

ΕΠΑΝΑΛΗΨΗ ΧΩΡΙΣ ΠΡΟΚΑΘΟΡΙΣΜΕΝΑ ΒΗΜΑΤΑ ΜΕΧΡΙ ΝΑ ΜΗΝ ΞΕΠΕΡΑΣΟΥΜΕ  
ΚΑΠΟΙΟ ΟΡΙΟ , ΛΙΣΤΕΣ, ΤΑΞΙΝΟΜΗΣΗ, ΔΥΑΔΙΚΗ ΑΝΑΖΗΤΗΣΗ

## Άσκηση 11

Φορτώνουμε ένα φορτηγάκι με δέματα διαφορετικού βάρους το καθένα. Το βάρος που μπορούμε να βάλουμε πάνω σε αυτό το φορτηγάκι δεν θα πρέπει να ξεπερνά τα 1500 κιλά. Το κάθε δέμα έχει ένα μοναδικό κωδικό που αποτελείται από δύο γράμματα και τέσσερις αριθμούς π.χ. ΠΤ3452. Να γράψετε πρόγραμμα σε ρυθμό το οποίο:

**α)** Να διαβάζει το Βάρος και τον Κωδικό του κάθε δέματος που θα φορτωθεί στο φορτηγάκι μέχρι να μην ξεπεράσουμε το όριο των 1500 κιλών.

**β)** Να τα τοποθετεί στις λίστες BA για το βάρος και CO για τον κωδικό.

**γ)** Να υπολογίζει και να εμφανίζει πόσο είναι το πλήθος των πακέτων που φορτώθηκαν

**δ)** Να υπολογίζει και να εμφανίζει το συνολικό βάρος που φορτώθηκε και το MO του βάρους των πακέτων (Θεωρήστε ότι φορτώθηκε τουλάχιστον ένα)

**ε)** Με τη βοήθεια της συνάρτησης bubbleSort να κάνει αύξουσα ταξινόμηση ως προς το βάρος και να παρουσιάζει όλα τα δέματα με τη σειρά από το ελαφρύτερο προς το βαρύτερο εμφανίζοντας σε κάθε γραμμή το κείμενο π.χ. “Το δέμα με κωδικό ΠΤ3452 έχει βάρος 34 κιλά”.

**στ)** Να διαβάζει το βάρος δέματος από το πληκτρολόγιο και χρησιμοποιώντας τη συνάρτηση binarySearch (πρέπει να τη γράψετε), η οποία επιστρέφει τη θέση ενός στοιχείου, να εμφανίζει τον κωδικό του δέματος που αντιστοιχεί στο βάρος ή το μήνυμα “δεν φορτώθηκε” αν αυτό δεν υπάρχει στην λίστα BA (θεωρείστε ότι κάθε βάρος που φορτώθηκε είναι μοναδικό)

Άσκηση 11

Φορτώνουμε ένα φορτηγάκι με δέματα διαφορετικού βάρους το καθένα. Το βάρος που μπορούμε να βάλουμε πάνω σε αυτό το φορτηγάκι δεν θα πρέπει να ξεπερνά τα 1500 κιλά. Το κάθε δέμα έχει ένα μοναδικό κωδικό που αποτελείται από δύο γράμματα και τέσσερις αριθμούς π.χ. ΠΤ3452. Να γράψετε πρόγραμμα σε ρυθμό το οποίο: **α)** Να διαβάζει το Βάρος και τον Κωδικό του κάθε δέματος που θα φορτωθεί στο φορτηγάκι μέχρι να μην ξεπεράσουμε το όριο των 1500 κιλών. **β)** Να τα τοποθετεί στις λίστες BA για το βάρος και CO για τον κωδικό. **γ)** Να υπολογίζει και να εμφανίζει πόσο είναι το πλήθος των πακέτων που φορτώθηκαν **δ)** Να υπολογίζει και να εμφανίζει το συνολικό βάρος που φορτώθηκε και το MO του βάρους των πακέτων (Θεωρήστε ότι φορτώθηκε τουλάχιστον ένα) **ε)** Με τη βοήθεια της συνάρτησης bubbleSort να κάνει αύξουσα ταξινόμηση ως προς το βάρος και να παρουσιάζει όλα τα δέματα με τη σειρά από το ελαφρύτερο προς το βαρύτερο εμφανίζοντας σε κάθε γραμμή το κείμενο π.χ. “Το δέμα με κωδικό ΠΤ3452 έχει βάρος 34 κιλά”. **στ)** Να διαβάζει το βάρος δέματος από το πληκτρολόγιο και χρησιμοποιώντας τη συνάρτηση binarySearch (πρέπει να τη γράψετε), η οποία επιστρέφει τη θέση ενός στοιχείου, να εμφανίζει τον κωδικό του δέματος που αντιστοιχεί στο βάρος ή το μήνυμα “δεν φορτώθηκε” αν αυτό δεν υπάρχει στην λίστα BA (θεωρείστε ότι κάθε βάρος που φορτώθηκε είναι μοναδικό)

