

**ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ
ΗΜΕΡΗΣΙΩΝ – ΕΣΠΕΡΙΝΩΝ ΕΠΑΓΓΕΛΜΑΤΙΚΩΝ ΛΥΚΕΙΩΝ
ΣΑΒΒΑΤΟ 27 ΙΟΥΝΙΟΥ 2020
ΕΞΕΤΑΖΟΜΕΝΟ ΜΑΘΗΜΑ:
ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ**

ΠΡΟΤΕΙΝΟΜΕΝΕΣ ΑΠΑΝΤΗΣΕΙΣ ΘΕΜΑΤΩΝ

moodle.sepchou.gr

ΘΕΜΑ Α

A1.

- α. Σωστό
- β. Λάθος
- γ. Λάθος
- δ. Σωστό
- ε. Σωστό

A2.

- 1. 0
- 2. 5
- 3. 5

A3.

- 1. True
- 2. 2
- 3. 1
- 4. 1.232
- 5. False
- 6. True

ΘΕΜΑ Β

B1.

1. len(array)-1
2. last
3. -1
4. pos

5. last
6. first
7. mid +1
8. pos

B2.

Το γ είναι 2

Το χ είναι 30

Το κ είναι 6

Το γ είναι 3

Το z είναι 9

B4.

1ος Τρόπος

```
def find_gr(L):  
    count = 0  
    for site in L:  
        if site[-3:] == ".gr":  
            count +=1  
    return count
```

2ος Τρόπος

```
def find_gr(L):  
    count = 0  
    for site in L:  
        if site[-3:] == ".":  
            if site[-2] == "g":  
                if site[-1:] == "r":  
                    count +=1  
    return count
```

3ος Τρόπος

```
def find_gr(L):  
    count = 0  
    for site in L:  
        if site[-3]+site[-2]+site[-1] == ".gr":  
            count +=1  
    return count
```

ΘΕΜΑ Γ

```
# Να υλοποιήσετε τη συνάρτηση EISITIRIO(), η οποία δέχεται το πλήθος
# των ενηλίκων και το πλήθος των παιδιών μιας παρέας και επιστρέφει το
# συνολικό κόστος των εισιτηρίων.
def EISITIRIO(enilikes, paidia):
    return enilikes*10 + paidia*5

syn_en = 0.0
syn_paidia= 0.0
theseis =500

#να εμφανίζει τις διαθέσιμες ελεύθερες θέσεις του θεάτρου
print theseis

#Η διαδικασία της εισόδου θεατών να τερματίζεται, όταν εισαχθεί ο αριθμός μείον ένα (-1)
#ως πλήθος ενηλίκων θεατών μιας παρέας. Ο αριθμός μείον ένα (-1) σημαίνει ότι είτε οι
#διαθέσιμες θέσεις δεν επαρκούν είτε δεν υπάρχουν άλλοι θεατές που επιθυμούν να
#εισέλθουν. Στην περίπτωση αυτή, δεν εισάγεται αριθμός παιδιών.

enilikes = input("πλήθος ενηλίκων ")

while enilikes != -1:
    paidia = input(" πλήθος παιδιών")
    kostos =EISITIRIO(enilikes,paidia)
    print 'συνολικό κόστος των εισιτηρίων' kostos
    syn_en +=enilikes
    syn_paidia += paidia
    sunolo += kostos
    #να εμφανίζει τις διαθέσιμες ελεύθερες θέσεις του θεάτρου
    theseis = theseis – enilikes – paidia
    print theseis
    enilikes = input("πλήθος ενηλίκων ")
```

#Να υπολογίζει και να εμφανίζει: α) Τα συνολικά έσοδα του θεάτρου

```
print 'Σύνολο εσόδων',synolo
```

#Το ποσοστό των παιδιών στο σύνολο των θεατών

```
synolo_theaton = syn_en + syn_paidia
```

```
pososto_p = syn_paidia/synolo_theaton *100.0
```

```
print 'Ποσοστό παιδιών', pososto_p, '%'
```

ΘΕΜΑ Δ

```
# Οι σωστές απαντήσεις βρίσκονται στη λίστα LI
LI=["α","δ","γ","β","δ","γ","β","α","δ","γ","β","δ","γ","β","α"]

SV = []

ON = []

for i in range(20):

    # Να διαβάξει από το πληκτρολόγιο το όνομά του και να το
    #εισάγει στη λίστα ON[ ]

    on=raw_input("Όνομα υποψηφίου: ")

    ON.append(on)

    # Να διαβάξει από το πληκτρολόγιο διαδοχικά τις δεκαπέντε (15) απαντήσεις

    synolo = 0

    for i in range(15):

        ap = raw_input()

        if ap == LI[i]: #Αν η επιλογή είναι η σωστή απάντηση βαθμολογείται με τρεις (3) μονάδες,

            synolo += 3

        elif ap != "ε" # λανθασμένη, αφαιρείται μία (1) μονάδα

            synolo -= 1:

        #αν έχει επιλεγεί το "ε", δεν βαθμολογείται [μηδέν (0) μονάδες].

    # Η συνολική βαθμολογία του εισάγεται στη λίστα SV[ ]

    SV.append(synolo)

#Να εμφανίζει τα ονόματα των υποψηφίων που είχαν βαθμολογία μεγαλύτερη ή ίση από τον

#μέσο όρο όλων των βαθμολογιών.

s = 0.0

for b in SV:

    s += b

mo = s/20.0
```

```
for l in range(20):  
    if SV[i] >= mo:  
        print ON[i]
```

Bubble Sort αύξουσα

```
def bubbleSortA(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(SV,ON)
```

```
print "Οι μαθητές με τις 3 υψηλότερες βαθμολογίες είναι: "
```

```
print ON[-1]
```

```
print ON[-2]
```

```
print ON[-3]
```