Ασκήσεις εργαστηρίου

Υπενθύμιση θεωρίας

Συναρτήσεις

Οι συναρτήσεις είναι ξεχωριστά κομμάτια κώδικα που παράγουν συγκεκριμένα αποτελέσματα. Μπορεί να δέχονται δεδομένα από τον κυρίως κώδικα και να επιστρέφουν αποτελέσματα. Στην Python ορίζονται ως εξής:

```
def addition(a,b):
    c=a+b
    return c
```

Αρχικά για να δηλώσουμε τη συνάρτηση γράφουμε def και μια ονομασία (η οποία ακολουθεί τους κανόνες ονομασίας της γλώσσας) και στη συνέχεια αν θέλουμε να δέχεται κάποιες μεταβλητές από τον κυρίως κώδικα τις ορίζουμε μέσα στην παρένθεση.
Η συνάρτηση δεν πρέπει να ταυτίζεται με τη μαθηματική έννοια. Μέσα στη συνάρτηση μπορεί να γραφεί οποιαδήποτε εντολή π.χ. if, for, while, ακόμα και να οριστεί άλλη συνάρτηση.
Εάν θέλουμε η συνάρτηση να επιστρέφει αποτέλεσμα στον κυρίως κώδικα το γράφουμε μετά το return.
Προσοχή: Εάν δεν οριστούν δεδομένα ως ορίσματα της συνάρτησης και ως αποτελέσματα (με return) η συνάρτηση δεν επικοινωνεί με τον κυρίως κώδικα. Τότε αν την καλέσουμε χωρίς να έχει κανένα print δεν δίνει κανένα αποτέλεσμα.
Τα δεδομένα μπορεί να είναι αριθμητικά και κείμενο σε μορφή απλής μεταβλητής ή ολόκληρης λίστας.

Ασκήσεις εργαστηρίου

Άσκηση 1

Γράψτε έναν κώδικα σε Python ο οποίος να διαβάζει αριθμούς και να ελέγχει εάν είναι μονοί ή ζυγοί και να εμφανίζει το κατάλληλο μήνυμα. Ο έλεγχος να γίνει με κατάλληλη συνάρτηση που θα γράψετε εσείς. Ο κώδικας θα σταματάει να ζητά αριθμούς αν δοθεί η τιμή 1000.

Άσκηση 2

Γράψτε έναν κώδικα σε Python ο οποίος να υπολογίζει το άθροισμα: $\sum_{n=1}^{N} \frac{1}{n^2}$

Ο υπολογισμός να γίνει μέσω κατάλληλης συνάρτησης που θα γράψετε εσεις με παράμετρο τον αριθμό των όρων (N).

Άσκηση 3

Γράψτε έναν κώδικα σε Python ο οποίος να υπολογίζει την ελάχιστη και τη μέγιστη τιμή των στοιχείων μίας λίστας. Ο υπολογισμός να γίνει μέσω κατάλληλης συνάρτησης που θα γράψετε εσείς.

Άσκηση 4

Γράψτε έναν κώδικα σε Python ο οποίος να υπολογίζει το άθροισμα των όρων μιας λίστας και τον μέσο όρο των τιμών των όρων της. Οι δύο υπολογισμοί θα γίνουν με ξεχωριστές συναρτήσεις.

Ασκήσεις εργαστηρίου

Άσκηση 5

Γράψτε έναν κώδικα σε Python ο οποίος να υπολογίζει την τυπική απόκλιση των τιμών των όρων μιας λίστας. Ο υπολογισμός θα γίνει με συνάρτηση ενώ για τον υπολογισμό του μέσου όρου που απαιτείται θα χρησιμοποιήσετε πάλι μια κατάλληλη συνάρτηση.

