

## Primeiro control de programación en Python de 2018

---

Escribe un programa en Python que monitorice as características da voz dunha persoa no tempo. Para simplificar o problema, o programa fará o seguinte:

1. Ler datos reais introducidos polo usuario mentres que o seu valor sexa positivo e almacenalos nun vector  $\mathbf{x}$ . Supón que estes datos representan a intensidade  $x_t$  da voz en distintos instantes  $t = 0, \dots, T - 1$  de tempo.
2. A transformada discreta de Fourier do vector  $\mathbf{x}$  descomponse na parte real (vector  $\mathbf{Re}$ ) e parte imaxinaria (vector  $\mathbf{Im}$ ), con compoñentes  $Re_k$  e  $Im_k$  para as frecuencias  $k = 0, \dots, N - 1$  (sendo  $N = T$  o número de frecuencias), dadas pola expresión:

$$Re_k = \sum_{n=0}^{N-1} x_n \cos\left(-\frac{2\pi kn}{N}\right) \quad k = 0, 1, \dots, N - 1 \quad (1)$$

$$Im_k = -\sum_{n=0}^{N-1} x_n \sin\left(-\frac{2\pi kn}{N}\right) \quad k = 0, 1, \dots, N - 1 \quad (2)$$

A amplitude da transformada de Fourier é un vector  $\mathbf{A}$  de  $N$  elementos con valores  $A_k$  dados por:

$$A_k = \sqrt{Re_k^2 + Im_k^2} \quad k = 0, \dots, N - 1 \quad (3)$$

3. Visualiza na pantalla os vectores  $\mathbf{x}$  e  $\mathbf{A}$ .
4. Calcula o número de elementos de  $\mathbf{A}$  que será necesario sumar para que se acade a metade da suma dos elementos de  $\mathbf{A}$  e visualiza na pantalla este número e os elementos sumados.

```
#!/usr/bin/python
#-*- coding: utf-8 -*-
from numpy import *

x=[]
xi=float(raw_input('Introduzca numero negativo para sair. xi= '))
while xi > 0:
    x.append(xi)
    xi=float(raw_input('Introduzca numero negativo para sair. xi= '))

N=len(x)
xa=array(x, 'float')
Re=zeros(N); n=arange(N); t=-2*pi/N
for k in n:
    Re[k]=dot(xa, cos(t*k*n))
Im=-Re;
A=sqrt(Re*Re+Im*Im)
print 'x= ', x
print 'Re= ', Re
print 'Im= ', Im
print 'A= ', A

sA=sum(A)/2
suma=0; i=0
while suma < sA:
    suma = suma + A[i]
    i=i+1
print 'Numero elementos sumados: ', i
print 'Elementos: ', A[:i]
```

# Primeiro control de programación en Python de 2018

---

Escribe no editor de texto unha secuencia de números reais nunha liña separados por espacios, como por exemplo o seguinte:

2 3 4 -5 7 2 5 9 4 1 8 4 2 6.2 7

Escribe un programa en Python que realice as seguintes operacións:

1. Pida ó usuario o nome do arquivo e lea o seu contido a un vector  $\mathbf{x}$ . Sexa  $n$  o número de elementos do vector.
2. Pida ó usuario un número enteiro  $m$  que sexa impar no intervalo  $(0, n)$ , non deixando avanzar no programa ate que o usuario introduza un número correcto. Sexa  $l$  o cociente da división enteira de  $m$  entre 2.
3. Realice un filtrado sobre o vector  $\mathbf{x}$ , é dicir, calcule un vector  $\mathbf{y}$  de dimensión  $n$  onde cada  $y_i$ , con  $i = 0, \dots, n - 1$ , sexa:

$$y_i = \begin{cases} \frac{1}{m} \sum_{k=-l}^{k=l} x_{i+k} & \text{se } l \leq i \leq n - l - 1 \\ x_i & \text{noutro caso} \end{cases} \quad (4)$$