# **-\*- coding: cp1253 -\*-**

# Όταν έχω προβλήματα που αναφέρουν ότι διαβάζουμε πράγματα μέχρι να ΜΗΝ ξεπεράσουμε κάποιο όριο, πρώτα βάζουμε μία μεταβλητή όριο ίση με αυτό το όριο

**orio=1500**

# Στη συνέχεια διαβάζουμε έξω από την επανάληψη , η οποία είναι χωρίς προκαθορισμένα βήματα (while) , το κριτήριο από το οποίο εξαρτάται η επανάληψη, δηλαδή το βάρος και μία λίγο πριν τελειώσει η επανάληψη

**b=float(input("Δώσε το βάρος"))**

# Όσο το βάρος που παίρνουμε από το πληκτρολόγιο είναι μικρότερο ή ίσο του υπόλοιπου ορίου που απέμεινε

**while b<=orio:**

# Διάβασε και τα υπόλοιπα στοιχεία δηλαδή τον κωδικό, οποίος είναι κείμενο αφού έχει έστω ένα γράμμα

**c=raw\_input("Δώσε τον κωδικό")**

# Κάθε φορά πρέπει να μειώνουμε το όριο κατά b ώστε το νέο βάρος που ελέγχουμε στη while να μην ξεπερνά το όριο που απέμεινε

**orio=orio-b**

# Ξαναδιαβάζω το κριτήριο της επανάληψης

**b=float(input("Δώσε το βάρος"))**

Άσκηση 11

Φορτώνουμε ένα φορτηγάκι με δέματα διαφορετικού βάρους το καθένα. Το βάρος που μπορούμε να βάλουμε πάνω σε αυτό το φορτηγάκι δεν θα πρέπει να ξεπερνά τα 1500 κιλά. Το κάθε δέμα έχει ένα μοναδικό κωδικό που αποτελείται από δύο γράμματα και τέσσερις αριθμούς π.χ. ΠΤ3452. Να γράψετε πρόγραμμα σε ρυθμό το οποίο: **α)** Να διαβάζει το Βάρος και τον Κωδικό του κάθε δέματος που θα φορτωθεί στο φορτηγάκι μέχρι να μην ξεπεράσουμε το όριο των 1500 κιλών. **β)** Να τα τοποθετεί στις λίστες BA για το βάρος και CO για τον κωδικό. **γ)** Να υπολογίζει και να εμφανίζει πόσο είναι το πλήθος των πακέτων που φορτώθηκαν **δ)** Να υπολογίζει και να εμφανίζει το συνολικό βάρος που φορτώθηκε και το ΜΟ του βάρους των πακέτων (Θεωρήστε ότι φορτώθηκε τουλάχιστον ένα) **ε)** Με τη βοήθεια της συνάρτησης bubbleSort να κάνει αύξουσα ταξινόμηση ως προς το βάρος και να παρουσιάζει όλα τα δέματα με τη σειρά από το ελαφρύτερο προς το βαρύτερο εμφανίζοντας σε κάθε γραμμή το κείμενο π.χ. “Το δέμα με κωδικό ΠΤ3452 έχει βάρος 34 κιλά”. **στ)** Να διαβάζει το βάρος δέματος από το πληκτρολόγιο και χρησιμοποιώντας τη συνάρτηση binarySearch (πρέπει να τη γράψετε), η οποία επιστρέφει τη θέση ενός στοιχείου, να εμφανίζει τον κωδικό του δέματος που αντιστοιχεί στο βάρος ή το μήνυμα “δεν φορτώθηκε” αν αυτό δεν υπάρχει στην λίστα BA (θεωρείστε ότι κάθε βάρος που φορτώθηκε είναι μοναδικό)

