

ΘΕΩΡΙΑ ΚΕΦΑΛΑΙΟ 4

- 1) Τι ονομάζεται Δομή ακολουθίας; (σελ 46)
- 2) Τι ονομάζεται Δομή επιλογής if (σελ 46)
- 3) Σχεδιάστε το Διάγραμμα ροής για την απλή δομή if (σελ 47)
- 4) Σχεδιάστε το Διάγραμμα ροής για τη δομή if-else (σελ 48)
- 5) Αναφέρετε τις δομές επιλογής που γνωρίζετε
- 6) Αναφέρετε τις Δομές επανάληψης που γνωρίζετε
- 7) Τι τύπος επανάληψης είναι η for και τι τύπος επανάληψης είναι η while (σελ 51)
- 8) Στην εντολή for χρησιμοποιείται η συνάρτηση _____ για τον καθορισμό των επαναλήψεων. (σελ 51)
- 9) Η range() είναι μια _____ συνάρτηση της γλώσσας Python, η οποία, ανάμεσα σε άλλα, χρησιμοποιείται για την υπόδειξη του αριθμού των επαναλήψεων που θα εκτελεστούν σε ένα _____. Η δομή της είναι της μορφής range (αρχή, μέχρι, _____) (σελ 52)
- 10) Σχεδιάστε το Διάγραμμα ροής για τη δομή επανάληψης while (σελ 53)
- 11) Η εντολή "import random" εισάγει μια βιβλιοθήκη συναρτήσεων για την παραγωγή _____. (σελ 54)
- 12) Για να ορίσουμε μια δική μας συνάρτηση χρησιμοποιούμε τη χαρακτηριστική λέξη _____, ακολουθεί ένα όνομα που ταυτοποιεί την εκάστοτε συνάρτηση και ένα ζευγάρι παρενθέσεων που μπορούν να περικλείουν ονόματα μεταβλητών, ενώ η γραμμή τελειώνει με _____ (σελ 57)
- 13) Σε μια συνάρτηση οι παράμετροι καθορίζονται μέσα στο ζευγάρι των παρενθέσεων στον ορισμό της συνάρτησης και διαχωρίζονται με _____. Όταν καλούμε τη συνάρτηση, δίνουμε και τις τιμές με τον ίδιο τρόπο, οι οποίες τιμές ονομάζονται _____.

ΑΠΑΝΤΗΣΕΙΣ

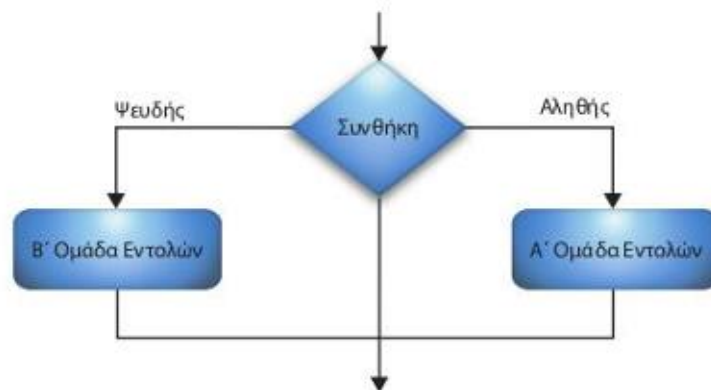
1) Δομή ακολουθίας : Πρόκειται για μια σειρά από εντολές που εκτελούνται η μία μετά την άλλη με τη σειρά. (σελ 46)

2) Η δομή επιλογής if (AN) χρησιμοποιείται, όταν θέλουμε να εκτελεστεί μια ακολουθία εντολών, μόνον, εφόσον πληρείται μία συγκεκριμένη συνθήκη. (σελ 46)

3) (σελ 47)



4) (σελ 48)



5) α) απλή δομή if, β) δομή if-else, γ) Πολλαπλή Επιλογή (elif), δ) Εμφωλευμένες δομές επιλογής

6) α) for , β) while

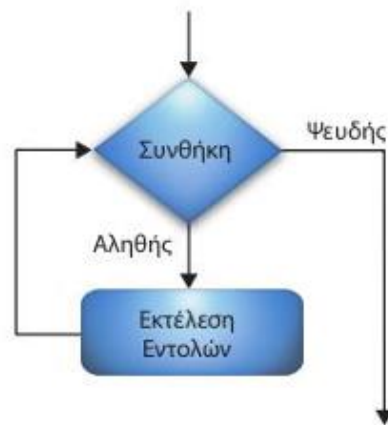
7) α) **for** : προκαθορισμένος, όπου το πλήθος των επαναλήψεων είναι δεδομένο

β) **while** : μη προκαθορισμένος, όπου το πλήθος των επαναλήψεων καθορίζεται κατά τη διάρκεια της εκτέλεσης των εντολών του σώματος της επανάληψης. (σελ 51)

8) α) **range()** (σελ 51)

9) α) ενσωματωμένη, β) βρόχο, γ) βήμα (σελ 52)

10) (σελ 53)



11) α) τυχαίων β) αριθμών. (σελ 54)

12) α) def, β) άνω και κάτω τελεία (:) (σελ 57)

13) α) κόμμα, β) ορίσματα (σελ 58) (διαφοροποιούνται από το κεφάλαιο των συναρτήσεων τα ορίσματα)

ΑΣΚΗΣΕΙΣ ΤΥΠΟΥ Α1

1) Να γράψετε στο τετράδιό σας τον αριθμό καθεμιάς από τις παρακάτω προτάσεις και δίπλα τη λέξη Σωστό αν είναι σωστή ή τη λέξη Λάθος αν είναι λανθασμένη.

1. Στην Python δεν έχει σημασία αν το όνομα μιας μεταβλητής είναι γραμμένο με κεφαλαία ή μικρά γράμματα.
2. Ένας μετρητής πρέπει να πάρει αρχική τιμή πριν από την εντολή επανάληψης.
3. Μπορούμε, στην ίδια μεταβλητή να εκχωρήσουμε αρχικά μία ακέραια τιμή και μετά μια συμβολοσειρά.
4. Η έκφραση $45 \% 10$ επιστρέφει 4.5
5. Η εντολή `open("words.txt", "w")` ανοίγει ένα αρχείο για εγγραφή και διατηρεί τα υπάρχοντα περιεχόμενά του.
6. Τα αλφαριθμητικά ή συμβολοσειρές στην Python είναι ακολουθίες από χαρακτήρες που έχουν σταθερό μέγεθος και μη μεταβαλλόμενα περιεχόμενα
7. Η Python παρέχει ένα μόνο τύπο υποπρογραμμάτων, τις συναρτήσεις, τις οποίες τις θεωρεί ως αντικείμενα.
8. Η Λίστα (List) ανήκει στους απλούς Τύπους Δεδομένων.
9. Η συνάρτηση `pow ( x , y )` επιστρέφει το ακέραιο πηλίκο και το ακέραιο υπόλοιπο της πράξης x δια y .
10. Η Συνάρτηση `open("words.txt", "a")`, αν δεν υπάρχει το αρχείο, το δημιουργεί, ενώ, αν υπάρχει, το ανοίγει σε κατάσταση προσθήκης δεδομένων στο τέλος του.
11. Ο αλγόριθμος της δυαδικής αναζήτησης δεν μπορεί να υλοποιηθεί σε ταξινομημένες λίστες.
12. Έστω ένα αρχείο `f` το οποίο έχει ανοιχτεί για ανάγνωση. Η εντολή `print f.read(1)` εκτυπώνει την πρώτη γραμμή του αρχείου στην οθόνη.
13. Η Python παρέχει ένα μόνο τύπο υποπρογραμμάτων, τις συναρτήσεις τις οποίες θεωρεί ως αντικείμενα.
14. Το πλεονέκτημα της απεριόριστης εμβέλειας είναι ότι περιορίζεται η ανεξαρτησία των υποπρογραμμάτων.
15. Στις λίστες τα δεδομένα που υπάρχουν δεν είναι απαραίτητα ίδιου τύπου μεταξύ τους.
16. Οι εντολές που περιλαμβάνονται μέσα στη δομή `while` θα εκτελεστούν τουλάχιστον μία (1) φορά.

17. 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 (:).
18. Αποδομητής (destructor) ονομάζεται η μέθοδος η οποία καταστρέφει αντικείμενα και ελευθερώνει τη μνήμη.
19. Η μέθοδος `fin.tell()` επιστρέφει έναν ακέραιο που περιέχει πάντα την τελευταία θέση του αρχείου.
20. Η λίστα, σε αντίθεση με τη συμβολοσειρά, είναι μία δυναμική δομή, στην οποία μπορούμε να προσθέτουμε ή να αφαιρούμε στοιχεία (mutable).
21. 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.
22. Η λειτουργία της ουράς είναι γνωστή στη βιβλιογραφία ως FIFO (First In First Out).
23. Η αρίθμηση των στοιχείων στις λίστες ξεκινάει από το 1.
24. Η συνάρτηση `open("words.txt", "r")` δημιουργεί το αρχείο `words.txt` αν αυτό δεν υπάρχει.
25. Η δυαδική αναζήτηση εφαρμόζεται στα στοιχεία μιας λίστας τα οποία βρίσκονται σε κάποια λογική διάταξη.
26. Η συνάρτηση `random` επιστρέφει έναν τυχαίο δεκαδικό ανάμεσα στο 0.0 και στο 1.0 (συμπεριλαμβανομένου του 0.0, αλλά όχι του 1.0)
27. Η δομή της στοίβας μπορεί να υλοποιηθεί στην Python με μια λίστα στην οποία οι εισαγωγές και οι εξαγωγές στοιχείων γίνονται μόνο από το ένα άκρο.
28. Η εντολή `elif` χρησιμοποιείται στην Python για τη σύνταξη σύνθετων δομών επιλογής.
29. Η `str()` δέχεται οποιαδήποτε αριθμητική τιμή και τη μετατρέπει σε ακέραιο αριθμό.
30. Οι τιμές που μεταβιβάζονται από ένα υποπρόγραμμα σε άλλο, λέγονται παράμετροι.
31. Αν ανοίξουμε ένα υπάρχον αρχείο με τη χρήση του ορίσματος `"a"`, τυχόν υπάρχοντα περιεχόμενά του θα διαγραφούν.
32. Κάθε συνάρτηση, όταν κληθεί, επιστρέφει πάντα κάποια τιμή.
33. Ο αλγόριθμος ταξινόμησης ευθείας ανταλλαγής μπορεί να τροποποιηθεί ώστε να τερματίσει, μόλις διαπιστώσει ότι η λίστα έχει ταξινομηθεί.
34. Οι μεταβλητές που μπορούν να χρησιμοποιηθούν σε οποιοδήποτε τμήμα ενός προγράμματος, ανεξάρτητα από το πού δηλώθηκαν, χαρακτηρίζονται ως καθολικές.
35. Η δομή `while` χρησιμοποιείται για μη προκαθορισμένο αριθμό επαναλήψεων.

- 36 . Η μέθοδος `L.pop()` προσθέτει ένα στοιχείο στο τέλος της λίστας `L` .
- 37 . Ο αριθμός `28.2E - 5` είναι ένας αριθμός κινητής υποδιαστολής .
- 38 . Η `int ( x )` μετατρέπει σε ακέραιο την αριθμητική τιμή `x` .
- 39 . Ένα αντικείμενο δημιουργείται από μία ειδική μέθοδο που ονομάζεται αποδομητής (`destructor`) .
- 40 . Στη γλώσσα προγραμματισμού `Python` για τη χρησιμοποίηση μιας μεταβλητής δεν απαιτείται η δήλωσή της .
- 41 . Στη γλώσσα προγραμματισμού `Python` χρησιμοποιούμε την εντολή `for` για να εκτελεστεί ένα τμήμα του κώδικα για έναν καθορισμένο αριθμό επαναλήψεων .
- 42 . Η συνάρτηση `row (4,2)` επιστρέφει την τιμή 8.
- 43 . Η μέθοδος λίστας `L.append ( object )`, όπου `L` το όνομα της λίστας χρησιμοποιείται για προσθήκη του στοιχείου `object` στο τέλος της λίστας `L` .
- 44 . Στον αντικειμενοστραφή προγραμματισμό τα χαρακτηριστικά (`attributes`) ενός αντικειμένου ονομάζονται και μέθοδοι.
- 45 . Το μειονέκτημα των καθολικών μεταβλητών (`global`) είναι ότι περιορίζουν την ανεξαρτησία των υποπρογραμμάτων.
- 46 . Η κλήση μιας συνάρτησης γίνεται με την εντολή `call` `όνομα_συνάρτησης()` .
- 47 . Η εκχώρηση τιμής σε μια μεταβλητή γίνεται με το σύμβολο `" == "` .
- 48 . Μία συνάρτηση μπορεί να κληθεί και μέσα από μία άλλη συνάρτηση.
- 49 . Η συνάρτηση `range (10, 1, - 2)` επιστρέφει τη λίστα `[10, 8, 6, 4, 2]` .

ΑΠΑΝΤΗΣΕΙΣ

1)

1. Σωστό

2. Σωστό

3. Σωστό

4. Λάθος

5. Λάθος

6. Σωστό

7. Σωστό

8. Λάθος

9. Λάθος

10. Σωστό

11. Λάθος

12. Λάθος

13. Σωστό

14. Λάθος

15. Σωστό

16. Λάθος

17. Σωστό

18. Σωστό

19. Λάθος

20. Σωστό

20. Λάθος

22. Σωστό

23. Λάθος

24. Λάθος

25. Σωστό

26. Σωστό

27. Σωστό

28. Σωστό

29. Λάθος

30. Σωστό

31. Λάθος

32. Λάθος

33. Σωστό

34. Σωστό

35. Σωστό

36. Λάθος

37. Σωστό

38. Σωστό

39. Λάθος

40. Σωστό

41. Σωστό

42. Λάθος

43. Σωστό

44. Λάθος

45. Σωστό

46. Λάθος

47. Λάθος

48. Σωστό

49. Σωστό

ΑΣΚΗΣΕΙΣ ΤΥΠΟΥ Α2

1) Ποιες λίστες αριθμών παράγουν οι παρακάτω συναρτήσεις range;

i. range(5)

ii. range(1, 5)

iii. range(5, -1, -2)

2) Να μετατρέψετε τις παρακάτω προτάσεις σε εντολές στην γλώσσα προγραμματισμού Python.

α. μειώστε την μεταβλητή x κατά 3 μονάδες.

β. εκχωρήστε στην μεταβλητή z το μέσο όρο των μεταβλητών a και b.

γ. αν η μεταβλητή k είναι μεγαλύτερη του μηδενός (0) να εμφανίζει «thetikos arithmos».

δ. να εμφανίσετε τη λέξη «kalimera».

