

ΑΡΧΗ 1ΗΣ ΣΕΛΙΔΑΣ

ΠΡΟΤΕΙΝΟΜΕΝΑ ΘΕΜΑΤΑ 5

ΛΥΣΕΙΣ

A1.

1. Σωστό
2. Σωστό
3. Λάθος
4. Σωστό
5. Σωστό

A2.

- 1) γ
- 2) α
- 3) ε
- 4) δ
- 5) στ

A3.

1)	not(B<6)	False
2)	(A<B) and C	True
3)	A>1000 or A<=7	True
4)	(2*B+4)>(A**2) and C	False
5)	(A != B) or not (A == B) and C	True

B1.

... συνάρτηση SUMP η οποία δέχεται έναν ακέραιο αριθμό N ...

def SUMP(N):

Άθροισμα άρα αρχική τιμή 0

SUM=0

Δημιουργώ τους περιττούς (μονούς) αριθμούς από το 1 μέχρι και το N, βάζω N+1 γιατί αν το N είναι μονός αριθμός θέλω να συμμετέχει στο άθροισμα

for i in range(1,N+1,2):

Το νέο άθροισμα είναι το παλιό άθροισμα συν το i

SUM=SUM+i

Επιστρέφω το αποτέλεσμα

return SUM

B2.

α)

6 3

ΑΡΧΗ 2ΗΣ ΣΕΛΙΔΑΣ

6 7
5 8
5 12

β)

Κάνω τη λύση σε δύο βήματα : πρώτα αλλάζω την εξωτερική for σε while. Ελέγχω ποια είναι η μεταβλητή από την εξαρτάται η επανάληψη (i) και της δίνω αρχική τιμή (i=10)

Επανάληψη μέχρι να μην ξεπεράσω το 6 (i>6). Εφόσον το βήμα είναι αρνητικό, η τιμή του i συνεχώς μειώνεται γι' αυτό και βάζω i>6 , γιατί θα το ξεπεράσω ξεκινώντας από το 10 και αφαιρώντας.

Στο τέλος λίγο πριν τελειώσει η επανάληψη μειώνω το i κατά 3 (i=i-3)

Βήμα 1ο

x=10

y=2

i=10

while i>6:

 x=x-i+6

 y=y-1

 j=2

 while j<=4:

 y=y+j

 print x,y

 j=j+2

 i=i-3

Βήμα 2° (Τελική Λύση)

Στην εσωτερική while βλέπω ότι η μεταβλητή από την οποία εξαρτάται η επανάληψη είναι η j

Παρατηρώ ότι ξεκινάει από 2, αυξάνει ανά 2 και σταματάει μόλις ξεπεράσει την τιμή 4

δηλαδή στην τιμή 5. Βάζω στην for στο μέχρι την τιμή την οποία δεν μπορεί να πάρει η μεταβλητή αλλά ούτε και να την ξεπεράσει. Αν έβαζα 4 δεν θα μπορούσε να την πάρει.

x=10

y=2

i=10

while i>6:

 x=x-i+6

 y=y-1

 for j in range(2,5,2):

 y=y+j

ΤΕΛΟΣ 2ΗΣ ΑΠΟ 5 ΣΕΛΙΔΕΣ

ΑΡΧΗ 3ΗΣ ΣΕΛΙΔΑΣ

```
    print x,y
    i=i-3
```

Γ.

-*- coding: cp1253 -*-

Γ1

```
def EPIDOT(OIK,KOST):
    if OIK<=20000:
        poso=KOST*70/100.0
    elif OIK<=40000:
        poso=KOST*60/100.0
    else:
        poso=KOST*50/100.0
    return poso
```

#Γ4

SUM=0

#Γ5

m1=0

#Γ6 Θα πρέπει να μετρήσω πόσες είναι όλες οι περιπτώσεις και αφού θα έχω και το SUM

#Μπορώ να βρω το ΜΟ

m=0

#Γ2

on=raw_input("Δώσε το ονοματεπώνυμο: ")

while on!="ΤΕΛΟΣ":

oik_eis=input("Δώσε το οικογενειακό εισόδημα: ")

kostos=input("Δώσε το κόστος εργασίας: ")