Άσκηση 6

Γράψτε έναν κώδικα σε Python ο οποίος να υπολογίζει το παραγοντικό n! (π.χ. 3!=3*2*1=6) με συνάρτηση. Στη συνέχεια χρησιμοποιήστε τη συνάρτηση του παραγοντικού για να κατασκευάσετε μία συνάρτηση που θα υπολογίζει το sin(x) μέσω της δυναμοσειράς:

$$sin(x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)!}$$

Άσκηση 7

Γράψτε έναν κώδικα σε Python ο οποίος να υπολογίζει τις συναρτήσεις cos(x) και atan(x) με βάση τις ακόλουθες εκφράσεις:

$$cos(x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n)!} \qquad arctan(x) = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)}$$

Ασκήσεις εργαστηρίου

Άσκηση 8

Γράψτε έναν κώδικα σε Python ο οποίος να ελέγχει αν τρεις πλευρές a,b,c μπορούν να σχηματίσουν τρίγωνο (a+b>c, a+c>b, b+c>a) και στη συνέχεια να ελέγχει αν το τρίγωνο είναι σκαληνό ή όχι (δηλαδή αν οι τρείς πλευρές δεν είναι ίσες μεταξύ τους) και να υπολογίζει το εμβαδό του τριγώνου από τη σχέση:

$$\mathrm{E} = \sqrt{s(s-a)(s-b)(s-c)} \qquad s = \frac{a+b+c}{2}$$

Όλοι οι έλεγχοι και οι υπολογισμοί να γίνουν με συναρτήσεις.

Άσκηση 9

Γράψτε έναν κώδικα σε Python ο οποίος να διαβάζει ένα κείμενο και να υπολογίζει τον αριθμό χαρακτήρων, λέξεων και προτάσεων. Ο κάθε υπολογισμός θα γίνεται με ξεχωριστή συνάρτηση.

Άσκηση 10

Γράψτε έναν κώδικα σε Python ο οποίος με χρήση συναρτήσεων να υπολογίζει τον αριθμό των φωνηέντων και συμφώνων σε ένα κείμενο και να μετατρέπει τα “a” σε “i” , τα “e” σε “o” και τα “b” σε “d”. Όλοι οι υπολογισμοί θα γίνουν με ξεχωριστές συναρτήσεις.

Λυμένες ασκήσεις

Άσκηση 1

Αφού ο κώδικας ζητά να γίνει έλεγχος χρειάζεται να χρησιμοποιηθεί η δομή ελέγχου if. Ο έλεγχος για ζυγούς και μονούς αριθμούς γίνεται μέσω του υπολοίπου της διαίρεσης με το 2. Ο έλεγχος θα γίνει με συνάρτηση η οποία θα καλείται από τον κυρίως κώδικα μέσα σε μια δομή επανάληψης η οποία θα ολοκληρώνεται όταν θα δοθεί η τιμή 1000. Γι' αυτό το λόγο είναι πιο εύκολο να χρησιμοποιήσουμε τη δομή while.

Δεδομένα εισόδου

```
def check(a):
    if a%2==0:
        print('Ζυγός')
    else:
        print('Μονός')

x=0
while (x!=1000):
    x=int(input())
    check(x)
```

Συνάρτηση

Κυρίως κώδικας

Κλήση συνάρτησης

Αλλιώς αν δεν χρησιμοποιούνταν η συνάρτηση, ο κώδικας θα γραφόταν ως εξής:

```
x=0
while (x!=1000):
  x=int(input())
  if x%2==0:
    print('Ζυγός')
  else:
    print('Μονός')
```

Λυμένες ασκήσεις

Άσκηση 2

Ο γνωστός κώδικας για υπολογισμό αθροίσματος πρέπει να γραφεί στη συνάρτηση με μόνο δεδομένο τον αριθμό των όρων που πρέπει να χρησιμοποιηθούν (N). Προσοχή, το n μεταβάλλεται αλλά η συνάρτηση αφορά μόνο το άθροισμα των όρων π.χ. $1/1^2+1/2^2+1/3^2+1/4^2+1/5^2+$ … και όχι τον όρο $1/n^2$ που είναι στο άθροισμα. Η μεταβολή του n θα γίνει με δομή for γιατί είναι γνωστά τα όρια του αθροίσματος. Επειδή το for φτάνει έως το προηγούμενο ακέραιο αριθμό πριν την τελική τιμή στο range, γράφουμε N+1.

```
def athroisma(N):
    total=0
    for n in range(1,N+1):
        total=total+1/n**2
    return total

k=50
a=athroisma(k)
```

Είναι απαραίτητο να χρησιμοποιηθεί το return για να επιστρέψει την τιμή του αθροίσματος η συνάρτηση.

