

ΦΥΛΛΑΔΙΟ 3 ΚΕΦ5 - 6 – 7 – 8 ΛΥΣΕΙΣ

1)

-*- coding: cp1253 -*-

Η παραπάνω εντολή είναι για να βγάξει σωστά τα ελληνικά

#(α)

```
EP=[]
MIS=[]
for i in range(30):
    epo=raw_input("Δώσε το ΕΠώνυμο")
    m=float(input("Δώσε το μισθό"))
    EP.append(epo)
    MIS.append(m)
```

#(β)

```
SUM=0.0
for i in range(len(MIS)):
    SUM=SUM+MIS[i]
MO=SUM/len(MIS)
print " Ο ΜΟ είναι ", MO
```

#(γ)

```
MAX=MIS[0]
MAXONOMA=EP[0]
for i in range(len(MIS)):
    if MIS[i]>MAX:
        MAX=MIS[i]
        MAXONOMA=EP[i]
print "Ο υπάλληλος ", MAXONOMA, " έχει το μέγιστο μισθό που είναι ", MAX
```

#(δ)

```
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(MIS,EP)
```

Εμφάνιση

```
for i in range(len(MIS)):
    print "Ο υπάλληλος ", EP[i], " έχει μισθό ", MIS[i]
```

#(ε)

```
print "Οι υπάλληλοι που έχουν μισθό πάνω από το ΜΟ είναι:"
for i in range(len(MIS)):
    if MIS[i]>MO:
        print EP[i]
```

#(στ)

```
print "Τους μικρότερους μισθούς έχουν οι υπάλληλοι"
print "ο", EP[0], " με μισθό ", MIS[0]
print "ο", EP[1], " με μισθό ", MIS[1]
print "ο", EP[2], " με μισθό ", MIS[2]
```

2)

-*- coding: cp1253 -*-

Η παραπάνω εντολή είναι για να βγάλει σωστά τα ελληνικά

#(α)

AM=[]

BA=[]

for i in range(20):

 am=int(input("Δώσε τον αριθμό μητρώου του μαθητή"))

 ba=float(input("Δώσε το βαθμό του μαθητή στο μάθημα της ιστορίας"))

 AM.append(am)

 BA.append(ba)

#(β)

SUM=0.0

for i in range(len(BA)):

 SUM=SUM+BA[i]

MO=SUM/len(BA)

print "Ο μέσος όρος είναι ",MO

#(γ)

m1=0

for i in range(len(BA)):

 if BA[i]<MO:

 m1=m1+1

print "Οι μαθητές που είχαν χαμηλότερο βαθμό από το MO της τάξης είναι ", m1

#(δ)

def bubbleSortF(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]

bubbleSortF(BA,AM)

for i in range(len(BA)):

 print "Ο μαθητής με τον AM",AM[i], "έχει βαθμό",BA[i]

#(ε)

def binarySearch(A,key):

 N=len(A)

 first=0

 last=N-1

 pos=-1

 while first<=last and pos==-1:

 mid=(first+last)/2

 if A[mid]==key:

 pos=mid

 elif A[mid]>key:

 last=mid-1

 else:

 first=mid+1

 return pos

thesi=binarySearch(BA,14)

print "Μαθητής που έχει το βαθμό 14 είναι αυτό με AM= ",AM[thesi]

3)

-*- coding: cp1253 -*-

Η παραπάνω εντολή είναι για να βγάξει σωστά τα ελληνικά

#(α)

```
f=open("poleis.txt","r")
```

```
POL=[]
```

```
for line in f:
```

#Κανονικά θα έγραφα POL.append(line) επειδή όμως παίρνει και την αλλαγή γραμμής θα πρέπει να αφαιρέσω τον τελευταίο χαρακτήρα

άρα γράφω από τον χαρακτήρα 0 μέχρι και τον χαρακτήρα len(line) -1

#Μην μπερδευτείτε από αυτό αφαιρώ μόνο τον χαρακτήρα \n που έχει κάθε γραμμή αρχείου στο τέλος της

```
    POL.append(line[0:len(line)-1])
```

```
f.close()
```