4. Visualice na pantalla os vectores  $\mathbf{x}$  e  $\mathbf{y}$ .
5. Calcule o número de elementos do vector  $\mathbf{x}$  que son maiores que os seus correspondentes (é dicir, situados na mesma posición) no vector  $\mathbf{y}$ . O programa debe sumar estes elementos e visualizalos na pantalla.

```
#!/usr/bin/python
#-*- coding: utf-8 -*-
from numpy import *
from sys import *
nome=raw_input('Nome do arquivo: ')
try:
    x=loadtxt(nome)
except IOError:
    print 'Erro lendo o arquivo: ', nome
    exit()
n=len(x)
m=int(raw_input('m= '))
while m%2 ==0 or m<0 or m>n:
    m=int(raw_input('m= '))
l=m/2
print 'n= ', n, ' m= ', m, ' l= ', l
xx=array(x, 'float')
y=xx.copy()
for i in range(l, n-l):
    y[i]=sum(x[i-l:i+l])/m
print 'x= ', x
print 'y= ', y
xmy=extract(x > y, x)
print 'x maior que y: ', xmy
print 'no. veces: ', len(xmy), ' suma= ', sum(xmy)
```

# Primeiro control de programación en Python de 2018

---

Escribe no editor de texto unha matriz, como por exemplo a seguinte:

```
2 3 4 -5 7 2 5 9
4 1 8 4 2 6.2 7 1
8 1 5 3 2 2 4 2
10 1 1 1 1 1 1 1
8 7 4 3 2 1 0 -2
1 9 8 9 8 1 0 -1
0 9 8 5 4 2 1 0
2 3 4 1 2 3 4 8
```

Escribe un programa en Python que realice as seguintes operacións:

1. Pida ó usuario o nome do arquivo e almacene o seu contido na matriz **a**. Sexan  $nf$  e  $nc$  o número de filas e columnas respectivamente da matriz **a**. Se  $nf$  ou  $nc$  son menores que 3, remata o programa.
2. Realice un filtrado sobre a matriz **a**, é dicir, calcule a matriz **b**, da mesma dimensión que **a**, onde cada  $b_{ij}$ , con  $i = 0, \dots, nf - 1, j = 0, \dots, nc - 1$  estea dado por:

$$b_{ij} = \begin{cases} a_{ij} & \text{se } i = 0, \text{ ou } j = 0, \text{ ou } i = nf - 1 \text{ ou } j = nc - 1 \\ \frac{1}{9} \sum_{k=-1}^{k=1} \sum_{l=-1}^{l=1} a_{(i+k)(j+l)} & \text{no resto dos casos} \end{cases} \quad (5)$$

3. Visualice na pantalla as matrices **a** e **b**.
4. Pida ó usuario un  $\epsilon$ , (número real positivo) e calcule o número de elementos do valor absoluto da matriz **b-a** que hai que sumar (percorrendo a matriz por filas) para que se supere  $\epsilon$ . Visualiza na pantalla o número de elementos e a súa suma.

Nota: Se probas co arquivo anterior e con  $\epsilon = 10$ , a suma obtida será 11.64 con 13 elementos.

```
#!/usr/bin/python
#-*- coding: utf-8 -*-
from numpy import *
from sys import *
nome = raw_input('Nome do arquivo: ')
try :
    a = loadtxt(nome)
except IOError :
    print 'Erro lendo o arquivo: ', nome
    exit()
if a.ndim != 2:
    print 'a non e unha matriz'
    exit()
[nf, nc] = a.shape
if nf < 4 or nc < 4 :
    print 'Matriz con menos de 4 filas ou columnas'
    exit()
b = a.copy()
for i in range( 1 , nf-1):
    for j in range( 1 , nc-1):
        aux = a[i-1:i+2 , j-1:j+2]
        print aux
        b[i,j] = sum(aux)/9
print 'a= ', a
```

```
print 'b= ' , b
c = abs(a - b)
x = c.flatten()
epsilon = abs(float(raw_input( 'Epsilon= ' )))
suma = 0
n = len(x)
for i in range(n):
    suma = suma + x[i]
    if suma > epsilon:
        break
print 'Suma= ' , suma
print 'No. elementos sumados= ' , i
```