Έλεγχος ορθότητας στο κόστος

while kostos<=0 or kostos>100000:

kostos=input("Δώσε το κόστος εργασίας: ")

#Γ3 Καλώ τη συνάρτηση με τις μεταβλητές του κυρίως προγράμματος

poso_epi=EPIDOT(oik_eis,kostos)

print "Ποσό επιδότησης: ",poso_epi

Το ποσό που θα πληρώσει είναι το αρχικό κόστος της εργασίας μείον την επιδότηση

poso_dikomou=kostos-poso_epi

print "Το ποσό που θα πληρώσει ο δικαιούχος είναι ",poso_dikomou

#Γ4

ΑΡΧΗ 4ΗΣ ΣΕΛΙΔΑΣ

```

SUM=SUM+poso_epi
#Γ5
if poso_epi>40000:
    m1+=1
#Γ6
m+=1
on=raw_input("Δώσε το ονοματεπώνυμο: ")
#Γ4
print "Το συνολικό ποσό επιδότησης από το Κράτος για όλους είναι :",SUM

#Γ5
print "Με πάνω από 40.000 € επιδοτήθηκαν ",m1," πολίτες"
#Γ6
MO=SUM/float(m)
print "Ο κάθε πολίτης πήρε κατά μέσο όρο ",MO," επιδότηση"

Δ.
#-*- coding: cp1253 -*-
#Δ1
N=input("Δώσε το πλήθος των αυτοκινήτων: ")

#Δ2
AK=[ ]
KOSTOS=[ ]
for i in range(N):
    ark=raw_input("Δώσε τον αριθμό κυκλοφορίας: ")
    ekost=float(input("Δώσε το ετήσιο κόστος συντήρησης: "))
    AK.append(ark)
    KOSTOS.append(ekost)

#Δ3 Η εύρεση του SUM μπορούσε να γίνει και μέσα στην προηγούμενη επανάληψη
SUM=0.0
for i in range(len(AK)):
    SUM=SUM+KOSTOS[i]
MO=SUM/N
print "Κατα μέσο όρο κάθε χρόνο ο Δήμος πληρώνει για κάθε αυτοκίνητο ",MO," €"

#Δ4
print "Αυτοκίνητα με ετήσιο κόστος πάνω από το MO"
```

ΤΕΛΟΣ 4ΗΣ ΑΠΟ 5 ΣΕΛΙΔΕΣ

ΑΡΧΗ 5ΗΣ ΣΕΛΙΔΑΣ

```
for i in range(N):
```

```
    #Αν το κόστος είναι μεγαλύτερο από το ΜΟ εμφάνισε τον αρ. κυκλοφορίας και το κόστος
```

```
    if KOSTOS[i]>MO:
```

```
        print AK[i],KOSTOS[i]
```

```
#Δ5 Θα πρέπει να χρησιμοποιήσω τη bubbleSort με δύο λίστες
```

```
# Θα τη γράψω με αύξουσα ταξινόμηση έτσι ώστε μόλις κάνω ταξινόμηση με βάση το κόστος
```

```
#Τα οχήματα με το μεγαλύτερο κόστος να είναι στις τελευταίες θέσεις (-1,-2,-3)
```

```
def bb2(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]
```

```
# Αύξουσα ταξινόμηση ως προς το κόστος, άρα βάζω πρώτη τη λίστα KOSTOS
```

```
bb2(KOSTOS,AK)
```

```
print "Τα οχήματα με το μεγαλύτερο κόστος είναι"
```

```
print AK[-1],KOSTOS[-1]
```

```
print AK[-2],KOSTOS[-2]
```

```
print AK[-3],KOSTOS[-3]
```

```
# Δ6 Θα κάνω απλά pop() και από τις δύο λίστες 3 φορές.
```

```
for i in range(3):
```

```
    AK.pop()
```

```
    KOSTOS.pop()
```

```
print "Αυτοκίνητα που θα παραμείνουν στο Δήμο: "
```

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

```
    print AK[i], KOSTOS[i]
```