#(β)

```
for i in range(len(POL)-1,-1,-1):
```

```
    print POL[i]
```

#(γ)

```
min_gram=1000
```

```
min_poli=""
```

```
for item in POL:
```

```
    if len(item)<min_gram:
```

```
        min_gram=len(item)
```

```
        min_poli=item
```

```
print "Η πόλη ", min_poli, " έχει τα λιγότερα γράμματα που είναι ",min_gram
```

#(δ)

```
max_gram=0
```

```
max_poli=""
```

```
for item in POL:
```

```
    if len(item)>max_gram:
```

```
        max_gram=len(item)
```

```
        max_poli=item
```

```
print "Η πόλη ", max_poli, " έχει τα περισσότερα γράμματα που είναι ",max_gram
```

#(ε)

```
count=0
```

```
for item in POL:
```

```
    if ("T" in item) or ("τ" in item):
```

```
        count=count+1
```

```
print "Οι πόλεις που περιέχουν το γράμμα Τ ή τ είναι ",count
```

4)

-*- coding: cp1253 -*-

Η παραπάνω εντολή είναι για να βγάξει σωστά τα ελληνικά

#(γ)

```
EP=[]
```

```
ON=[]
```

```

BA=[]
#(α)

for i in range(20):
    ep=raw_input("Δώσε το επώνυμο")
    on=raw_input("Δώσε το όνομα")
    #(β)
    b=input("Δώσε το βαθμό")
    while b<0 or b>100:
        b=input("Δώσε το βαθμό")

    #(γ)
    EP.append(ep)
    ON.append(on)
    BA.append(b)

#(δ)
SUM=0.0
for i in range(len(BA)):
    SUM=SUM+BA[i]
MO=SUM/20
print "MO=",MO

#(ε)
print "Πάνω από το MO πήραν οι μαθητές"
for i in range(len(BA)):
    if BA[i]>MO:
        print EP[i],ON[i]

#(στ)
MAX=-1
for item in BA:
    if item>MAX:
        MAX=item
print "Ο μεγαλύτερος βαθμός της τάξης είναι",MAX

#(ζ)
print "Αυτό το βαθμό τον έχουν οι μαθητές"
for i in range(len(BA)):
    if BA[i]==MAX:
        print EP[i],ON[i]

```

5)
-*- coding: cp1253 -*-
Η παραπάνω εντολή είναι για να βγάλει σωστά τα ελληνικά

```

#(α)
ON=[]

#(β)
NIKES=[]
HTTES=[]

#(δ)
SYN=[]

```

#(α)

```
on=raw_input("Δώσε το όνομα")
```

```
while on!="TELOS":
```

```
    ON.append(on)
```

```
    n=int(input("Δώσε τις νίκες"))
```

```
    h=int(input("Δώσε τις ήττες"))
```

```
    NIKES.append(n)
```

```
    HTTES.append(h)
```

#(γ)

```
sb=n*2+h*1
```

```
print "Η ομάδα ",on," έχει ",sb," βαθμούς "
```

#(δ)

```
SYN.append(sb)
```

#(α)

```
on=raw_input("Δώσε το όνομα")
```

#(ε)

```
def bubbleSort(A,B,C,D):
```

```
    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]
```

```
                C[j],C[j-1]=C[j-1],C[j]
```

```
                D[j],D[j-1]=D[j-1],D[j]
```

```
bubbleSort(SYN,ON,NIKES,HTTES)
```

#(στ)

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

```
    print "Η ομάδα",ON[i], " έχει ",NIKES[i]," νίκες και ",HTTES[i]," ήττες και συνολική βαθμολογία
```

```
",SYN[i]
```

#(ζ)

```
print "Κάτω από 4 νίκες έχουν οι ομάδες:"
```

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

```
    if NIKES[i]<4:
```

```
        print ON[i]
```

#(η)

```
f=open("apotelesmata.txt","w")
```

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

```
    f.write("Η ομάδα "+ON[i]+ " έχει "+str(NIKES[i])+" νίκες και "+str(HTTES[i])+" ήττες και συνολική  
βαθμολογία "+str(SYN[i])+"\n")
```

```
f.close()
```

