

Ασκήσεις εργαστηρίου

Υπενθύμιση θεωρίας

Μεταβλητές κειμένου (strings)

Στην Python εκτός από τις ακέραιες, πραγματικές, μιγαδικές και λογικές μεταβλητές υπάρχουν και μεταβλητές κειμένου (strings). Γράφονται μέσα σε εισαγωγικά " " και μπορούν να οριστούν κάποιες πράξεις με αυτές όπως σύγκριση (==, !=, and). Άρα μπορούν να χρησιμοποιηθούν ως συνθήκη στο if και το while.

Επειδή οι μεταβλητές κειμένου στην Python είναι case sensitive δηλαδή το "Text" δεν ισούται με το "TEXT", μπορούμε να μετατρέπουμε όλες τις μεταβλητές κειμένου ώστε να γράφονται με μικρά γράμματα για να διευκολύνουν τις συγκρίσεις με την εντολή .lower() ως εξής:

Αν a="Hello" τότε το a.lower() ισούται με hello.

Επίσης, μπορούμε να προσδιορίσουμε το αρχικό γράμμα μιας μεταβλητής κειμένου με την εντολή .startswith(). Δηλαδή αν a = "name" τότε η συνθήκη a.startswith("n") είναι αληθής.

Για να εντοπίσουμε αν ένα ή περισσότερα γράμματα βρίσκονται σε μια μεταβλητή κειμένου χρησιμοποιούμε το in π.χ. η συνθήκη "m" in a είναι αληθής.

Για να συνδυάσουμε μαζί δύο μεταβλητές κειμένου σε μια εντολή print τις ενώνουμε με το σύμβολο + (κάτι που δε γίνεται για ακέραιες ή πραγματικές μεταβλητές).

Ασκήσεις εργαστηρίου

Υπενθύμιση θεωρίας

Π.χ. αν `a= "Hello"` και `b="user"`, το `print(a,b)` δίνει `Hello user` ενώ το `print(a+b)` `Hellouser`.

Τέλος, με την εντολή `str()` μπορούμε να μετατρέψουμε οποιαδήποτε εντολή σε κείμενο, ώστε να συνδυάσουμε αν θέλουμε κείμενο και αριθμητικές τιμές:

```
a = 5
```

```
print("Ο αριθμός α ισούται με:" + str(a))
```

Λίστα

Η λίστα είναι μία δομή δεδομένων στη Python που υποστηρίζει περισσότερες από μία τιμές. Μοιάζει σαν έννοια με τον πίνακα της γραμμικής άλγεβρας. Στη γενική περίπτωση μπορεί να είναι πολυδιάστατη αλλά αρχικά θα χρησιμοποιήσουμε λίστες μίας διάστασης.

Η λίστα γράφεται με τετραγωνικές αγκύλες `[]` και οι τιμές διαχωρίζονται με κόμμα: `a=[1,2,3,4,5]`

Η αρίθμηση των στοιχείων της λίστας στην Python θεωρείται ότι αρχίζει με το 0 άρα το πρώτο στοιχείο είναι το `a[0]`, το τρίτο το `a[2]` κτλ.

Όταν θέλουμε να χρησιμοποιήσουμε στη γενική περίπτωση μια λίστα με κάθε στοιχείο της το στοιχείο γράφεται ως `a[i]` όπου `i` ο δείκτης του στοιχείου.

Με τη λίστα δεν μπορούμε να κάνουμε πράξεις με τον ίδιο τρόπο όπως οι απλές μεταβλητές, δηλαδή `a+1`, `a-1`, `a/2`.

Όμως αν εκτελέσουμε την εντολή `print (a)` εμφανίζονται όλα τα στοιχεία της λίστας.

Μία χρήσιμη εντολή για λίστες είναι η `len()`. Γράφοντας `len(a)` υπολογίζεται το πλήθος των στοιχείων της λίστας.

Ασκήσεις εργαστηρίου

Άσκηση 1

Γράψτε έναν κώδικα σε Python ο οποίος να διαβάσει αριθμούς από μια λίστα και να διπλασιάζει την τιμή των μονών ενώ να διαιρεί με 2 την τιμή των ζυγών.

