**ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΑΣΚΗΣΕΙΣ**

**1.**Να χαρακτηρίσετε καθεμία από τις παρακάτω προτάσεις ως **Σωστή ή Λάθος.**

* Η int(x) μετατρέπει σε ακέραιο την αριθμητική τιμή x.
* H type(5/3) επιστρέφει float.
* Η εντολή χ=5>3 είναι μια αποδεκτή εντολή στη γλώσσα Python.
* H εντολή 10%3 επιστρέφει την τιμή 3.
* Η εντολή str(5) επιστρέφει την τιμή 5.0
* Στη δομή επιλογής μπορεί, μπορεί μία ή περισσότερες εντολές να μην εκτελεστούν ποτέ.
* Η εντολή for χρησιμοποιεί τη συνάρτηση range() για τον καθορισμό των επαναλήψεων
* Οι συναρτήσεις είναι επαναχρησιμοποιήσιμα μέρη προγραμμάτων.
* Οι ονομασίες που δίνουμε στη γραμμή ορισμού της συνάρτησης ονομάζονται ορίσματα.
* Οι συναρτήσεις randint(1,10) και randrange(1,10) είναι ισοδύναμες.
* Ο αλγόριθμος της δυαδικής αναζήτησης μπορεί να εφαρμοσθεί σε οποιαδήποτε λίστα.
* Ο αλγόριθμος της δυαδικής αναζήτησης είναι πιο γρήγορος από τον αλγόριθμο της σειριακής αναζήτησης.
* Όλες τις συναρτήσεις η Python τις θεωρεί αντικείμενα.
* Μια συνάρτηση δεν πρέπει να έχει ορισθεί πριν χρησιμοποιηθεί.
* Η κλήση μιας συνάρτησης είναι παράκαμψη στη ροή της εκτέλεσης.
* Μια συνάρτηση πρέπει να επιστρέφει οπωσδήποτε ένα αποτέλεσμα.
* Οι καθολικές μεταβλητές είναι περιορισμένης εμβέλειας.
* Οι παράμετροι μιας συνάρτησης είναι καθολικές μεταβλητές.
* Μια καθολική μεταβλητή μπορεί να χρησιμοποιηθεί στο εσωτερικό μιας συνάρτησης.

**2.** Να γράψετε το ισοδύναμο τμήμα προγράμματος χρησιμοποιώντας την εντολή while στη θέση της for και αντίστροφα.

Α) s=0 β) s=0

 i=1 for i in range(30,0,-2):

 while i<=6: s=s+i\*\*2

 i+=1

 print i

 s=s+i

**3.** ∆ίνονται οι παρακάτω εντολές σε Python. Τι πιστεύετε ότι θα εµφανιστεί στην οθόνη µετά την εκτέλεσή τους;

1. x=[9,2,6,3] 5. z.append(30) 9. print z

2. y=[24,3,29] 6. print z 10. z.insert(3,4)

3. z=x+y 7. print z[3] 11. print z

4. print z 8. z.pop(4)

**4.**Τι θα εμφανιστεί στην οθόνη εάν εκτελέσουμε το ακόλουθο πρόγραμμα:

def changelist(mylist):

 mylist.append([5,10,15])

 print “Τιμές μέσα στη συνάρτηση”, mylist

 return

mylist=[10,20,30]

changelist(mylist)

print “Τιμές έξω από τη συνάρτηση”, mylist

Έξοδος:

**5.** a. Τι θα εμφανιστεί στην οθόνη αν εκτελέσουμε το παρακάτω πρόγραμμα;

x **=** 50

**def** **func**(x):

 **print**('Το x είναι', x)

 x **=** 2

 **print**('Άλλαξα το τοπικό x σε', x)

func(x)

**print**('Το x είναι ', x)

Έξοδος:

**b.** Τι θα εμφανιστεί στην οθόνη αν εκτελέσουμε το παρακάτω πρόγραμμα;

x **=** 20

**def** **func**():

 **global** x

 **print**('Το x είναι', x)

 x **=** 3

 **print**('Άλλαξα το καθολικό x σε', x)

func()

**print**('Η τιμή του x είναι', x)

Έξοδος:

**6.** Να γραφεί πρόγραμμα Python που να δημιουργεί μία λίστα 50 θέσεων με πρώτο στοιχείο το 1 και τα καθένα από τα επόμενα στοιχεία να είναι διπλάσιο από το προηγούμενο (δηλ.[1,2,4,8,16….]).

