

**ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ  
ΗΜΕΡΗΣΙΩΝ ΚΑΙ ΕΣΠΕΡΙΝΩΝ ΕΠΑΓΓΕΛΜΑΤΙΚΩΝ ΛΥΚΕΙΩΝ  
ΠΕΜΠΤΗ 26 ΣΕΠΤΕΜΒΡΙΟΥ 2019**

**ΕΞΕΤΑΖΟΜΕΝΟ ΜΑΘΗΜΑ:  
ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ**

**ΣΥΝΟΛΟ ΣΕΛΙΔΩΝ: ΕΠΤΑ (7)**

**ΘΕΜΑ Α**

- Α1.** Να χαρακτηρίσετε τις προτάσεις που ακολουθούν, γράφοντας στο τετράδιό σας, δίπλα στο γράμμα που αντιστοιχεί σε κάθε πρόταση, τη λέξη **Σωστό**, αν η πρόταση είναι σωστή, ή τη λέξη **Λάθος**, αν η πρόταση είναι λανθασμένη.
- α.** Οι εντολές που περιλαμβάνονται μέσα στη δομή **while** θα εκτελεστούν τουλάχιστον μία (1) φορά.
  - β.** Μία συνάρτηση ορίζεται με τη λέξη κλειδί **def** που την ακολουθεί ένα όνομα το οποίο την ταυτοποιεί, ένα ζεύγος παρενθέσεων, που μπορεί να περιέχει ονόματα μεταβλητών, και τελειώνει με διπλή τελεία (:).
  - γ.** Αποδομητής (destructor) ονομάζεται η μέθοδος η οποία καταστρέφει αντικείμενα και ελευθερώνει τη μνήμη.
  - δ.** Η μέθοδος `fin.tell()` επιστρέφει έναν ακέραιο που περιέχει πάντα την τελευταία θέση του αρχείου.
  - ε.** Η λίστα, σε αντίθεση με τη συμβολοσειρά, είναι μία δυναμική δομή, στην οποία μπορούμε να προσθέτουμε ή να αφαιρούμε στοιχεία (mutable).
  - στ.** Η δομή της ουράς μπορεί να υλοποιηθεί στην Python με μία λίστα στην οποία οι εισαγωγές και οι εξαγωγές στοιχείων γίνονται μόνο από το ένα άκρο.

**Μονάδες 12**

ΑΡΧΗ 2ΗΣ ΣΕΛΙΔΑΣ

**A2.** Η στήλη **A** περιέχει συναρτήσεις, ενώ η στήλη **B** περιέχει το αποτέλεσμά τους. Να γράψετε στο τετράδιό σας τους αριθμούς **1, 2, 3, 4** από τη στήλη **A** και δίπλα ένα από τα γράμματα **α, β, γ, δ, ε** της στήλης **B**, που δίνει τη σωστή αντιστοίχιση.

Σημειώνεται ότι ένα γράμμα από τη στήλη **B** θα περισσέψει.

<b>ΣΤΗΛΗ Α ΣΥΝΑΡΤΗΣΗ</b>	<b>ΣΤΗΛΗ Β ΑΠΟΤΕΛΕΣΜΑ</b>
<b>1. float(10)</b>	<b>α. 10.0</b>
<b>2. pow(2,3)</b>	<b>β. 5</b>
<b>3. abs(-10)</b>	<b>γ. 8</b>
<b>4. int(5.6)</b>	<b>δ. 10</b>
	<b>ε. 5.6</b>

**Μονάδες 8**

**A3.** Να χαρακτηρίσετε τις λογικές εκφράσεις που ακολουθούν, γράφοντας στο τετράδιό σας, δίπλα στο γράμμα που αντιστοιχεί σε κάθε έκφραση, τη λέξη **True**, αν η πρόταση είναι αληθής, ή τη λέξη **False**, αν η πρόταση είναι ψευδής.

