

Ασκήσεις Φυλλάδιο 2 - Λύσεις (Κεφάλαιο 4)

(MO, MIN, MAXONOMA, MINONOMA, MAX, SUM, μερικά αθροίσματα, κλιμακωτή χρέωση, εμφωλευμένες επαναλήψεις, μετρητές)

1.

-*- coding: cp1253 -*-

Η παραπάνω εντολή είναι για να βγάξει σωστά τα ελληνικά

Σύνολο κιλών = SUM με αρχική τιμή 0 (γ)

SUM=0

Θα κρατήσουμε στη μεταβλητή MAX το βάρος του βαρύτερου κοντέινερ και στη μεταβλητή MAXTYPOS
#τον τύπο του βαρύτερου κοντέινερ

#Στην αρχή βάζουμε μία τιμή πολύ μικρή (0) και θεωρούμε ότι ο τύπος είναι το κενό (" ") - Αρχικοποίηση
#τιμών (ε)

MAX=0

MAXTYPOS=" "

Θα κρατήσουμε στη μεταβλητή MIN το βάρος του ελαφρύτερου κοντέινερ και στη μεταβλητή MINTYPOS
#τον τύπο του ελαφρύτερου κοντέινερ

#Στην αρχή βάζουμε μία τιμή πολύ μεγάλη (32501) και θεωρούμε ότι ο τύπος είναι το κενό (" ") -
#Αρχικοποίηση τιμών (στ)

MIN=32501

MINTYPOS=" "

#mA για να μετρήσω τα κοντέινερ τύπου A, mB και mC αντίστοιχα για τα τύπου B και C. Στην αρχή έχω από
#όλα 0 - Αρχικοποίηση (ζ)

mA=mB=mC=0

Τα κιλά που έχω συνολικά από κάθε τύπου είναι μερικά αθροίσματα. Στην ουσία θέλω SUM μόνο για τα A
#π.χ. SUMA και SUMB μόνο για τα B

και SUMC μόνο για τα C. Αρχικοποίηση: Στην αρχή όλα τα αθροίσματα είναι 0 αφού δεν έχω πάρει ακόμη
#το βάρος κανενός κοντέινερ (η)

SUMA=SUMB=SUMC=0

Για 300 κοντέινερ

for i in range(300):

 # Διάβασε τον τύπο (t) που θα είναι κείμενο και το μεικτό βάρος (mb) που θα είναι κινητής
#υποδιαστολής (α)

 t=raw_input("Δώσε τον τύπο του κοντέινερ (A ή B ή C):")

 mb=float(input("Δώσε το μεικτό βάρος του κοντέινερ:"))

 # Έλεγχος ορθότητας στο βάρος του κοντέινερ (β)

 while mb<=0 or mb>=32500:

 mb=float(input("Ξαναδώσε το μεικτό βάρος του κοντέινερ (πριν έδωσες λάθος τιμή):"))