**7.** Να συµπληρώσετε τα κενά στο παρακάτω πρόγραµµα σε Python, έτσι ώστε να εκτελεί τον αλγόριθµο της δυαδικής αναζήτησης και να επιστρέφει τη θέση ενός στοιχείου key σε µια λίστα array ταξινομημένη κατά **φθίνουσα** σειρά . Αν το στοιχείο δεν υπάρχει να επιστρέφει -1.

 def binarySearch( array, key ) :

 first = \_\_\_\_\_\_\_\_\_\_\_

 last = \_----------------

 pos = \_\_\_\_\_\_\_\_\_\_\_\_

 while \_\_\_\_\_\_\_\_\_\_ and pos\_\_\_\_\_\_\_\_\_\_ :

 mid = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

 if array[ mid ] == key :

 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

 elif \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ :

 first = \_\_\_\_\_\_\_\_\_\_\_\_\_

 else :

 last = \_\_\_\_\_\_\_\_\_\_\_\_\_

 return pos

**8.** Δίνεται παρακάτω η λίστα L.

L: 0 1 2 3 4 5 6 7 8 9 10 11

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2 | 5 | 8 | 14 | 20 | 27 | 32 | 36 | 43 | 47 | 52 | 60 |

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 60

β) για τον αριθμό 4.

**9.** Δίνεται η παρακάτω λίστα Α με 6 αριθμούς. Να εκτελέσετε τον αλγόριθμο ταξινόμησης ευθείας ανταλλαγής για την ταξινόμηση των αριθμών σε αύξουσα σειρά, συμπληρώνοντας παράλληλα τα κενά στον παρακάτω πίνακα, έτσι ώστε να φαίνονται τα στοιχεία της λίστας αμέσως μετά από κάθε πέρασμα του αλγορίθμου.

 **Α[0] Α[1] Α[2] Α[3] Α[4] Α[5]**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **24** | **17** | **6** | **9** | **3** | **7** |
| **1ο** **πέρασμα** |  |  |  |  |  |  |
| **2ο** **πέρασμα** |  |  |  |  |  |  |
| **3ο** **πέρασμα** |  |  |  |  |  |  |
| **4ο** **πέρασμα** |  |  |  |  |  |  |
| **5ο** **πέρασμα** |  |  |  |  |  |  |

**10.** Μια εταιρεία κινητής τηλεφωνίας ακολουθεί την παρακάτω πολιτική τιμών στην κλιμακωτή χρέωση των τηλεφωνημάτων ανά μήνα :

|  |
| --- |
| Πάγιο 10 € |
| Χρόνος τηλεφωνημάτων (δευτερόλεπτα) | Χρονοχρέωση ( ευρώ/δευτερόλεπτο) |
| 1 έως 500 | 0,07 |
| 501 έως 800 | 0,05 |
| 801 και άνω | 0,02 |

Να γραφεί πρόγραμμα το οποίο :

α. να διαβάζει το όνομα και τη μηνιαία χρονική διάρκεια των τηλεφωνημάτων κάθε συνδρομητή. Η καταχώρηση τελειώνει αν δώσουμε όνομα το ‘’(null χαρακτήρας).

β. να υπολογίζει τη μηνιαία χρέωση κάθε συνδρομητή και να εμφανίζει το μήνυμα “Η ΜΗΝΙΑΙΑ ΧΡΕΩΣΗ ΕΙΝΑΙ” και τη μηνιαία χρέωση.

γ. να υπολογίζει το μέσο όρο των χρημάτων που πλήρωσαν οι συνδρομητές.

δ. να υπολογίζει το πλήθος των συνδρομητών που πλήρωσαν πάνω από 50 ευρώ.

**11.** 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:

a. Θα αποθηκεύει στη μεταβλητή cap την αρχική χωρητικότητα του σκληρού δίσκου.

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

c. 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.

d. Στο τέλος θα εμφανίζει:

 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 GΒ,

 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 μέινει χώρος στον σκληρό δίσκο, να εμφανίζει το μήνυμα ‘Ο δίσκος έχει γεμίσει.