6)

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

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

#(α)

```
WRES=range(1,25)
```

#(γ)

THERM=[]

#(δ)

SUM=0.0

#(β)

for i in range(24):

**# Το παρακάτω το κάνω για να μου δώσει τη θερμοκρασία και να αναγράφεται και η ώρα
#Θα μπορούσα να έβαζα απλά "Δώσε τη θερμοκρασία" και θα ήταν ολόσωστο**

th=float(input("Δώσε τη θερμοκρασία για την ώρα "+str(WRES[i])))

#(γ)

THERM.append(th)

#(δ)

SUM=SUM+th

#(δ)

MO=SUM/24.0

print "Η μέση θερμοκρασία της ημέρας είναι ",MO

#(ε)

def bubblesort2(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]

bubblesort2(THERM,WRES)

for i in range(len(THERM)):

print "Την ώρα ", WRES[i], "είχαμε θερμοκρασία ", THERM[i]

(στ) Αφού οι λίστες ταξινομήθηκαν κατά φθίνουσα σειρά οι τρεις πρώτες τιμές θα είναι και οι μεγαλύτερες

print "Οι τρεις μεγαλύτερες θερμοκρασίες είναι:"

for i in range(0,3):

print "Την ώρα ", WRES[i], "είχαμε θερμοκρασία ", THERM[i]

7)

-*- coding: cp1253 -*-

Η παραπάνω εντολή είναι για να βγάξει σωστά τα ελληνικά

#(α)

EP=[]

ON=[]

THL=[]

ep=raw_input("Δώσε το επώνυμο")

while ep!="TELOS":

on=raw_input("Δώσε το ;όνομα")

#Το τηλέφωνο εφόσον δεν συμμετέχει σε πράξεις μπορώ να το διαβάζω ως κείμενο

#ώστε να κρατάει και τα αρχικά μηδενικά

thl=raw_input("Δώσε το τηλέφωνο")

EP.append(ep)

ON.append(on)

THL.append(thl)

```
ep=raw_input("Δώσε το επώνυμο")
```

β)

```
def bubblesort3(A,B,C):
    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]
                C[j],C[j-1]=C[j-1],C[j]

bubblesort3(EP,ON,THL)
```

γ)

```
for i in range(len(EP)):
    print "Ο μαθητής ", EP[i], ON[i]," έχει τηλέφωνο ",THL[i]
```

δ)

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

ε)

```
ep1=raw_input("Δώσε το επώνυμο")
pos=binarySearch(EP,ep1)
if pos==-1:
    print "Δεν υπάρχει"
else:
    print "Το τηλέφωνο του μαθητή ",ep1, ' είναι ', THL[pos]
```

στ)

```
def bubblesort3(A,B,C):
    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]
                C[j],C[j-1]=C[j-1],C[j]
            if A[j]==A[j-1]:
                if B[j]<B[j-1]:
                    A[j],A[j-1]=A[j-1],A[j]
                    B[j],B[j-1]=B[j-1],B[j]
                    C[j],C[j-1]=C[j-1],C[j]

bubblesort3(EP,ON,THL)
```

```

for i in range(len(EP)):
    print "Ο μαθητής ", EP[i], ON[i]," έχει τηλέφωνο ",THL[i]
8)
# -*- coding: cp1253 -*-
# Η παραπάνω εντολή είναι για να βγάξει σωστά τα ελληνικά

 #(α)
AR=[]

 #(γ)
POSO=[]
for i in range(20):
    #Ο αριθμός κυκλοφορίας είναι κείμενο αφού περιέχει και γράμματα
    ar=raw_input("Δώσε τον αριθμό κυκλοφορίας")
    AR.append(ar)
     #(β)
    pal=input("Δώσε την παλαιότητα του αυτοκινήτου")
    ke=input("Δώσε τα κυβικά εκατοστά ")
    if pal>=1 and pal<=10:
        if ke<1000:
            poso=90
        elif ke>=1000 and ke<=1999:
            poso=130
        elif ke>1999:
            poso=190
    elif pal>10:
        if ke<1000:
            poso=120
        elif ke>=1000 and ke<=1999:
            poso=240
        elif ke>1999:
            poso=300

     #(γ)
    POSO.append(poso)

 #(δ)
SUM=0.0
for i in range(len(POSO)):
    SUM=SUM+POSO[i]
print "Το συνολικό ποσό που πρέπει να πληρώσει η εταιρεία είναι ",SUM

 #(ε)
MIN=POSO[0]
for i in range(1,len(POSO)):
    if POSO[i]<MIN:
        MIN=POSO[i]
print "Το μικρότερο ποσό που πρέπει να πληρώσει η εταιρεία είναι ", MIN

 #(στ)
print "Τα οχήματα που θα πληρώσουν ", MIN, "€ είναι : "
for i in range(len(POSO)):
    if POSO[i]==MIN:
        print AR[i]

 #(ζ)
for i in range(len(AR)):
    arithmos=AR[i]