3) Αντιστοιχήστε τα στοιχεία της Στήλης Α με τα στοιχεία της Στήλης Β

ΣΤΗΛΗ Α	ΣΤΗΛΗ Β
1. and	α. Σχεσιακός Τελεστής
2. sqrt()	β. Αριθμητικός Τελεστής
3. or	γ. Πράξη σύζευξης
4. %	δ. Πράξη διάζευξης
5. abs()	ε. Συνάρτηση που επιστρέφει την απόλυτη τιμή
	στ. Συνάρτηση που επιστρέφει την τετραγωνική ρίζα ενός αριθμού

4) Να γράψετε στο τετράδιό σας το αποτέλεσμα που εμφανίζεται στην οθόνη μετά την εκτέλεση του παρακάτω προγράμματος :

Πρόγραμμα	Οθόνη Η/Υ
L=[13,5] L=L+[6,24] print L L.append(20) print L L.pop(2) print L L.pop() print L L.insert(0,1) print L	α. β. γ. δ. ε.

5) Να χαρακτηρίσετε καθεμιά από τις ακόλουθες λογικές εκφράσεις ως True ή False αν $x=5$ και $y=2$:

α . $x==6$

β . $x>4$ and $y!=1$

γ . $x<=5$ or $(y*2<3)$

δ . $(x>10$ and $y<3)$ or $(2*y>4)$

ε . $x>12$ and $y<10$ or $(3*y>9)$

6) Αντιστοιχήστε τα στοιχεία της Στήλης Α με τα στοιχεία της Στήλης Β

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

7) 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. $\text{not}(4<9)$

β. $4<\text{len}(\text{"καλημέρα"})$

γ. $(4>5)$ or $(9>2)$

δ. $(4==4)$ and $\text{not}(4>9)$

ε. $\text{pow}(3,0)==9-8$

8) Αντιστοιχήστε τα στοιχεία της Στήλης Α με τα στοιχεία της Στήλης Β

ΣΤΗΛΗ Α	ΣΤΗΛΗ Β
1. $12 / 4 \% 2$	α. True
2. $\text{not}(56<=12)$	β. False
3. $45 / 10$	γ. 4.5
4. $(12 < 11)$ and $(23 > 10)$	δ. 1
5. $45.0 / 10$	ε. 4
6. $2 * (5 \% 4) + 4 / (1 + 3)$	στ. 5
	ζ. 3

9) Δίνεται το παρακάτω τμήμα προγράμματος Python:

```
for i in range (0, 100, 5)
```

```
    print i
```

Το τμήμα αυτό του προγράμματος εμφανίζει διαδοχικά τους αριθμούς 0, 5, 10, ... , 95. Να τροποποιήσετε τον παραπάνω κώδικα έτσι ώστε αυτοί να εμφανίζονται σε αντίστροφη σειρά.

10) Να χαρακτηρίσετε καθεμιά από τις ακόλουθες λογικές εκφράσεις ως True ή False.

α. $34 != 45$

β. $56 <= 12$

γ. $(12 < 11)$ and $(23 > 10)$

δ. $(12 < 11)$ or $(23 > 10)$

ε. $\text{not}(56 <= 12)$

11) Δίνεται το παρακάτω τμήμα προγράμματος Python:

```
for x in range (A, M, B):
```

```
    print x
```

Για καθεμιά από τις παρακάτω περιπτώσεις, να γράψετε στο τετράδιό σας τις τιμές των A, M, B, έτσι ώστε το αντίστοιχο τμήμα προγράμματος να εμφανίζει όλους :

α. τους ακέραιους από 1 μέχρι και 80 (αύξουσα σειρά)

β. τους ακέραιους από 50 μέχρι και 20 (φθίνουσα σειρά)

γ. τους περιττούς ακέραιους από 81 μέχρι και 151 (αύξουσα σειρά)

δ. τους ακέραιους από -50 μέχρι και -5 (αύξουσα σειρά)

ε. τους θετικούς ακέραιους που είναι μικρότεροι του 200 και πολλαπλάσιοι του 7 (αύξουσα σειρά).

12) Αντιστοιχήστε τα στοιχεία της Στήλης Α με τα στοιχεία της Στήλης Β

ΣΤΗΛΗ Α	ΣΤΗΛΗ Β
1 divmod()	α. Σχεσιακός Τελεστής
2 not	β. Αριθμητικός Τελεστής
3 ==	γ. Τελεστής Λογικής Πράξης
4 %	δ. Συνάρτηση Ενσωματωμένη
	ε. Μη Ενσωματωμένη Συνάρτηση

13) Δίνεται λίστα που περιέχει όλα τα θετικά πολλαπλάσια του 3 μέχρι και το 99. Το τμήμα προγράμματος Python που ακολουθεί αντιγράφει τα στοιχεία της λίστας σε ένα νέο αρχείο κειμένου, με όνομα Pol.txt. Κάθε στοιχείο γράφεται σε μια διαφορετική γραμμή. Στο τμήμα αυτό υπάρχουν υπογραμμισμένα κενά τα οποία έχουν αριθμηθεί.

```
pollaplasia=range(3,       (1),       (2) )  
myfile=open("Pol.txt",       (3) )  
for number in pollaplasia:  
    myfile.       (4) (       (5) (number) + "\n")  
myfile.       (6) ()
```

Να γράψετε στο τετράδιό σας τους αριθμούς(1), (2), (3), (4),(5) και(6) που αντιστοιχούν στα κενά του παραπάνω τμήματος προγράμματος και δίπλα σε κάθε αριθμό, αυτό που πρέπει να συμπληρωθεί.

14) Σε μια μεταβλητή τύπου ακεραίου(integer) με όνομα x αποθηκεύεται η βαθμολογία ενός μαθητή. Οι επιτρεπτές τιμές είναι από 1 μέχρι και 20. Να γράψετε στο τετράδιό σας ποια από τις παρακάτω εκφράσεις ελέγχει αυτή τη συνθήκη.

α) $(x \leq 1) \text{ and } (x \geq 20)$

β) $(x \geq 1) \text{ or } (x \leq 20)$

γ) $(x \geq 1) \text{ and } (x \leq 20)$

δ) $(x \leq 1) \text{ or } (x \leq 20)$

15) Αντιστοιχήστε τα στοιχεία της Στήλης Α με τα στοιχεία της Στήλης Β

ΣΤΗΛΗ Α	ΣΤΗΛΗ Β
1. str ()	α. Λογικός τελεστής
2. True	β. Συγκριτικός τελεστής
3. " False "	γ. Λογική τιμή
4. or	δ. Συμβολοσειρά
5. ==	ε. Αριθμητικός τελεστής
	στ . Συνάρτηση μετατροπής μιας τιμής σε συμβολοσειρά

16) Να γράψετε στο τετράδιό σας το αποτέλεσμα που εμφανίζεται στην οθόνη μετά την εκτέλεση καθεμιάς από τις παρακάτω εντολές :

α . range(2,10)

β . range(2,10,3)

γ . a = " abc "

print a*2

δ . x= 2

y=3

print 2 * x+y

ε . a = 2

print a **3

17) Να χαρακτηρίσετε καθεμιά από τις ακόλουθες λογικές εκφράσεις ως True ή False αν $x = 3$ και $y = 1$:

α . not(x>y)

β . (x>5) or (y<2)

γ . (x!=5) and (y!=0)

δ . (x<y) or (x**2>y)

ε . x<len(" abc ")

18) Δίνεται η παρακάτω κλάση:

class K inito:

```
def __init__(self, marka, model):  
    self.marka=marka  
    self.model=model  
  
def fortizi ( self ):  
    print "το κινητό φορτίζει"
```

Με βάση την παραπάνω ορισμένη κλάση:

α) Ποιος είναι ο κατασκευαστής (constructor) της κλάσης .

β) Να προσθέσετε την ιδιότητα `cpu _ cores` που αντιπροσωπεύει το πλήθος των πυρήνων του επεξεργαστή και την ιδιότητα `cam _ resolution` που αντιπροσωπεύει την ανάλυση της κάμερας σε Mpixel ώστε να αρχικοποιούνται στον κατασκευαστή .

γ) Να δημιουργήσετε ένα στιγμιότυπο της κλάσης, δηλαδή ένα αντικείμενο με όνομα `phone 1` του οποίου οι τιμές των ιδιοτήτων του θα οριστούν κατά τη δημιουργία του ως εξής:

`marka = " orange " , model = " S 3 " , cpu _ cores = 4 , cam _ resol ution = 10 .`

19) Αντιστοιχήστε τα στοιχεία της Στήλης Α με τα στοιχεία της Στήλης Β

ΣΤΗΛΗ Α	ΣΤΗΛΗ Β
1. <code>range(1,10,1)</code>	α. <code>[10]</code>
2. <code>range(10,1,-1)</code>	β. <code>[]</code>
3. <code>range(10,-1,-1)</code>	γ. <code>[10,9,8,7,6,5,4,3,2]</code>
4. <code>range(10,0,-1)</code>	δ. <code>[1,2,3,4,5,6,7,8,9]</code>
5. <code>range(10,10,1)</code>	ε. <code>[10,9,8,7,6,5,4,3,2,1,0]</code>
	στ. <code>[10,9,8,7,6,5,4,3,2,1]</code>

20) Να μεταφέρετε και να συμπληρώσετε στο τετράδιό σας τον παρακάτω πίνακα με τα αποτελέσματα των πράξεων μεταξύ τριών μεταβλητών X , Y , Z .

X	Y	Z	$X > Y$ and not($Y <= Z$)	$X > 2$ and $Y < 4$ or $Z >= 5$
10	5	3		
2	5	6		
12	3	5		

21) Ο παρακάτω αλγόριθμος επιστρέφει τη θέση του στοιχείου `key` αν υπάρχει μέσα στη λίστα `array`, σε διαφορετική περίπτωση επιστρέφει `-1`

```
def binarySearch( array, key ) :
    first = 0
    last = len(array) - 1
    pos = (1)
    while first <= last and pos (2) -1 :
        mid = ( first + last ) / 2
        if array[ mid ] == key :
            pos = (3)
        elif array[ mid ] (4) key :
            first = mid + 1
        else :
            last = mid - 1
    return pos
```

Στο τμήμα προγράμματος υπάρχουν υπογραμμισμένα κενά τα οποία έχουν αριθμηθεί. Να γράψετε στο τετράδιό σας τους αριθμούς 1, 2, 3 και 4 που αντιστοιχούν στα κενά του παραπάνω τμήματος προγράμματος και δίπλα σε κάθε αριθμό αυτό που πρέπει να συμπληρωθεί ώστε να υλοποιείται σωστά η δυαδική αναζήτηση που επιστρέφει τη θέση `pos` ενός στοιχείου `key` μέσα σε μία λίστα `array`.

22) Αντιστοιχήστε τα στοιχεία της Στήλης Α με τα στοιχεία της Στήλης Β

ΣΤΗΛΗ Α	ΣΤΗΛΗ Β
1. x	α. Συμβολοσειρά (string)
2. 256.14	β. Μεταβλητή
3. 28	γ. Μιγαδικός αριθμός
4. True	δ. Αριθμός κινητής υποδιαστολής (float)
5. "True"	ε. Τιμή Λογικού τύπου (boolean)
	στ. Ακέραιος (integer)

23) Δίνονται τα παρακάτω τρία (3) τμήματα προγραμμάτων :

1. a = 5 while a - 1 != 4: print a a += 1	2. i = 3 while i <= 12: i += 2 print i	3. y = 2 while y > - 3: print y y - = 1
---	--	---

Να γράψετε τον αριθμό του τμήματος προγράμματος και δίπλα το πλήθος των επαναλήψεων που θα πραγματοποιηθούν σε καθένα από αυτά.

ΑΠΑΝΤΗΣΕΙΣ

1) i. 0, 1, 2, 3, 4 ii. 1, 2, 3, 4 iii. 5, 3, 1	2) α. $x=x-3$ β. $z=(a+b)/2.0$ γ. if $k>0$: print "thetikos arithmos" δ. print "kalimera"
---	--

3) 1. γ 2. στ 3. δ 4. β 5. ε	4) α. [13, 5, 6, 24] β. [13, 5, 6, 24, 20] γ. [13, 5, 24, 20] δ. [13, 5, 24] ε. [1, 13, 5, 24]
--	--

5) 1. False 2. True 3. True 4. False 5. False	6) 1 α 2 γ 3 δ 4 β
---	---------------------------------------

7) α. False β. True γ. True δ. True ε. True	8) 1 δ 2 α 3 ε 4 β 5 γ 6 ζ
---	---

9) for i in range (95, -5, -5): print i	10) α. True β. False γ. False δ. True ε. True
--	---

11) α. A=1, M=81, B=1 β. A=50, M=19, B=-1 γ. A=81, M=152, B=2 δ. A=-50, M=-6, B=1 ε. A=7, M=200, B=7	12) 1δ 2γ 3α 4β
--	------------------------------------

13) (1) 100 (2) 3 (3) 'w' (4) write (5) str (6) close	14) γ
15) 1. στ 2. γ 3. δ 4. α 5. β	16) α. 2 3 4 5 6 7 8 9 β. 2 5 8 γ. abcabc δ. 7 ε. 8
17) α. False β. True γ. True δ. True ε. False	
18) α) Ο κατασκευαστής της κλάσης είναι η ειδική μέθοδος <code>def __init__(self,marka,model)</code> β) <pre>def __init__(self,marka,model,cpu_cores,cam_resolution): self.marka=marka self.model=model self.cpu_cores=cpu_cores self.cam_resolution=cam_resolution</pre> γ) <code>phone1= Kinito('orange', 'S3', 4 , 10)</code>	
19) 1. δ 2. γ 3. ε 4. στ 5. β	20) True False False True False True
21) 1) -1 2) == 3) mid 4) <	22) 1. β 2. δ 3. στ 4. ε 5. α
23) 1. καμία 2. 5 3. 5	

ΑΣΚΗΣΕΙΣ ΤΥΠΟΥ Β

1) Έστω το παρακάτω τμήμα προγράμματος. Να βρείτε τις τιμές όλων των μεταβλητών σε κάθε επανάληψη:

```
K=-5
X=-7
while X<=0:
    M=K+ 4
    if M<= 0 :
        M,X=X,M
    else:
        X=6*M
    K=K+1
```

2) Έστω το παρακάτω τμήμα προγράμματος. Να βρείτε τις τιμές όλων των μεταβλητών σε κάθε επανάληψη:

```
def F1(Y,X):
    while Y<5:
        print X+2*Y
        Y+=2

X=1
Y=2
F1(X,Y)
```

3) Τι θα εμφανίσει το παρακάτω πρόγραμμα;

```
def F2(X,Y,Z):

    X=Y-Z
    Z=X+Y
    print X,Y,Z

X=27
Y=2
Z=13

F2(X,Y,Z)
print X,Y,Z

F2(Z, X, Y)

F2(Y, Z, X)
```

