

ΑΡΧΗ 1ΗΣ ΣΕΛΙΔΑΣ

ΠΡΟΤΕΙΝΟΜΕΝΑ ΘΕΜΑΤΑ 2

ΛΥΣΕΙΣ

A1.

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

A2.

1. β
2. δ
3. στ
4. ε
5. α

P	Q	S	(P AND Q) OR S	P OR (Q AND (NOTS))
True	True	True	True	True
True	False	False	False	True
False	True	True	True	False
False	False	False	False	False

A4.

Δομή Ακολουθίας, Επιλογής και Επανάληψης.

B1

α.

S= 10 i= 10

S= 18 i= 8

S= 24 i= 6

S= 28 i= 4

ΤΕΛΟΣ 1ΗΣ ΑΠΟ 6 ΣΕΛΙΔΕΣ

ΑΡΧΗ 2ΗΣ ΣΕΛΙΔΑΣ

β.
x=4
S=0
i=10
while i>3:
 S=S+i
 x=S%i
 print "S=",S,
 print " i=",i
 i=i-2

B2

α. Ο κατασκευαστής είναι:

```
def __init__(self, eponymo,onoma,misthos):  
  
    self.eponymo=eponymo  
    self.onoma=onoma  
    self.misthos=misthos
```

β. Οι ιδιότητες είναι :

eponymo,onoma,misthos

γ. Οι μέθοδοι είναι:

```
def afksisemistho(self,amount):  
  
    self.misthos=self.misthos+amount  
  
def emfanise_stoixeia(self):  
    print "Επώνυμο ",self.eponymo  
    print "Όνομα ",self.onoma  
    print "Μισθός ",self.misthos
```

δ. Έχουμε μόνο αλλαγή στο:

```
def __init__(self, eponymo, onoma, misthos,xronia):  
  
    self.onoma=onoma  
    self.eponymo=eponymo  
    self.misthos=misthos  
    self.xronia=xronia
```

ΤΕΛΟΣ 2ΗΣ ΑΠΟ 6 ΣΕΛΙΔΕΣ

ΑΡΧΗ 3ΗΣ ΣΕΛΙΔΑΣ

ε.

```
def afxisexronia(self,amount):
```

```
    self.xronia=self.xronia+amount
```

στ.

```
I. yp1= ypallilos("ΑΝΤΩΝΙΟΥ","ΑΝΤΩΝΙΟΣ",1500,15)
```

```
II. yp2= ypallilos("ΠΑΠΑΔΟΠΟΥΛΟΥ","ΕΛΕΝΗ",2000,20)
```

```
z. yp2. afksisemistho(200)
```

γ.

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

```
SUMK=0.0 #Συνολικό ποσό κρατήσεων
```

```
#Μικρότερο ποσό κρατήσεων στην αρχή βάζω μία πολύ μεγάλη τιμή
```

```
MINK=10000
```

```
MINON=""
```

```
#3 μετρητές για 3 κατηγορίες υπαλλήλων
```

```
m1=m2=m3=0
```

```
for i in range(30):
```

```
    onep=raw_input("Δώσε το ονοματεπώνυμο")
```

```
    bm=float(input("Δώσε το βασικό μισθό"))
```

```
    #Έλεγχος ορθότητας
```

```
    while not(bm>=1 and bm<=10000): # ή bm<1 or bm>10000
```

```
        bm=float(input("Δώσε το βασικό μισθό"))
```

```
    if bm>=1 and bm<=1000: #θα μπορούσε να είναι και bm<=1000, μας καλύπτει ο έλεγχος ορθότητας
```

```
        pk=bm*10/100
```

```
        #Υπολογισμός ποσού κρατήσεων
```

```
        m1=m1+1
```

```
    elif bm>1000 and bm<=2000:
```

```
        pk=bm*15/100
```

```
        m2=m2+1
```

```
    elif bm>2000: #θα μπορούσε να είναι και else μας καλύπτει ο έλεγχος ορθότητας
```