# -\*- coding: cp1253 -\*-

#Στην αρχή οι λίστες μας είναι κενές

BA=[]

CO=[]

orio=1500

b=float(input("Δώσε το βάρος"))

while b<=orio:

    c=raw\_input("Δώσε τον κωδικό")

# Βάλε μέσα στη λίστα BA το βάρος και στη λίστα CO τον κωδικό που διάβασες από το πληκτρολόγιο

    BA.append(b)

    CO.append(c)

orio=orio-b

b=float(input("Δώσε το βάρος"))

Άσκηση 11

Φορτώνουμε ένα φορτηγάκι με δέματα διαφορετικού βάρους το καθένα. Το βάρος που μπορούμε να βάλουμε πάνω σε αυτό το φορτηγάκι δεν θα πρέπει να ξεπερνά τα 1500 κιλά. Το κάθε δέμα έχει ένα μοναδικό κωδικό που αποτελείται από δύο γράμματα και τέσσερις αριθμούς π.χ. ΠΤ3452. Να γράψετε πρόγραμμα σε ρυθμό το οποίο: **α)** Να διαβάζει το Βάρος και τον Κωδικό του κάθε δέματος που θα φορτωθεί στο φορτηγάκι μέχρι να μην ξεπεράσουμε το όριο των 1500 κιλών. **β)** Να τα τοποθετεί στις λίστες BA για το βάρος και CO για τον κωδικό. **γ)** Να υπολογίζει και να εμφανίζει πόσο είναι το πλήθος των πακέτων που φορτώθηκαν **δ)** Να υπολογίζει και να εμφανίζει το συνολικό βάρος που φορτώθηκε και το MO του βάρους των πακέτων (Θεωρήστε ότι φορτώθηκε τουλάχιστον ένα) **ε)** Με τη βοήθεια της συνάρτησης bubbleSort να κάνει αύξουσα ταξινόμηση ως προς το βάρος και να παρουσιάζει όλα τα δέματα με τη σειρά από το ελαφρύτερο προς το βαρύτερο εμφανίζοντας σε κάθε γραμμή το κείμενο π.χ. “Το δέμα με κωδικό ΠΤ3452 έχει βάρος 34 κιλά”. **στ)** Να διαβάζει το βάρος δέματος από το πληκτρολόγιο και χρησιμοποιώντας τη συνάρτηση binarySearch (πρέπει να τη γράψετε), η οποία επιστρέφει τη θέση ενός στοιχείου, να εμφανίζει τον κωδικό του δέματος που αντιστοιχεί στο βάρος ή το μήνυμα “δεν φορτώθηκε” αν αυτό δεν υπάρχει στην λίστα BA (θεωρείστε ότι κάθε βάρος που φορτώθηκε είναι μοναδικό)

```
# -*- coding: cp1253 -*-
```

```
BA=[]
```

```
CO=[]
```

```
orio=1500
```

```
b=float(input("Δώσε το βάρος"))
```

```
while b<=orio:
```

```
    c=raw_input("Δώσε τον κωδικό")
```

```
    BA.append(b)
```

```
    CO.append(c)
```

```
    orio=orio-b
```

```
    b=float(input("Δώσε το βάρος"))
```

**# Από τη στιγμή που γέμισαν οι λίστες μας (δηλαδή έξω από την επανάληψη) το πλήθος των πακέτων που φορτώθηκαν είναι όσα είναι και τα στοιχεία στη λίστα BA ή στην CO (έχουν το ίδιο πλήθος)**

```
print "Το πλήθος των πακέτων που φορτώθηκαν είναι", len(BA)
```