Άσκηση 2

Γράψτε έναν κώδικα σε Python ο οποίος να αποθηκεύει σε λίστα N αριθμούς που του δίνει ο χρήστης και να υπολογίζει το μέσο όρο τους και την τυπική απόκλιση. Στο τέλος θα εμφανίζει μήνυμα εάν ο μέσος όρος είναι θετικός ή αρνητικός αριθμός.

$$(\bar{x}) = \frac{\sum x_i}{n}, (s) = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n - 1}}$$

Άσκηση 3

Γράψτε έναν κώδικα σε Python ο οποίος να υπολογίζει το μέτρο N-διάστατου διανύσματος $x = (x_1, x_2, \dots, x_n)$ που είναι αποθηκευμένο σε λίστα.

$$\|x\|_2 = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$$

Ασκήσεις εργαστηρίου

Άσκηση 4

Γράψτε έναν κώδικα σε Python ο οποίος να υπολογίζει το εσωτερικό γινόμενο 2 διανυσμάτων ίδιας διάστασης N και να ελέγχει αν είναι κάθετα μεταξύ τους, δηλαδή αν έχουν εσωτερικό γινόμενο 0.

$$\vec{a} \cdot \vec{b} = a_1 b_1 + a_2 b_2 + \dots + a_n b_n$$

Άσκηση 5

Γράψτε έναν κώδικα σε Python ο οποίος να διαβάζει N αριθμούς από λίστα, να υπολογίζει και να εμφανίζει τον μικρότερο και τον μεγαλύτερο αριθμό και τη θέση τους στη λίστα.

Άσκηση 6

Γράψτε έναν κώδικα σε Python ο οποίος να διαβάζει αριθμούς από μια λίστα, να εντοπίζει τους αριθμούς που είναι ζυγοί και μεγαλύτεροι του 10 και να εμφανίζει το πλήθος τους.

Ασκήσεις εργαστηρίου

Άσκηση 7

Γράψτε έναν κώδικα σε Python ο οποίος να διαβάζει μία λίστα με μη-αριθμητικές τιμές τις οποίες θα θεωρεί ξεχωριστές λέξεις, να υπολογίζει πόσες λέξεις έχουν 5 γράμματα και άνω και πόσες αρχίζουν από α.

Άσκηση 8

Γράψτε έναν κώδικα σε Python ο οποίος αρχικά να ρωτάει το χρήστη αν θέλει να υπολογίσει το τετράγωνο ("Τετράγωνο" ή "ΤΕΤΡΑΓΩΝΟ") ή την τετραγωνική ρίζα ("Ρίζα" ή "ΡΙΖΑ") του αριθμού και εάν δώσει σωστή απάντηση να προχωρήσει στον υπολογισμό, αλλιώς να εμφανίσει μήνυμα σφάλματος.

Λυμένες ασκήσεις

Άσκηση 1

Αφού η άσκηση ζητάει να διαβαστεί ένα πλήθος αριθμών πρέπει να χρησιμοποιηθεί κάποια δομή επανάληψης. Ενώ δε δίνεται συγκεκριμένος αριθμός επαναλήψεων, πρέπει όμως η λίστα να έχει συγκεκριμένο μέγεθος καθώς θα την ορίσουμε στην αρχή του κώδικα. Οπότε μπορεί να χρησιμοποιηθεί η δομή for. Επειδή επίσης αναφέρεται ένας έλεγχος για ζυγούς και μονούς αριθμούς πρέπει στη δομή επανάληψης να χρησιμοποιηθεί και μια δομή επιλογής (if).

Ο έλεγχος για ζυγούς ή μονούς αριθμούς απαιτεί μόνο δύο επιλογές, δηλαδή if-else. Εάν ο αριθμός είναι ζυγός, τότε το υπόλοιπο της διαίρεσης με το 2 είναι 0, άρα $i\%2==0$. Αλλιώς αν είναι μονός είτε $i\%2!=0$ είτε $i\%2==1$.

Οπότε ο κώδικας γράφεται ως εξής:

`a = [1,2,5,8,10,12,14]` ← Ορίζουμε μια λίστα.

```
for i in range(len(a)):
    ← if (a[i]%2==0):
        ← a[i]=a[i]/2 ...
    ← else:
        ← a[i]=2*a[i]
```

Με χρήση του len γίνονται τόσες επαναλήψεις όσα τα στοιχεία της λίστας

Έλεγχος ζυγών και μονών

Δομή επανάληψης για να διαβαστούν όλα τα στοιχεία της λίστας

Λυμένες ασκήσεις

Άσκηση 2

Στη συγκεκριμένη περίπτωση η λίστα θα δημιουργηθεί με βάση δεδομένα που δίνει ο χρήστης. Επειδή όμως η λίστα χρειάζεται κάποια αρχικοποίηση (αργότερα θα δούμε και εναλλακτικό τρόπο) καλό είναι στην αρχή να οριστεί με μηδενικά στοιχεία τα οποία θα αλλάξουν όταν ο χρήστης δώσει τις N τιμές).

