

Segundo control de programación en Python de 2021

Crea un archivo de texto con números enteros, onde as liñas poden ter un número distinto de elementos, como por exemplo:

```
1 3 6 5
9 2
4 7 3 12
5
2 8
9 4 13
```

Escribe un programa chamado `exame1.py` que realice as seguintes operacións:

1. Pida por teclado o nome do arquivo de texto, e lea elementos de dito arquivo ata que a suma dos elementos múltiplos de 3 sexa maior que 20 ou se acade a final do arquivo. Almacena os elementos lidos múltiplos de 3 no vector **x** e visualiza **x** na pantalla. Sexa n o número de elementos do vector **x**.
2. Constrúe un vector **y** de dimensión n con números aleatorios enteros no intervalo $[0, M]$, onde M é o valor máximo do vector **x**.
3. Define unha función co nome `analizaVector(...)`, tes que decidir os argumentos, que calcule a media dos elementos de un vector, os elementos menores que a media e as súas posicións no vector.
4. Desde o programa principal, aplica a función `analizaVector()` aos vectores **x** e **y**. Sexan μ_x e μ_y os valores medios de **x** e **y** respectivamente, **mx** e **my** os vectores cos elementos menores que a media para ambos vectores, e **px** e **py** as posicións dos elementos de **mx** e **my** nos vectores **x** e **y**. Representa gráficamente os valores dos vectores **x** e **y** con asteriscos azuis e verdes respectivamente, unha liña vermella co valor de μ_x e os valores de **mx** con círculos vermellos. Pon títulos, enreixado e lendas ó gráfico.

```
from numpy import *
from matplotlib.pyplot import *
from numpy.random import *
from sys import *
nome=input('Nome do arquivo: ')
try:
    suma=0
    f=open(nome, 'r')
    rematar=False
    x=[]
    for linha in f:
        aux=linha.rsplit()
        for i in aux:
            y=int(i)
            if y%3 == 0:
                suma += y
                x.append(y)
            if suma > 20:
                rematar=True
                break
        if rematar:
            break
    print('Suma = ', suma)
    f.close()
except IOError:
    print('Erro abrindo arquivo %s\n' % nome)
```

```
    exit()
print('x= ', x)
n=len(x); mx=max(x);
y=randint(0, mx+1, n)
print('y= ', y)

def analizaVector(v):
    mv=mean(v)
    pv=where(v < mv)[0]
    z=v[pv]
    return [mv, z, pv]

[mediax, mx, px] = analizaVector(array(x))
[mediay, my, py] = analizaVector(y)
figure(1); clf()
t=range(n)
plot(t, x, 'b*', label='Puntos x')
plot(t, y, 'g*', label='Puntos y')
plot([0,n-1],[mediax, mediax], 'r-', label='Media x')
plot(px, mx, 'ro', label='Menores x')
legend()
xlabel('posicion'); ylabel('valor'); grid(True); show()
```

Segundo control de programación en Python de 2021

Escribe un programa en Python3 chamado `exame2.py` que realice o seguinte:

1. Pida polo teclado dous números enteiros n e m no intervalo $[2, 10]$ comprobando que os números son correctos (por exemplo $n=3$ e $m=5$).
2. Calcula unha matriz \mathbf{a} , onde cada elemento a_{ij} , con $i = 0, \dots, n - 1$, $j = 0, \dots, m - 1$, defínese como:

$$a_{ij} = \begin{cases} \exp\left(\frac{2\pi j}{100}\right) & i = 0 \\ \sin\left(\frac{3.5\pi(i+j)}{n}\right) & \text{se } i \leq j \\ a_{(i-1)j} & \text{resto dos casos} \end{cases}$$