4) Τι θα εμφανίσει το παρακάτω πρόγραμμα;

```
def F3(K, T):  
    for x in range(1, K):  
        T=T-1  
        print T  
    return T  
  
S=0  
T=0  
  
for x in range(5, 1, -2):  
    S=S+F3(x, T)  
  
print S
```

5) Τι θα εμφανίσει το παρακάτω πρόγραμμα;

```
B=[5, 8]  
  
print B + [10,20,30, range(-5,-8,-2)]
```

6) Τι θα εμφανίσει το παρακάτω πρόγραμμα;

```
X= [range(5, 20, 5), range(-5,-10,-2), range(3) ]  
  
print X  
  
print X[ 1 ] [ 2 ]  
  
print X[2]
```

7) Έστω το παρακάτω τμήμα προγράμματος. Να βρείτε τις τιμές όλων των μεταβλητών σε κάθε επανάληψη:

```
K=8  
L=11  
M=0  
  
while L>0:  
    if L % 2 ==1:  
        M=M+K  
    K=K*2  
    L=L//2  
    print "L",L  
    print "K", K  
    print "M", M  
    print M
```

ΑΠΑΝΤΗΣΕΙΣ

1)

K	M	X
-5		-7
	-1	
-4	-7	-1
	0	
-3	-1	0
-2	1	6

2)

X έξω	Y έξω	X μέσα	Y μέσα
1	2		
		2	1
		2	3
		2	5

3) -11 2 -9

 27 2 13

 25 27 52

 -14 13 -1

4) -4 -2 -6

5) [5, 8, 10, 20, 30, [-5, -7]]

6) [[5, 10, 15], [-5, -7, -9], [0, 1, 2]]

-9

[0, 1, 2]

7)

L	K	M
11	8	0
5	16	8
2	32	24
1	64	24
0	128	88

ΑΣΚΗΣΕΙΣ ΤΥΠΟΥ Β

1) Δίνεται το παρακάτω πρόγραμμα σε γλώσσα προγραμματισμού Python.

```
i=10
sum=0
while i<=100 :
    sum =sum + i
    i=i+20
print i, sum
```

Να γράψετε στο τετράδιό σας :

- A. Ποια είναι η αρχική τιμή της μεταβλητής i;
- B. Ποιες είναι οι διαδοχικές τιμές που θα πάρουν οι μεταβλητές i, sum;
- C. Ποιο είναι το περιεχόμενο των μεταβλητών i, sum στο τέλος του προγράμματος;

2) Δίνεται το παρακάτω τμήμα προγράμματος σε γλώσσα προγραμματισμού Python:

```
x=input("Dose timi sto x'")
if x==1 :
    y=x+5
if x==2 :
    y=x* 5+8
if x==3 :
    y=2* x-x
if x==4 :
    y=(x+ x* 5) // 7
if x>4 :
    y=(x // 3)+(x % 3)
print y
```

B1. Έστω ότι η θετική ακέραια μεταβλητή εισόδου x ($x>0$) παίρνει τις ακόλουθες τιμές :
α)1, β)7, γ)4, δ)12, ε)3, στ)2.

Να γράψετε στο τετράδιό σας τις τιμές της μεταβλητής εξόδου γ που θα εμφανιστούν στην οθόνη για κάθε μία από τις παραπάνω τιμές εισόδου.

3) Δίνεται ο παρακάτω αλγόριθμος σε λογικό διάγραμμα, όπου οι μεταβλητές X, A, B, C είναι ακέραιες:

```
X=10
A=20
while X>0:
    if A%2 ==0:
        A=A+3
    else:
        A=A-1
    X=X // 2
    print X, A
B=X+A
C= 2 * A
print B, C
```

A) Να μεταφέρετε στο τετράδιό σας και να συμπληρώσετε τον παρακάτω πίνακα με τις τιμές των μεταβλητών X, A που εμφανίζονται σε κάθε επανάληψη.

	X	A
ΑΡΧΙΚΕΣ ΤΙΜΕΣ	10	20
1 ^η επανάληψη		
2 ^η επανάληψη		
3 ^η επανάληψη		
4 ^η επανάληψη		

B) Ποιες είναι οι τιμές των μεταβλητών B, C που θα εμφανιστούν;

4) Δίνεται το παρακάτω πρόγραμμα σε γλώσσα προγραμματισμού Python:

```
k=32
m=10
while k>=8 :
    k , a= divmod(k, 2)
    m=m+k
    print k, m
```

Να μεταφέρετε στο τετράδιό σας και να συμπληρώσετε τον παρακάτω πίνακα με τις τιμές των μεταβλητών k, m, που εμφανίζονται σε κάθε επανάληψη.

	k	m
ΑΡΧΙΚΕΣ ΤΙΜΕΣ	32	10
1 ^η επανάληψη		
2 ^η επανάληψη		
3 ^η επανάληψη		

ΑΠΑΝΤΗΣΕΙΣ

1)		2)	
i	sum	1 → 6	
10	0	7 → 3	
30	10	4 → 3	
50	40	12 → 4	
70	90	3 → 3	
90	160	2 → 18	
110	250		

3)

	X	A	B	C
ΑΡΧΙΚΕΣ ΤΙΜΕΣ	10	20		
1 ^η επανάληψη	5	23		
2 ^η επανάληψη	2	22		
3 ^η επανάληψη	1	25		
4 ^η επανάληψη	0	24	24	28

4)

	k	m
ΑΡΧΙΚΕΣ ΤΙΜΕΣ	32	10
1 ^η επανάληψη	16	26
2 ^η επανάληψη	8	34
3 ^η επανάληψη	4	38

5)

plithos	sum	x	z	y
0	0	100.0	0	
1	1	50	0	1
2	6	25	1	5
3	14	12	0	8
4	24	6	0	10
5	27	3	1	3

ΑΣΚΗΣΕΙΣ ΤΥΠΟΥ B

B1. Δίνεται η παρακάτω συνάρτηση σε γλώσσα προγραμματισμού Python που υλοποιεί το αλγόριθμο της δυαδικής αναζήτησης ενός στοιχείου `key` μέσα σε μία λίστα `array`.

```
def binarySearch( array, key ) :  
    first = ( 1 )  
  
    last = ( 2 )  
  
    found = ( 3 )  
  
    while first <= last and not found :  
        mid = ( first + last ) / 2  
  
        if array[ mid ] == key :  
            found = True  
  
        elif array[ mid ] < key :  
            first = ( 4 )  
  
        else :  
            last = ( 5 )  
  
    return found
```

Στο τμήμα προγράμματος υπάρχουν υπογραμμισμένα κενά τα οποία έχουν αριθμηθεί. Να γράψετε στο τετράδιό σας 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 δυαδική αναζήτηση.

B2. Δίνεται το πρόγραμμα:

```
x=4
```

```
S=0
```

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

```
    S=S+i
```

```
    x=S%i
```

```
    print "S=",S,
```

```
    print "x=", x
```

α. Να γράψετε στο τετράδιό σας ότι 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.

B3. Δίνεται η παρακάτω κλάση:

```
class ypallilos:

    def __init__(self, eponymo, onoma,misthos):

        self.eponymo=eponymo

        self.onoma=onoma

        self.misthos=misthos

    def afksisemistho(self,amount):

        self.misthos=self.misthos+amount

    def emfanise_stoixeia(self):

        print "Επώνυμο ",self.eponymo

        print "Όνομα ",self.onoma

        print "Μισθός ",self.misthos
```

α . Ποιος είναι ο κατασκευαστής (constructor) της κλάσης;

β Ποιες είναι οι ιδιότητες της κλάσης;

γ. Ποιες είναι οι μέθοδοι της κλάσης.

δ. Να προσθέσετε την ιδιότητα `χρονια` που θα αφορά τα χρόνια υπηρεσίας του υπαλλήλου και να αρχικοποιείται στον κατασκευαστή (Δεν χρειάζεται να εμφανίζεται στην `emfanise_stoixeia()`) .

ε . Να προσθέσετε μία μέθοδο `afxisexchronia(self,amount)` η οποία να αυξάνει τα χρόνια υπηρεσίας κατά `amount`.

στ. Να δημιουργήσετε τα παρακάτω στιγμιότυπα της κλάσης:

I. Αντικείμενο με όνομα αντικειμένου `yp1` και Επώνυμο `"ΑΝΤΩΝΙΟΥ"` , όνομα `"ΑΝΤΩΝΙΟΣ"`, Μισθό 1500 και χρόνια υπηρεσίας 15 .

II. Αντικείμενο με όνομα αντικειμένου `yp2` και Επώνυμο `"ΠΑΠΑΔΟΠΟΥΛΟΥ"` , όνομα `"ΕΛΕΝΗ"`, Μισθό 2000 και χρόνια υπηρεσίας 20 .

ζ. Να καλέσετε την κατάλληλη μέθοδο, ώστε ο μισθός της ΠΑΠΑΔΟΠΟΥΛΟΥ να αυξηθεί κατά 200.

B4. A) Οι παρακάτω συναρτήσεις αφορούν την υλοποίηση της Ουράς. Να τις ξαναγράψετε στο τετράδιό σας συμπληρώνοντας τα κενά:

```
def enqueue(queue, item) :
```

```
    _____
```

```
def dequeue(queue) :
```

```
    _____
```

```
def isEmpty(queue) :
```

```
    _____
```

```
def createQueue( ) :
```

```
    _____
```

B) Τι θα εμφανιστεί στην οθόνη του Η/Υ μετά την εκτέλεση των παρακάτω εντολών

```
A=createQueue()
```

```
enqueue(A,10)
```

```
enqueue(A,20)
```

```
print dequeue(A)
```

```
enqueue(A,40)
```

```
if not isEmpty(A):
```

```
    enqueue(A,87)
```

```
    enqueue(A,-6)
```

```
print dequeue(A)
```

```
print dequeue(A)
```

ΑΠΑΝΤΗΣΕΙΣ

B1	B2 α.	B2 β.
1. 0 2. len(array) – 1 3. False 4. mid + 1 5. mid – 1	S= 10 x= 0 S= 18 x= 2 S= 24 x= 0 S= 28 x= 0	x=4 S=0 i=10 while i>3: S=S+i x=S%i print "S=",S print " x=",x i=i-2
B3 α.		B3 β.
Ο κατασκευαστής είναι: <pre>def __init__(self, eponymo,onoma,misthos): self.eponymo=eponymo self.onoma=onoma self.misthos=misthos</pre>		Οι ιδιότητες είναι : eponymo, onoma, misthos
B3 γ.		B3 ε.
Οι μέθοδοι είναι: <pre>def afksisemistho(self,amount): self.misthos=self.misthos+amount def emfanise_stoixeia(self): print "Επώνυμο ",self.eponymo print "Όνομα ",self.onoma print "Μισθός ",self.misthos</pre>		<pre>def afxisexronia(self,amount): self.xronia=self.xronia+amount</pre>
B3 δ.		B3 ζ.
Έχουμε μόνο αλλαγή στο: <pre>def __init__(self, eponymo, onoma, misthos, xronia): self.onoma=onoma self.eponymo=eponymo self.misthos=misthos self. xronia = xronia</pre>		γρ2. afksisemistho(200)
B3 στ.		
I. γρ1= γρallilos("ΑΝΤΩΝΙΟΥ","ΑΝΤΩΝΙΟΣ",1500,20) II. γρ2= γρallilos("ΠΑΠΑΔΟΠΟΥΛΟΥ"," ΕΛΕΝΗ",2000,20)		

B4 A
<pre>def enqueue(queue, item) : queue = queue.append(item) def dequeue(queue) : return queue.pop(0) def isEmpty(queue) : return len(queue) == 0 def createQueue() : return []</pre>
B4 B
10 20 40

B4 Γ

```
class Queue :  
    def __init__(self) :  
        self.items = []  
    def enqueue(self, item) :  
        self.items.append( item )  
    def dequeue(self) :  
        return self.items.pop( 0 )  
    def isEmpty(self) :  
        return len(self.items)==0
```

B4 Δ

```
A=Queue()  
A.enqueue(10)  
A.enqueue(20)  
print A.dequeue()  
A.enqueue(40)  
if not A.isEmpty():  
    A.enqueue(87)  
    A.enqueue(-6)  
print A.dequeue()  
print A.dequeue()
```

ΑΣΚΗΣΕΙΣ ΘΕΩΡΙΑ ΣΥΜΒΟΛΟΣΕΙΡΕΣ ΛΙΣΤΕΣ

1) Τι θα εμφανιστεί στην οθόνη του υπολογιστή μετά την εκτέλεση της εντολής: `print 3*"test"`

- A) Σφάλμα
- B) `3*"test"`
- Γ) `testtesttest`
- Δ) `3test`

2) Τι θα εμφανιστεί στην οθόνη του υπολογιστή μετά την εκτέλεση του παρακάτω τμήματος προγράμματος;

```
S="new"  
print "E" not in S
```

- A) True
- B) False

3) Τι τιμή θα έχει η σύγκριση: `"dimitris">"dimitra"`

- A) True
- B) False

4) Τι θα εμφανιστεί στην οθόνη του υπολογιστή μετά την εκτέλεση του παρακάτω τμήματος προγράμματος;

```
A="epal"  
print len(A)
```

- A) 3
- B) 4

5) Τα αλφαριθμητικά ή συμβολοσειρές στην Python είναι ακολουθίες από χαρακτήρες που δεν έχουν σταθερό μέγεθος και έχουν μεταβαλλόμενα περιεχόμενα. Σ Λ

6) Μια δομή δεδομένων μπορεί να οριστεί ως ένα σχήμα οργάνωσης σχετικών μεταξύ τους στοιχείων δεδομένων. Σ Λ

7) Τι τιμή θα έχει η σύγκριση: `str(480)=='480'`

- A) True
- B) False

8) Τι εμφανιστεί στην οθόνη του υπολογιστή μετά την εκτέλεση της εντολής: `print int("20")+5`

- A) "20"+5
- B) 25
- Γ) 205
- Δ) "20"5

9) Τι θα εμφανιστεί στην οθόνη του υπολογιστή μετά την εκτέλεση του παρακάτω τμήματος προγράμματος;

```
A="TEST ENA"  
print A[1],A[6]
```

- A) T E
- B) E N
- Γ) T N
- Δ) E A

10) Τι εμφανιστεί στην οθόνη του υπολογιστή μετά την εκτέλεση της εντολής: `print "10"+"10"`

- A) 20
- B) "20"
- Γ) "10"+"10"
- Δ) 1010

11) Ποιο από τα παρακάτω προγράμματα παρουσιάζει τα στοιχεία μιας λίστας L με την αντίστροφη σειρά

```
L=[12,15,2,17,23,48]
```

```
N=len(L)
for i in range(1,N-1,-1):
    print L[i]
```

A)

```
L=[12,15,2,17,23,48]
```

```
N=len(L)
for i in range(N,-1,-1):
    print L[i]
```

B)

```
L=[12,15,2,17,23,48]
```

```
N=len(L)
for i in range(N-1,-1,-1):
    print L[i]
```

Γ)

```
L=[12,15,2,17,23,48]
```

```
N=len(L)
for i in range(N-1,0,-1):
    print L[i]
```

Δ)

12) Η συνάρτηση list (String) επιστρέφει μια λίστα με στοιχεία τους χαρακτήρες της συμβολοσειράς string.