Ο υπολογισμός του μέσου όρου θα γίνει σε δύο βήματα: στην αρχή θα υπολογιστεί το άθροισμα όλων των τιμών και στο τέλος θα διαιρεθεί το άθροισμα με το N. Για τον υπολογισμό του αθροίσματος θα χρειαστεί κάποια βοηθητική μεταβλητή, έστω total.

Ο υπολογισμός της τυπικής απόκλισης θα γίνει σε επόμενο στάδιο γιατί χρειάζεται να έχει υπολογιστεί πρώτα ο μέσος όρος. Το άθροισμα $\sum(x_i - \bar{x})^2$ σε επόμενο βήμα θα διαιρεθεί με N-1 και τελικά θα υπολογιστεί η τετραγωνική ρίζα του αποτελέσματος.

Οπότε ο κώδικας θα γραφεί ως εξής:

```
N=5 ← Ορισμός του N. Εναλλακτικά θα μπορούσε ο χρήστης να δίνει και αυτή την τιμή: N = int(input())
total=0 ← Μηδενισμός του αθροίσματος
a=[0]*N ← Αρχικοποίηση της λίστας με N μηδενικά στοιχεία
for i range(N):
    a[i]=float(input("Δώσε αριθμό:"))
    total=total+a[i]
```

Λυμένες ασκήσεις

Άσκηση 2

average=total/N ← Ο υπολογισμός του μέσου όρου γίνεται έξω από το for
print(average) ← Προαιρετικά μηδενίζουμε την τιμή του μέσου όρου

variance=0 ← Βοηθητική μεταβλητή αθροίσματος για τον
for i in range(N) υπολογισμό της διακύμανσης

variance=variance+(a[i]-average)**2

variance=variance/(N-1)

std_dev=(variance)**0.5 ← Η τυπική απόκλιση είναι η τετραγωνική ρίζα της
print(std_dev) διακύμανσης

Λυμένες ασκήσεις

Άσκηση 3

Ο κώδικας αυτός είναι αρκετά όμοιος με τον προηγούμενο. Αν παρατηρήσουμε τον τύπο του μέτρου, χρειάζεται δύο βήματα για τον υπολογισμό, δηλαδή ένα άθροισμα και μια τετραγωνική ρίζα. Αυτή τη φορά αντί να υπολογιστεί το άθροισμα των τιμών υπολογίζεται το άθροισμα των τετραγώνων τους.

Οπότε ο κώδικας γράφεται ως εξής:

```
x= [1,2,-5,6,1]
norm=0
for i range(len(x)):
    norm=norm+x[i]**2
norm=norm**(1/2)
print(norm)
```

Λυμένες ασκήσεις

Άσκηση 5

Ο υπολογισμός του μέγιστου και ελάχιστου αριθμού γίνεται με κατάλληλους ελέγχους μέσα σε μια δομή επανάληψης στην οποία διαβάζονται ένα-ένα τα στοιχεία της λίστας. Συνήθως για τον υπολογισμό του μέγιστου ορίζουμε αρχικά μια βοηθητική μεταβλητή με αρκετά χαμηλή τιμή ώστε να είναι χαμηλότερη των αναμενόμενων τιμών. Στη συνέχεια συγκρίνουμε κάθε αριθμό με αυτή την τιμή και εάν είναι μεγαλύτερος αποθηκεύεται σε αυτή την μεταβλητή. Το ίδιο κάνουμε και για τον ελάχιστο αριθμό αλλά την αντίστοιχη μεταβλητή την αρχικοποιούμε με κάποια αρκετά μεγάλη τιμή ώστε οι τιμές της λίστας να είναι μικρότερες από αυτή.

Οπότε ο κώδικας γράφεται ως εξής:

```
x = [1,-2,5,6,7,8,10,-5]
```

```
nmax = -100000
```

```
nmin = 100000
```

```
for i in range(len(x)):
```

```
    if (x[i]>nmax):
```

```
        nmax=x[i]
```



Αποθήκευση τιμής μεγίστου

Λυμένες ασκήσεις

Άσκηση 5

`maxr=i` ← Αποθήκευση θέσης μεγίστου στη λίστα
`if (x[i]<nmin):`
`nmin=x[i]` ← Αποθήκευση τιμής ελάχιστου
`minr=i` ← Αποθήκευση θέσης ελαχίστου στη λίστα