- α. not(4<9)**
- β. 4<len("καλημέρα")**
- γ. (4>5) or (9>2)**
- δ. (4==4) and not(4>9)**
- ε. pow(3,0)==9-8**

**Μονάδες 5**

**ΘΕΜΑ Β**

**B1.** Δίνεται το παρακάτω τμήμα προγράμματος Python:

```
num=int(input("Δώστε αριθμό"))
check=num%2
if check==0:
    print "ο αριθμός είναι άρτιος"
else:
    print "ο αριθμός είναι περιττός"
```

Ποιο θα είναι το αποτέλεσμα του παραπάνω τμήματος προγράμματος για καθεμία από τις παρακάτω εισόδους;

- α. `num=3.5`
- β. `num=8`
- γ. `num=4.8`

**Μονάδες 6**

**B2.** Δίνεται το παρακάτω τμήμα προγράμματος Python:

```
i=0
p=1
prod=1
while i<8:
    if i<=5:
        p=2
    else:
        p=1
    prod=prod*p
    i=i+2
```

ΑΡΧΗ 4ΗΣ ΣΕΛΙΔΑΣ

Επίσης δίνεται ο παρακάτω πίνακας τιμών στον οποίο έχουν συμπληρωθεί κάποια κενά. Να μεταφέρετε στο τετράδιό σας τον παρακάτω πίνακα τιμών και να συμπληρώσετε τις τιμές στα κενά εκτελώντας τις εντολές του τμήματος προγράμματος. Να προσθέσετε στον πίνακα όσες γραμμές είναι απαραίτητες.

	<b>i</b>	<b>prod</b>	<b>p</b>
Αρχικά	0	1	1
1 <sup>n</sup> επανάληψη			2
...			

**Μονάδες 11**

**B3.** Το παρακάτω τμήμα προγράμματος Python υπολογίζει το γινόμενο  $1*2*3*4*5$  :

**mult**=       <sup>(1)</sup>

**for i in range** (       <sup>(2)</sup> ,       <sup>(3)</sup> ):

**mult**=**mult**\*       <sup>(4)</sup>

**print** "Το αποτέλεσμα είναι ",**mult**

Στο τμήμα προγράμματος υπάρχουν τέσσερα (4) κενά, τα οποία έχουν αριθμηθεί και υπογραμμιστεί. Να γράψετε στο τετράδιό σας τον αριθμό του κενού και δίπλα τι πρέπει να συμπληρωθεί, ώστε το τμήμα προγράμματος να εκτελεί σωστά τη λειτουργία του.

**Μονάδες 8**

**ΘΕΜΑ Γ**

Ένα ξενοδοχείο διαθέτει 50 δωμάτια και λειτουργεί 100 ημέρες τον χρόνο. Θεωρείστε ότι το κόστος διανυκτέρευσης είναι 80€ για κάθε δωμάτιο.

Να γράψετε πρόγραμμα σε γλώσσα προγραμματισμού Python το οποίο να πραγματοποιεί τα παρακάτω:

**Γ1.** Για κάθε μέρα να διαβάζει το πλήθος των αναχωρήσεων (δωμάτια που αδειάζουν) ελέγχοντας την εγκυρότητα των δεδομένων, δηλαδή, ότι το πλήθος των αναχωρήσεων είναι μικρότερο ή ίσο από το πλήθος των κατειλημμένων δωματίων.

**Μονάδες 6**

**Γ2.** Για κάθε μέρα να διαβάξει το πλήθος των αφίξεων (δωμάτια που γεμίζουν) ελέγχοντας την εγκυρότητα των δεδομένων, δηλαδή, ότι το πλήθος των αφίξεων δεν μπορεί να είναι μεγαλύτερο από το πλήθος των κενών δωματίων.

**Μονάδες 6**

**Γ3.** Να υπολογίζει και να εμφανίζει το πλήθος των ημερών που το ξενοδοχείο είχε πληρότητα 100%, δηλαδή και τα πενήντα (50) δωμάτια ήταν κατειλημμένα.

**Μονάδες 5**

**Γ4.** Να υπολογίζει και να εμφανίζει τον μέσο όρο του πλήθους των δωματίων (μέση πληρότητα) που ήταν κατειλημμένα στο διάστημα λειτουργίας των εκατό (100) ημερών.

**Μονάδες 4**

**Γ5.** Να υπολογίζει και να εμφανίζει τα συνολικά έσοδα του ξενοδοχείου από τις διανυκτερεύσεις για το διάστημα λειτουργίας των εκατό (100) ημερών.

**Μονάδες 4**

### **ΘΕΜΑ Δ**

Μία τράπεζα, για τις ανάγκες εξυπηρέτησης των πελατών της, διατηρεί δύο παράλληλες ταξινομημένες λίστες κατά αύξουσα σειρά με βάση τον αριθμό λογαριασμού. Μία λίστα με όνομα LOG[ ], η οποία περιλαμβάνει τους αριθμούς λογαριασμών των πελατών της, και μία λίστα με όνομα ΚΑΤΑΤΗ[ ], η οποία περιλαμβάνει το υπόλοιπο του λογαριασμού κάθε πελάτη. Θεωρείστε ότι όλοι οι αριθμοί λογαριασμών που υπάρχουν στη λίστα LOG[ ] είναι μοναδικοί θετικοί ακέραιοι.

