

ΤΕΣΤ ΠΡΟΣΟΜΟΙΩΣΗΣ

ΘΕΜΑ Α

A1. Να γράψετε στο τετράδιό σας τον αριθμό καθεμιάς από τις παρακάτω προτάσεις **1-5** και δίπλα τη λέξη **Σωστό**, αν είναι σωστή, ή τη λέξη **Λάθος**, αν είναι λανθασμένη.

1. Η σειριακή αναζήτηση χρησιμοποιείται αποκλειστικά στις ταξινομημένες λίστες. Λ
2. Η εντολή επανάληψης for μπορεί να χρησιμοποιηθεί, όταν έχουμε άγνωστο αριθμό επαναλήψεων. Λ
3. Τα στοιχεία μιας λίστας μπορούν να αποτελούνται από δεδομένα διαφορετικού τύπου. Σ
4. Μία συνάρτηση μπορεί να καλείται από μία άλλη συνάρτηση ή από το κύριο πρόγραμμα. Σ
5. Κατά την εκτέλεση ενός προγράμματος μπορεί να αλλάζει η τιμή και ο τύπος μιας μεταβλητής Σ

Μονάδες 5

A2. Εντοπίστε το λάθος στον πιο κάτω κώδικα

```
S=0
for i in range(2,10,0):
    S=S+i
print S
```

Μονάδες 5

A3. Δίνεται το παρακάτω πρόγραμμα και η συνάρτηση:

```
def func(b,a,c):
    a=a+2
    b=b-3
    c=a+b
    print a,b,c

a=input("A:")
b=input("B:")
c=input("C:")
func(a,b,c)
print a,b,c
```

Τι θα εμφανιστεί κατά την εκτέλεση του προγράμματος, αν ως τιμές εισόδου δοθούν οι αριθμοί 5, 7, 10;

Μονάδες 6

Απάντηση

9 2 11

5 7 10

A4. Να γράψετε στο τετράδιό σας τους αριθμούς της **Στήλης Α** και δίπλα το γράμμα της **Στήλης Β** που αντιστοιχεί σωστά. Στη **Στήλη Β** υπάρχει ένα επιπλέον στοιχείο.

Στήλη Α	Στήλη Β
1. True	α. λογικός τελεστής
2. and	β. μεταβλητή
3. $a > 12$	γ. αλφαριθμητική σταθερά
4. arithmos_paidion	δ. λογική τιμή
5. $\leq$	ε. συγκριτικός τελεστής
	στ. συνθήκη

Μονάδες 5

1-δ, 2-α, 3-στ, 4-β, 5-ε

A5. Αν $a = 5$, $b = 7$ και $c = 10$, να χαρακτηρίσετε στο τετράδιό σας τις παρακάτω προτάσεις ως True ή False.

- 1.** `(not (a + 2 >= b)) or b + 3 == c`
- 2.** `a + 2 * b < 20 and 2 * a == c`

Μονάδες 4

1. True, 2.True

ΘΕΜΑ Β

B1. Δίνεται το παρακάτω τμήμα προγράμματος Python:

```
for x in range (A, M, B):  
    print x
```

Για καθεμιά από τις παρακάτω περιπτώσεις, να γράψετε στο τετράδιό σας τις τιμές των **A**, **M**, **B**, έτσι ώστε το αντίστοιχο τμήμα προγράμματος να εμφανίζει όλους :

- α.** τους ακέραιους από 1 μέχρι και 80 (αύξουσα σειρά)
- β.** τους ακέραιους από 50 μέχρι και 20 (φθίνουσα σειρά)
- γ.** τους περιττούς ακέραιους από 81 μέχρι και 151 (αύξουσα σειρά)
- δ.** τους ακέραιους από -50 μέχρι και -5 (αύξουσα σειρά)
- ε.** τους θετικούς ακέραιους που είναι μικρότεροι του 200 και πολλαπλάσιοι του 7 (αύξουσα σειρά).