```

```
if arithmos[0:3]=='KZN':
    print AR[i], POSO[i]
```

9)

*- coding: cp1253 -*-

Η παραπάνω εντολή είναι για να βγάξει σωστά τα ελληνικά

#(α)

EP=[]

GL=[]

MA=[]

ep=raw_input("Δώσε το επώνυμο")

#(β)

while ep!="TELOS":

glo=float(input("Δώσε το βαθμό στη Γλώσσα"))

math=float(input("Δώσε το βαθμό στα Μαθηματικά"))

EP.append(ep)

GL.append(glo)

MA.append(math)

#(β)

ep=raw_input("Δώσε το επώνυμο")

#(γ)

SUMGL=0

SUMMA=0

for i in range(len(EP)):

SUMGL=SUMGL+GL[i]

SUMMA=SUMMA+MA[i]

MOGL=SUMGL/len(GL)

MOMA=SUMMA/len(MA)

print "Ο μέσος όρος στη γλώσσα είναι ",MOGL

print "Ο μέσος όρος στα μαθηματικά είναι ",MOMA

#(δ)

MESOS=[]

for i in range(len(EP)):

mesosoros=(GL[i]+MA[i])/2.0

MESOS.append(mesosoros)

print "Ο/Η διαγωνιζόμενος/νη ", EP[i], " έχει μέσο όρο από τα δύο μαθήματα ",mesosoros

#(ε)

m1=0

for i in range(len(EP)):

if MESOS[i]>10:

m1=m1+1

print "Πάνω από 10 μέσο όρο έχουν ", m1," διαγωνιζόμενοι/νες"

#(στ)

m2=0

for i in range(len(EP)):

if MA[i]>GL[i]:

m2=m2+1

print "Έγραψαν καλύτερα στα Μαθηματικά απ'ότι στη Γλώσσα ", m1," διαγωνιζόμενοι/νες"

10)

-*- coding: cp1253 -*-

Η παραπάνω εντολή είναι για να βγάλει σωστά τα ελληνικά

#(α)

```
def countS(word):
    symfona="ΒΓΔΖΘΚΛΜΝΞΠΡΣΤΦΧΨβγδζθκλμνξπρστφχψ"
    count=0
    for letter in word:
        if letter in symfona:
            count=count+1
    return count
```

#(β)

```
def bubbleSort(A):
    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]
```

#(γ)

```
WORDS=[]
word=raw_input("Δώσε τη λέξη")
while word!="TELOS":

    WORDS.append(word)
    word=raw_input("Δώσε τη λέξη")
```

#(δ)

```
MAX=countS(WORDS[0]) # Θα μπορούσε να είναι και MAX=-1
MAXLEXI=WORDS[0] #Θα μπορούσε να είναι και " "
for i in range(len(WORDS)):
    arithmos_symfonwn=countS(WORDS[i])
    if arithmos_symfonwn>MAX:
        MAX=arithmos_symfonwn
        MAXLEXI=WORDS[i]
print "Η λέξη με τα περισσότερα σύμφωνα είναι ",MAXLEXI," και έχει ",MAX," σύμφωνα"
```