Επειδή η συνάρτηση επιστρέφει τιμή μπορούμε να αναθέσουμε την τιμή αυτή σε μια μεταβλητή.

Λυμένες ασκήσεις

Άσκηση 3

Θα χρησιμοποιηθούν δύο ξεχωριστές συναρτήσεις για τον υπολογισμό της ελάχιστης και της μέγιστης τιμής της λίστας. Η ελάχιστη και η μέγιστη τιμή θα υπολογιστούν με ελέγχους σε κάθε στοιχείο της λίστας, άρα θα χρειαστεί επανάληψη η οποία θα είναι το for γιατί το μέγεθος της λίστας είναι γνωστό, καθώς και if για να συγκρίνεται η τιμή του κάθε στοιχείου με την ελάχιστη ή μέγιστη μέχρι τη συγκεκριμένη στιγμή.

```
def elahisto(a):
    amin=1000000000
    for i in range(len(a)):
        if a[i]<amin:
            amin=a[i]
    return amin
```

Συγκρίνουμε όλα τα στοιχεία με μια πολύ μεγάλη τιμή ώστε να υπολογιστεί η ελάχιστη τιμή. Εναλλακτικά συγκρίνουμε με την τιμή του 1ου στοιχείου.

```
def megisto(a):
    amax=-1000000000
    for i in range(len(a)):
        if a[i]>amax:
            amax=a[i]
    return amax
```

Συγκρίνουμε όλα τα στοιχεία με μια πολύ μικρή τιμή ώστε να υπολογιστεί η μέγιστη τιμή. Εναλλακτικά συγκρίνουμε με την τιμή του 1ου στοιχείου.

```
a=[1,2,3,5,7,-10,6,-100]
b=elahisto(a)
c=megisto(a)
```

Κυρίως κώδικας

Λυμένες ασκήσεις

Άσκηση 4

Θα χρησιμοποιήσουμε το γνωστό τρόπο για υπολογισμό αθροισμάτων στη μία συνάρτηση και στην άλλη θα υπολογίσουμε τον μέσο όρο ως το πηλίκο αυτού του αθροίσματος με τον αριθμό των στοιχείων της λίστας. Ο υπολογισμός του μέσου όρου μπορεί να γίνει με διάφορους εναλλακτικούς τρόπους.

```
def athroisma(a):
    total=0
    for i in range(len(a)):
        total=total+a[i]
    return total

def mesos(asum,N):
    aver=asum/N
    return aver

a=[1,2,3,5,6,-1,-10]
b=athroisma(a)
c=mesos(b, len(a))
```

```
def athroisma(a):
    total=0
    for i in range(len(a)):
        total=total+a[i]
    return total

def mesos(a):
    asum=athroisma(a)
    N=len(a)
    aver=asum/N
    return aver

a=[1,2,3,5,6,-1,-10]
c=mesos(a)
```

Καλούμε συνάρτηση μέσα σε άλλη συνάρτηση

Λυμένες ασκήσεις

Άσκηση 6

Αρχικά θα γράψουμε τη συνάρτηση για το παραγοντικό η οποία μπορεί να γραφεί αλλάζοντας τον κώδικα του αθροίσματος και στη συνέχεια θα γράψουμε τη δυναμοσειρά με βάση τον κώδικα του αθροίσματος.

```
def paragontiko(n):
    total=1
    for i in range(n,1,-1):
        total=total*i
    return total

def sin(x):
    total=0
    for n in range(0,10):
        arithmitis=(-1)**n*x**(2*n+1)
        paronomastis=paragontiko(2*n+1)
        total=total+arithmitis/paronomastis
    return total

a=sin(0.5)
print(a)
```

Το γινόμενο πρέπει να είναι αρχικά 1 για να μην μηδενιστεί

Ακολουθώντας τον ορισμό του παραγοντικού υπολογίζουμε το γινόμενο με αντίθετη σειρά π.χ. 3!=3*2*1=6

Το άπειρο όριο της δυναμοσειράς είναι ένας αριθμός που μπορεί να δώσει την απαιτούμενη ακρίβεια.

Αν θέλουμε, μπορούμε να αναλύσουμε τη δυναμοσειρά σε αριθμητή και παρονομαστή