Άσκηση 11

Φορτώνουμε ένα φορτηγάκι με δέματα διαφορετικού βάρους το καθένα. Το βάρος που μπορούμε να βάλουμε πάνω σε αυτό το φορτηγάκι δεν θα πρέπει να ξεπερνά τα 1500 κιλά. Το κάθε δέμα έχει ένα μοναδικό κωδικό που αποτελείται από δύο γράμματα και τέσσερις αριθμούς π.χ. ΠΤ3452. Να γράψετε πρόγραμμα σε ρυθμό το οποίο: **α)** Να διαβάζει το Βάρος και τον Κωδικό του κάθε δέματος που θα φορτωθεί στο φορτηγάκι μέχρι να μην ξεπεράσουμε το όριο των 1500 κιλών. **β)** Να τα τοποθετεί στις λίστες BA για το βάρος και CO για τον κωδικό. **γ)** Να υπολογίζει και να εμφανίζει πόσο είναι το πλήθος των πακέτων που φορτώθηκαν **δ)** Να υπολογίζει και να εμφανίζει το συνολικό βάρος που φορτώθηκε και το MO του βάρους των πακέτων (Θεωρήστε ότι φορτώθηκε τουλάχιστον ένα) **ε)** Με τη βοήθεια της συνάρτησης bubbleSort να κάνει αύξουσα ταξινόμηση ως προς το βάρος και να παρουσιάζει όλα τα δέματα με τη σειρά από το ελαφρύτερο προς το βαρύτερο εμφανίζοντας σε κάθε γραμμή το κείμενο π.χ. “Το δέμα με κωδικό ΠΤ3452 έχει βάρος 34 κιλά”. **στ)** Να διαβάζει το βάρος δέματος από το πληκτρολόγιο και χρησιμοποιώντας τη συνάρτηση binarySearch (πρέπει να τη γράψετε), η οποία επιστρέφει τη θέση ενός στοιχείου, να εμφανίζει τον κωδικό του δέματος που αντιστοιχεί στο βάρος ή το μήνυμα “δεν φορτώθηκε” αν αυτό δεν υπάρχει στην λίστα BA (θεωρείστε ότι κάθε βάρος που φορτώθηκε είναι μοναδικό)

```
# -*- coding: cp1253 -*-
```

```
BA=[]
```

```
CO=[]
```

```
orio=1500
```

```
b=float(input("Δώσε το βάρος"))
```

```
while b<=orio:
```

```
    c=raw_input("Δώσε τον κωδικό")
```

```
    BA.append(b)
```

```
    CO.append(c)
```

```
    orio=orio-b
```

```
    b=float(input("Δώσε το βάρος"))
```

```
print "Το πλήθος των πακέτων που φορτώθηκαν είναι", len(BA)
```

```
# Άθροισμα και MO λίστας - Δες άσκηση 9 διαφάνεια 7
```

```
SUM=0.0
```

```
for i in range(len(BA)):
```

```
    SUM=SUM+BA[i]
```

```
print "Το συνολικό βάρος που φορτώθηκε είναι ", SUM
```

```
MO=SUM/len(BA)
```

```
print "MO=",MO
```