- A) True
- B) False

13) Έστω ότι έχουμε τη λίστα L=[2,5,8]. Ποιο θα είναι το αποτέλεσμα της σύγκρισης: 8 in L

- A) True
- B) False

14) Αν έχουμε μία λίστα L, η εντολή L+= [10] προσθέτει το στοιχείο 10 στην αρχή της λίστας L Σ Λ

15) Η αρίθμηση των στοιχείων σε μια λίστα ξεκινάει από το 1. Σ Λ

16) Μπορούμε να έχουμε σε μια λίστα ακόμα και στοιχεία διαφορετικού τύπου π.χ. αριθμούς, συμβολοσειρές κλπ. Σ Λ

17) Θεωρούμε ότι έχουμε μία λίστα L. Η εντολή L.append(object)

- A) Αφαιρεί το τελευταίο στοιχείο της λίστας L
- B) Προσθέτει το στοιχείο object στο τέλος της λίστας L.
- Γ) Προσθέτει το στοιχείο object στην αρχή της λίστας L.
- Δ) Όλα τα παραπάνω

18) Δίνεται το παρακάτω τμήμα προγράμματος. Τι θα εμφανιστεί στην οθόνη του υπολογιστή;

```
L=[1,5,7,9,2]
L.append(13)
L.pop(2)
L.pop()
L.insert(2,4)
print L
```

- A) [1, 5, 4, 9, 2]
- B) [1, 7, 4, 9, 2]
- Γ) [1, 5, 9, 2, 2]
- Δ) [1, 7, 9, 2, 2]

19) Δίνεται η λίστα L=[4,5,9] και η λίστα K=[1,2,3]. Τι θα εμφανιστεί στη οθόνη του υπολογιστή μετά την εκτέλεση της εντολής: print L+K

- A) [1,2,3,4,5,6]
- B) [4,5,9][1,2,3]
- Γ) [4,5,9]+[1,2,3]
- Δ) [4,5,9,1,2,3]

20) Η λίστα είναι μια δυναμική δομή στην οποία μπορούμε να προσθέτουμε ή να αφαιρούμε στοιχεία (mutable) Σ Λ

21) Ποια από τις παρακάτω συναρτήσεις δέχεται μία λίστα σαν είσοδο και επιστρέφει σωστά τη μέγιστη τιμή της;

```
def MAX(L):  
    N=len(L)  
    Max=L[0]  
    for i in range(N):  
        if Max>L[i]:  
            Max=L[i]  
    return Max
```

A)

```
def MAX(L):  
    N=len(L)  
    Max=L[0]  
    for i in range(N):  
        if Max<L[i]:  
            Max=L[i]  
    return Max
```

B)

```
def MAX(L):  
    N=len(L)  
    Max=L[0]  
    for i in range(N-1):  
        if Max>L[i]:  
            Max=L[i]  
    return Max
```

Γ)

```
def MAX(L):  
    N=len(L)  
    Max=L[0]  
    for i in range(N-1):  
        if Max<L[i]:  
            Max=L[i]  
    return Max
```

Δ)

22) Δίνεται το παρακάτω τμήμα προγράμματος. Τι θα εμφανιστεί στην οθόνη του υπολογιστή ;

```
def enqueue(queue, item) :  
    queue = queue.append( item )
```

```
def dequeue(queue) :  
    return queue.pop( 0 )
```

```
def isEmpty(queue) :  
    return len(queue) == 0
```

```
def createQueue( ) :  
    return []
```

```
L=createQueue()  
enqueue(L,3)  
enqueue(L,5)  
dequeue(L)  
enqueue(L,8)  
enqueue(L,11)  
dequeue(L)
```

```
print L
```

A) [11, 8]

B) [3, 8]

Γ) [8, 3]

Δ) [8, 11]

23) Δίνεται το παρακάτω τμήμα προγράμματος. Τι θα εμφανιστεί στην οθόνη του υπολογιστή ;

```
def push(stack, item) :  
    stack.append( item )
```

```
def pop(stack) :  
    return stack.pop( )
```

```
def isEmpty(stack) :  
    return len(stack) == 0
```

```
def createStack( ) :  
    return []
```

```
L=createStack()  
push(L,5)  
push(L,10)  
push(L,13)  
pop(L)  
push(L,15)  
print L
```

- A) [10, 13, 15]
- B) [5, 10, 15]
- Γ) [15, 10, 5]
- Δ) [15, 13, 10]

24) Η δομή της Στοιβάς μπορεί να υλοποιηθεί στην Python με μια λίστα στην οποία οι εισαγωγές και οι εξαγωγές στοιχείων γίνονται μόνο από το ένα άκρο Σ Λ

25) Το σύνολο των επικεφαλίδων των συναρτήσεων το οποίο είναι διαθέσιμο στον προγραμματιστή που χρησιμοποιεί τη Στοιβά, το ονομάζουμε διεπαφή ή διασύνδεση (interface) της δομής αυτής Σ Λ

26) Στην δομή δεδομένων Ουρά οι λειτουργίες εισαγωγής και εξαγωγής είναι γνωστές στη βιβλιογραφία ως ώθηση (push) και απώθηση (pop). Σ Λ

27) Η λειτουργία της Ουράς χαρακτηρίζεται ως LIFO (Last In First Out) Σ Λ

28) Μια δομή δεδομένων που χρησιμοποιείται για την μοντελοποίηση και προσομοίωση πραγματικών φαινομένων, είναι η δομή της Ουράς. Τα φαινόμενα αυτά μελετώνται από διάφορους κλάδους των Μαθηματικών και της Πληροφορικής, όπως είναι η Θεωρία Ουρών (Queueing theory) και η Επιχειρησιακή Έρευνα (Operations Research) Σ Λ

29) Η Στοιβά είναι μία δομή δεδομένων στην οποία το στοιχείο που προστέθηκε τελευταίο είναι και το πρώτο που θα εξαχθεί Σ Λ

ΑΠΑΝΤΗΣΕΙΣ

1	testtesttest
2	True
3	True
4	4
5	ΛΑΘΟΣ
6	ΣΩΣΤΟ
7	True
8	25
9	E N
10	1010
11	Γ
12	True
13	True
14	ΛΑΘΟΣ
15	ΛΑΘΟΣ
16	ΣΩΣΤΟ
17	B
18	A
19	Δ
20	ΣΩΣΤΟ
21	B
22	Δ
23	B
24	ΣΩΣΤΟ
25	ΣΩΣΤΟ
26	ΛΑΘΟΣ
27	ΛΑΘΟΣ
28	ΣΩΣΤΟ
29	ΣΩΣΤΟ

ΘΕΩΡΗΤΙΚΕΣ ΑΣΚΗΣΕΙΣ ΣΥΝΑΡΤΗΣΕΙΣ

1) Σύμφωνα με τον Τμηματικό Προγραμματισμό , μπορούμε να γράψουμε ένα πρόγραμμα ως ένα σύνολο από μικρότερα κομμάτια προγράμματος Σ Λ

2) Μια συνάρτηση μπορεί να καλείται από διάφορα σημεία του κύριου προγράμματος απαγορεύεται όμως να καλείται μέσα από μια άλλη συνάρτηση Σ Λ

3) Ποια είναι τα βασικά χαρακτηριστικά που πρέπει να έχει ένα υποπρόγραμμα;

A) Να έχει μόνο ένα σημείο εισόδου από το οποίο δέχεται τα δεδομένα του.

B) Το (υπο)πρόγραμμα το οποίο καλεί ένα άλλο υποπρόγραμμα σταματάει την εκτέλεσή του όσο εκτελείται το καλούμενο υποπρόγραμμα. Μόνο ένα υποπρόγραμμα μπορεί να εκτελείται σε μια χρονική στιγμή.

Γ) Ο έλεγχος επιστρέφει στο (υπο)πρόγραμμα το οποίο καλεί, όταν το καλούμενο υποπρόγραμμα σταματήσει να εκτελείται.

Δ) Όλα τα παραπάνω

4) Η Python παρέχει ένα μόνο τύπο υποπρογραμμάτων, τις συναρτήσεις, τις οποίες τις θεωρεί ως αντικείμενα. Σ Λ

5) Η εμβέλεια (scope) μιας μεταβλητής αναφέρεται στο τμήμα του προγράμματος που μπορεί αυτή να έχει πρόσβαση Σ Λ

6) Οι μεταβλητές που ισχύουν μόνο για το υποπρόγραμμα στο οποίο δηλώθηκαν ονομάζονται καθολικές (global) Σ Λ

7) Ένα άρθρωμα είναι ένα Python αντικείμενο. Πιο απλά είναι ένα αρχείο αποτελούμενο από κώδικα Python και μπορεί να ορίσει συναρτήσεις, κλάσεις και μεταβλητές. Σ Λ

8) Για να χρησιμοποιήσουμε ένα άρθρωμα στο πρόγραμμά μας, θα πρέπει να το "εισάγουμε" σε αυτό. Αυτό γίνεται με τη δήλωση import. Σ Λ

9) from math import sqrt

Μετά την παραπάνω εντολή, η εντολή που ακολουθεί είναι σωστή;
print sqrt(25)

Σ Λ

10) Ένα άρθρωμα έχει μέσα του πολλές πρότυπες βιβλιοθήκες Σ Λ

11)

```
def allaxe(x):  
    x=13
```

```
x=2  
allaxe(x)  
print x
```

Τι θα εμφανιστεί στη οθόνη του υπολογιστή μετά την εκτέλεση του παραπάνω προγράμματος

- A) 13
- B) 2
- Γ) Τίποτα
- Δ) Θα εμφανίζει συντακτικό λάθος

12)

```
def allaxe(x):  
    x=10  
    print x,y
```

```
x=2  
y=3
```

```
allaxe(x)
```

Τι θα εμφανιστεί στη οθόνη του υπολογιστή μετά την εκτέλεση του παραπάνω προγράμματος

- A) Τίποτα
- B) Θα εμφανίζει συντακτικό λάθος
- Γ) 2 3
- Δ) 10 3

13)

```
def allaxe(x):  
    x=4  
    y=5  
    print "Από μέσα ",x,y
```

```
x=10  
y=3
```

```
allaxe(x)  
print "Από έξω ",x,y
```

Τι θα εμφανιστεί στη οθόνη του υπολογιστή μετά την εκτέλεση του παραπάνω προγράμματος

A) Από μέσα 4 5

Από έξω 10 3

B) Από μέσα 10 3

Από έξω 10 3

Γ) Από μέσα 10 5

Από έξω 10 3

Δ) Από μέσα 4 3

Από έξω 10 3

14)

```
def allaxe(x):  
    global y  
    x=2  
    y=200  
    print "Από μέσα ",x,y
```

```
x=1  
y=7
```

```
allaxe(x)  
print "Από έξω ",x,y
```

Τι θα εμφανιστεί στη οθόνη του υπολογιστή μετά την εκτέλεση του παραπάνω προγράμματος

A) Από μέσα 2 200

Από έξω 1 7

Β) Από μέσα 2 200
Από έξω 2 200

Γ) Από μέσα 2 200
Από έξω 1 200

Δ) Από μέσα 2 7
Από έξω 1 7

15)

```
import random  
for i in range(50):  
    print random.randrange(6)
```

Τι θα εμφανιστεί στην οθόνη του υπολογιστή μετά την εκτέλεση του παραπάνω προγράμματος;

- Α) Θα εμφανίζει συντακτικό λάθος
Β) Θα εμφανίσει 50 τυχαίους ακέραιους από το 0 έως και το 6
Γ) Θα εμφανίσει 50 τυχαίους ακέραιους από το 1 έως και το 6
Δ) Θα εμφανίσει 50 τυχαίους ακέραιους από το 0 έως και το 5

ΑΠΑΝΤΗΣΕΙΣ

1	Σ
2	Λ
3	Όλα τα παραπάνω
4	Σ
5	Σ
6	Λ
7	Σ
8	Σ
9	Σ
10	Λ
11	2
12	10 3
13	Από μέσα 4 5 Από έξω 10 3
14	Από μέσα 2 200 Από έξω 1 200
15	Θα εμφανίσει 50 τυχαίους ακέραιους από το 0 έως και το 5

ΑΣΚΗΣΕΙΣ ΤΥΠΟΥ Γ ΚΑΙ Δ

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΗΜΕΡΗΣΙΩΝ ΚΑΙ ΕΣΠΕΡΙΝΩΝ ΕΠΑΛ 2009

1) Μία εταιρεία κινητής τηλεφωνίας ακολουθεί ανά μήνα την πολιτική τιμών, που φαίνεται στον παρακάτω πίνακα:

Πάγιο 4,5 €	
ΑΡΙΘΜΟΣ ΜΗΝΥΜΑΤΩΝ	ΧΡΕΩΣΗ ΑΝΑ ΜΗΝΥΜΑ
1-50	0,10 €
51-150	0,08 €
από 151 και άνω	0,05 €

Να αναπτύξετε αλγόριθμο σε Python ο οποίος:

- Να διαβάζει τον αριθμό των μηνυμάτων ενός συνδρομητή στο τέλος ενός μήνα.
- Να υπολογίζει τη μηνιαία χρέωση του συνδρομητή. Η χρέωση των μηνυμάτων είναι κλιμακωτή με βάση τον παραπάνω πίνακα.
- Να εμφανίζει (τυπώνει) τη λέξη «ΧΡΕΩΣΗ» και τη μηνιαία χρέωση του συνδρομητή.

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΗΜΕΡΗΣΙΩΝ ΚΑΙ ΕΣΠΕΡΙΝΩΝ ΕΠΑΛ 2010

2) Να γράψετε στο τετράδιό σας πρόγραμμα σε γλώσσα Python, το οποίο:

- Να διαβάζει το επώνυμο του υποψηφίου με τη χρήση κατάλληλου μηνύματος.
- Να διαβάζει τους βαθμούς στα τέσσερα μαθήματα του υποψηφίου με τη χρήση κατάλληλου μηνύματος.
- Να υπολογίζει το μέσο όρο και να κατατάσσει τον υποψήφιο σε μια από τις τέσσερις κατηγορίες, ανάλογα με την επίδοσή του, δηλαδή:

κλίμακα	1-150	151-300	301-350	351-400
χαρακτηρισμός	D	C	B	A

Δ 4. Να εμφανίζει σε ποια κατηγορία ανήκει ο κάθε υποψήφιος.

