

Primeiro control de programación en Python

Escribe un programa en Python que lea por teclado unha matriz a coa función `input()` de dimensións 4×4 , calcule a suma dos elementos pares da matriz e conte o número de elementos impares na matriz. O programa ten que visualizar os resultados na pantalla.

Por exemplo, se introducimos a matriz `[[1,2,4,6],[4,3,7,5],[4,8,0,5],[6,4,7,1]]`, o resultado sería a suma 38 e o número de elementos impares 7.

```
#!/usr/bin/python
#-*- coding: utf-8 -*-
a=input("Matriz a introducida como lista: ")
n=4
suma = 0
contador = 0
for i in range(n):
    for j in range(n):
        if a[i][j] % 2:
            contador += 1
        else:
            suma = suma + a[i][j]
print "Suma elementos pares: ", suma
print 'Numero elementos impares: ', contador
```

Primeiro control de programación en Python

Escribe un programa que lea números x_i por teclado ate que sumen máis de 100. O programa ten que construír unha lista \mathbf{v} (de tamaño igual ó número de elementos lidos) onde cada elemento v_i desta lista ven dado pola expresión:

$$v_i = \begin{cases} 1 & x_i > 10 \\ -1 & x_i < -10 \\ 100 & \text{Resto dos casos} \end{cases}$$

O programa debe mostrar por pantalla os números x_i introducidos e a lista \mathbf{v} . Por exemplo, se introducimos a secuencia de números: 4, 12, -23, 34, 6, -4, -12, 45, 67, a lista resultante sería: [100, 1, -1, 1, 100, 100, -1, 1, 1].

```
#!/usr/bin/python
#-*- coding: utf-8 -*-
suma = 0
x = []
while suma <= 100:
    xi = float(raw_input('Introduce numero: '))
    suma = suma + xi
    x.append(xi)
n=len(x)
print 'Numero de elementos: ', n
v = [0]*n
for i in range(n):
    if x[i] > 10:
        v[i] = 1
    elif x[i] < -10:
        v[i] = -1
    else:
        v[i] = 100
print 'Lista numeros lidos: ', x
print 'Lista resultante: ', v
```

Primeiro control de programación en Python

Escribe un programa que lea por teclado unha lista de números \mathbf{x} coa función `input()`. Sexa n a lonxitude de \mathbf{x} , construe unha matriz b de dimensións $n \times n$, onde cada elemento b_{ij} da matriz ven dado pola expresión:

$$b_{ij} = \begin{cases} \sum_{k=0}^{i-1} x_k & i > j \\ \sum_{k=j}^{n-1} x_k & j \geq i \end{cases}$$

Visualiza a matriz na pantalla mostrando unha fila en cada liña. Por exemplo, se introducimos a lista $\mathbf{x}=[1,2,3,4,5,6]$ o programa ten que construír a matriz b seguinte:

```
21 20 18 15 11 6
3 20 18 15 11 6
6 6 18 15 11 6
10 10 10 15 11 6
15 15 15 15 11 6
21 21 21 21 21 6
```

```
#!/usr/bin/python
#-*- coding: utf-8 -*-
v=input('Introduce v como unha lista: ')
n=len(v)
b = []
for i in range(n):
    b.append([0.0]*n)
for i in range(n):
    for j in range(n):
        if i > j:
            b[i][j] = sum(v[:i+1])
        else:
            b[i][j] = sum(v[j:n])
for i in range(n):
    print b[i]
```

Primeiro control de programación en Python

Escribe un programa que lea un número enteiro n por teclado e o descompoña nas súas cifras. Para isto, o programa debe dividir iterativamente o número n entre 10, e os restos destas divisións son as cifras (por orde inversa). En cada división, o cociente pasa a ser o dividendo da seguinte división. O proceso debe continuar ata que o cociente sexa nulo. O programa debe almacenar as cifras nunha lista chamada c , mostrar tódalas cifras, e logo mostrar as cifras pares e as impares en liñas distintas da terminal.

Por exemplo, se $n = 1234$, dividimos 1234 entre 10 e obteño cociente 123 con resto 4 (primeira cifra); o cociente 123 divídese novamente por 10 e obteño cociente 12 con resto 3 (segunda cifra); divido 12 entre 10 e obteño cociente 1 e resto 2 (terceira cifra); divido 1 entre 10, obteño cociente 0 e resto 1 (cuarta cifra), e remato. Polo tanto, a lista de pares debe ser [2,4] e a de impares [1,3].

```
n=input('n? ')
c=[];nc=0
while n>0:
    c.append(n%10);n=n/10;nc=nc+1
    print c[-1]
print 'pares:'
for i in range(nc):
    if 0==c[i]%2:
        print c[i],
print '\nimpares:'
for i in range(nc):
    if 1==c[i]%2:
        print c[i]
```

Primeiro control de programación en Python

Escribe un programa que lea por teclado repetidamente números enteros, almacenando los valores pares e impares en dúas listas, chamadas `par` e `impar` respectivamente. O proceso debe repetirse ata que o número de valores lidos pares e impares sexa igual, momento en que debe mostrar por pantalla as dúas listas `par` e `impar` en liñas distintas.

Por exemplo, se introduces os números 2,4,5,6,8,3,1,7, o programa debe rematar de ler números logo de introducir o 7, e mostrar as listas [2, 4, 6, 8] e [5, 3, 1, 7] como pares e impares respectivamente.

```
par=[];npar=0
impar=[];nimpar=1
while npar!=nimpar:
    x=input('x? ')
    if 0==x%2:
        par.append(x)
    else:
        impar.append(x)
    npar=len(par);nimpar=len(impar)
print 'npar=',npar,'nimpar=',nimpar
print 'par=',par
print 'impar=',impar
```

Primeiro control de programación en Python

Escribe un programa que lea números x_i por teclado ate que sumen máis de 100. O programa ten que construír unha lista v (de tamaño igual ó número de elementos lidos menos un) onde cada elemento desta lista ven dado pola expresión:

$$v_i = \begin{cases} 1 & x_{i+1} - x_i > 0 \\ -1 & x_{i+1} - x_i < 0 \\ 100 & \text{Resto dos casos} \end{cases}$$

Por exemplo, se introducimos a secuencia de números: 4, 14, 14, 14, 3, -15, 23, 34, 34, a lista resultante sería: [1, 100, 100, -1, -1, 1, 1, 100].

```
#!/usr/bin/python
#-*- coding: utf-8 -*-
suma = 0
x = []
while suma <= 100:
    xi = float(raw_input('Introduce numero: '))
    suma = suma + xi
    x.append(xi)
n=len(x)
print 'Numero de elementos: ', n
y = [0]*(n-1)
for i in range(n-1):
    if x[i+1]-x[i] > 0:
        y[i] = 1
    elif x[i+1]-x[i] < 0:
        y[i] = -1
    else:
        y[i] = 100
print 'Lista numeros lidos: ', x
print 'Lista resultante: ', y
```