#(ε)

```
bubbleSort(WORDS)

for i in range(len(WORDS)):
    print WORDS[i]
```

#(στ)

```
f=open("lexeis.txt","w")
m1=1
for i in range(len(WORDS)):
    f.write(str(m1)+"."+WORDS[i]+"\\n")
    m1=m1+1

f.close()
```

11)

-*- coding: cp1253 -*-
Η παραπάνω εντολή είναι για να βγάλει σωστά τα ελληνικά

#(α)

```
BA=[]
FY=[]
f=raw_input("Δώσε το φύλο")
```

#(β)

```
while f!="ΤΕΛΟΣ":
    b=float(input("Δώσε το βαθμό"))
    FY.append(f)
    BA.append(b)
    f=raw_input("Δώσε το φύλο")
```

#(γ)

```
BAA=[]
BAK=[]
for i in range(len(BA)):
    if FY[i]=="Κ":
        BAK.append(BA[i])
    else:
        BAA.append(BA[i])
```

```
print "Οι βαθμοί των αγοριών ",BAA
print "Οι βαθμοί των κοριτσιών ",BAK
```

#(δ)

```
def bubbleSort(A):
    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]
```

```
bubbleSort(BAA)
bubbleSort(BAK)
print " Ο μέγιστος βαθμός των αγοριών είναι ο ",BAA[-1]," και ο ελάχιστος ο ",BAA[0]
print " Ο μέγιστος βαθμός των κοριτσιών είναι ο ",BAK[-1]," και ο ελάχιστος ο ",BAK[0]
```

12)

-*- coding: cp1253 -*-
Η παραπάνω εντολή είναι για να βγάλει σωστά τα ελληνικά

#(α)

```
ON=[]
NOM=[]
PL=[]
on=raw_input("ΠΟΛΗ ή ΧΩΡΙΟ: ")
```

#(β)

```
while on!="ΤΕΛΟΣ":
    nomos=raw_input("ΝΟΜΟΣ: ")
```

```
pl=int(input("ΠΛΗΘΥΣΜΟΣ: "))
ON.append(on)
NOM.append(nomos)
PL.append(pl)
on=raw_input("ΠΟΛΗ ή ΧΩΡΙΟ: ")
```

#(γ)

```
print "Χωριά ή Πόλεις με πληθυσμό πάνω από 4000 κατοίκους"
for i in range(len(ON)):
    if PL[i]>4000:
        print ON[i],NOM[i]
```

#(δ)

```
SUM=0
for i in range(len(ON)):
    SUM=SUM+PL[i]
print "Ο συνολικός πληθυσμός είναι ",SUM
```

#(ε)

```
def bubbleSort(A,B,C):
    N=len(A)
    for i in range(N-1):
        for j in range(N-1,i,-1):
            #Αλλαγή φοράς > για φθίνουσα ταξινόμηση
            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]
                C[j],C[j-1]=C[j-1],C[j]
```

```
bubbleSort(PL,ON,NOM)
print "Οι 3 πόλεις με το μεγαλύτερο πληθυσμό είναι :"
```

```
print "ΟΝΟΜΑ: ",ON[0]," ΝΟΜΟΣ: ", NOM[0]," ΠΛΗΘΥΣΜΟΣ:", PL[0]
print "ΟΝΟΜΑ: ",ON[1]," ΝΟΜΟΣ: ", NOM[1]," ΠΛΗΘΥΣΜΟΣ:", PL[1]
print "ΟΝΟΜΑ: ",ON[2]," ΝΟΜΟΣ: ", NOM[2]," ΠΛΗΘΥΣΜΟΣ:", PL[2]
```

#(στ)

```
f=open("thess.txt","w")
for i in range(len(ON)):
    if NOM[i]=="ΘΕΣΣΑΛΟΝΙΚΗΣ":
        f.write(ON[i]+" "+str(PL[i])+"\n")
f.close()
```

#(ζ)

```
for i in range(len(ON)):
    ONOMA=ON[i]
    if ONOMA[0:3]=="NEA":
        print ON[i], PL[i]
```