3. Calcula unha matriz \mathbf{b} onde cada fila b_i , con $i = 0, \dots, n - 1$, sexa igual á fila $n - 1 - i$ da matriz \mathbf{a} . Garda no arquivo `saida2.txt` a matriz \mathbf{b} (cunha fila en cada liña, un ancho de campo de 10 e dúas cifras decimais) e a traza da matriz \mathbf{b} ó final do arquivo.
4. Define unha función `procesaMatriz(...)`, cos argumentos axeitados, que almacene nun vector os elementos que hai que sumar (percorrendo a matriz por filas) para que a suma supere o valor s , sendo s a suma dos elementos da matriz dividido por 10.
5. Desde o programa principal, aplica a función `procesaMatriz()` a matriz \mathbf{a} para calcular o vector \mathbf{xa} , e a matriz \mathbf{b} para calcular \mathbf{xb} . Concatena os vectores \mathbf{xa} e \mathbf{xb} , calcula o valor medio μ do vector concatenado, e representa gráficamente este vector con puntos azuis, marcando con puntos vermellos os elementos do vector maiores que μ . Pon título e lenda ó gráfico.

```
from numpy import *
from sys import *
from matplotlib.pyplot import *
n=0
while n <= 2 or n >= 10:
    n=abs(int(input('n= ')))
m=0
while m <= 2 or m >= 10:
    m=abs(int(input('m= ')))
a=zeros([n,m])
for i in range(n):
    for j in range(m):
        if i==0:
            a[i,j] = exp(2*pi*j/100)
        elif i <= j:
            a[i,j] = sin(pi*3.5*(i+j)/n)
        else:
            a[i,j] = a[i-1, j]
print('a= ', a)
b=zeros([n,m])
for i in range(n):
    b[i]=a[n-1-i].copy()
try:
    savetxt('saida2.txt', b, '%10.2f')
    f=open('saida2.txt', 'a')
    f.write('Traza matriz: %f\n' % trace(b))
    f.close()
except IOError:
```

```

    print('Erro escribindo en saida2.dat')
    exit()
def procesaMatriz(d):
    x=d.flatten()
    s=sum(x)/10
    suma = 0
    nx=len(x); y=[]
    for i in range(nx):
        suma = suma + x[i]
        if suma > s:
            break
        y.append(x[i])
    return y
xa=procesaMatriz(a)
xb=procesaMatriz(b)
print('xa= ', xa)
print('xb= ', xb)
x=concatenate((xa, xb))
t=range(len(x))
mu=mean(x)
p=where(x > mu)[0]
figure(1); clf()
plot(t, x, 'b*', label='x')
plot(p, x[p], 'r*', label= 'maiores media')
title('Grafico de puntos')
legend(); show()

```

Segundo control de programación en Python de 2021

Escribe co editor de texto un arquivo con números (as liñas conteñen o mesmo número de elementos). Por exemplo co seguinte contido:

```
2 6 8 4 15
12 3 6 4 7
3 5 4 6 9
1 2 3 4 5
```

Escribe un programa chamado `exame4.py` que realice as seguintes operacións:

1. Pida o nome do arquivo por teclado e lea o arquivo á matriz **a**, comprobando que non hai erros na lectura. Define unha matriz **b**, das mesmas dimensións que **a**, con números enteiros aleatorios no intervalo $[mi, ma]$, sendo mi e ma os valores mínimo e máximo da matriz **a** respectivamente.
2. Define unha función `calculosMatriz(...)` cos argumentos axeitados, que almacene nos vectores **x** e **y** os elementos da matriz que son múltiplos de 2 e de 3 respectivamente. Calcula o vector **z** de lonxitude n (sendo n o tamaño do vector **y**), onde cada elemento z_i , con $i = 0, \dots, n - 1$, será o número de elementos do vector **x** que hai que sumar para que dita suma acade o valor y_i (se se chega ó final do vector **x**, volver a empezar desde o principio).
3. Aplica a función `calculosMatriz()` á matriz **a** e despois á matriz **b**. Escribe no arquivo `saida4.txt` os vectores **p** para ambas matrices (un vector en cada liña e como números enteiros). Representa nunha figura os valores dos vectores **p** para ambas matrices (os valores da matriz **a** en azul e os valores da matriz **b** en verde).

```
from numpy import *
from matplotlib.pyplot import *
from sys import *
from numpy.random import *
nome=input('Nome do arquivo: ')
try:
    a=loadtxt(nome);
except IOError:
    exit('Erro lendo arquivo %s'%nome)
[nf, nc]=a.shape
mia=a.min(); maa=a.max()
b=randint(mia, maa, [nf, nc])
#*****
def calculosMatriz(a):
    x=extract(a%2==0,a)
    y=extract(a%3==0,a)
    print('x= ', x); print('y= ', y)
    n=len(x); m=len(y)
    p=zeros(m)
    for i in range(m):
        k=0; nc=0; suma=0
        while suma < y[i]:
            suma = suma + x[k]
            k = k+1; nc = nc+1
            if k==n:
                k=0
        p[i] = nc
    return p
#*****
pa = calculosMatriz(a)
```

```
pb = calculosMatriz(b)
try:
    fin=open('saida4.txt', 'w')
    n=len(pa); aux=''
    for i in range(n):
        aux= aux + '%d '%pa[i]
    aux=aux+'\n'
    fin.write(aux)
    n=len(pb); aux=''
    for i in range(n):
        aux= aux + '%d '%pb[i]
    aux=aux+'\n'
    fin.write(aux)
    fin.close()
except IOError:
    print('Erro escribindo en saida4.txt'); exit()
figure(1); clf()
plot(pa, 'b*', label='Vector pa')
plot(pb, 'g*', label='Vector pb')
grid(True); legend(); show()
```

Segundo control de programación en Python de 2021

Escribe co editor de texto o arquivo `datos5.txt` con números enteiros. Por exemplo co seguinte contido:

```
1 3 -6 5
9 2
4 7 3 12
5
2 -8
9 4 13
```

Escribe un programa chamado `exame5.py` que realice as seguintes operacións:

1. Lea o arquivo `datos5.txt` e almacena no vector \mathbf{x} os elementos positivos. Visualiza na pantalla o vector \mathbf{x} , a súa mediana e desviación típica.
2. Sexa n a dimensión do vector \mathbf{x} , calcula un vector \mathbf{y} definido por:

$$y_i = \frac{x_i + x_{n-i-1}}{2}, \quad i = 0, \dots, n-1 \quad (1)$$

3. Calcula a ecuación da recta $y = mx + b$ que minimiza o erro cuadrático medio utilizando o conxunto de puntos $\{(x_i, y_i), i = 0, \dots, n-1\}$. Representa gráficamente o conxunto de puntos en azul e a ecuación da recta como unha liña vermella. Pon enreixado, títulos nos eixos e lendas ó gráfico. Garda o gráfico co nome `figura5.png`.
4. Define unha función `calculos(...)`, debes decidir os argumentos axeitados, que calcule os vectores \mathbf{z} e \mathbf{iz} . Define o vector \mathbf{d} , de lonxitude n , con elementos $d_i = |x_i - y_i|$, con $i = 0, \dots, n-1$. Crea outro vector \mathbf{z} que conteña os elementos d_i , tal que $d_i \in [\min(d), \text{mean}(d)]$, é dicir, os elementos de \mathbf{d} que están entre o valor mínimo de \mathbf{d} e o valor medio de \mathbf{d} . Crea tamén o vector \mathbf{iz} cos índices (posicións) dos elementos do vector \mathbf{z} no vector \mathbf{d} .
5. Desde o programa principal, chama a función `calculos()` e visualiza na pantalla os vectores \mathbf{z} e \mathbf{iz} .

```
from numpy import *
from matplotlib.pyplot import *
from sys import *
try:
    f=open('datos5.txt', 'r')
    x=[]
    for linha in f:
        aux=linha.rsplit()
        for i in aux:
            y=int(i)
            if y > 0:
                x.append(y)
    f.close()
except IOError:
    exit('Erro lendo arquivo datos5.txt')
print('x= ', x);
print('Mediana=%f, Desviacion tipica= %f\n'%(median(x), std(x)))
n=len(x); y=zeros(n)
for i in range(n):
    y[i]=(x[i]+x[n-i-1])/2
p=polyfit(x,y, 1)
print('y= ', y)
print('y=%.2f x + %.2f' % (p[0], p[1]))
```

```
yr=polyval(p, x)
figure(1); clf()
plot(x, y, 'b*', label='Puntos')
plot(x, yr, 'r-', label='Recta')
grid(True); xlabel('x'); ylabel('y')
legend(); savefig('figure5.png')
#####
def calculos(x,y):
    n=len(x)
    d=abs(array(x)-y)
    print('d= ', d)
    mind=d.min(); meand=mean(d)
    iz=where((d >= mind) & (d<=meand))[0]
    z=d[iz]
    return [z, iz]
#####
[z, iz]=calculos(x,y)
print('z= ', z)
print('iz= ', iz)
```