 # Το νέο άθροισμα είναι το παλιό + το νέο μεικτό βάρος που μόλις πήρα από το πληκτρολόγιο (γ)

 SUM=SUM+mb

 # εύρεση μεγίστου (ε)

 if mb>MAX:

```
MAX=mb
MAXTYPOS=t
```

```
# εύρεση ελαχίστου (στ)
```

```
if mb<MIN:
```

```
    MIN=mb
```

```
    MINTYPOS=t
```

```
# Αν έχω ένα κοντέινερ τύπου A αύξησε το mA κατά 1. Το ίδιο και για τα υπόλοιπα (ζ)
```

```
if t=="A":
```

```
    mA=mA+1
```

```
# Έχω τύπο A άρα αύξησε το συνολικό βάρος των A κατά mb (η)
```

```
    SUMA=SUMA+mb
```

```
elif t=="B":
```

```
    mB=mB+1
```

```
# Έχω τύπο B άρα αύξησε το συνολικό βάρος των B κατά mb (η)
```

```
    SUMB=SUMB+mb
```

```
elif t=="C":
```

```
    mC=mC+1
```

```
# Έχω τύπο C άρα αύξησε το συνολικό βάρος των C κατά mb (η)
```

```
    SUMC=SUMC+mb
```

```
# Εμφάνισε το SUM (γ)
```

```
print "Το σύνολο των κιλών που φορτώθηκαν στο πλοίο είναι ", SUM
```

```
# Βρες το μέσο όρο (MO) και εμφάνισέ τον (δ)
```

```
MO=SUM/300.0
```

```
print " Το κάθε κοντέινερ είναι κατά μέσο όρο ", MO, " κιλά"
```

```
#Εμφάνισε το φορτίο με το μέγιστο βάρος και τον τύπο του (ε)
```

```
print "Το φορτίο με το μέγιστο βάρος είναι ", MAX,"κιλά και είναι τύπου",MAXTYPOS
```

```
#Εμφάνισε το φορτίο με το ελάχιστο βάρος και τον τύπο του (στ)
```

```
print "Το φορτίο με το ελάχιστο βάρος είναι ", MIN,"κιλά και είναι τύπου",MINTYPOS
```

```
#Εμφάνισε πόσα κοντέινερ έχω από κάθε τύπο (ζ)
```

```
print "Έχω ", mA, " κοντέινερ τύπου A"
```

```
print "Έχω ", mB, " κοντέινερ τύπου B"
```

```
print "Έχω ", mC, " κοντέινερ τύπου C"
```

```
#Εμφάνισε πόσο είναι το συνολικό βάρος σε κάθε τύπο (η)
```

```
print " Τα κοντέινερ τύπου A έχουν συνολικό βάρος", SUMA
```

```
print " Τα κοντέινερ τύπου B έχουν συνολικό βάρος", SUMB
```

```
print " Τα κοντέινερ τύπου C έχουν συνολικό βάρος", SUMC
```

```
2.
```

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

```
# Η παραπάνω εντολή είναι για να βγάξει σωστά τα ελληνικά
```

```
# (ε) Πόσους μαθητές έχουμε από κάθε κατηγορία - 4 κατηγορίες άρα 4 μετρητές
```

```
# Στην αρχή έχω για κάθε κατηγορία 0 μαθητές
```

m1=m2=m3=m4=0

(στ) Σύνολο λεπτών = SUM όλων των λεπτών. Στην αρχή είναι 0
SUM=0.0

(η) Θέτω τα περισσότερα λεπτά που έπαιξε κάποιος MAX και δίνω αρχική τιμή 0
MAX=0

(θ) Θέτω τα λιγότερα λεπτά που έπαιξε κάποιος MIN και δίνω αρχική τιμή 1500 - πάνω από τα λεπτά μιας ημέρας (1440)
MIN=1500

Το σύνολο των λεπτών που έπαιξαν τα αγόρια είναι άθροισμα και το ονομάζω SUMA (ι)
SUMA=0.0

#Για να βρω το MO των λεπτών των αγοριών θέλω το άθροισμα των λεπτών όλων των αγοριών δια όσα είναι (ια)
Άρα βάζω ένα μετρητή mA ο οποίος να μετράει τα αγόρια - Στην αρχή έχω 0 αγόρια
mA=0

Το σύνολο των λεπτών που έπαιξαν τα κορίτσια είναι άθροισμα και το ονομάζω SUMK (ιβ)
SUMK=0.0

#Για να βρω το MO των λεπτών των κοριτσιών θέλω το άθροισμα των λεπτών όλων των κοριτσιών δια όσα είναι (ιγ)
Άρα βάζω ένα μετρητή mK ο οποίος να μετράει τα κορίτσια - Στην αρχή έχω 0 κορίτσια
mK=0

Για να βρω το ποσοστό των παιδιών που παίζουν πάνω από 60 λεπτά πρέπει πρώτα να τα μετρήσω άρα m60 (ιδ)
m60=0