**12.** Ένας από τους πιο γνωστούς και χρήσιμους αλγορίθμους της Πληροφορικής είναι ο αλγόριθμος της συγχώνευσης των στοιχείων δυο ταξινομημένων λιστών σε μία νέα, επίσης ταξινομημένη, λίστα. Ο αλγόριθμος αξιοποιεί το γεγονός ότι οι αρχικές λίστες είναι ήδη ταξινομημένες, ώστε να μη χρειαστεί να ταξινομήσει από την αρχή την τελική λίστα, κάτι το οποίο έχει σημαντικό υπολογιστικό κόστος για πολύ μεγάλες λίστες.

def merge( A, B ) :

L = ( **1** )

while A != [ ] and B != ( **2** ) :

 if A[0] < ( **3** ) :

L.append(**(4))**

 else :

 L.append( B.pop(0))

 return **( 5 )**

Στο τμήμα προγράμματος υπάρχουν υπογραμμισμένα κενά τα οποία έχουν αριθμηθεί. Να γράψετε 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 συγχώνευση διατεταγμένων λιστών.

**13.** Δύο από τους συλλόγους που έχουν ιδρυθεί στη Μάνη είναι ‘Η Αγία Τριάς’ και ο ‘Διρός’. Να γραφεί πρόγραμμα που :

να διαβάζει τα ονόματα των μελών του συλλόγου ‘Η Αγία Τριάς’ και να τα αποθηκεύει σε μια λίστα list1. Επίσης, να διαβάζει τα ονόματα των μελών του συλλόγου ‘Διρός’ και να τα αποθηκεύει σε μια άλλη λίστα list2. Δίνεται ότι τα ονόματα που ανήκουν σε ένα σύλλογο είναι όλα διαφορετικά μεταξύ τους, ενώ κάποιος μπορεί να ανήκει και στους 2 συλλόγους.

Α)Να δημιουργεί συνάρτηση που να δέχεται ως παραμέτρους μια λίστα και μια συμβολοσειρά και να επιστρέφει True αν η συγκεκριμένη συμβολοσειρά είναι στοιχείο της λίστας. Διαφορετικά να επιστρέφει False.

Β)Με χρήση της προηγούμενης συνάρτησης να εμφανίζει τα ονόματα αυτών που ανήκουν στο σύλλογο ‘Αγία Τριάς’ και δεν ανήκουν στο σύλλογο ‘Διρός.

Γ)Να εμφανίζει τα ονόματα αυτών που ανήκουν στο σύλλογο ‘Διρός’ και δεν ανήκουν στο σύλλογο ‘Αγία Τριάς’.

Δ) Να εμφανίζει τα ονόματα αυτών που ανήκουν και στους 2 συλλόγους.

**14.** Σε μια έρευνα που πραγματοποίησε το υπουργείο Παιδείας κατέγραψε το πλήθος των μαθητών ανά σχολείο που σταμάτησαν τις σπουδές τους η λήξη του πρώτου τετράμηνου και στο τέλος του β’ τετραμήνου. Να γραφεί πρόγραμμα το οποίο:

Α. δημιουργεί την κλάση schools η οποία περιλαμβάνει τις παρακάτω ιδιότητες:

* Το όνομα του σχολείου
* Το πλήθος των μαθητών που σταμάτησαν στο πρώτο τετράμηνο το σχολείο
* Το πλήθος των μαθητών που σταμάτησαν στο δεύτερο τετράμηνο το σχολείο
* Τον ετήσιο αριθμό μαθητών που σταμάτησαν το σχολείο
* Μια μεταβλητή κλάσης που μετράει πόσα αντικείμενα της κλάσης υπάρχουν.

 Και τις παρακάτω μεθόδους:

* Μέθοδο που κατασκευάζει τα αντικείμενα και τα αρχικοποιεί εκτός από το ετήσιο πλήθος μαθητών που το θέτει 0.
* Μέθοδο που υπολογίζει τον ετήσιο αριθμό μαθητών που σταμάτησαν κσι ενημερώνει την αντίστοιχη ιδιότητα.
* Μέθοδο που εμφανίζει την ονομασία του σχολείου και τον ετήσιο αριθμό των μαθητών που σταμάτησαν.

Β. Για κάθε σχολείο που συμμετείχε στην έρευνα:

1. διαβάζει το όνομά του, το πλήθος των μαθητών που σταμάτησαν στο πρώτο τετράμηνο και αυτών που σταμάτησαν το δεύτερο εξάμηνο.