Δ5. Η επανάληψη θα συνεχίζεται έως ότου στη θέση του επωνύμου γραφτεί η λέξη: τέλος .

Υποδείξεις :

- Η αποδεκτή βαθμολογία σε κάθε μάθημα είναι από 1 ... 400 και δε χρειάζεται να γίνει έλεγχος ορθότητας τιμών
- Όπου απαιτείται επανάληψη, να γίνει χρήση μόνο της εντολής while.

ΑΣΚΗΣΗ 1

```
M=input("Δώσε αριθμό μηνυμάτων ")

if M>=1 and M<= 50:
    Xr=M*0.10 + 4.5

if M>=51 and M<= 150:
    Xr=50*0.10 + (M-50)*0.08 + 4.5

if M>=151:
    Xr=50*0.10 + 100*0.08 + (M-150)*0.05 + 4.5

print "ΧΡΕΩΣΗ ",Xr
```

ΑΣΚΗΣΗ 2

```
On=raw_input("Δώσε ονομα")

while On!="τελος":

    V1=input("Δώσε βαθμό 1")

    V2=input("Δώσε βαθμό 2")

    V3=input("Δώσε βαθμό 3")

    V4=input("Δώσε βαθμό 4")

    MO=(V1+V2+V3+V4)/4.0

    if MO>=1 and MO<=150:
        print "D"
    if MO>=151 and MO<=300:
        print "C"
    if MO>=301 and MO<=350:
        print "B"
    if MO>=351 and MO<=400:
        print "A"

    On=raw_input("Δώσε ονομα")
```

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΗΜΕΡΗΣΙΩΝ ΚΑΙ ΕΣΠΕΡΙΝΩΝ ΕΠΑΛ 2011

1) Το σύνολο των σχολείων μιας πόλης αποφάσισε να επισκεφθούν το Ενυδρείο της περιοχής τους. Η χρέωση για την είσοδο των μαθητών ανά σχολείο θα γίνει σύμφωνα με τον παρακάτω πίνακα:

Τιμή εισιτηρίου ανά μαθητή 10 Ευρώ	
1 έως και 20 μαθητές	Κανένα δωρεάν εισιτήριο
21 έως και 40 μαθητές	Δώρο το κόστος 5 εισιτηρίων
41 και άνω μαθητές	Δώρο το κόστος 9 εισιτηρίων

Να γραφεί αλγόριθμος ο οποίος:

Γ1. Να διαβάζει το πλήθος N των σχολείων της πόλης.

Γ 2. Για καθένα από τα σχολεία να διαβάζει το όνομά του και το πλήθος των μαθητών που θα συμμετάσχουν.

Γ 3. Να υπολογίζει το πληρωτέο ποσό κάθε σχολείου ανάλογα με το πλήθος των μαθητών του.

Γ4. Να εμφανίζει το όνομα και το ποσό πληρωμής κάθε σχολείου.

Υποδείξεις :

α. Ο υπολογισμός δεν γίνεται κλιμακωτά. Για παράδειγμα σχολείο 50 μαθητών θα τύχει δώρου 9 εισιτηρίων και θα πληρώσει 410 Ευρώ.

β. Δεν χρειάζεται να γίνει έλεγχος ορθότητας τιμών.

2) Μια εταιρεία παραγωγής γραφικής ύλης που διανέμει τα προϊόντα της μέσω πωλητών της, επιθυμεί στο τέλος της χρονιάς να ελέγξει την απόδοσή τους.

Να γράψετε στο τετράδιό σας πρόγραμμα σε γλώσσα Python το οποίο:

Δ1. Να διαβάζει το όνομα του πωλητή.

Δ2. Το ανωτέρω (Δ 1) να επαναλαμβάνεται έως ότου δοθεί για όνομα πωλητή η τιμή 'ΤΕΛΟΣ'.

Δ3. Κατά τη διάρκεια της επανάληψης να διαβάζεται το ποσό των ετήσιων πωλήσεων κάθε πωλητή και μετά το τέλος των επαναλήψεων να έχουν υπολογιστεί τα ακόλουθα:

α) Το πλήθος των πωλητών με πωλήσεις ≥ 50000 Ευρώ.

β) Το πλήθος των πωλητών με πωλήσεις < 50000 Ευρώ.

γ) Το συνολικό ποσό των πωλήσεων όλων των πωλητών.

Δ4. Στο τέλος των επαναλήψεων να:

α) τυπώσει το πλήθος των πωλητών με πωλήσεις ≥ 50000 Ευρώ

β) τυπώσει το πλήθος των πωλητών με πωλήσεις < 50000 Ευρώ

γ) τυπώσει το συνολικό ποσό των πωλήσεων όλων των πωλητών

δ) υπολογίσει και να τυπώσει το μέσο όρο των πωλήσεων όλων των πωλητών.

Υποδείξεις :

α. Η εταιρεία διαθέτει τουλάχιστον έναν πωλητή.

β. Για την επανάληψη να γίνει χρήση της εντολής while.

γ. Στις εντολές εισόδου και εξόδου να υπάρχουν τα κατάλληλα μηνύματα.

ΑΣΚΗΣΗ 1

```

N=input("Δώσε αριθμό σχολείων")

for x in range (N):
    on=raw_input(" Δώσε όνομα σχολείου")
    a=input(" Δώσε αριθμό μαθητών")

    if a >=1 and a<=20:
        Xr=a*10

    if a >=21 and a<=40:
        Xr=(a-5)*10

    if a >=41 :
        Xr=(a-9)*10

print "Το σχολείο ",on," θα πληρώσει ",Xr

```

ΑΣΚΗΣΗ 2

```

c1=0
c2=0
S=0.0

ON=input("Δώσε όνομα πωλητή")

while ON!="ΤΕΛΟΣ":

    E=input("Δώσε ετήσιες πωλήσεις ")

    if E>=50000:
        c1+=1
    else:
        c2+=1

    S=S+E

    ON=input("Δώσε όνομα πωλητή")

print "πλήθος πωλητών με πωλήσεις >=50000 ",c1

print "πλήθος πωλητών με πωλήσεις <50000 ",c2

print "συνολικό ποσό πωλήσεων ",S

print "μέσος όρος πωλήσεων ",S/(c1+c2)

```

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΗΜΕΡΗΣΙΩΝ ΚΑΙ ΕΣΠΕΡΙΝΩΝ ΕΠΑΛ 2012

1) Μια επιχείρηση έχει 50 υπαλλήλους. Η διοίκηση έκλεισε συμφωνία για την παροχή πακέτων σύνδεσης με πρόσβαση στο Διαδίκτυο μέσω κινητού τηλεφώνου για κάθε υπάλληλο. Το πακέτο έχει **πάγιο 3 ευρώ** και η χρέωση είναι κλιμακωτή με βάση τον παρακάτω πίνακα:

Όγκος δεδομένων σε GB (Gigabyte)	Ευρώ ανά GB
έως και 2	10
3 έως και 6	2,5
7 και άνω	1,5

Να γραφεί αλγόριθμος σε Python, ο οποίος:

Γ1. Να διαβάζει το όνομα και τον όγκο δεδομένων σε GB κάθε υπαλλήλου.

Γ 2. Να υπολογίζει για καθέναν τη χρέωσή του, με το πάγιο.

Γ 3. Να εμφανίζει το όνομα και τη συνολική του χρέωση.

Γ 4. Να υπολογίζει και να εμφανίζει το μέσο όρο της χρέωσης όλων των υπαλλήλων, με το πάγιο.

Υποδείξεις :

α. Δε χρειάζεται να γίνει έλεγχος ορθότητας τιμών, ούτε να υπάρχουν τα κατάλληλα μηνύματα στις εντολές εισόδου και εξόδου.

β. Ο όγκος δεδομένων παίρνει ακέραιες τιμές.

γ. Παράδειγμα χρέωσης: ένας υπάλληλος με όγκο δεδομένων 5GB θα χρεωθεί 27.5 ευρώ χωρίς το πάγιο.

2) Μια ναυτιλιακή εταιρεία σε ένα οχηματαγωγό της πλοίο και μόνο σε σχέση με τα οχήματα, εφαρμόζει την τιμολογιακή πολιτική που φαίνεται στον παρακάτω πίνακα:

Τύπος οχήματος	Χρέωση ανά όχημα
Μηχανή	10 ευρώ
Αυτοκίνητο ΙΧ	20 ευρώ
Φορτηγό	30 ευρώ

Ο οδηγός δεν πληρώνει εισιτήριο, ενώ κάθε επιπλέον επιβάτης του οχήματος πληρώνει 5 ευρώ.

Να γραφεί πρόγραμμα σε Python, το οποίο:

Δ1. Να διαβάζει τον τύπο του οχήματος ('Μ' για μηχανή, 'Α' για αυτοκίνητο, 'Φ' για φορτηγό) και τον αριθμό των επιβατών του (μαζί με τον οδηγό).

Δ2. Να υπολογίζει το κόστος για κάθε όχημα, στο οποίο να συμπεριλαμβάνεται και το κόστος των επιβατών.

Δ3. Η διαδικασία (Δ1-Δ2) επαναλαμβάνεται για όλα τα οχήματα και μέχρι να δοθεί η τιμή 'ΤΕΛΟΣ' στον τύπο του οχήματος.

Δ4. Μετά την επανάληψη να εμφανίζονται:

α) Το πλήθος των φορτηγών.

β) Η συνολική χρέωση όλων των οχημάτων μαζί με τους επιβάτες τους.

Υποδείξεις :

α. Να γίνει έλεγχος ορθότητας τιμών στον τύπο του οχήματος.

β. Το πρόγραμμα δεν ασχολείται με τους επιβάτες άνευ οχήματος.

γ. Για την επανάληψη να γίνει χρήση της εντολής while.

δ. Στο πλοίο εισέρχεται τουλάχιστον ένα όχημα, ενώ όλα τα οχήματα είναι με οδηγό.

ΑΣΚΗΣΗ 1

```
S=0.0
for x in range(50):
    ON=raw_input("Δώσε όνομα")
    D=input("Δώσε όγκο δεδομένων ")

    if D>=1 and D<= 2:
        Xr=D*10 + 3

    if D>=3 and D<= 6:
        Xr=2*10 + (D-2)*2.50 + 3

    if D>=7:
        Xr=2*10 + 5*2.50 + (D-7)*1.50 + 3

    print "Ο Υπάλληλος ",ON, "θα πληρώσει ",Xr

    S=S+Xr

print "μέσος όρος χρέωσης όλων των υπαλλήλων",Xr/50
```

ΑΣΚΗΣΗ 2

```
S=0
SF=0

OX=raw_input("Δώσε τύπο οχήματος")
while OX not in ["M", "A", "Φ"]:
    OX=raw_input("Δώσε ΣΩΣΤΟ τύπο οχήματος")

while OX!="ΤΕΛΟΣ" :

    A=input("Δώσε αριθμό επιβατών ")

    if OX=="M":
        K=10 + 5*(A-1)

    if OX=="A":
        K=20 + 5*(A-1)

    if OX=="Φ":
        K=30 + 5*(A-1)
        SF+=1

    S=S+K

    OX=raw_input("Δώσε τύπο οχήματος")
    while OX not in ["M", "A", "Φ"]:
        OX=raw_input("Δώσε ΣΩΣΤΟ τύπο οχήματος")

print "συνολική χρέωση όλων των οχημάτων",S

print "πλήθος των φορτηγών:",SF
```

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΗΜΕΡΗΣΙΩΝ ΚΑΙ ΕΣΠΕΡΙΝΩΝ ΕΠΑΛ 2013

1) Ένα σχολείο πρόκειται να πάει μια εκπαιδευτική επίσκεψη. Στην προσφορά, που έγινε από ένα ταξιδιωτικό γραφείο, περιλαμβάνεται το κόστος ανά μαθητή, χωρίς τη διατροφή ξενοδοχείο, όπως φαίνεται στον παρακάτω πίνακα:

Αριθμός μαθητών	Κόστος ανά μαθητή
Από 1 έως και 100	80€
Από 101 και πάνω	60€

Αν το σχολείο επιλέξει, να έχουν οι μαθητές και διατροφή στο ξενοδοχείο, το κόστος αυξάνεται κατά 30€ ανά μαθητή.

Να γραφεί πρόγραμμα σε γλώσσα προγραμματισμού Python, το οποίο:

Γ1. Να διαβάζει τον αριθμό των μαθητών, που θα συμμετάσχουν στην εκπαιδευτική επίσκεψη. Επίσης, να διαβάζει την απάντηση του σχολείου σχετικά με τη διατροφή των μαθητών ('ΝΑΙ' αν επιθυμούν, 'ΟΧΙ' αν δεν επιθυμούν διατροφή).

Γ2. Να υπολογίζει και να εμφανίζει το συνολικό κόστος της εκπαιδευτικής επίσκεψης, χωρίς τη διατροφή.

Γ3. Να υπολογίζει το συνολικό κόστος της διατροφής, στην περίπτωση που το σχολείο την έχει επιλέξει.

Γ4. Να υπολογίζει και να εμφανίζει το συνολικό κόστος της εκπαιδευτικής επίσκεψης.

Υποδείξεις :

α. Να γίνει έλεγχος ορθότητας τιμών στην απάντηση του σχολείου σχετικά με τη διατροφή

β. Αν ο αριθμός των μαθητών είναι μικρότερος ή ίσος του 100, τότε όλοι οι μαθητές θα πληρώσουν από 80€ ο καθένας, ενώ αν ο αριθμός τους είναι μεγαλύτερος του 100, τότε όλοι οι μαθητές θα πληρώσουν από 60€ ο καθένας.

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΕΠΑ.Λ.

2) Μια αεροπορική εταιρεία πρόκειται να δρομολογήσει μία πτήση με αεροπλάνο 100 θέσεων. Τα λειτουργικά έξοδα της πτήσης είναι 5.000€ και όλα τα εισιτήρια έχουν διατεθεί. Η τιμή του εισιτηρίου, που πλήρωσε ο κάθε επιβάτης, μπορεί να κυμαινόταν από 20€ έως και 200€.

Να γράψετε στο τετράδιό σας, έναν αλγόριθμο σε Python, ο οποίος:

Δ1. Να διαβάζει το όνομα του κάθε επιβάτη.

Δ2. Να διαβάζει την τιμή του εισιτηρίου του κάθε επιβάτη με έλεγχο ορθότητας τιμών.

Δ3. Να υπολογίζει τις συνολικές εισπράξεις της εταιρείας από τη συγκεκριμένη πτήση.