ΤΕΛΟΣ 3ΗΣ ΑΠΟ 6 ΣΕΛΙΔΕΣ

ΑΡΧΗ 4ΗΣ ΣΕΛΙΔΑΣ

```
pk=bm*20/100
m3=m3+1
```

```
#Υπολογισμός τελικού μισθού
tm=bm-pk
#Εμφάνιση αποτελεσμάτων:
print "Ο υπάλληλος ",onep," έχει ποσό κρατήσεων ",pk,"και τελικο μισθό ",tm
```

```
SUMK=SUMK+pk
#Εύρεση ελαχίστου ποσού κρατήσεων και ονοματεπώνυμου που το έχει
if pk<MINK:
    MINK=pk
    MINON=onep
```

```
#ΜΟ κρατήσεων
MOK=SUMK/30
```

```
print "Σύνολο κρατήσεων",SUMK
print "Μέσος όρος κρατήσεων",MOK
```

```
print "Ο Υπάλληλος ",onep,"έχει τις ελάχιστε κρατήσεις που είναι ",MINK
```

```
print "Στην κατηγορία με βασικό μισθό από 1-1000 έχουμε",m1,'υπαλλήλους'
print "Στην κατηγορία με βασικό μισθό πάνω από 1000-2000 έχουμε",m2,'υπαλλήλους'
print "Στην κατηγορία με βασικό μισθό πάνω από 2000 έχουμε",m3,'υπαλλήλους'
```

```
Δ.
# **-coding:cp1253 -*-
```

```
#Ακολουθεί η bubble sort για δύο παράλληλες λίστες
```

```
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]
```

```
#-----
TYPOS=[]
HMERES=[]
```

ΤΕΛΟΣ 4ΗΣ ΑΠΟ 6 ΣΕΛΙΔΕΣ

ΑΡΧΗ 5ΗΣ ΣΕΛΙΔΑΣ

```

#Σύνολο - Άθροισμα ημερών για κάθε τύπο
sumA=sumB=sumG=0

typos=raw_input("Δώσε τον τύπο του ιστιοπλοϊκού")
while typos!="TELOS":
    h=int(input("Δώσε τις ημέρες που ενοικιάστηκε")) # int γιατί έχω ολόκληρες ημέρες

    #ke= κόστος ενοικίασης
    if typos=="A":
        ke=h*200
        #Αύξησε το σύνολο ημερών του A κατά h
        sumA=sumA+h

    elif typos=="B":
        ke=h*300
        #Αύξησε το σύνολο ημερών του B κατά h
        sumB=sumB+h

    elif typos=="Γ":
        ke=h*400
        #Αύξησε το σύνολο ημερών του Γ κατά h
        sumG=sumG+h

    print "Το κόστος ενοικίασης είναι",ke

    TYPOS.append(typos)
    HMERES.append(h)

    typos=raw_input("Δώσε τον τύπο του ιστιοπλοϊκού")

print "Το σύνολο ημερών που ενοικιάστηκε ο κάθε τύπος είναι "
print "A",sumA
print "B",sumB
print "Γ",sumG

#εγγράφει σε ένα αρχείο synola.txt τα στοιχεία του Δ2
f=open("synola.txt","w") # Με w γιατί πρώτη φορά το δημιουργούμε
f.write("A"+" "+str(sumA)+"\n")
# Το sumA πρέπει να γίνει κείμενο και το \n για αλλαγή γραμμής
f.write("B"+" "+str(sumB)+"\n")
f.write("Γ"+" "+str(sumG)+"\n")

```

ΤΕΛΟΣ 5ΗΣ ΑΠΟ 6 ΣΕΛΙΔΕΣ

ΑΡΧΗ 6ΗΣ ΣΕΛΙΔΑΣ

```
f.close()
```

```
#Αύξουσα ταξινόμηση ως προς τον τύπο  
bb2(TYPOS,HMERES)
```

```
f=open("synola.txt","a") # Με a για προσθήκη στο τέλος  
for i in range(len(TYPOS)):  
    f.write(TYPOS[i]+" "+str(HMERES[i])+"\n")  
f.close()
```

ΤΕΛΟΣ 6ΗΣ ΑΠΟ 6 ΣΕΛΙΔΕΣ