# Primeiro control de programación en Python de 2018

---

Escribe un programa en Python que realice as seguintes operacións:

1. Lea continuamente por teclado números enteiros e os almacene no vector  $\mathbf{x}$ , sen considerar os números negativos e o 0. Deixa de ler cando, na secuencia que se lea, existen dous números consecutivos iguais, non introducindo no vector o elemento repetido. Por exemplo, a seguinte secuencia:

1 3 5 -2 3 4 3 7 4 0 1 8 8

xera o seguinte vector  $\mathbf{x} = \{1, 3, 5, 3, 4, 3, 7, 4, 1, 8\}$ .

2. Lea un número enteiro do teclado  $y$ , e visualice na pantalla se  $y$  está contido en  $\mathbf{x}$  e, en caso afirmativo, o número de veces que aparece.
3. Sexa  $n$  o número de elementos de  $\mathbf{x}$ . Calcula un vector  $\mathbf{z}$ , de dimensión  $n$ , onde cada elemento  $z_i$ , con  $i = 0, \dots, n - 1$ , ven dado pola expresión:

$$z_i = \sum_{j=0}^i \frac{x_j}{x_{n-1-j}} \quad (6)$$

4. Visualice na pantalla os vectores  $\mathbf{x}$  e  $\mathbf{z}$ , e os elementos  $z_i$  que verifican que  $z_i > x_i/2$ .

```
##*-coding:utf-8*-
from numpy import *
#x=[];xi1=0
#while True:
#    xi=input('x(>0)? ')
#    if xi==xi1 and xi>0:
#        break
#    if xi>0:
#        x.append(xi)
#        xi1=xi
#-----
x=[];xi1=-1;xi=0
while xi!=xi1:
    if xi>0:
        xi1=xi
    xi=input('x(>0)? ')
    if xi>0:
        x.append(xi)
del x[-1]
#-----
print 'x=',x
y=input('y? ')
print y,'aparece',x.count(y),'veces'
#-----
x=array(x);n=len(x);z=zeros(n)
for i in range(n):
    s=0
    for j in range(i):
        s=s+float(x[j])/x[n-j-1]
    z[i]=s
print 'z=',z
#-----
print 'z>x/2=',extract(z>x/2,z)
```

# Primeiro control de programación en Python de 2018

---

Escribe un programa en Python que realice as seguintes operacións:

1. Xere un número enteiro aleatorio  $n$  no intervalo  $[3, 15]$  e unha matriz cadrada  $\mathbf{a}$  de orde  $n$ , onde cada elemento  $a_{ij}$  ven dado pola expresión:

$$a_{ij} = \begin{cases} (i+2)je^{-i} & \text{se } j > i \\ \sin(i^2(j+1))\sqrt{(i+1)^3} & \text{se } j < i \\ \log((i+1)^3) & \text{se } j = i \end{cases} \quad (7)$$

2. Lea un número real positivo  $x$  do teclado e calcule o número de elementos da matriz  $\mathbf{a}$  (percorréndoa por columnas) que se necesita sumar para que a suma sexa maior que  $x$ . Se a suma dos elementos da matriz  $\mathbf{a}$  é menor que  $x$ , repetir o proceso as veces que sexa necesario.
3. Visualice na pantalla a matriz  $\mathbf{a}$  e o número de elementos sumados.

```
#!/usr/bin/python
#-*- coding: utf-8 -*-
from numpy import *
from random import *

n=randint(3, 15)
a=zeros([n,n])
for i in range(n):
    for j in range(i):
        a[i,j]=sin(i*i*(j+1))/sqrt((i+1)**3)
    for j in range(i+1, n):
        a[i,j]=float((i+2)*j)/exp(i)
    a[i,i]=log((i+1)**3)
print 'a= ', a
x=abs(float(raw_input('Introduce numero real: ')))
y=a.flatten('F')
m=len(y)
suma = 0
contador = 0
while suma < x:
    for i in range(m):
        suma = suma + y[i]
        contador += 1
    if suma > x:
        break
print 'No. elementos sumados: ', contador
```