Δ4. Να υπολογίζει και να εμφανίζει το όνομα του επιβάτη με τη μικρότερη τιμή εισιτηρίου. Να θεωρήσετε, ότι η μικρότερη τιμή είναι μοναδική.

Δ5. Να εμφανίζει το μήνυμα «ΚΕΡΔΟΣ», αν οι συνολικές εισπράξεις είναι μεγαλύτερες από τα λειτουργικά έξοδα ή το μήνυμα «ΖΗΜΙΑ», αν οι συνολικές εισπράξεις είναι μικρότερες από τα λειτουργικά έξοδα ή το μήνυμα «ΜΗΔΕΝΙΚΟ ΑΠΟΤΕΛΕΣΜΑ», αν είναι ίσα.

Υποδείξεις :

Δε χρειάζεται να υπάρχουν τα κατάλληλα μηνύματα στις εντολές εισόδου και εξόδου, εκτός του ερωτήματος Δ 5.

ΑΣΚΗΣΗ 1

```
N=input("Δώσε αριθμό μαθητών")

D=raw_input("Θα υπάρξει διατροφή των μαθητών;")
while D not in ["ΝΑΙ","ΟΧΙ"]:
    D=raw_input("Δώσε ΝΑΙ ή ΟΧΙ")

if N>=1 and N<=100:
    Xr=N*80
if N>100:
    Xr=N*60

print "κόστος χωρίς τη διατροφή",Xr

if D=="ΝΑΙ":
    XD=30*N
    print "κόστος διατροφή",XD

print "συνολικό κόστος της εκπαιδευτικής επίσκεψης",Xr+XD
```

ΑΣΚΗΣΗ 2

```
min1=201

S=0
for x in range(100):
    ON=raw_input("Δώσε ονομα")
    T=input("Δώσε τιμή εισιτηρίου ")
    while T<20 or T>200:
        T=input("Δώσε ΣΩΣΤΗ τιμή εισιτηρίου ")

    S=S+T

    if T<min1:
        min1=T
        onoma=ON

print "Ο επιβάτης",onoma, "έχει το φθηνότερο εισιτήριο"

if S<5000:
    print "ΚΕΡΔΟΣ"
if S>5000:
    print "ΖΗΜΙΑ"
if S==5000:
    print "ΜΗΔΕΝΙΚΟ ΑΠΟΤΕΛΕΣΜΑ"
```

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΗΜΕΡΗΣΙΩΝ ΚΑΙ ΕΣΠΕΡΙΝΩΝ ΕΠΑΛ 2014

1) Μια ασφαλιστική εταιρία στον κλάδο του αυτοκινήτου προσφέρει τις παρακάτω τιμές ετήσιου ασφαλιστρου ανάλογα με τον κυβισμό του αυτοκινήτου.

Κυβισμός σε cc	Τιμή ετήσιου ασφαλιστρου σε €
Έως και 1000	150
Από 1001 έως και 2000	200
Πάνω από 2000	300

Σε περίπτωση που ο οδηγός είναι νέος, δηλαδή έχει ηλικία μικρότερη ή ίση των 23 ετών, τότε τα ασφάλιστρα αυξάνονται κατά 40€. Στην τιμή που προκύπτει από όλα τα προηγούμενα προστίθεται ΦΠΑ 23%.

Να γραφεί αλγόριθμος σε Python, ο οποίος:

Γ1. Να διαβάζει τον κυβισμό ενός αυτοκινήτου καθώς και την ηλικία του οδηγού.

Γ2. Να υπολογίζει τα ετήσια ασφάλιστρα του αυτοκινήτου με βάση τον κυβισμό του και την ηλικία του οδηγού.

Γ3. Να υπολογίζει και να εμφανίζει το τελικό ποσό ετήσιων ασφαλιστρων συμπεριλαμβανομένου του ΦΠΑ.

Υποδείξεις :

Δε χρειάζεται να γίνει έλεγχος ορθότητας τιμών, ούτε να υπάρχουν κατάλληλα μηνύματα στις εντολές εισόδου και εξόδου.

ΑΣΚΗΣΗ 1

```
K=input("Δώσε κυβισμό αυτοκινήτου ")  
  
H=input("Δώσε την ηλικία του οδηγού")  
  
if K>=1 and K<=1000:  
    A=150  
  
if K>=1001 and K<=2000:  
    A=200  
  
if K>2000:  
    A=300  
  
if H<=23:  
    A=A+40  
  
SA=A + A*23/100  
  
print "τελικό ποσό ετήσιων ασφαλίσεων ",SA
```

ΑΣΚΗΣΗ 2

```
S=0  
c=0  
  
for x in range(143):  
    N=raw_input("Δώσε νομό")  
    M=input("Δώσε πλήθος μαθητών ")  
    A=input("Δώσε πλήθος των απόντων μαθητών")  
  
    S=S+(M-A)  
  
    if A==0:  
        c=c+1  
  
print "Σύνολο μαθητών που προσήλθαν στις εξετάσεις",S  
  
print "πλήθος εξεταστικών κέντρων χωρίς απόντες",c
```

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΗΜΕΡΗΣΙΩΝ ΚΑΙ ΕΣΠΕΡΙΝΩΝ ΕΠΑΛ 2015

1) Ένα κατάστημα ηλεκτρονικών ειδών προσφέρει τα παρακάτω ποσοστά εκπτώσεων, ανάλογα με τον τρόπο πληρωμής που θα επιλέξει ο πελάτης:

Τρόπος πληρωμής	Ποσοστό έκπτωσης (%)
"ΜΕΤΡΗΤΑ"	20
"ΚΑΡΤΑ"	10
"ΔΟΣΕΙΣ"	0

Να γράψετε έναν αλγόριθμο σε Python, ο οποίος:

Γ1. Να διαβάζει με τη χρήση κατάλληλων μηνυμάτων:

α) τη συνολική αξία πριν από την έκπτωση των προϊόντων που αγόρασε ο πελάτης.

β) τον τρόπο πληρωμής.

(Δε χρειάζεται να γίνει έλεγχος ορθότητας τιμών)

Γ2. Να υπολογίζει το ποσό της έκπτωσης ανάλογα με τον τρόπο πληρωμής.

Γ3. Να εμφανίζει το ποσό της έκπτωσης.

Γ4. Να υπολογίζει και να εμφανίζει το τελικό ποσό πληρωμής.

Γ5. Στην περίπτωση που το τελικό ποσό πληρωμής είναι μεγαλύτερο από 200€, να εμφανίζει το μήνυμα «Κερδίσατε Δώρο».

2) Το Υπουργείο Περιβάλλοντος αποφάσισε να παρακολουθήσει για τριάντα (30) ημέρες τα επίπεδα ενός ρύπου στην ατμόσφαιρα, πραγματοποιώντας μία μέτρηση την ημέρα.

Έχουν καθοριστεί τρία επίπεδα μόλυνσης με βάση την τιμή του ρύπου, όπως φαίνεται στον παρακάτω πίνακα:

Τιμές ρύπου	Επίπεδα μόλυνσης
έως και 1	«Φυσιολογικό»
πάνω από 1 έως και 2	«Οριακό»
πάνω από 2	«Επικίνδυνο»

Να γράψετε ένα πρόγραμμα σε γλώσσα προγραμματισμού Python, το οποίο:

Δ1. Για κάθε μία από τις τριάντα (30) ημέρες να διαβάζει την τιμή του ρύπου με τη χρήση κατάλληλου μηνύματος (δε χρειάζεται να γίνεται έλεγχος ορθότητας τιμών).

Δ2. Να εμφανίζει για κάθε μέρα το επίπεδο μόλυνσης ανάλογα με την τιμή του ρύπου.

Δ3. Να υπολογίζει και να εμφανίζει το πλήθος των ημερών κατά τη διάρκεια των οποίων η τιμή του ρύπου ξεπέρασε την τιμή 3.

Δ4. Να υπολογίζει και να εμφανίζει τον μέσο όρο των τιμών του ρύπου για το διάστημα των τριάντα (30) ημερών.

ΑΣΚΗΣΗ 1

```
A=input("Δώσε συνολική αξία πριν απο την έκπτωση ")  
T=raw_input("Δώσε τρόπο πληρωμής")
```

```
if T=="ΜΕΤΡΗΤΑ":  
    E=A*20/100
```

```
if T=="ΚΑΡΤΑ":  
    E=A*10/100
```

```
if T=="ΔΟΣΕΙΣ":  
    E=0
```

```
print "ποσό της έκπτωσης",E
```

```
print "τελικό ποσό πληρωμής", A-E
```

```
if A-E >200:  
    print "Κερδίσατε Δώρο"
```

ΑΣΚΗΣΗ 2

```
c=0  
S=0.0
```

```
for x in range(30):  
    R=input("Δώσε τιμή ρύπου ")
```

```
    S=S+R
```

```
    if R<=1:  
        print "επίπεδο μόλυνσης : Φυσιολογικό"
```

```
    if R>1 and R<=2:  
        print "επίπεδο μόλυνσης : Οριακό"
```

```
    if R>2:  
        print "επίπεδο μόλυνσης : Επικίνδυνο"
```

```
    if R>3:  
        c=c+1
```

```
print "μέσος όρος των τιμών του ρύπου ",S/30
```

2) Σε ένα διαγωνισμό χορού συμμετέχουν διαγωνιζόμενοι από όλη τη χώρα. Στην πρώτη φάση του διαγωνισμού κάθε διαγωνιζόμενος βαθμολογείται από τρεις (3) κριτές. Ο διαγωνιζόμενος προκρίνεται στην επόμενη φάση, αν ο μέσος όρος των τριών βαθμολογιών του είναι μεγαλύτερος ή ίσος του επτά (7).

Να γράψετε πρόγραμμα σε γλώσσα προγραμματισμού Python, το οποίο:

Δ1. Να διαβάζει το επώνυμο κάθε διαγωνιζομένου και τις βαθμολογίες που έλαβε από τους τρεις κριτές. Η επανάληψη συνεχίζεται έως ότου δοθεί για επώνυμο διαγωνιζομένου η τιμή «ΤΕΛΟΣ».

Δ2. Να υπολογίζει το μέσο όρο των βαθμολογιών κάθε διαγωνιζομένου. Εάν ο διαγωνιζόμενος περνάει στην επόμενη φάση, να εμφανίζει το επώνυμο και το μέσο όρο του.

Δ3. Να υπολογίζει και να εμφανίζει το επώνυμο του διαγωνιζομένου με το μεγαλύτερο μέσο όρο. Να θεωρήσετε ότι αυτός ο διαγωνιζόμενος είναι μοναδικός.

Δ4. Να υπολογίζει και να εμφανίζει το πλήθος των διαγωνιζομένων που δεν πέρασαν στην επόμενη φάση.

Υποδείξεις για το Θέμα:

α. Δεν χρειάζεται να γίνει έλεγχος ορθότητας τιμών, ούτε να υπάρχουν κατάλληλα μηνύματα στις εντολές εισόδου και εξόδου.

β. Στο διαγωνισμό συμμετέχει τουλάχιστον ένας διαγωνιζόμενος. Να θεωρήσετε ότι δεν υπάρχουν διαγωνιζόμενοι με το ίδιο επώνυμο.

γ. Να θεωρήσετε ότι η βαθμολογία κάθε κριτή είναι μεγαλύτερη ή ίση του ένα (1).

ΑΣΚΗΣΗ 1

```
N=input("Δώσε αριθμό των μαθητών που θα συμμετάσχουν")
while N>200:
    N=input("Δώσε τιμή έως και 200")

if N>=1 and N<=20:
    K=5*N

if N>=21 and N<=80:
    K=4*N

if N>80:
    K=3*N

if N>160:
    E=K*5/100
    print "ποσό της έκπτωσης",E
    print "το τελικό κόστος για το σχολείο",K-E

else:
    print "ΔΕΝ ΔΙΚΑΙΟΥΣΤΕ ΕΚΠΤΩΣΗ"
```

ΑΣΚΗΣΗ 2

```
max1=0
c=0

ON=raw_input("Δώσε επώνυμο")

while ON!="ΤΕΛΟΣ":

    V1=input("Δώσε βαθμό 1")
    V2=input("Δώσε βαθμό 2")
    V3=input("Δώσε βαθμό 3")

    ON=raw_input("Δώσε επώνυμο")

    MO=(V1+V2+V3)/3.0

    if MO>=7:
        print ON
        print MO
    else:
        c=c+1

    if MO>max1:
        max1=MO
        onoma=ON

print "Ο διαγωνιζόμενος με το μεγαλύτερο μέσο όρο",onoma

print "δεν πέρασαν στην επόμενη φάση",c
```

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΗΜΕΡΗΣΙΩΝ ΚΑΙ ΕΣΠΕΡΙΝΩΝ ΕΠΑΛ 2017

1) Σε ένα κεντρικό ΚΤΕΟ προσέρχονται για τεχνικό έλεγχο τριών τύπων οχήματα: Φορτηγά, Επιβατικά και Μοτοσυκλέτες. Οι τακτικοί πελάτες μπορούν να γίνουν μέλη του ΚΤΕΟ και να έχουν έκπτωση στο κόστος ελέγχου. Το κόστος ελέγχου υπολογίζεται σύμφωνα με τον παρακάτω πίνακα:

Τύπος οχήματος	Μέλη ΚΤΕΟ	Μη Μέλη ΚΤΕΟ
Φορτηγά	70 ευρώ	80 ευρώ
Επιβατικά	40 ευρώ	50 ευρώ
Μοτοσυκλέτες	25 ευρώ	30 ευρώ

Να γράψετε πρόγραμμα σε γλώσσα προγραμματισμού Python , το οποίο, για μία συγκεκριμένη ημέρα :

Γ1. Να διαβάσει τον τύπο του οχήματος για κάθε όχημα που προσέρχεται καθώς και αν ο πελάτης είναι μέλος του ΚΤΕΟ ή όχι. Η διαδικασία αυτή τερματίζεται όταν δοθεί ως τύπος οχήματος η λέξη « TELOS ».

(Για Φορτηγό θα διαβάσει το « F » , για Επιβατικό το « E » και για Μοτοσυκλέτα το « M » . Δεν απαιτείται έλεγχος εγκυρότητας εισαγωγής των δεδομένων .)

Γ2. Να υπολογίζει το πλήθος και τις εισπράξεις του ΚΤΕΟ για κάθε τύπο οχήματος και να τα εμφανίζει με κατάλληλα μηνύματα. Για παράδειγμα:

Φορτηγά 10 750 ευρώ

Επιβατικά 20 900 ευρώ

Μοτοσυκλέτες 10 295 ευρώ

(Οι παραπάνω τιμές , όπως και ο τρόπος εμφάνισης -στοίχισης δίνονται ενδεικτικά .)

Γ3. Να υπολογίζει και να εμφανίζει το πλήθος όλων των οχημάτων καθώς και το συνολικό ποσό εισπραξης του ΚΤΕΟ.

