x
        x = 3
    elif x == 3:
        y = 2*x
        x = 2*y
    else:
        y = x
    print('x = ' + str(x))
    print('y = ' + str(y))

def thing_two(l, v):
    for i in range(len(l)):
        l[i] += v
        v = v-1

def main():
    thing_one(7)
    thing_one(3)
    thing_one(1)
    l = [20, 40, 30]
    v = 6
    thing_two(l, v)
    print('l = ' + str(l))
    print('v = ' + str(v))

main()
```

Απάντηση:

Όταν τρέξει το πρόγραμμα θα δώσει σαν output:

```
7          # thing_one(7)
x = 12     # thing_one(3)
y = 6
x = 3      # thing_one(1)
y = -2
l = [26, 45, 34] # thing_two(l,v)
v = 6
```

Παρατηρήσεις:

Οι κλήσεις στις συναρτήσεις thing_one(7) και thing_one(3) παράγουν printout που είναι αναμενόμενο. Ωστόσο η κλήση thing_one(1) θέλει προσοχή γιατί ενώ υπάρχει η συνθήκη για x==1, το x αλλάζει και γίνεται 2. Επομένως θα εξεταστεί η συνθήκη για x==2 και εκεί το y γίνεται -2 και το x έχει τιμή 3.

Η περίπτωση του thing_two(l,v) είναι η πιο περίεργη. Η v είναι απλή μεταβλητή και στο όρισμα της συνάρτησης περνά μόνο η τιμή της. Αντίθετα η l είναι η μεταβλητή της list και στο όρισμα με την l περνούμε τη διεύθυνση στη μνήμη του υπολογιστή (ουσιαστικά περνούμε έναν pointer στη μνήμη). Μέσα στη συνάρτηση thing_two(), η μεταβλητή v είναι απλά μια τοπική μεταβλητή με αρχική τιμή αυτή που περνά σαν όρισμα. Ότι αλλαγή γίνεται στη τιμή της είναι τοπική στη συνάρτηση και όταν επιστρέφει στο κύριο πρόγραμμα η τιμή αυτή χάνεται και η μόνη που μένει είναι η v=6 που ήταν η αρχική τιμή. Για την list l ωστόσο τα πράγματα είναι διαφορετικά, γιατί τόσο στο main όσο και στο thing_two εξετάζεται η θέση της μνήμης και αυτή αλλάζει με τις τιμές 26, 45, 34. Επομένως η l και στο main θα έχει τις ίδιες τιμές.

Άσκηση 2 [7μ]

Να γράψετε ένα πρόγραμμα το οποίο περιέχει μια συνάρτηση με όνομα `sort_list(mylist)`, το οποίο χρησιμοποιεί μία λίστα αριθμών, `mylist`, για να την ταξινομήσει κατά φθίνουσα σειρά. (Προσοχή: θα πρέπει να γράψετε τη δική σας συνάρτηση ταξινόμησης). Η συνάρτηση επιστρέφει ως αποτέλεσμα την ταξινομημένη λίστα. Για έλεγχο των αποτελεσμάτων, θα πρέπει να τυπώνετε στην οθόνη του υπολογιστή την αρχική λίστα και την ταξινομημένη σε δύο στήλες, τη μία δίπλα στην άλλη. Τα στοιχεία της λίστας θα πρέπει να είναι σε ένα file με όνομα `mylist.dat` το οποίο θα πρέπει να διαβάζει το πρόγραμμά σας.

Απ: δείτε την επόμενη σελίδα.

Άσκηση 3 [2μ]

(α) Να εξηγήσετε τη διαφορά μεταξύ των τελεστών `//` και `/` και να δώσετε ένα παράδειγμα για το που μπορεί να χρειαστεί ο τελεστής `//`.

Απ: `/` τελεστής διαίρεσης με αποτέλεσμα πραγματικό ενώ `//` είναι ο τελεστής δίνει αποτέλεσμα ακέραιο της διαίρεσης που οδηγεί σε στρογγυλοποίηση του αποτελέσματος στον προηγούμενο ακέραιο. Χρησιμοποιείται σε περιπτώσεις που η διαίρεση ακεραίων απαιτεί να δώσει αποτέλεσμα ακέραιο όπως για παράδειγμα τη θέση σε μια λίστα.

(β) Στις αεροπορικές εταιρείες υπάρχει περιορισμός ως προς τις διαστάσεις των αποσκευών που μπορεί να πάρει κάποιος στην καμπίνα του αεροπλάνου. Υποθέστε ότι οι μεταβλητές αυτές είναι *length*, *width* και *depth* και οι προδιαγραφές είναι 60cmx35cmx21cm αντίστοιχα. Να γράψετε μια λογική έκφραση που να συνδυάζει τις τρεις μεταβλητές σε αποτέλεσμα που είναι *True* αν η αποσκευή πληροί τις προδιαγραφές και *False* αν δεν τις πληροί.

Απ: `length <= 60 and width <= 35 and depth <= 21`

Απάντηση του ερωτήματος 2:

```
#!/usr/bin/python3
import numpy as np
def short_list(mylist):
    copylist = []
    for i in range(len(mylist)):
        copylist.append(mylist[i]) # copy the original list in a new list
    #sort
    for i in range(len(copylist)-1):
        for j in range(i,len(copylist)):
            if copylist[j]>copylist[i] :
                tmp = copylist[i]
                copylist[i] = copylist[j]
                copylist[j] = tmp
    return copylist

filename="mylist.dat"
inpfiler=open(filename,"r")
alist = []
for line in inpfiler:
    numb = int(line.strip())
    alist.append(numb)
# Sort the list
sortlist = short_list(alist)
for i in range(len(alist)):
    print("%5d %5d"%(alist[i],sortlist[i]))
```