

Primeiro control de programación en Python (2017)

Crea co editor un arquivo de texto chamado `datos11.dat` co seguinte contido:

```
1 0 -1
0 -2 1
1 -1 0
2 -1 3
```

Escribe un programa en Python chamado `exame11.py` que:

1. Lea o arquivo `datos1.dat` a unha matriz a .
2. Crear o vector v concatenando as filas 1 e 2 da matriz a , e o vector w concatenando as filas 3 e 4.
3. Calcula o vector x , da mesma dimensión que v e w , onde cada elemento x_i ven dado pola expresión:

$$x_i = \begin{cases} v_i w_i & \text{se } w_i > v_i \\ v_i \cos(10w_i) & \text{resto} \end{cases} \quad (1)$$

4. Calcula repetidamente os valores de $u_{i-1} = (v \cdot w)^i$ con $i = 1 \dots n$ (onde \cdot representa o produto escalar) e sendo n o número de repeticións necesarias para que:

$$s = \sum_{i=0}^{n-1} u_i > 10^6 \quad (2)$$

e almacenaos no vector u .

5. Mostra por pantalla o vector u e a suma s .

NOTA: co arquivo anterior deberías obter:

```
x= [ 5.94,  5.40,  0, 0, 2, 3 ]
u= [6, 36, 216, 1296, 7776, 46656, 279936, 1679616]
s= 2015538
```

```
#!/usr/bin/python
#-*- coding: utf-8 -*-
from numpy import *
a=loadtxt('datos11.txt', 'int')
v=concatenate((a[0],a[1]))
w=concatenate((a[2],a[3]))
n=len(v)
x=zeros(n)
for i in range(n):
    if w[i] > v[i]:
        x[i]=v[i]*w[i]
    else:
        x[i]=(v[i]+10)*cos(w[i])
print 'x=', x
# alternativa
# where(w > v, v*w, (v+10)*cos(w))
u=[]
i=1
while sum(u) < 10000000:
    u.append(dot(v,w)**i)
    i=i+1
print 'u= ', u
print 's= ', sum(u)
```

Primeiro control de programación en Python (2017)

Escribe un programa en Python chamado `exame12.py` que:

1. Pida o usuario o nome dun arquivo de texto que conteña unha matriz de números e almacénaos na matriz a .
2. Calcula o número de filas n e columnas m da matriz a . Se a matriz a non é cadrada, converter en cadrada da seguinte maneira: se $n > m$, repetir as $n - m$ primeiras columnas á dereita da matriz (supón que a matriz ten $n - m$ columnas), e, en caso contrario, repetir o mesmo proceso coas filas.
3. Percorrer a matriz cadrada resultante por filas e ir sumando os seus elementos. Naqueles elementos onde a suma sexa un número par almacenar o valor da suma nun vector x . Ó mesmo tempo realizar noutra variable p o produto desas sumas.
4. O programa debe mostrar ó usuario a matriz cadrada resultante, o vector x e o valor de p .

NOTA: Se probas cun arquivo de texto coa seguinte matriz:

```
1 2 3
4 5 6
```

o resultado debería ser:

```
b= [[ 1.  2.  3.]
     [ 4.  5.  6.]
     [ 4.  5.  6.]]
x= [6.0, 10.0, 30.0, 36.0]
p= 64800.0
```

```
#!/usr/bin/python
#-*- coding: utf-8 -*-
from numpy import *
nome=raw_input('Nome do arquivo: ')
a=loadtxt(nome)
n, m=a.shape
if n > m:
    b=hstack((a, a[:, m-(n-m):m]))
if m > n:
    b=vstack((a, a[n-(m-n):n, :]))
n=b.shape[0]
p=1
suma =0
x=[]
for i in range(n):
    for j in range(n):
        suma = suma+ b[i, j]
        if suma % 2 == 0:
            x.append(suma)
            p = p * suma
print 'b= ', b
print 'x= ', x
print 'p= ', p
print 'p= ', prod(x)
```

Primeiro control de programación en Python (2017)

Crea un arquivo de texto chamado `matriz1.dat` co seguinte contido:

```
1 2 3 4 5
6 7 8 9 8
7 6 5 4 3
2 3 4 5 1
3 6 5 1 2
```

Escrebe un programa en Python chamado `exame13.py` que:

1. Lea o arquivo `matriz1.dat` e o almacene na variable a .
2. Calcule un vector \mathbf{x} con elemento de a nos que se obtén un resto de 3 cando se dividen por 4.
3. Calcule unha matriz b , das mesmas dimensións de a , onde:

$$b_{ij} = \begin{cases} a_{ij}a_{ji} & \text{se } a_{ij} < a_{ji} \\ \frac{a_{ij}}{a_{ji}} & \text{se } a_{ij} > a_{ji} \\ a_{ij} & \text{se } a_{ij} = a_{ji} \end{cases} \quad (3)$$

4. Dividir o vector \mathbf{x} (cada elemento de \mathbf{x}) por 2 mentres a suma dos elementos do vector \mathbf{x} sexa maior que 10. Visualiza na pantalla o número de veces n que se dividiu o vector \mathbf{x} .
5. Mostrar ó usuario na pantalla o vector \mathbf{x} final e a matriz b .

```
#!/usr/bin/python
#-*- coding: utf-8 -*-
from numpy import *
a=loadtxt('matriz1.dat')
x=extract(a%4==3, a)
b=a.copy()
nf, nc=a.shape
for i in range(nf):
    for j in range(nc):
        if a[i,j] < a[j,i]:
            b[i,j]=a[i,j]*a[j,i]
        elif a[i,j] > a[j,i]:
            b[i,j]=a[i,j]/a[j,i]
n=0
while sum(x) > 10:
    x=x/2
    n= n+1
print 'b= ', b
print 'x= ', x
print 'n= ', n
```