Γ4. Να υπολογίζει και να εμφανίζει με κατάλληλο μήνυμα τον αριθμό των μελών του ΚΤΕΟ και των μη μελών που προσήλθαν για τεχνικό έλεγχο την συγκεκριμένη ημέρα.

2) Σε μια Ολυμπιάδα Πληροφορικής συμμετέχουν πενήντα (50) μαθητές. Κάθε μαθητής που συμμετέχει λαμβάνει μια τελική βαθμολογία από 1 έως και 100 ακέραιες μονάδες.

Να γράψετε πρόγραμμα σε γλώσσα προγραμματισμού Python , το οποίο:

Δ1. Να διαβάζει το ονοματεπώνυμο κάθε μαθητή και τη βαθμολογία που έλαβε. Τα στοιχεία αυτά καταχωρίζονται στις λίστες NAME και VATHMOS αντίστοιχα. Να γίνει έλεγχος ορθότητας ότι δηλαδή η βαθμολογία που καταχωρίζεται είναι από 1 έως και 100.

Δ2. Να υπολογίζει και να εμφανίζει τον μέσο όρο (ΜΟ) της βαθμολογίας όλων των μαθητών.

Δ3. Να εντοπίζει και να εμφανίζει τα ονοματεπώνυμα και τη βαθμολογία όλων των μαθητών των οποίων η τελική βαθμολογία είναι μεγαλύτερη ή ίση από τον μέσο όρο (ΜΟ) .

Δ4. Να εντοπίζει και να εμφανίζει με κατάλληλο μήνυμα την υψηλότερη βαθμολογία και τα ονοματεπώνυμα των μαθητών που έχουν αυτή τη βαθμολογία .

ΑΣΚΗΣΗ 1

```
XF=0
XE=0
XM=0
SF=0
SE=0
SM=0
M1=0
M2=0

T=raw_input("Δώσε τύπο οχήματος")

while T!="TELOS ":

    M=raw_input("είναι μέλος του ΚΤΕΟ ;")

    if T=="F" and M=="Y":
        XF=XF+70
        SF+=1
        M1+=1
    if T=="F" and M=="N":
        XF=XF+80
        SF+=1
        M2+=1

    if T=="E" and M=="Y":
        XE=XE+40
        SE+=1
        M1+=1
    if T=="E" and M=="N":
        XE=XE+50
        SE+=1
        M2+=1

    if T=="M" and M=="Y":
        XM=XM+25
        SM+=1
        M1+=1
    if T=="M" and M=="N":
        XM=XM+30
        SM+=1
        M2+=1

    T=raw_input("Δώσε τύπο οχήματος")

print "Φορτηγά ",SF, XF, "ευρώ"
print "Επιβατικά ",SE, XE, "ευρώ"
print "Μοτοσυκλέτες ",SM, XM, "ευρώ"

print "πλήθος όλων των οχημάτων ", SF+SE+SM
print "συνολικό ποσό είσπραξης ", XF+XE+XM

print "αριθμός μελών ", M1, "αριθμός μη μελών ", M2
```

ΑΣΚΗΣΗ 2

```
NAME=[]
VATHMOS=[]

for x in range(50):
    name=str(raw_input("Δωσε ονομα"))
    vathmos=input("Δωσε βαθμό")
    while vatnos<1 or vathmos>100:
        vathmos=input("Δωσε ΣΩΣΤΟ βαθμό")

    NAME.append(name)
    VATHMOS.append(vathmos)

S=0.0
for x in VATHMOS:
    S=S+x

MO=S/50

print "μέσος όρος",MO

for x in range(50):
    if VATHMOS[x]>=MO:
        print NAME[x], VATHMOS[x]

max1=VATHMOS[0]
for x in range(1,50,1):
    if VATHMOS[x]> max1:
        max1=VATHMOS[x]

for x in range(50):
    if VATHMOS[x]==max1:
        print NAME[x]
```

ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΗΜΕΡΗΣΙΩΝ ΚΑΙ ΕΣΠΕΡΙΝΩΝ ΕΠΑΛ 2017

1) Σε μια εθελοντική δράση δενδροφύτευσης συμμετέχουν διάφορα σχολεία.

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

Γ1. Να διαβάζει, με χρήση μιας δομής επανάληψης, το όνομα του σχολείου και το πλήθος των εθελοντών του. Η διαδικασία αυτή τερματίζεται όταν δοθεί ως όνομα του σχολείου η λέξη «TELOS» (θεωρείστε ότι συμμετέχουν τουλάχιστον 2 σχολεία).

Γ2. Να εμφανίζει το όνομα του σχολείου και το πλήθος των εθελοντών για το σχολείο που έχει τους περισσότερους και για το σχολείο που έχει τους λιγότερους εθελοντές (θεωρείστε ότι ο αριθμός των εθελοντών είναι διαφορετικός και μικρότερος του 100 για κάθε σχολείο).

Γ3. Να υπολογίζει και να εμφανίζει με κατάλληλο μήνυμα το μέσο όρο του αριθμού των εθελοντών όλων των σχολείων.

Γ4. Να υπολογίζει και να εμφανίζει με κατάλληλο μήνυμα το πλήθος των λεωφορείων που θα χρειαστούν για τη μεταφορά των εθελοντών, αν κάθε λεωφορείο διαθέτει 50 θέσεις.

2) Μια εταιρεία κατασκευής υπολογιστών παράγει 20 διαφορετικά μοντέλα υπολογιστών.

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

Δ1. Να διαβάζει για κάθε μοντέλο το όνομά του και το πλήθος των πωλήσεών του κατά το προηγούμενο έτος. Τα στοιχεία αυτά καταχωρίζονται στις λίστες με ονόματα MODELO και POLISEIS αντίστοιχα.

Δ2. Να υπολογίζει και να εμφανίζει το σύνολο των πωλήσεων όλων των μοντέλων της εταιρείας για το προηγούμενο έτος.

Δ3. Να ταξινομεί με χρήση του αλγόριθμου ταξινόμησης της ευθείας ανταλλαγής (φυσσαλίδα-bubble sort) τις δύο λίστες σε φθίνουσα σειρά ως προς το πλήθος των πωλήσεων.

Δ4. Να δέχεται το όνομα ενός μοντέλου από το πληκτρολόγιο, να εντοπίζει τις πωλήσεις του και να εμφανίζει το όνομα και το πλήθος των πωλήσεών του. Στη συνέχεια να εμφανίζει τα ονόματα και τις πωλήσεις όλων των μοντέλων που οι πωλήσεις τους είναι μεγαλύτερες ή ίσες από τις πωλήσεις του παραπάνω μοντέλου. Η εμφάνιση να γίνεται σε αύξουσα σειρά ως προς τις πωλήσεις.

ΑΣΚΗΣΗ 1

```
max1=0
min1=100
SumE=0
sxoleia=0

N= raw_input(" Doste to onoma tou sxoleiou")

while N != "TELOS":
    E=int(input("Dwste aritho ethelontwn"))

    if E > max1:
        max1=E
        MaxName=N

    if E < min1:
        min1=E
        MinName=N

    SumE=SumE + E

    sxoleia= sxoleia + 1

    N = raw_input(" Dwste to onoma tou sxoleiou")

print MaxName, max1, MinName, min1

MO=float(SumE)/sxoleia

print " o mesos oros ethelontwn einai:", MO

bus=SumE / 50
if SumE % 50 !=0 :
    bus+=1

print " ta lewforeia pou tha xreiasoun einai:", bus
```

ΑΣΚΗΣΗ 2

```
MODELO=[]
POLISEIS=[]
for i in range (20):
    M=raw_input("Dwste onoma modelou")
    P=int(input("Dwste upologistes pou poulithikan"))
    MODELO.append (M)
    POLISEIS.append(P)

S=0
for x in POLISEIS:
    S=S+x
print S

for i in range (1, 20, 1):
    for j in range (19, i-1,-1):
        if POLISEIS[j-1]<POLISEIS[j]:
            POLISEIS[j-1],POLISEIS[j]=POLISEIS[j],POLISEIS[j-1]
            MODELO[j-1], MODELO[j]=MODELO[j],MODELO[j-1]

Onoma=raw_input("Dwste onoma modelou")

for x in range (20):
    if MODELO[x]==Onoma:
        POL=POLISEIS[x]
        print MODELO[x],POLISEIS[x]

for x in range (19, -1,-1):
    if POLISEIS[x] >= POL:
        print MODELO[x], POLISEIS[x]
```

ΥΠΟΛΕΙΠΟΜΕΝΕΣ ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΗΜΕΡΗΣΙΩΝ ΚΑΙ ΕΣΠΕΡΙΝΩΝ ΕΠΑΛ 2017

1) Σε μια εξέταση του μαθήματος της Αγγλικής Γλώσσας εξετάζονται πενήντα (50) μαθητές προφορικά και γραπτά. Οι μαθητές βαθμολογούνται από το 0.0 έως και το 20.0 σε κάθε εξέταση (προφορικά, γραπτά).

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

Γ1. Σε δομή επανάληψης να διαβάσει το ονοματεπώνυμο , την προφορική και τη γραπτή βαθμολογία κάθε μαθητή. Δεν απαιτείται έλεγχος ορθότητας εισαγωγής τιμών.

Γ2. Να εμφανίζει τα ονοματεπώνυμα των μαθητών που έχουν άθροισμα προφορικής και γραπτής βαθμολογίας μεγαλύτερο από το 19.5.

Γ3. Να υπολογίζει και να εμφανίζει το πλήθος των μαθητών που η γραπτή βαθμολογία τους είναι μεγαλύτερη από την προφορική τους .

Γ4. Να υπολογίζει και να εμφανίζει τον μέσο όρο της γραπτής βαθμολογίας και τον μέσο όρο της προφορικής βαθμολογίας όλων των μαθητών.

2) Το Υπουργείο Πολιτισμού διατηρεί στατιστικά στοιχεία για το θέατρο της Αρχαίας Επιδαύρου σχετικά με τον τίτλο κάθε παράστασης και το πλήθος των θεατών που την παρακολούθησαν (κάθε παράσταση παρουσιάζεται μόνο μία φορά και υπάρχει τουλάχιστον μία παράσταση) .

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

Δ1. Να διαβάσει τον τίτλο κάθε παράστασης και το πλήθος των θεατών που την παρακολούθησαν. Τα στοιχεία αυτά να καταχωρίζονται στις λίστες με ονόματα PAR και S _ P αντίστοιχα. Να γίνεται έλεγχος ορθότητας για το πλήθος των θεατών που εισάγεται έτσι ώστε να είναι θετικός αριθμός. Η εισαγωγή των στοιχείων θα τερματίζεται όταν δοθεί ως τίτλος παράστασης η λέξη « TELOS » .

Δ2. Να εντοπίζει και να εμφανίζει τον τίτλο της παράστασης με το μέγιστο πλήθος θεατών. Να θεωρήσετε ότι υπάρχει μία μόνο τέτοια παράσταση.

Δ3. Να υπολογίζει και εμφανίζει τον μέσο όρο των θεατών όλων των παραστάσεων .

Δ4. Κάθε παράσταση με πλήθος θεατών μεγαλύτερο ή ίσο από 1000 άτομα επιδοτείται με 10000€, ενώ κάθε παράσταση με πλήθος θεατών μικρότερο των 1000 ατόμων επιδοτείται με 5000€. Να υπολογίσετε και να εμφανίσετε το συνολικό ποσό της επιδότησης που θα διαθέσει το Υπουργείο Πολιτισμού.

ΑΣΚΗΣΗ 1

```
S1=0
S2=0
c=0
for x in range(50):
    name=str(raw_input("Δωσε ονομα"))

    vathmos1=float(input("Δωσε γραπτο βαθμό"))
    while vatnos1<0.0 or vathmos1>20.0:
        vathmos1=float(input("Δωσε ΣΩΣΤΟ βαθμό"))

    vathmos2=float(input("Δωσε προφορικό βαθμό"))
    while vatnos2<0.0 or vathmos2>20.0:
        vathmos2=float(input("Δωσε ΣΩΣΤΟ βαθμό"))

    if vathmos1 + vathmos2 >19.5:
        print name

    if vathmos1 > vathmos2 :
        c=c+1

    S1=S1+vathmos1
    S2=S2+vathmos2

print "πλήθος με γραπτή βαθμολογία μεγαλύτερη από την προφορική", c

print "μέσο όρο γραπτής βαθμολογίας ",S1/20

print "μέσο όρο προφορικής βαθμολογίας ",S2/20
```

ΑΣΚΗΣΗ 2

```
PAR=[]
S_P=[]

T=str(raw_input("Δωσε τίτλο παράστασης"))

while T!="TELOS":
    P=input("Δώσε πλήθος θεατών ")
    while P<=0:
        P=input("Δώσε ΣΩΣΤΟ πλήθος θεατών ")

    PAR.append(T)
    S_P.append(P)

EP=0
S=S_P[0]
max1=S_P[0]

for x in range(1,len(PAR)):
    if S_P[x]>max1:
        max1=S_P[x]
        titlos=PAR[x]

    S=S+S_P[x]

    if S_P[x]>=1000:
        EP=EP+10000
    else:
        EP=EP+5000

print "παράστασης με το μέγιστο πλήθος θεατών",titlos

print "μέσο όρο θεατών ", S/float(len(S_P))

print "συνολικό ποσό επιδότησης ", EP
```

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΗΜΕΡΗΣΙΩΝ – ΕΣΠΕΡΙΝΩΝ ΕΠΑΛ 2018

1) Διαφημιστική εταιρεία επιθυμεί να παραγγείλει κεφαλαία γράμματα του ελληνικού αλφαβήτου για τη δημιουργία επιγραφών. Για το λόγο αυτό κάνει στατιστική εκτίμηση της συχνότητας εμφάνισης των γραμμάτων χρησιμοποιώντας τη λίστα GRAM η οποία περιέχει στη σειρά τα κεφαλαία ελληνικά γράμματα, δηλαδή $GRAM = ['A', 'B', 'Γ', \dots, 'Ψ', 'Ω']$.

Να γράψετε πρόγραμμα σε γλώσσα προγραμματισμού Python, το οποίο:

Γ1. Να διαβάζει από το πληκτρολόγιο διαδοχικά δύο επιγραφές με κεφαλαία ελληνικά γράμματα και να τις συνενώνει στη μεταβλητή `erig`.

Γ2. Να δημιουργεί μια κενή λίστα με όνομα SUMA και στη συνέχεια με μια επαναληπτική διαδικασία να καταχωρίζει σε αυτή 24 στοιχεία με τιμή μηδέν (0). Η λίστα SUMA θα χρησιμοποιηθεί στα επόμενα ερωτήματα, για την αποθήκευση του αριθμού που δείχνει πόσες φορές υπάρχει κάθε γράμμα (συχνότητα εμφάνισης) στη μεταβλητή `erig`. Κάθε θέση της λίστας SUMA αντιστοιχεί, με την ίδια σειρά, σε ένα γράμμα της λίστας GRAM.