Έχω επανάληψη χωρίς προκαθορισμένα βήματα οπότε χρησιμοποιώ while
Διαβάζω μία φορά από έξω το κριτήριο της επανάληψης και μία λίγο πριν τελειώσει η επανάληψη
Το κριτήριο της επανάληψης είναι " Όσο το φύλο δεν είναι η λέξη "TELOS" , άρα διαβάζω
μία φορά από έξω το φύλο και βάζω μία ίδια εντολή λίγο πριν τελειώσει η επανάληψη (α) και (β)
Το φύλο είναι γράμματα άρα αλφαριθμητικό άρα χρησιμοποιώ raw_input

f=raw_input("Δώσε το φύλο")
while f!="TELOS":

 # Διαβάζω τα υπόλοιπα μέσα στη while, εκτός από αυτό από το οποίο εξαρτάται η επανάληψη (α)

 # Βάζω (h) για ηλικία και (lep) για λεπτά

 h=input("Δώσε την ηλικία σε έτη")

 # Έλεγχος ορθότητας γράφω μία while - όσο δεν είναι αποδεκτή τιμή

 # Όσο τα έτη είναι κάτω από 10 ή πάνω από 18 να ξαναζητάει την ηλικία

 # Βάζω ή (or) γιατί δεν θέλω ταυτόχρονα να είναι και κάτω από 0 και πάνω από 18

 # αυτό δεν αληθεύει ποτέ (δηλαδή αν είχα and) (γ)

 while h<10 or h>18:

 h=input("Δώσε την ηλικία σε έτη")

 # (α) Διάβασε τα λεπτά που παίζει

 lep=float(input("Δώσε τα λεπτά που παίζει συνήθως σε μία ημέρα"))

 # (δ) χωρίζω 4 περιπτώσεις

 if lep<15:

 print "Εξάρτηση Μικρή"

 # (ε) Βρήκα έναν από την πρώτη κατηγορία άρα αυξάνω τον m1 κατά 1

 m1=m1+1

```

elif lep>=15 and lep<=30:
    print "Εξάρτηση Μέτρια"
    # (ε) Βρήκα έναν από την δεύτερη κατηγορία άρα αυξάνω τον m2 κατά 1
    m2=m2+1
elif lep>30 and lep<=60:
    print "Εξάρτηση Μεγάλη"
    # (ε) Βρήκα έναν από την τρίτη κατηγορία άρα αυξάνω τον m3 κατά 1
    m3=m3+1
elif lep>60 :
    print "Εξάρτηση Πάρα πολύ μεγάλη"
    # (ε) Βρήκα έναν από την τέταρτη κατηγορία άρα αυξάνω τον m4 κατά 1
    m4=m4+1

```

```

# Σύνολο λεπτών = Σύνολο λεπτών + τα νέα λεπτά που μόλις διάβασα (στ)
SUM=SUM+lep

```

```

# Υπολογισμός μεγίστου στα λεπτά (η)
if lep>MAX:
    MAX=lep

```

```

#Υπολογισμός ελαχίστου στα λεπτά (θ)
if lep<MIN:
    MIN=lep

```

```

#Αν αυτό που έχω τώρα είναι αγόρι:
if f=="Α":
    #Αύξησε το άθροισμα των αγοριών κατά lep (ι)
    SUMA=SUMA+lep
    # Αύξησε το πλήθος των αγοριών κατά 1 (ια)
    mA=mA+1
else:
    #Αν αυτό που έχω είναι κορίτσι
    #Αύξησε το άθροισμα των κοριτσιών κατά lep (ιβ)
    SUMK=SUMK+lep
    # Αύξησε το πλήθος των κοριτσιών κατά 1 (ιγ)
    mK=mK+1

```

```

#Αν ένα παιδί παίζει πάνω από 60 λεπτά πρέπει να το μετρήσω (ιδ)
if lep>60:
    m60=m60+1

```

```

#μία ίδια εντολή λίγο πριν τελειώσει η επανάληψη (α) και (β)
f=raw_input("Δώσε το φύλο")
# Τέλος επανάληψης WHILE

```

```

# (στ) Εμφάνιση του συνόλου των λεπτών
print " Οι μαθητές παίζουν συνολικά κάθε μέρα ", SUM ," λεπτά"