Η τράπεζα επιθυμεί να γνωρίζει το ύψος των καταθέσεων και των αναλήψεων που πραγματοποιήθηκαν κατά τη διάρκεια λειτουργίας του ταμείου.

Να γράψετε τμήμα προγράμματος και τις απαραίτητες συναρτήσεις σε γλώσσα προγραμματισμού Python που να πραγματοποιεί τα παρακάτω:

**Δ1.** Να διαβάξει για κάθε αιτούμενη συναλλαγή τον αριθμό λογαριασμού, το είδος της συναλλαγής (1-κατάθεση, 2-ανάληψη) και το ποσό συναλλαγής. Η επαναληπτική διαδικασία τερματίζεται (κλείσιμο του ταμείου), όταν δοθεί ως αριθμός λογαριασμού ο αριθμός μηδέν (0).

**Μονάδες 3**

**Δ2.** Να αναζητά τον αριθμό λογαριασμού στη λίστα LOG[ ] και να εμφανίζει το τρέχον υπόλοιπό του από τη λίστα ΚΑΤΑΤΗ[ ].

**Μονάδες 7**

**Δ3.** Να υλοποιήσετε συνάρτηση **money()**, η οποία θα καλείται σε περίπτωση που η αιτούμενη συναλλαγή είναι ανάληψη, ώστε να ελέγξει αν επαρκεί το υπόλοιπο του λογαριασμού για να πραγματοποιηθεί η συναλλαγή. Αν το υπόλοιπο επαρκεί, να γίνεται η ανάληψη, σε διαφορετική περίπτωση να ενημερώνεται ο πελάτης με κατάλληλο μήνυμα. Αν η αιτούμενη συναλλαγή είναι κατάθεση, να πραγματοποιείται η συναλλαγή.

**Μονάδες 10**

**Δ4.** Να υπολογίζει και να εμφανίζει το άθροισμα των καταθέσεων και των αναλήψεων, μετά το κλείσιμο του ταμείου.

**Μονάδες 5**

Σημείωση:

Δεν απαιτούνται έλεγχοι ορθότητας δεδομένων.

**Επισημάνσεις για τα θέματα Γ και Δ**

Στην Python η ομάδα εντολών, η οποία εκτελείται μέσα σε μια δομή (επιλογής, επανάληψης, συνάρτησης, κ.λπ.), καθορίζεται ως ένα μπλοκ εντολών με τη χρήση μιας ίδιας εσοχής σε σχέση με την αρχική γραμμή της δομής. Για να είναι εμφανής η ύπαρξη της ίδιας εσοχής, **συστήνεται** οι μαθητές στο γραπτό τους να σημειώνουν τις εντολές που ανήκουν στο ίδιο μπλοκ με μία κάθετη γραμμή μπροστά από αυτές.

Παράδειγμα:

```
while x<4:  
    | b=b+1  
    | a=a*x  
    | print a, x  
    | if b%2 == 0:  
    | | x=x+1  
    | | print a,x  
print b
```

**ΟΔΗΓΙΕΣ ΓΙΑ ΤΟΥΣ ΕΞΕΤΑΖΟΜΕΝΟΥΣ**

1. Στο τετράδιο να γράψετε μόνο τα προκαταρκτικά (ημερομηνία, εξεταζόμενο μάθημα). **Να μην αντιγράψετε τα θέματα στο τετράδιο.**
2. Να γράψετε το ονοματεπώνυμό σας στο πάνω μέρος των φωτοαντιγράφων, αμέσως μόλις σας παραδοθούν. **Δεν επιτρέπεται να γράψετε** καμία άλλη σημείωση. Κατά την αποχώρησή σας να παραδώσετε μαζί με το τετράδιο και τα φωτοαντίγραφα.
3. Να απαντήσετε **στο τετράδιό σας** σε όλα τα θέματα, **μόνο με μπλε ή μόνο με μαύρο στυλό ανεξίτηλου μελανιού.**
4. Κάθε επιστημονικά τεκμηριωμένη απάντηση είναι αποδεκτή.
5. Διάρκεια εξέτασης: τρεις (3) ώρες μετά τη διανομή των φωτοαντιγράφων.
6. Ώρα δυνατής αποχώρησης: **17:00.**

**ΚΑΛΗ ΕΠΙΤΥΧΙΑ**

**ΤΕΛΟΣ ΜΗΝΥΜΑΤΟΣ**