Γ3. Να υπολογίζει τη συχνότητα εμφάνισης κάθε γράμματος της λίστας GRAM που περιέχεται στη μεταβλητή `erig` και να ενημερώνει την αντίστοιχη θέση της λίστας SUMA με την τιμή αυτή.

Γ4. α) Να εμφανίζει ποια γράμματα πρέπει να παραγγελθούν και σε ποια ποσότητα.

β) Να υπολογίζει και να εμφανίζει το πλήθος των γραμμάτων που έχουν μηδενικό πλήθος εμφανίσεων και δεν θα παραγγελθούν.

Σημείωση: α) Η λίστα GRAM θα πρέπει να οριστεί στο πρόγραμμα που θα αναπτύξετε.

β) Δεν απαιτούνται έλεγχοι ορθότητας δεδομένων.

2) Δίνεται λίστα A η οποία περιέχει ονόματα πόλεων και τη μέγιστη θερμοκρασία τους σε μία συγκεκριμένη ημέρα . Η λίστα έχει την παρακάτω δομή (ενδεικτικά):

Πάτρα, 25, Λάρισα, 27

Συνεπώς τα στοιχεία που βρίσκονται σε άρτιες θέσεις περιέχουν ονόματα πόλεων και τα στοιχεία που βρίσκονται σε περιττές θέσεις περιέχουν σε ακέραιο αριθμό τη μέγιστη θερμοκρασία της πόλης .

Να αναπτύξετε πρόγραμμα σε γλώσσα προγραμματισμού Python , το οποίο:

Δ1. Να διαβάζει την λίστα A και ανάλογα αν είναι άρτια ή περιττή θέση, να καταχωρεί τα ονόματα των πόλεων σε μία λίστα POL και τις αντίστοιχες θερμοκρασίες τους σε μία λίστα THER.

Δ2. Να υπολογίζει και να εμφανίζει το μέσο όρο των θερμοκρασιών όλων των πόλεων.

Δ3 . Να ταξινομεί τη λίστα THER με χρήση του αλγορίθμου ταξινόμησης της ευθείας ανταλλαγής (φουσαλίδα – bubblesort) σε φθίνουσα σειρά ως προς τις θερμοκρασίες αναδιατάσσοντας συγχρόνως τη λίστα POL έτσι ώστε να διατηρείται η αντιστοίχιση πόλεων - θερμοκρασιών .

Δ4 . Θεωρώντας ότι μπορεί να υπάρχουν περισσότερες από μία πόλεις με την ίδια θερμοκρασία, να εμφανίζει την υψηλότερη θερμοκρασία που έχει καταχωρηθεί και τα ονόματα των πόλεων που έχουν αυτή τη θερμοκρασία.

ΑΣΚΗΣΗ 1

```
GRAM = ['Α','Β','Γ','Δ','Ε','Ζ','Η','Θ','Ι','Κ','Λ','Μ','Ν','Ξ','Ο','Π','Ρ','Σ','Τ','Υ','Φ','Χ','Ψ','Ω']
epig1 = raw_input('Δώσε την 1η επιγραφή:')
epig2 = raw_input('Δώσε την 2η επιγραφή:')
epig = epig1 + epig2

SUMA = []
for i in range(24):
    SUMA = SUMA + [0]

for letter in epig:
    for i in range(24):
        if letter == GRAM[i]:
            SUMA[i] = SUMA[i] + 1
pl = 0
for i in range(24):
    if SUMA[i]>0:
        print GRAM[i],':',SUMA[i]
    else:
        pl = pl + 1
print "Πλήθος γραμμάτων με μηδενικό πλήθος εμφανίσεων:",pl
```

ΑΣΚΗΣΗ 2

```
POL = []
THER = []
count = 0

i = 0
for x in A:
    i = i + 1
    if i % 2 == 0 :
        POL.append(x)
    else:
        THER.append(float(x))

S = 0.0
for x in THER:
    S = S + x
print "ΜΟ:", S / len(THER)

N=len(THER)
for i in range(1,N, 1):
    for j in range(N-1,i-1,-1):
        if THER[j] > THER[j-1]:
            THER[j], THER[j-1] = THER[j-1] , THER[j]
            POL [j], POL [j-1] = POL [j-1] , POL [j]

print "Υψηλότερη θερμοκρασία:", THER[0]

for x in range(N):
    if THER[x] == THER[0]:
        print POL[x]
```

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΗΜΕΡΗΣΙΩΝ – ΕΣΠΕΡΙΝΩΝ ΕΠΑΛ 2019

1) Σε έναν αγώνα ταχύτητας αυτοκινήτων συμμετέχουν στον προκριματικό 35 οδηγοί με τα αυτοκίνητά τους. Στον τελικό αγώνα θα συμμετάσχουν όσοι οδηγοί σημειώσουν επίδοση μικρότερη ή ίση από 180 δευτερόλεπτα που αποτελεί το όριο πρόκρισης. Κάθε οδηγός έχει μέχρι τέσσερις (4) προσπάθειες για να πετύχει το όριο πρόκρισης. Αν πετύχει σε μία προσπάθεια, σταματά και δεν συνεχίζει τις υπόλοιπες προσπάθειες.

Να γράψετε πρόγραμμα σε γλώσσα προγραμματισμού Python, το οποίο:

Γ1. Για κάθε οδηγό να διαβάζει το όνομά του και διαδοχικά τον χρόνο των προσπαθειών του μέχρι να πετύχει την κατάλληλη επίδοση ή να συμπληρωθεί ο αριθμός των προσπαθειών που δικαιούται.

Γ2. Για κάθε οδηγό να εμφανίζει το όνομά του και αν προκρίθηκε τον χρόνο πρόκρισής του, διαφορετικά να εμφανίζει το μήνυμα “ΜΗ ΣΥΜΜΕΤΟΧΗ”.

Γ3. Να υπολογίζει και να εμφανίζει το πλήθος των προκριθέντων οδηγών, καθώς και το μέσο όρο των χρόνων πρόκρισης που πέτυχαν. Υποθέστε ότι υπάρχει τουλάχιστον ένας.

Γ4. Να βρίσκει και να εμφανίζει το όνομα του οδηγού με το μικρότερο χρόνο πρόκρισης και τον χρόνο αυτό (Υποθέστε ότι είναι μοναδικός).

2) Σε μία αποθήκη σταθμού τρένων υπάρχει ένας πεπερασμένος αριθμός από κιβώτια εμπορευμάτων. Τα κιβώτια πρόκειται να φορτωθούν σε άδεια βαγόνια ενός τρένου. Κάθε βαγόνι έχει όριο χωρητικότητας **2000 λίτρα**. Η φόρτωση ακολουθεί την παρακάτω διαδικασία:

Σε κάθε βαγόνι φορτώνονται διαδοχικά κιβώτια με προκαθορισμένη σειρά, μέχρι να συμπληρωθεί το όριο χωρητικότητάς του. Ένα κιβώτιο φορτώνεται στο βαγόνι μόνο εάν ο όγκος του μαζί με τον όγκο των ήδη φορτωμένων κιβωτίων δεν ξεπερνούν το όριο χωρητικότητας του βαγονιού. Διαφορετικά η φόρτωση του βαγονιού ολοκληρώνεται και το κιβώτιο φορτώνεται στο επόμενο βαγόνι. Αν δεν υπάρχει άλλο διαθέσιμο βαγόνι, το κιβώτιο παραμένει στην αποθήκη.

Η παραπάνω διαδικασία επαναλαμβάνεται μέχρι να τελειώσουν τα κιβώτια ή να μην υπάρχει άλλο διαθέσιμο βαγόνι.

Να αναπτύξετε πρόγραμμα σε γλώσσα προγραμματισμού Python, το οποίο:

Δ1. α. Να διαβάζει τον αριθμό των διαθέσιμων βαγονιών του τρένου.

β. Να διαβάζει τον όγκο κάθε κιβωτίου της αποθήκης σε λίτρα και να το εισάγει σε μία λίστα με όνομα QUE, έως ότου εισαχθεί ως όγκος κιβωτίου ο αριθμός μηδέν (0).

Δ2. Να υπολογίζει και να εμφανίζει, με κατάλληλο μήνυμα, για κάθε βαγόνι που χρησιμοποιήθηκε, το πλήθος και τον συνολικό όγκο των κιβωτίων που περιέχει. Κάθε κιβώτιο που φορτώνεται αφαιρείται από τη λίστα

Δ3. Να υπολογίζει τον αριθμό των βαγονιών που χρησιμοποιήθηκαν. Αν φορτώθηκαν όλα τα κιβώτια, να υπολογίζει το συνολικό όγκο τους και να εμφανίζει με κατάλληλο μήνυμα, τον αριθμό των βαγονιών που χρησιμοποιήθηκαν και τον συνολικό όγκο. Διαφορετικά, αν δεν φορτώθηκαν όλα, να υπολογίζει και να εμφανίζει το πλήθος των κιβωτίων που παρέμειναν στη λίστα (αποθήκη) καθώς και τον συνολικό όγκο σε όλα τα βαγόνια που δεν αξιοποιήθηκε.

Θεωρείστε ότι κανένα κιβώτιο δεν έχει όγκο μεγαλύτερο από 2.000 λίτρα

ΑΣΚΗΣΗ 1

```
pl = 0
Sum = 0.0
Min = 181
#Γ1
for i in range(35):
    onoma = raw_input("Δώσε όνομα:")
    epidosi = input("Δώσε επίδοση:")
    pr = 1
    while epidosi > 180 and pr < 4:
        epidosi = input("Δώσε επίδοση:")
        pr = pr + 1

    #Γ2
    if epidosi <= 180:
        print onoma, epidosi
        #Γ3
        pl = pl + 1
        Sum = Sum + epidosi
    else:
        print onoma, "ΜΗ ΣΥΜΜΕΤΟΧΗ"
#Γ4
    if epidosi < Min:
        Min = epidosi
        onomaMin = onoma
#Γ3
print pl, Sum/pl

#Γ4
print onomaMin, Min
```

ΑΣΚΗΣΗ 2

```
QUE = []
#Δ1
N = input("Δώσε αριθμό βαγονιών:")

v = input("Δώσε τον όγκο του κιβωτίου:")
while v != 0:
    QUE.append(v)
    v = input("Δώσε τον όγκο του κιβωτίου:")

#Δ2
Q1 = []
Q2 = []
i = 1
SumV = 0
while i <= N and QUE != []:
    pl = 0
    Sum = 0
    v = QUE[0]
    while Sum + v <= 2000 and QUE != []:
        Sum = Sum + v
        QUE.pop(0)

        pl = pl + 1
        if QUE != []:
            v = QUE[0]
    Q1.append(pl)
    Q2.append(Sum)
    SumV = SumV + Sum
    i = i + 1

for i in range(len(Q1)):
    print "Το βαγόνι",i+1,"είχε",Q1[i],"κιβώτια και συνολικό όγκο",Q2[i]

#Δ3
if QUE == []:
    print "Χρησιμοποιήθηκαν",len(Q1),"βαγόνια με συνολικό όγκο",SumV
else:
    print "Παρέμειναν στην αποθήκη",len(QUE),"κιβώτια"
    print "Ο συνολικός όγκος των βαγονιών που δεν αξιοποιήθηκε είναι",2000*N - SumV
```

ΠΑΝΕΛΛΑΔΙΚΕΣ ΕΞΕΤΑΣΕΙΣ ΗΜΕΡΗΣΙΩΝ – ΕΣΠΕΡΙΝΩΝ ΕΠΑΛ 2020

1) Σε μία παράσταση υπαίθριου δημοτικού θεάτρου το εισιτήριο εισόδου είναι για τους ενήλικες δέκα (10) ευρώ και για τα παιδιά πέντε (5) ευρώ. Η χωρητικότητα του θεάτρου είναι πεντακόσια (500) άτομα.

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

Γ1.

α) Για κάθε άτομο ή παρέα εισερχόμενων θεατών, να εμφανίζει τις διαθέσιμες ελεύθερες θέσεις του θεάτρου και να διαβάζει από το πληκτρολόγιο το πλήθος των ενηλίκων και ακολούθως το πλήθος των παιδιών της παρέας .

β) Να καλεί τη συνάρτηση EISITIRIO () , η οποία υπολογίζει το συνολικό κόστος των εισιτηρίων, και στη συνέχεια το πρόγραμμα να εμφανίζει το κόστος αυτό. Η λειτουργία της συνάρτησης αυτής περιγράφεται στο ερώτημα Γ3.

γ) 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. Θεωρήστε ότι υπάρχει τουλάχιστον ένας θεατής και σε κάθε παρέα υπάρχει τουλάχιστον ένας ενήλικας.

Γ2. Να υπολογίζει και να εμφανίζει:

α) Τα συνολικά έσοδα του θεάτρου.

β) Το ποσοστό των παιδιών στο σύνολο των θεατών

Γ3. 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.

ΑΣΚΗΣΗ 1

```
def Eisitirio(E,P):  
    return E*10+P*5  
  
SP=0  
T=500  
S=0  
  
E=input("Δώσε αριθμό ενηλίκων")  
while E!=-1:  
    P=input("Δώσε αριθμό παιδιών")  
  
    Sp=Sp+P  
    Xr=Eisitirio(E,P)  
  
    print "Διαθέσιμες ελεύθερες θέσεις", T  
    print "Κόστος εισιτηρίων ",Xr  
  
    S=S+Xr  
    T=T-E-P  
  
    E=input("Δώσε αριθμό ενηλίκων ή -1 ")  
  
print "Τα συνολικά έσοδα του θεάτρου", S  
  
print "Το ποσοστό των παιδιών στο σύνολο των θεατών", 100*Sp/(500-T)
```

ΑΣΚΗΣΗ 2

```
ON=[]
SV=[]

for x in range(20):
    on=raw_input("Δώσε όνομα")
    ON.append(on)
    S=0
    for y in range(15):
        a=raw_input("Δώσε απάντηση")
        if a=="ε":
            m=0
        elif a=="LI[y]":
            m=3
        else:
            m=-1
        S=S+m
    SV.append(S)

S=0.0
for x in SV:
    S=S+x

MO=S/20

for x in range(20):
    if SV[x]>=MO:
        print ON[x]

N=20
for i in range(1,N,1):
    for j in range (N-1,i-1,-1):
        if SV[j]>SV[j-1]:
            SV[j],SV[j-1]=SV[j-1],Sv[j]
            ON[j],ON[j-1]=ON[j-1],ON[j]

print ON[0], ON[1], ON[2]
```