```

```

# (ζ) Υπολογισμός και εμφάνιση ΜΟ. Έχω το SUM μου λείπει το πλήθος όλων των μαθητών
#Θα μπορούσα να βάλω ένα μετρητή (ρι) που να αυξάνει κατά ένα κάθε φορά που εκτελείται η επανάληψη.
#Εδώ δεν χρειάζεται γιατί έχω ήδη μετρήσει όλες τις κατηγορίες (m1, m2, m3 ,m4) οπότε το σύνολο των
# Μαθητών είναι:
ri=m1+m2+m3+m4

```

```
# Θα πρέπει να προβλέψω την περίπτωση που το ρl είναι 0 (Αν μου έδωσε αμέσως για φύλο τη λέξη "TELOS")
```

```
# Γιατί διαιρέση με 0 δεν γίνεται, οπότε δεν μπορώ αμέσως να πω MO=SUM/ρl
```

```
if ρl==0:
```

```
    print "Δεν υπάρχει Μέσος όρος γιατί δεν μου έδωσες κανένα μαθητή/τρια"
```

```
else:
```

```
    MO=SUM/ρl
```

```
    print "Τα παιδιά παίζουν κατά μέσο όρο ", MO ," λεπτά"
```

```
#Εμφάνιση πόσους μαθητές έχουμε από κάθε κατηγορία (ε)
```

```
print " Οι μαθητές/τριες που έχουν μικρή εξάρτηση είναι: ", m1
```

```
print " Οι μαθητές/τριες που έχουν μέτρια εξάρτηση είναι: ", m2
```

```
print " Οι μαθητές/τριες που έχουν μεγάλη εξάρτηση είναι: ", m3
```

```
print " Οι μαθητές/τριες που έχουν Πάρα πολύ μεγάλη εξάρτηση είναι: ", m4
```

```
# Εμφάνιση τα περισσότερα λεπτά (η)
```

```
print "Τα περισσότερα λεπτά που έπαιξε κάποιος/α μαθητής/τρια είναι",MAX," λεπτά"
```

```
# Εμφάνιση τα λιγότερα λεπτά (θ)
```

```
print "Τα λιγότερα λεπτά που έπαιξε κάποιος/α μαθητής/τρια είναι",MIN," λεπτά"
```

```
#Υπολογισμός και Εμφάνιση του MO των αγοριών (ια)
```

```
# Αν δεν έχω καθόλου αγόρια (mA==0) τότε θα πρέπει να βγάξει το μήνυμα "Δεν υπάρχει MO αφού δεν έδωσες αγόρια"
```

```
if mA==0:
```

```
    print "Δεν υπάρχει MO αφού δεν έδωσες αγόρια"
```

```
#Αλλιώς : βρες το MO των λεπτών των αγοριών και εμφάνισέ τον
```

```
else:
```

```
    MOA=SUMA/mA
```

```
    print "Ο Μέσος όρος των λεπτών που παίζουν τα αγόρια είναι ", MOA
```

```
#Υπολογισμός και Εμφάνιση του MO των κοριτσιών (iy)
```

```
# Αν δεν έχω καθόλου κορίτσια (mK==0) τότε θα πρέπει να βγάξει το μήνυμα "Δεν υπάρχει MO αφού δεν έδωσες κορίτσια"
```

```
if mK==0:
```

```
    print "Δεν υπάρχει MO αφού δεν έδωσες κορίτσια"
```

```
#Αλλιώς : βρες το MO των λεπτών των κοριτσιών και εμφάνισέ τον
```

```
else:
```

```
    MOK=SUMK/mK
```

```
    print "Ο Μέσος όρος των λεπτών που παίζουν τα κορίτσια είναι ", MOK
```

```
# (ιδ) Έχω m60 Παιδιά που παίζουν πάνω από 60 λεπτά στον Η/Υ σε σύνολο mA+mK παιδιά
```

```
# Σε ένα χαρτί με απλή μέθοδο των τριών λέω :
```

```
# Σε σύνολο (mA+mK) παιδιά πάνω από 60 λεπτά παίζουν m60 παιδιά
```

```
# Από κάτω στα 100 πόσα Χ; και κάνω χιαστί οπότε προκύπτει ο παρακάτω τύπος:
```

```
#Το float το βάζω επειδή είναι όλα ακέραιοι για να γίνει σωστά η διαίρεση
```

```
X=(float(m60)/(mA+mK))*100
```

```
print "Το ποσοστό των παιδιών που παίζουν πάνω από 60 λεπτά την ημέρα είναι",X," %"
```

3.

