

**ΕΠΑΝΑΛΗΠΤΙΚΑ ΘΕΜΑΤΑ 2022**
Β' ΦΑΣΗ**E_3.ΠΕΛ3Ε(ε)****ΤΑΞΗ: 3^η ΤΑΞΗ ΕΠΑ.Λ.****ΜΑΘΗΜΑ: ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ/
ΕΙΔΙΚΟΤΗΤΑΣ****Ημερομηνία: Σάββατο 16 Απριλίου 2022****Διάρκεια Εξέτασης: 3 ώρες****ΕΚΦΩΝΗΣΕΙΣ****ΘΕΜΑ Α**

A1. Να χαρακτηρίσετε τις προτάσεις που ακολουθούν, γράφοντας στο τετράδιό σας, δίπλα στον αριθμό που αντιστοιχεί σε κάθε πρόταση, τη λέξη **Σωστό**, αν η πρόταση είναι σωστή, ή τη λέξη **Λάθος**, αν η πρόταση είναι λανθασμένη.

1. Η συνάρτηση **pow(a,b)** επιστρέφει το ακέραιο πηλίκο και το ακέραιο υπόλοιπο της διαίρεσης του **a** με το **b**.
2. Ο υπαρκτικός τελεστής **in** ελέγχει αν ένα αντικείμενο ανήκει σε ένα σύνολο αντικειμένων.
3. Σύμφωνα με την τεχνική του Τμηματικού Προγραμματισμού, μπορούμε να γράψουμε ένα πρόγραμμα ως ένα σύνολο από μικρότερα κομμάτια προγράμματος.
4. Η δομή της ουράς μπορεί να υλοποιηθεί στην Python με μια λίστα στην οποία οι εισαγωγές και οι εξαγωγές στοιχείων γίνονται μόνο από το ένα άκρο.
5. Κατασκευαστής (constructor) ονομάζεται μια ειδική μέθοδος η οποία καταστρέφει ένα αντικείμενο και απελευθερώνει την αντίστοιχη μνήμη που είχε δεσμευτεί κατά τη δημιουργία του.

Μονάδες 10

- A2.** Να γράψετε στο τετράδιό σας τους αριθμούς **1, 2, 3, 4, 5** από τη στήλη **A** του παρακάτω πίνακα και δίπλα ένα από τα γράμματα **α, β, γ, δ, ε, στ** της στήλης **B**, που δίνει τη σωστή αντιστοίχιση. Σημειώνεται ότι ένα γράμμα από τη στήλη **B** θα περισσέψει.

ΣΤΗΛΗ Α	ΣΤΗΛΗ Β
1. open(myfile, 'w')	α. Επιστρέφει έναν ακέραιο αριθμό που περιέχει την τρέχουσα θέση μέσα στο αρχείο.
2. open(myfile, 'r+')	β. Αλλαγή της τρέχουσας θέσης μέσα στο αρχείο.
3. seek()	γ. Άνοιγμα αρχείου για εγγραφή, με διατήρηση των προηγούμενων περιεχομένων.
4. tell()	δ. Ανάγνωση πλήθους χαρακτήρων από αρχείο.
5. read()	ε. Άνοιγμα αρχείου για ανάγνωση και για εγγραφή.
	στ. Άνοιγμα αρχείου για εγγραφή, με διαγραφή των προηγούμενων περιεχομένων.

Μονάδες 10

- A3.** Σε μια μεταβλητή τύπου κινητής υποδιαστολής (float) με όνομα ther αποθηκεύεται η θερμοκρασία ενός χώρου σε βαθμούς Κελσίου. Οι επιτρεπτές τιμές είναι από 15,5 μέχρι και 30 βαθμούς. Να γράψετε στο τετράδιό σας ποια από τις παρακάτω εκφράσεις ελέγχει αυτή τη συνθήκη.

- α) (ther <= 15.5) and (ther >= 30.0)
 β) (ther <= 15.5) or (ther >= 30.0)
 γ) (ther >= 15.5) or (ther <= 30.0)
 δ) (ther >= 15.5) and (ther <= 30.0)

Μονάδες 5

ΘΕΜΑ Β

B1. Δίνεται η παρακάτω κλάση:

```
class Football_Player:
    def __init__(self, name, team):
        self.name = name
        self.team = team
```

όπου name το όνομα του ποδοσφαιριστή και team η ομάδα που αγωνίζεται αυτή τη στιγμή.

