

ΔΙΑΓΩΝΙΣΜΑ ΠΡΟΣΟΜΟΙΩΣΗ
ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ
Γ' ΕΠΑΛ

ΘΕΜΑ Α:

A1]Να χαρακτηρίσετε τις παρακάτω προτάσεις με Σ ή Λ. (10 ΜΟΝ)

1. Η δυαδική αναζήτηση (binary Search) χρησιμοποιείται μόνο σε ταξινομημένες συλλογές δεδομένων.
2. Η συνθήκη $7*2==len('MichaelJordan')$ δίνει αποτέλεσμα True
3. Η εμβέλεια (scope) μιας μεταβλητής αναφέρεται στο τμήμα του προγράμματος που μπορεί αυτή να έχει πρόσβαση.
4. Μία συνάρτηση στην Python θα πρέπει υποχρεωτικά όταν κληθεί να επιστρέφει ένα αποτέλεσμα (κάποια τιμή).
5. Η λογική παράσταση $x<5$ έχει πάντα την τιμή False ανεξαρτήτως της τιμής του x (Για οποιαδήποτε τιμή του x).
6. Σ' ένα πρόγραμμα Python μπορούμε να εκχωρούμε διαφορετικούς τύπους τιμών σε μία μεταβλητή κατά τη διάρκεια της εκτέλεσής του.
7. Η συνάρτηση str() δέχεται μόνο ακέραιους και τους μετατρέπει σε συμβολοσειρές.
8. Στη συνάρτηση range(αρχή, μέχρι ,βήμα) η ένδειξη της αρχής είναι προαιρετική ενώ η ένδειξη του βήματος υποχρεωτική.
9. Το μειονέκτημα των τοπικών μεταβλητών είναι ότι περιορίζεται η ανεξαρτησία των υποπρογραμμάτων.
10. Οι συμβολοσειρές είναι ακολουθίες από χαρακτήρες και έχουν σταθερό μέγεθος και μη μεταβαλλόμενα περιεχόμενα.

A2] α. Δίνεται το παρακάτω τμήμα προγράμματος σε Python στο οποίο έχουν αριθμηθεί οι εντολές του:

1. `x=10`
2. `y=3`
3. `print x, y`
4. `t=x`
5. `x=y`
6. `y=t`
7. `print x, y`

Να ξαναγράψετε το παραπάνω τμήμα αντικαθιστώντας τις εντολές 4-6 με μια μόνο εντολή εκχώρησης ώστε να παράγει το ίδιο αποτέλεσμα με το δοθέν τμήμα.

(5 MON)

β. Δίνεται το παρακάτω τμήμα προγράμματος σε Python στο οποίο έχουν αριθμηθεί οι εντολές του:

1. `x=10`
2. `y=3`
3. `if x > y :`
4. `z=true`
5. `else :`
6. `z=false`
7. `print z`

Να ξαναγράψετε το παραπάνω τμήμα αντικαθιστώντας τις εντολές 3-6 με δύο εντολές απλής επιλογής ώστε να παράγει το ίδιο αποτέλεσμα με το δοθέν τμήμα.

(6 MON)

A3] Δίνεται το παρακάτω τμήμα προγράμματος

1. `x=0`

2. $y=0$

3. while $y<90$:

4. $y=y+20$

5. $x=x+y$

6. print x,y

α) Να γράψετε πόσες φορές θα εκτελεστεί η γραμμή 4.

β) Ποιες θα είναι οι τιμές του x,y σε κάθε επανάληψη.

(9 MON)

A4] Για καθένα από τα παρακάτω ερωτήματα να γράψετε το γράμμα α-στ και δίπλα το αποτέλεσμα που θα έχει η εντολή.

α. float(100)

β. int(3.99)

γ. str(8.25)

δ. abs(-12)

ε. divmod(4, 12)

στ. pow(3, 3)

(10 MON)

ΘΕΜΑ Β:

B1. Δίνεται το πρόγραμμα: (10 MON)

$x=4$

$S=0$

for i in range(10,3,-2):

$S=S+i$

$x=S\%i$

print "S=",S,

print " i=",i

α. Να γράψετε στο τετράδιό σας ό,τι ακριβώς εμφανίζεται στην οθόνη κατά την εκτέλεση του παραπάνω προγράμματος.

β. Να ξαναγράψετε το παραπάνω πρόγραμμα, χρησιμοποιώντας την εντολή επανάληψης while αντί της εντολής επανάληψης for έτσι ώστε να εμφανίζει το ίδιο αποτέλεσμα.