Segundo control de programación en Python de 2021

Escribe co editor de texto o arquivo `datos6.txt` con números enteiros. Por exemplo co seguinte contido:

```
2 6 8 4
12 3 6 4
9 5 4 6
7 2 3 4
```

Escribe un programa chamado `exame6.py` que realice as seguintes operacións:

1. Lea o arquivo `datos6.txt` á matriz `a`, rematando a execución se ésta non é cadrada. Visualiza na pantalla a matriz `a`, o seu rango e o seu determinante.
2. Define a función `xeraVectores(...)`, cos argumentos axeitados, que a partir dunha matriz calcule dous vectores `x` e `y`. O vector `x` componse dos elementos do triángulo inferior sen a diagonal principal e o vector `y` cos elementos do triángulo superior da matriz sen a diagonal principal.
3. Dende o programa principal, chama a función `xeraVectores()` para calcular os vectores `x` e `y` a partir da matriz `a`. Representa gráficamente os valores de `x` en azul e os valores de `y` en verde e en vermello os valores nos que $x_i > y_i$, con $i = 0, \dots, n - 1$, onde n é a dimensión dos vectores `x` e `y`.
4. Determina o número de elementos de `x` que hai que sumar, comezando polo principio de `x`, e o número de elementos de `y` que hai que restar a $sy = \sum_{i=0}^{n-1} y_i$, comezando polo final de `y`, para que a primeira suma supere á segunda. Mostra por pantalla o valor final de ambas sumas e os índices dos elementos de `x` e `y` nos que rematou o proceso.

```
from numpy import *
from matplotlib.pyplot import *
from sys import *
try:
    a=loadtxt('datos6.txt');
    [nf, nc]=a.shape
    if nf != nc:
        exit('Matriz non cadrada')
except IOError:
    exit('Erro lendo arquivo datos6.txt')
#####
def xeraVectores(a):
    n=size(a,1)
    x=[]; y=[]
    for i in range(n):
        for j in range(n):
            if i < j:
                x.append(a[i,j])
            elif i > j:
                y.append(a[i,j])
    return [array(x),array(y)]
#####
[x, y]=xeraVectores(a)
print('x= ', x); print('y= ', y)
n=len(x); t=list(range(n))
figure(1); clf()
plot(t, x, 'b*', label='Vector x')
plot(t, y, 'g*', label='Vector y')
ind=where(x > y)[0]
```

```
plot(ind, x[ind], 'r*', label='x > y')
grid(True); legend(); show()
#####
sx=0;i=0;sy=sum(y);j=n-1
while sx<=sy:
    sx+=x[i];sy-=y[j]
    i+=1;j-=1
print('sx=%g sy=%g i=%i j=%i'%(sx,sy,i,j))
```