- α) Να προσθέσετε την ιδιότητα age που αντιπροσωπεύει την ηλικία του ποδοσφαιριστή (μον. 1) και την ιδιότητα apps που αντιπροσωπεύει τον αριθμό των συμμετοχών του στην ομάδα που αγωνίζεται αυτή τη στιγμή (μον. 1), ώστε να αρχικοποιούνται στον κατασκευαστή της κλάσης.
- β) Να δημιουργήσετε τη μέθοδο Set_App η οποία θα ορίζεται μέσα στην κλάση και θα αυξάνει κατά ένα (1) τον αριθμό των συμμετοχών του ποδοσφαιριστή (μον. 3).
- γ) Να δημιουργήσετε δύο (2) στιγμιότυπα της κλάσης, δηλαδή δύο αντικείμενα με ονόματα P1 και P2 των οποίων οι τιμές των ιδιοτήτων τους θα οριστούν κατά τη δημιουργία τους ως εξής:
Αντικείμενο P1: name='Lionel Messi', team='PSG', age=34, apps=18 (μον. 4)
Αντικείμενο P2: name='Cristiano Ronaldo', team='Man Utd', age=37, apps=24 (μον. 4)
- δ) Για τα παραπάνω αντικείμενα να καλέσετε τη μέθοδο Set_App, μία φορά για το καθένα (μον. 2).

Μονάδες 15

B2. Ο αλγόριθμος ταξινόμησης ευθείας ανταλλαγής (φουσαλίδα/bubblesort) μπορεί να βελτιωθεί έτσι ώστε να τερματίζει μόλις η λίστα ταξινομηθεί πλήρως. Για το λόγο αυτό χρησιμοποιείται μια λογική μεταβλητή η οποία μας πληροφορεί αν η λίστα είναι ταξινομημένη στο τέλος κάθε περάσματος, ελέγχοντας εάν έχει γίνει αντιμετάθεση στοιχείων ή όχι. Δίνεται η παρακάτω συνάρτηση η οποία πραγματοποιεί ταξινόμηση μιας λίστας σε αύξουσα διάταξη και τερματίζει με τη βοήθεια της λογικής μεταβλητής isSorted μόλις διαπιστώσει

ότι η λίστα είναι ταξινομημένη. Στη συνάρτηση αυτή υπάρχουν κενά τα οποία έχουν αριθμηθεί από το 1 μέχρι το 6.

```
def Optimized_Bubblesort(L):
    N = ....(1)....
    isSorted = False
    i = 0
    while i < ....(2).... and not ....(3).... :
        isSorted = ....(4)....
        for j in range(N-1, ....(5)...., -1):
            if L[j] < L[j-1]:
                L[j], L[j-1] = L[j-1], L[j]
            isSorted = ....(6)....
        i += 1
```

Να γράψετε στο τετράδιό σας τους αριθμούς (1), (2), (3), (4), (5) και (6) που αντιστοιχούν στα κενά της παραπάνω συνάρτησης και δίπλα σε κάθε αριθμό, αυτό που πρέπει να συμπληρωθεί ώστε η συνάρτηση να εκτελεί σωστά τη λειτουργία της.

Μονάδες 6

B3. Δίνεται το παρακάτω τμήμα προγράμματος σε γλώσσα Python:

```
def grammata(x):
    return len(x)

a = 'προφιτερόλ'
b = 'ραβανί'
c = grammata(a)
d = grammata(b)
print c+d
print a[ : 3]
print 'Θα ήθελα', c, b 'και', d, a
print 'Επίσης μαζί με το', b, 'θα ήθελα και', d, 'μπάλες παγωτό βανίλια'
```

Να γράψετε στο τετράδιό σας τι θα εμφανίσει το παραπάνω τμήμα προγράμματος κατά την εκτέλεσή του.

Μονάδες 4

ΘΕΜΑ Γ

Ένα μεζεδοπωλείο σερβίρει συνολικά **είκοσι (20)** πιάτα τα οποία είναι αποθηκευμένα στη λίστα MENU και οι αντίστοιχες τιμές των πιάτων, σε ευρώ, είναι αποθηκευμένες στη λίστα PRICE. Στο τέλος της ημέρας, ο ιδιοκτήτης του μεζεδοπωλείου χρησιμοποιεί τη λίστα PAR για να υπολογίσει τα συνολικά έσοδα από όλες τις παραγγελίες. Στη λίστα PAR αποθηκεύεται ένας ακέραιος αριθμός που αντιπροσωπεύει πόσες μερίδες σερβιρίστηκαν συνολικά από το κάθε πιάτο της λίστας MENU κατά τη διάρκεια της ημέρας. Κάθε θέση της λίστας PAR αντιστοιχεί, με την ίδια σειρά, σε ένα είδος πιάτου της λίστας MENU και στην τιμή του πιάτου που βρίσκεται στη λίστα PRICE.