Μονάδες 10

- α.** A=1 M=81 B=1
- β.** A=50 M=19 B=-1
- γ.** A=81 M=152 B=2

δ. A=-50 M=-4 B=1
ε. A=7 M=200 B=7

B2. Δίνεται το παρακάτω τμήμα προγράμματος σε γλώσσα προγραμματισμού Python που υλοποιεί ταξινόμηση ευθείας ανταλλαγής (bubble sort) στα στοιχεία της λίστας lista κατά φθίνουσα σειρά:

```
N=len(lista)
for i in range(1,_(1)_,_(2)_):
    for j in range(N-1,_(3)_,-1):
        if lista[_(4)_] > lista[j-1]:
            lista[j], lista[j-1] = lista[j-1], lista[j]
```

Στο τμήμα προγράμματος υπάρχουν υπογραμμισμένα κενά τα οποία έχουν αριθμηθεί. Να γράψετε στο τετράδιό σας τους αριθμούς **1, 2, 3, 4** και **5** που αντιστοιχούν στα κενά του παραπάνω τμήματος προγράμματος και δίπλα σε κάθε αριθμό αυτό που πρέπει να συμπληρωθεί ώστε να υλοποιείται σωστά η ταξινόμηση.

Μονάδες 10

1. N, 2. 1, 3. i -1, 4. j, 5. >

B3. Δίνεται η επόμενη λίστα **A** :

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
A:	1	1	2	3	5	8	13	21	34	55	89	94	96	99

Να γράψετε στο τετράδιό σας τους αριθμούς της λίστας που θα συγκριθούν κατά την εκτέλεση του αλγορίθμου της δυαδικής αναζήτησης σε κάθε μία από τις παρακάτω περιπτώσεις:

- α) για τον αριθμό 100 (μον. 4)
β) για τον αριθμό 1 (μον. 3)

Μονάδες 5

α. 13, 89, 96, 99
β. 13, 2, 1

ΘΕΜΑ Γ

Ένας συλλέκτης γραμματοσήμων επισκέπτεται στο διαδίκτυο το αγαπημένο του ηλεκτρονικό κατάστημα φιλοτελισμού προκειμένου να αγοράσει γραμματόσημα. **Προτίθεται να ξοδέψει μέχρι 1500 ευρώ.**

Να γράψετε πρόγραμμα το οποίο:

α. Για κάθε γραμματόσημο, να διαβάζει την τιμή και την προέλευσή του (ελληνικό/ξένο) και να επιτρέπει την αγορά του, εφόσον η τιμή του δεν υπερβαίνει το διαθέσιμο υπόλοιπο χρημάτων. Διαφορετικά να τερματίζει τυπώνοντας το μήνυμα «ΤΕΛΟΣ ΑΓΟΡΩΝ».

ΣΗΜΕΙΩΣΗ: Δεν απαιτείται έλεγχος εγκυρότητας για τα δεδομένα εισόδου.

Μονάδες 10

β. Να τυπώνει:

1. Το συνολικό ποσό που ξόδεψε ο συλλέκτης.

Μονάδες 2

2. Το πλήθος των ελληνικών και το πλήθος των ξένων γραμματοσήμων που αγόρασε.

Μονάδες 4

3. Το ποσό που περίσσεψε, εφόσον υπάρχει, διαφορετικά το μήνυμα «ΕΞΑΝΤΛΗΘΗΚΕ ΟΛΟ ΤΟ ΠΟΣΟ».

Μονάδες 4

grammatosima.py

```
diathesimo=1500
exoda=0
k1=0
k2=0

print "Αγορά νέου γραμματοσήμου"
timi=input("Δώσε την τιμή του γραμματοσήμου: ")
while timi<=diathesimo :
    proeleysi=raw_input("Δώσε προέλευση (Gr για Ελληνικό/Χη για ξένο): ")

    if proeleysi=='Gr' :
        k1 += 1
    else :
        k2 += 1

    exoda+=timi
    diathesimo -= timi

    print "Αγορά νέου γραμματοσήμου"
    timi=input("Δώσε την τιμή του γραμματοσήμου: ")

print "ΤΕΛΟΣ ΑΓΟΡΩΝ"

print "Συνολικό ποσό που ξόδεψε ο συλλέκτης:",exoda
print "Πλήθος Ελληνικών γραμματοσήμων που αγόρασε:",k1
print "Πλήθος ξένων γραμματοσήμων που αγόρασε:",k2

if diathesimo >0 :
    print "Ποσό που περισεψε:",diathesimo
else :
    print "ΕΞΑΝΤΛΗΘΗΚΕ ΟΛΟ ΤΟ ΠΟΣΟ"
```