B2]Να γραφεί συνάρτηση σε Python που θα δέχεται ένα τριψήφιο αριθμό x και θα επιστρέφει το άθροισμα των ψηφίων του.π.χ. Αν δίνεται ο αριθμός 235 να επιστρέφει τον αριθμό $2+3+5=10$. (10 ΜΟΝ)

ΘΕΜΑ Γ:

Ενα βιβλιοπωλείο έχει καταχωρίσει στη λίστα TITLE τους τίτλους κάθε ενός από τα 1500 διαφορετικά βιβλία που διαθέτει, και στη λίστα ANT τον αριθμό των αντιτύπων από το κάθε βιβλίο. Το βιβλιοπωλείο έχει απόθεμα και από τα 1500 βιβλία, δηλαδή στη λίστα ANT, δεν υπάρχει κάποιο βιβλίο με μηδενικό πλήθος αντιτύπων. Να γράψετε πρόγραμμα σε γλώσσα Python, το οποίο:

Γ1. Θα διαβάσει τον τίτλο και τον αριθμό αντιτύπων κάθε βιβλίου και θα τα καταχωρίζει στις παραπάνω λίστες. (4 Μον)

Γ2. Θα ταξινομή τις λίστες ως προς τα αντίτυπα, κατά αύξουσα σειρά, με τη βοήθεια του αλγορίθμου ευθείας ανταλλαγής (φουσαλίδα/bubblesort). (7Μον)

Γ3. Θα διαβάσει τον τίτλο (titlos) ενός βιβλίου που ζήτησε ένας πελάτης, και με τη βοήθεια του αλγορίθμου της σειριακής αναζήτησης, θα εμφανίζει το μήνυμα «ΔΙΑΘΕΣΙΜΟ», αν ο συγκεκριμένος τίτλος υπάρχει στη λίστα TITLE, διαφορετικά θα εμφανίζει το μήνυμα «ΜΗ ΔΙΑΘΕΣΙΜΟ». (5 Μον)

Γ4. Με τη βοήθεια του αλγορίθμου της δυαδικής αναζήτησης, θα εμφανίζει κατάλληλο μήνυμα, αν υπάρχει κάποιο βιβλίο με πενήντα (50) διαθέσιμα αντίτυπα. (Μονάδες 5) Γ5. Θα υπολογίζει και θα εμφανίζει τον τίτλο των βιβλίων με περισσότερα από εκατό (100) αντίτυπα. (4Μον)

ΘΕΜΑ Δ:

Ένας σκληρός δίσκος έχει συνολική χωρητικότητα 500 GB. Ο ιδιοκτήτης του ξεκινάει να αποθηκεύει αρχεία στον σκληρό δίσκο, ο οποίος είναι άδειος αρχικά. Η αποθήκευση θα σταματήσει όταν η χωρητικότητα του δίσκου δεν θα επαρκεί για την αποθήκευση του αρχείου που διαβάζεται. Να γράψετε πρόγραμμα σε Python το οποίο:

Δ1. Θα αποθηκεύει στη μεταβλητή `cap` την αρχική χωρητικότητα του σκληρού δίσκου. (3 MON)

Δ2. Θα διαβάζει επαναληπτικά το μέγεθος (σε GB) του αρχείου που θέλει να αποθηκεύσει ο ιδιοκτήτης, και θα ελέγχει αν μπορεί να αποθηκευτεί στον σκληρό δίσκο.(4 MON)

Δ3. Αν η αποθήκευση μπορεί να πραγματοποιηθεί, θα εμφανίζει το μήνυμα 'Επιτυχής αποθήκευση'. Αν η αποθήκευση δε μπορεί να πραγματοποιηθεί, θα εμφανίζει 'Δεν υπάρχει αρκετός χώρος γι' αυτό το αρχείο.' (4 MON)

Δ4. Στο τέλος θα εμφανίζει: (9 MON)

- α. το πλήθος των αρχείων που αποθηκεύτηκαν,
- β. το μέγεθος του μεγαλύτερου αρχείου που αποθηκεύτηκε,
- γ. το ποσοστό των αρχείων που είχε μέγεθος το πολύ 1 GB,
- δ. το μέσο όρο των μεγεθών όλων των αρχείων που αποθηκεύτηκαν στον δίσκο,
- ε. τον ελεύθερο χώρο που έμεινε στον σκληρό δίσκο.

Αν δεν έχει μείνει χώρος στον σκληρό δίσκο, να εμφανίζει το μήνυμα 'Ο δίσκος έχει γεμίσει.'

ΚΑΛΗ ΕΠΙΤΥΧΙΑ!!!