Να αναπτύξετε πρόγραμμα σε γλώσσα προγραμματισμού Python, το οποίο:

- Γ1.** Να διαβάζει τα ονόματα των 20 πιάτων και τις τιμές τους και να τα αποθηκεύει στις λίστες MENU και PRICE αντίστοιχα. Επίσης, στην ίδια επαναληπτική διαδικασία, να αποθηκεύει 20 στοιχεία με τιμή μηδέν (0) στη λίστα PAR. Δεν χρειάζεται να γίνει έλεγχος εγκυρότητας στις τιμές που εισάγονται.

Μονάδες 5

- Γ2.** Να διαβάζει επαναληπτικά τον αριθμό του τραπέζιού το οποίο πρόκειται να παραγγείλει. Η επανάληψη θα τερματίζει όταν δοθεί ως αριθμός τραπέζιού το μηδέν (0). Στη συνέχεια, να διαβάζει τον αριθμό των μερίδων που παρήγγειλε το κάθε τραπέζι από το κάθε πιάτο της λίστας MENU και να ενημερώνει την αντίστοιχη θέση της λίστας PAR με τον αριθμό αυτό.

Μονάδες 7

- Γ3.** Μετά το τέλος των παραγγελιών, να υπολογίζει και να εμφανίζει το όνομα του πιάτου με τις περισσότερες παραγγελίες και το όνομα εκείνου με τις λιγότερες.

Μονάδες 8

- Γ4.** Να υπολογίζει και να εμφανίζει τα συνολικά έσοδα της ημέρας, χρησιμοποιώντας τις λίστες PRICE και PAR.

Μονάδες 5

ΘΕΜΑ Δ

Ο Εθνικός Οργανισμός Δημόσιας Υγείας (ΕΟΔΥ) καταγράφει τα ημερήσια κρούσματα Covid-19 σε δύο (2) παράλληλες λίστες. Η καταγραφή των κρουσμάτων γίνεται ανά νομό. Στη λίστα NOM αποθηκεύονται τα ονόματα των 51 νομών της Ελλάδας και στη λίστα KR αποθηκεύεται ο αριθμός των κρουσμάτων (ακέραιου τύπου) που εντοπίστηκε σε κάθε νομό.

Να αναπτύξετε πρόγραμμα σε γλώσσα προγραμματισμού Python, το οποίο:

Δ1. Για μία συγκεκριμένη ημέρα, να διαβάζει τα ονόματα των 51 νομών και τον αριθμό των κρουσμάτων που εντοπίστηκε σε κάθε νομό και να τα αποθηκεύει στις λίστες NOM και KR αντίστοιχα. Να γίνεται έλεγχος εγκυρότητας για τον αριθμό των κρουσμάτων ώστε να είναι θετικός αριθμός.

Μονάδες 5

Δ2. Να υπολογίζει και να εμφανίζει το σύνολο των κρουσμάτων σε ολόκληρη τη χώρα.

Μονάδες 4

Δ3. Να καλεί τη συνάρτηση **Sorting()** η οποία ταξινομεί σε αλφαβητική σειρά τη λίστα NOM ως προς τα ονόματα των νομών με χρήση του αλγορίθμου ταξινόμησης ευθείας ανταλλαγής (φυσαλίδα/bubblesort), αναδιατάσσοντας ταυτόχρονα τη λίστα KR έτσι ώστε να διατηρείται η αντιστοιχία νομών – αριθμού κρουσμάτων. Να υλοποιήσετε τη συνάρτηση **Sorting()** που χρειάζεται για το σκοπό αυτό.

Μονάδες 8

Δ4. Να ζητάει και να διαβάζει το όνομα ενός νομού. Στη συνέχεια θα καλεί τη συνάρτηση **Searching()** η οποία θα αναζητάει το όνομα αυτό στη λίστα NOM με χρήση του αλγορίθμου της δυαδικής αναζήτησης (binary search) επιστρέφοντας τη θέση στην οποία βρίσκεται ο νομός. Στην περίπτωση που βρει το νομό στη λίστα NOM, να εμφανίζει τον αντίστοιχο αριθμό κρουσμάτων από τη λίστα KR, ενώ σε αντίθετη περίπτωση να εμφανίζει ανάλογο μήνυμα. Να υλοποιήσετε τη συνάρτηση **Searching()** που χρειάζεται για το σκοπό αυτό.

Μονάδες 8