ΘΕΜΑ Δ

Το Πανελλήνιο Σχολικό Δίκτυο παρέχει πρόσβαση στο Διαδίκτυο (Ίντερνετ) σε 150.000 μαθητές και διατηρεί τα στοιχεία τους, καθώς και στατιστικά στοιχεία, σχετικά με την πρόσβασή τους στο Διαδίκτυο. Να κατασκευάσετε πρόγραμμα το οποίο:

Δ1. Για κάθε μαθητή να διαβάζει:

α) τον αλφαριθμητικό κωδικό του και τον καταχωρεί στη λίστα **CODE**

β) το φύλο του, «**A**» αν είναι αγόρι και «**K**» αν είναι κορίτσι, και τον καταχωρεί στη λίστα **FYLO**

γ) τον συνολικό χρόνο πρόσβασής του στο Διαδίκτυο, για ένα έτος, και τον καταχωρεί στη λίστα **TIME**.

Δ2. Να εμφανίζει τον **κωδικό του αγοριού** με το μεγαλύτερο συνολικό χρόνο πρόσβασης και, στη συνέχεια, τον **κωδικό του κοριτσιού** με το μεγαλύτερο συνολικό χρόνο πρόσβασης, καλώντας τη συνάρτηση **maxPos()**, που περιγράφεται στο ερώτημα **Δ3**, **μία φορά για τα αγόρια** και **μία για τα κορίτσια**.

Δ3. Να αναπτύξετε συνάρτηση **maxPos()** η οποία:

α) να δέχεται ως παραμέτρους: την λίστα **FYLO**, τη λίστα **TIME** και τον χαρακτήρα «**A**» ή «**K**» που αντιστοιχεί στο φύλο

β) να βρίσκει τη **θέση** της μέγιστης τιμής του ετήσιου χρόνου πρόσβασης αγοριών ή κοριτσιών, ανάλογα με την τιμή «**A**» ή «**K**» του φύλου και

γ) να επιστρέφει τη θέση της μέγιστης τιμής

(Σημείωση: Δεν απαιτείται έλεγχος εγκυρότητας. Να θεωρήσετε ότι όλες οι εισαγωγές γίνονται σωστά και όλες οι συνολικές τιμές χρόνου πρόσβασης είναι μοναδικές).

#Λύση για 5 μαθητές

CODE=[]

FYLO=[]

TIME=[]

for i in range(5) :

 c=raw_input("Code: ")

 f=raw_input("Sex: ")

 t=input("Time: ")

 CODE.append(c)

 FYLO.append(f)

 TIME.append(t)

print CODE

print FYLO

print TIME

#CODE=["M1","M2","M3","M4","M5"]

#FYLO=["K","K","A","K","A"]

#TIME=[200,500,700,400,300]

def maxPos(F,T,char):

 MAX=0

 for i in range(5) :

 if F [i]==char:

 if T [i]>MAX:

 MAX=T [i]

 pos=i

return pos

posA=maxPos(FYLO,TIME,"A")

posK=maxPos(FYLO,TIME,"K")

print CODE[posA]

print CODE[posK]

Έξοδος

Code: M1

Sex: K

Time: 200

Code: M2

Sex: K

Time: 500

Code: M3

Sex: A

Time: 700

Code: M4

Sex: K

Time: 400

Code: M5

Sex: A

Time: 300

['M1', 'M2', 'M3', 'M4', 'M5']

['K', 'K', 'A', 'K', 'A']

[200, 500, 700, 400, 300]

M3

M2