Άσκηση 11

Φορτώνουμε ένα φορηγάκι με δέματα διαφορετικού βάρους το καθένα. Το βάρος που μπορούμε να βάλουμε πάνω σε αυτό το φορηγάκι δεν θα πρέπει να ξεπερνά τα 1500 κιλά. Το κάθε δέμα έχει ένα μοναδικό κωδικό που αποτελείται από δύο γράμματα και τέσσερις αριθμούς π.χ. ΠΤ3452. Να γράψετε πρόγραμμα σε ρυθμό το οποίο: **α)** Να διαβάζει το Βάρος και τον Κωδικό του κάθε δέματος που θα φορτωθεί στο φορηγάκι μέχρι να μην ξεπεράσουμε το όριο των 1500 κιλών. **β)** Να τα τοποθετεί στις λίστες BA για το βάρος και CO για τον κωδικό. **γ)** Να υπολογίζει και να εμφανίζει πόσο είναι το πλήθος των πακέτων που φορτώθηκαν **δ)** Να υπολογίζει και να εμφανίζει το συνολικό βάρος που φορτώθηκε και το MO του βάρους των πακέτων (Θεωρήστε ότι φορτώθηκε τουλάχιστον ένα) **ε)** Με τη βοήθεια της συνάρτησης bubbleSort να κάνει αύξουσα ταξινόμηση ως προς το βάρος και να παρουσιάζει όλα τα δέματα με τη σειρά από το ελαφρύτερο προς το βαρύτερο εμφανίζοντας σε κάθε γραμμή το κείμενο π.χ. “Το δέμα με κωδικό ΠΤ3452 έχει βάρος 34 κιλά”. **στ)** Να διαβάζει το βάρος δέματος από το πληκτρολόγιο και χρησιμοποιώντας τη συνάρτηση binarySearch (πρέπει να τη γράψετε), η οποία επιστρέφει τη θέση ενός στοιχείου, να εμφανίζει τον κωδικό του δέματος που αντιστοιχεί στο βάρος ή το μήνυμα “δεν φορτώθηκε” αν αυτό δεν υπάρχει στην λίστα BA (θεωρείστε ότι κάθε βάρος που φορτώθηκε είναι μοναδικό)

```
# -*- coding: cp1253 -*-
BA=[]
CO=[]
orio=1500
b=float(input("Δώσε το βάρος"))
while b<=orio:
    c=raw_input("Δώσε τον κωδικό")
    BA.append(b)
    CO.append(c)
    orio=orio-b
    b=float(input("Δώσε το βάρος"))
print "Το πλήθος των πακέτων που φορτώθηκαν είναι", len(BA)
SUM=0.0
for i in range(len(BA)):
    SUM=SUM+BA[i]
print "Το συνολικό βάρος που φορτώθηκε είναι ", SUM
MO=SUM/len(BA)
print "MO=",MO
```

**#Πρώτα γράφω τη συνάρτηση bubbleSort για 2 λίστες αφού τόσες (παράλληλες) έχω**

```
def bubbleSort(A,B):
    N=len(A)
    for i in range(N-1):
        for j in range(N-1,i,-1):
            if A[j]<A[j-1]:
                A[j],A[j-1]=A[j-1],A[j]
                B[j],B[j-1]=B[j-1],B[j]
```

**# Στη συνέχεια κάνω αύξουσα ταξινόμηση ως προς το βάρος άρα καλώ την bubbleSort και βάζω πρώτη τη λίστα με το βάρος**

```
bubbleSort(BA,CO)
```

**# Απομένει μόνο να εμφανίσω τον κωδικό και το βάρος του κάθε δέματος, άρα διατρέχω τις λίστες**

```
for i in range(len(BA)):
    print "Το δέμα με κωδικό ", CO[i], " έχει βάρος ", BA[i], " κιλά"
```

Άσκηση 11

Φορτώνουμε ένα φορτηγάκι με δέματα διαφορετικού βάρους το καθένα. Το βάρος που μπορούμε να βάλουμε πάνω σε αυτό το φορτηγάκι δεν θα πρέπει να ξεπερνά τα 1500 κιλά. Το κάθε δέμα έχει ένα μοναδικό κωδικό που αποτελείται από δύο γράμματα και τέσσερις αριθμούς π.χ. ΠΤ3452. Να γράψετε πρόγραμμα σε ρύθση το οποίο: **α)** Να διαβάζει το Βάρος και τον Κωδικό του κάθε δέματος που θα φορτωθεί στο φορτηγάκι μέχρι να μην ξεπεράσουμε το όριο των 1500 κιλών. **β)** Να τα τοποθετεί στις λίστες BA για το βάρος και CO για τον κωδικό. **γ)** Να υπολογίζει και να εμφανίζει πόσο είναι το πλήθος των πακέτων που φορτώθηκαν **δ)** Να υπολογίζει και να εμφανίζει το συνολικό βάρος που φορτώθηκε και το MO του βάρους των πακέτων (Θεωρήστε ότι φορτώθηκε τουλάχιστον ένα) **ε)** Με τη βοήθεια της συνάρτησης bubbleSort να κάνει αύξουσα ταξινόμηση ως προς το βάρος και να παρουσιάζει όλα τα δέματα με τη σειρά από το ελαφρύτερο προς το βαρύτερο εμφανίζοντας σε κάθε γραμμή το κείμενο π.χ. “Το δέμα με κωδικό ΠΤ3452 έχει βάρος 34 κιλά”. **στ)** Να διαβάζει το βάρος δέματος από το πληκτρολόγιο και χρησιμοποιώντας τη συνάρτηση binarySearch (πρέπει να τη γράψετε), η οποία επιστρέφει τη θέση ενός στοιχείου, να εμφανίζει τον κωδικό του δέματος που αντιστοιχεί στο βάρος ή το μήνυμα “δεν φορτώθηκε” αν αυτό δεν υπάρχει στην λίστα BA (θεωρείστε ότι κάθε βάρος που φορτώθηκε είναι μοναδικό)

```
# -- coding: utf-8 --
b=<
C=
size=1500
while b<=size
    print "Το βάρος των πακετων που φορτωσαν ειναι", b
    b+=1
    print "Το συνολικό βάρος που φορτωσαν ειναι", S
    print "MO", MO
end

bubbleSort(CO)
for i in range(0,CO)
    print "Το δέμα με κωδικό ", CO[i], " έχει βάρος ", BA[i], " κιλά"
```