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

```
# Η παραπάνω εντολή είναι για να βγάξει σωστά τα ελληνικά
```

```
# (α) Συνολικό βάρος = SUM. Στην αρχή είναι 0
```

```
SUM=0.0
```

```
# Βάζω mA μετρητή για να μετράω τα δοχεία τύπου A και mB για τα τύπου B και στην αρχή έχω 0 και από τους δύο τύπους (β)
```

```
mA=mB=0
```

```
# Έχω επανάληψη χωρίς προκαθορισμένα βήματα οπότε χρησιμοποιώ while (α)
```

```
# Η επανάληψή μου εξαρτάται από κάτι το οποίο υπολογίζω (SUM), άρα διαβάζω μία φορά όλα τα δεδομένα μου μέσα στην επανάληψη
```

```
# Η επανάληψη επαναλαμβάνεται μέχρι το SUM να ξεπεράσει το 1000, όσο δηλαδή το SUM είναι <=1000.
```

```
# Ο τύπος του δοχείου είναι γράμμα, άρα αλφαριθμητικό, άρα χρησιμοποιώ raw_input
```

```
while SUM<=1000:
```

```
    typos=raw_input("Δώσε τον τύπο του δοχείου (A ή B)")
```

```
    if typos=="A":
```

```
        SUM=SUM+17
```

```
        # Αύξησε το mA κατά 1 γιατί έχω ένα ακόμα δοχείο τύπου A (β)
```

```
        mA+=1 #Είναι το ίδιο με το mA=mA+1
```

```
    else:
```

```
        SUM=SUM+8.5
```

```
        # Αύξησε το mB κατά 1 γιατί έχω ένα ακόμα δοχείο τύπου B (β)
```

```
        mB+=1 #Είναι το ίδιο με το mB=mB+1
```

```
# Εμφάνισε πόσα δοχεία τύπου A και πόσα δοχεία τύπου B φορτώθηκαν
```

```
print "Φορτώθηκαν ", mA, ' δοχεία τύπου A και ', mB, ' δοχεία τύπου B'
```

```
# Για να βρώ τα κιλά που μπήκαν σε δοχεία τύπου A (SUMA) πολλαπλασιάζω τα δοχεία τύπου A επί 17 (γ)
```

```
SUMA=mA*17
```

```
# Για να βρώ τα κιλά που μπήκαν σε δοχεία τύπου B (SUMB) πολλαπλασιάζω τα δοχεία τύπου B επί 8.5 (γ)
```

```
SUMB=mB*8.5
```

```
# Εμφανίζω πόσα κιλά μπήκαν σε δοχεία τύπου A και σε δοχεία τύπου B (γ)
```

```
print "Τα κιλά που μπήκαν σε δοχεία τύπου A είναι ", SUMA
```

```
print "Τα κιλά που μπήκαν σε δοχεία τύπου B είναι ", SUMB
```

```
#Υπολογίζω τα συνολικά έσοδα se (δ)
```

```
se=mA*100+mB*50
```

```
# Εμφανίζω τα συνολικά έσοδα se (δ)
```

```
print "Τα συνολικά έσοδα είναι ", se, " €"
```

4.