Primeiro control de programación en Python (2017)

Escribe un programa en Python chamado `exame14.py` que:

1. Lea un número enteiro n maior que 10 por teclado (comprobando que o usuario introduce o número correcto).
2. Xere unha matriz a de orde n con números enteiros no intervalo $[0,10]$.
3. Calcule unha matriz b de números reais, da mesma orde que a , onde cada elemento de b ven dado pola expresión:

$$b_{ij} = \begin{cases} \frac{1}{9} \sum_{k=-1}^{k=1} \sum_{l=-1}^{l=1} a(i+k, j+l) & i = 1, \dots, n-2 \quad j = 1, \dots, n-2 \\ a_{ij} & \text{resto dos casos} \end{cases} \quad (4)$$

4. Pide ó usuario un nome dun arquivo, e garda nel as matrices a e b (cada fila nunha liña e a matriz b a continuación da matriz a).

```
#!/usr/bin/python
#-*- coding: utf-8 -*-
from numpy import *
from numpy.random import *
n=0
while n <= 10:
    n=int(raw_input('n= '))
a=round_(10*rand(n,n))
b=array(a, float)
for i in range(1, n-1):
    for j in range(1, n-1):
        suma = 0
        for k in range(-1, 2):
            for l in range(-1, 2):
                suma = suma + a[i+k, j+l]
        b[i, j]=suma/9
# outra alternativa
b2=array(a, float)
for i in range(1, n-1):
    for j in range(1, n-1):
        b2[i, j]=sum(a[i-1:i+2, j-1:j+2])/9
print b2
nome=raw_input('Nome arquivo: ')
savetxt(nome, vstack((a, b)))
#savetxt(nome, vstack((a, b)), '%5.2f') # saída con 2 decimais
```

Primeiro control de programación en Python (2017)

Escribe un programa en Python chamado `exame15.py` que:

1. Lea polo teclado un vector de números enteiros onde os números poden estar repetidos (supón que o usuario introduce correctamente o vector). Almacena o vector lido na variable `x`.
2. Calcula un vector `y`, onde cada elemento y_i sexa o número de veces que o valor i está contido no vector `x`. Para que todos os elementos de `x` se conten algunha vez, a dimensión de `y` será igual ó máximo de `x` máis 1. Mostrar por pantalla o vector `y`.
3. Mostra por pantalla o valor de `x` que se repite máis veces e o número de veces que se repite.
4. Montra en pantalla o número de `x` que se repite máis veces en posicións contiguas.

Por exemplo, se probas co vector `x=2,3,2,2,3,3,4,4,4,4,2,2,8,1,1` deberías obter:

```
y= [ 0.  2.  6.  3.  4.  0.  0.  0.  1.]
0 numero 2 repitese 6.0 veces
0 numero 4 repitese 4 veces contiguas
```

```
#!/usr/bin/python
#-*- coding: utf-8 -*-
from numpy import *
x=array(input('x= '))
n=max(x)+1
y=zeros(n)
for i in range(n):
    y[i]=sum(x == i)
print 'y= ', y
print '0 numero ', argmax(y), ' repitese ', max(y), ' veces'
nrm=0 # numero de repeticions maximas
i=0; nx=len(x)
while i < nx-1:
    nr = 1
    i=i+1
    while i<nx and x[i] == x[i-1]:
        i= i+1
        nr = nr+1
    if nr > nrm:
        nrm= nr
        ni=x[i-1]
print '0 numero ', ni, ' repitese ', nrm, ' veces contiguas'
```

Primeiro control de programación en Python (2017)

Escribe un programa en Python chamado `exame16.py` que lle pida ó usuario a introducción de dous vectores \mathbf{x} e \mathbf{y} coa función `input()`, sendo \mathbf{y} de menor tamaño que \mathbf{x} . O programa debe comprobar que o tamaño de \mathbf{y} sexa menor que o de \mathbf{x} , mostrando unha mensaxe se non se cumpre. Cando a dimensión de \mathbf{x} e maior que a de \mathbf{y} , o programa debe calcular o seguinte:

1. Mostrar unha mensaxe na pantalla informando se o vector \mathbf{y} está contido en \mathbf{x} .
2. Sea n a dimensión de \mathbf{x} , calcula a matriz a de orde n definida ó multiplicar o vector columna \mathbf{x} polo vector fila \mathbf{x} e dado pola expresión:

$$a_{ij} = \sum_{i=0}^{i=n-1} \sum_{j=0}^{j=n-1} x_i x_j \quad (5)$$

3. Guarda no arquivo `saida16.txt` a matriz a .

NOTA: se utilizas o vector $\mathbf{x} = [2,3,4,2,4,3,1]$ e o vector $\mathbf{y} = [2,4,3]$ diría que \mathbf{y} está contido en \mathbf{x} , e se usas $\mathbf{y} = [1,2,3]$ diría que non está contido.

```
#!/usr/bin/python
#-*- coding: utf-8 -*-
from numpy import *
x=array(input('x= '))
y=array(input('y= '))
nx=len(x); ny=len(y)
if ny > nx:
    print 'Dimension de y maior que de x'
else:
    contido=False
    for i in range(nx-ny):
        if all(y == x[i:i+ny]):
            contido=True
            break
    if contido:
        print 'Vector y contido en vector x'
    else:
        print 'Vector y non cotido en vector x'
# matriz a
a=zeros([nx,nx])
for i in range(nx):
    for j in range(nx):
        a[i,j]=x[i]*x[j]
savetxt('saida16.txt', a)
#savetxt('saida16.txt', a, '%5.1f')
```