**b1=float(input("Δώσε το βάρος που ψάχνεις"))** #Διάβασε ένα βάρος από το πληκτρολόγιο  
**#Γράψε την binarySearch που να επιστρέφει τη θέση του στοιχείου key μέσα στην λίστα array ή -1 αν δεν υπάρχει**

```
def binarySearch(array,key):
    first=0
    last=len(array)-1
    pos=-1
    while first<=last and pos==-1:
        mid=(first+last)/2
        if array[mid]==key:
            pos=mid
        elif array[mid]<key:
            first=mid+1
        else:
            last=mid-1
    return pos
```

**# Καλώ την binarySearch για να βρω τη θέση του b1 μέσα στη λίστα BA. ΠΡΟΣΟΧΗ:** Αυτή λειτουργεί σωστά επειδή η λίστα BA είναι ήδη ταξινομημένη με αύξουσα σειρά ως προς το βάρος  
**thesi=binarySearch(BA,b1)**  
**if thesi== -1:** #Αν η θέση είναι -1 σημαίνει ότι δεν υπάρχει στη λίστα BA οπότε  
 print "Δεν φορτώθηκε"  
**else:** #Αλλιώς εμφανίζω τον κωδικό στην αντίστοιχη θέση (thesi)  
 print "Ο κωδικός του δέματος με βάρος ",b1," είναι ",CO[thesi]



Άσκηση 11

# ΤΟ ΤΕΛΙΚΟ ΠΡΟΓΡΑΜΜΑ

```
# -*- coding: cp1253 -*-
BA=[]
CO=[]
orio=1500
b=float(input("Δώσε το βάρος"))
while b<=orio:
    c=raw_input("Δώσε τον κωδικό")
    BA.append(b)
    CO.append(c)
    orio=orio-b
    b=float(input("Δώσε το βάρος"))
print "Το πλήθος των πακέτων που φορτώθηκαν είναι", len(BA)
SUM=0.0
for i in range(len(BA)):
    SUM=SUM+BA[i]
print "Το συνολικό βάρος που φορτώθηκε είναι ", SUM
MO=SUM/len(BA)
print "MO=",MO
def bubbleSort(A,B):
    N=len(A)
    for i in range(N-1):
        for j in range(N-1,i,-1):
            if A[j]<A[j-1]:
                A[j],A[j-1]=A[j-1],A[j]
                B[j],B[j-1]=B[j-1],B[j]
```

```
bubbleSort(BA,CO)
for i in range(len(BA)):
    print "Το δέμα με κωδικό ", CO[i]," έχει βάρος ", BA[i], " κιλά"

b1=float(input("Δώσε το βάρος που ψάχνεις"))

def binarySearch(array,key):
    first=0
    last=len(array)-1
    pos=-1
    while first<=last and pos==-1:
        mid=(first+last)/2
        if array[mid]==key:
            pos=mid
        elif array[mid]<key:
            first=mid+1
        else:
            last=mid-1
    return pos

thesi=binarySearch(BA,b1)
if thesi==-1:
    print "Δεν φορτώθηκε"
else:
    print "Ο κωδικός του δέματος με βάρος ",b1," είναι ",CO[thesi]
```