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

```
# Η παραπάνω εντολή είναι για να βγάξει σωστά τα ελληνικά
```

```
#(δ) Να υπολογίζει και να εμφανίζει πόσο φόρο θα πληρώσουν όλοι μαζί οι αγρότες του χωριού. SUMF
```

SUMF=0

#(στ) Να υπολογίζει και να εμφανίζει το ονοματεπώνυμο του αγρότη που θα πληρώσει το #μεγαλύτερο φόρο και το φόρο αυτό . (MAXF και MAXFONOMA)

MAXF=0
MAXFONOMA=""

#(ζ) Να υπολογίζει και να εμφανίζει το ονοματεπώνυμο του αγρότη που θα πληρώσει το #μικρότερο φόρο και το φόρο αυτό . (MINF και MINFONOMA)

MINF=1000000000
MINFONOMA=""

#Υπάρχουν 100 αγρότες και για κάθε ένα να διαβάζει το ονοματεπώνυμό του (α)

for i in range(100):
 ONEP=raw_input("Δώσε το ονοματεπώνυμο")

Να διαβάζει τον κωδικό αριθμό κάθε χωραφιού και τα στρέμματα μέχρι να μας δοθεί για κωδικός ο αριθμός 0

#Και να υπολογίζει και να εμφανίζει τα συνολικά στρέμματα που έχει ο συγκεκριμένος αγρότης

SUMA (β)

SUMA=0
k=int(input("Δώσε τον κωδικό αριθμό"))
while k!=0:
 s=float(input("Δώσε τα στρέμματα"))
 SUMA=SUMA+s
 k=int(input("Δώσε τον κωδικό αριθμό"))

print "Ο αγρότης ",ONEP," έχει ",SUMA," στρέμματα"

#Να υπολογίζει και να εμφανίζει το συνολικό φόρο (κλιμακωτά) που πρέπει να πληρώσει ο συγκεκριμένος αγρότης. (γ)

if SUMA>=1 and SUMA<=100:
 f=SUMA*1
elif SUMA>100 and SUMA<=300:
 f=100*1+(SUMA-100)*2
elif SUMA>300:
 f=100*1+200*2+(SUMA-300)*3

print "Ο αγρότης ",ONEP," πρέπει να πληρώσει φόρο ",f," €"

#(δ) Να υπολογίζει και να εμφανίζει πόσο φόρο θα πληρώσουν όλοι μαζί οι αγρότες του χωριού.

SUMF

SUMF=SUMF+f

#(στ) Να υπολογίζει και να εμφανίζει το ονοματεπώνυμο του αγρότη που θα πληρώσει το #μεγαλύτερο φόρο και το φόρο αυτό . (MAXF και MAXFONOMA)

if f>MAXF:
 MAXF=f
 MAXFONOMA=ONEP

#(ζ) Να υπολογίζει και να εμφανίζει το ονοματεπώνυμο του αγρότη που θα πληρώσει το #μικρότερο φόρο και το φόρο αυτό . (MINF και MINFONOMA)

if f<MINF:
 MINF=f
 MINFONOMA=ONEP

#(δ) Να υπολογίζει και να εμφανίζει πόσο φόρο θα πληρώσουν όλοι μαζί οι αγρότες του χωριού. SUMF

```
print "Όλοι οι αγρότες του χωριού θα πληρώσουν ",SUMF," φόρο"
```

#(ε) Να υπολογίζει και να εμφανίζει κατά μέσο όρο πόσο φόρο θα πληρώσει ο κάθε αγρότης.

MO=SUMF/100.0

```
print "Κάθε αγρότης θα πληρώσει κατά μέσο όρο ",MO," €"
```

#(στ) Να υπολογίζει και να εμφανίζει το ονοματεπώνυμο του αγρότη που θα πληρώσει το

#μεγαλύτερο φόρο και το φόρο αυτό . (MAXF και MAXFONOMA)

```
print "Ο αγρότης ",MAXFONOMA," έχει το μεγαλύτερο φόρο που είναι ",MAXF
```

#(στ) Να υπολογίζει και να εμφανίζει το ονοματεπώνυμο του αγρότη που θα πληρώσει το

#μικρότερο φόρο και το φόρο αυτό . (MINF και MINFONOMA)

```
print "Ο αγρότης ",MINFONOMA," έχει το μικρότερο φόρο που είναι ",MINF
```