2. δημιουργεί αντικείμενο με τα παραπάνω δεδομένα και το αποθηκεύει στη λίστα sch.

Το τέλος του διαβάσματος γίνεται με την εισαγωγή ‘’ στην ονομασία του σχολείου.

Γ. Διατρέχει τη λίστα και πραγματοποιεί τα παρακάτω:

1. εμφανίζει το όνομα κάθε σχολείου και το συνολικό αριθμό των μαθητών που σταμάτησαν τις σπουδές τους.

2. εμφανίζει την ονομασία του σχολείου με τις περισσότερες ετήσιες απώλειες μαθητών.

**15.** a)Να υλοποιήσετε τις παρακάτω λειτουργίες της δομής δεδομένων “Στοίβα” συμπληρώνοντας κατάλληλα τις συναρτήσεις.

 def push(stack, item) : def isEmpty(stack) :

 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

def pop(stack) : def createStack( ) :

 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

 b) Να γράψετε στο τετράδιό σας τα γράμματα α, β από τη στήλη Α και τους αριθμούς 1, 2, 3, 4, 5, 6 από τη στήλη Β, έτσι ώστε να αντιστοιχίσετε τη σωστή λειτουργία σε κάθε δομή δεδομένων

|  |  |
| --- | --- |
| **Στήλη Α** |  **Στήλη Β** |
| α. Στοίβα | 1. Εισαγωγή |
| 2. Ώθηση |
| 3. Απώθηση |
| β. Ουρά | 4. Εξαγωγή |
| 5. FIFO |
| 6. LIFO |

**16.** a) Να γραφεί συνάρτηση vowels η οποία δέχεται ως παράμετρο μια λέξη και θα επιστρέφει το πλήθος των φωνηέντων.

b) Να γραφεί πρόγραμμα το οποίο :

b1) διαβάζει ονόματα πόλεων και θα τα καταχωρεί σε μια λίστα μέχρι να δοθεί ως όνομα η λέξη ΤΕΛΟΣ

b2) εμφανίζει τη λέξη που σχηματίζεται από το τελευταίο γράμμα κάθε στοιχείου της λίστας.

**17.** Μια μεταφορική εταιρεία διαθέτει 4 τύπους κοντέινερ για τις μεταφορές που πραγματοποιεί. Η χωρητικότητα κάθε τύπου φαίνεται στον παρακάτω πίνακα:

|  |  |
| --- | --- |
| **Τύπος** | **Χωρητικότητα** |
| Α | 20 |
| Β | 10 |
| C | 5 |
| D | 1 |

Να γραφεί πρόγραμμα Python το οποίο:

Α. να διαβάζει για καθένα από τους 30 πελάτες της μεταφορικής την ονομασία του πελάτη και τον αριθμό των τεμαχίων για μεταφορά και να τα καταχωρεί στις λίστες NAME και ΤΕΜ αντίστοιχα.

Β. να υπολογίζει και να εμφανίζει για κάθε πελάτη πόσα κοντέινερ από κάθε τύπο πρέπει να χρησιμοποιηθούν και να τα καταχωρεί στις λίστες ΤΑ, ΤΒ, ΤC, TD αντίστοιχα. Να θεωρήσετε ότι κάθε τύπος γεμίζει ώστε να μη μπορεί να χωρέσει άλλα μηχανήματα πριν περάσουμε στον επόμενο τύπο μικρότερης χωρητικότητας.

Γ. Να υπολογίζει και να εμφανίζει πόσα κοντέινερ κάθε τύπου θα χρειαστούμε για όλους τους πελάτες μαζί.

Δ. να εμφανίζει τις ονομασίες των πελατών ταξινομημένες κατά φθίνουσα σειρά ως προς τον συνολικό αριθμό τεμαχίων για μεταφορά.

**18.** Να γραφεί πρόγραμμα Python το οποίο να διαβάζει έναν εξαψήφιο αριθμό και να εμφανίζει το μικρότερο αριθμό που να έχει τα ίδια ακριβώς ψηφία. Για παράδειγμα, αν ο αρχικός αριθμός είναι ο 84652 ο τελικός αριθμός που θα προκύψει θα είναι 24568. (Υπόδειξη: δημιουργούμε μια λίστα με στοιχεία τα ψηφία του αριθμού) .