

Segundo control de programación en Python de 2020

Crea un archivo de texto con dúas columnas de números, como por exemplo:

```
1 3.5
3 4
2.3 7
2 1
10 6
```

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

1. Pida por teclado o nome do arquivo de texto, lea o arquivo e garde a primeira columna no vector **x** e a segunda columna no vector **y**. Visualiza ambos vectores na pantalla.
2. Define unha función chamada `matriz(...)` que calcule unha matriz **a** a partir de un vector **z** de modo que a_{ij} , con $i, j = 0, \dots, n-1$, sexa o número de elementos de **z** que hai que sumar (comezando no índice $n-1$ ate o índice 0 e volviendo a comezar polo final) para superar $5(i+1)^2(j+2)$.
3. O programa principal ten que calcular as matrices **b** e **d** chamando a función `matriz(...)` cos vectores **x** e **y** respectivamente.
4. Garda no arquivo `saida.txt` as matrices **b** e **d** como enteiros de ancho 5 (cada fila nunha liña).
5. Calcula tres vectores **fila**, **col** e **valor** coa fila, columna e valor dos elementos da matriz **b** que son maiores que os elementos da matriz **d** nas mesmas posicións. Visualiza na pantalla os tres vectores.
6. Finalmente, convirte a matriz **b** no vector **v** concatenando por filas, e a matriz **d** no vector **w** concatenando por columnas. Calcula os coeficientes a e b que mellor axustan a recta $w = a + bv$ e visualiza nun gráfico a recta en vermello e os puntos como asteriscos azuis. Por etiquetas ós eixos, lendas e enreixado.

```
from numpy import *
from matplotlib.pyplot import *
nome=input('Nome do arquivo: ')
try:
    b=loadtxt(nome)
    x=b[:,0]
    y=b[:,1]
except IOError:
    print('Erro abrindo arquivo %s\n' % nome)
    exit()
print('x= ', x)
print('y= ', y)
def matriz(z):
    n=len(z); a=zeros([n,n])
    for i in range(n):
        for j in range(n):
            m=0; t=5*(i+1)**2*(j+2)
            k=n-1; suma=0
            while suma <= t:
                suma = suma + z[k]
                m = m +1; k=k-1
            if k<0:
                k=n-1
            a[i,j]=m
    return a
b=matriz(x)
```

```
d=matriz(y)
c=vstack((b,d))
savetxt('saida1.txt', c, '%5i')
aux=where(b > d)
fila=aux[0]; col=aux[1]; valor=b[fila, col]; n=len(fila)
for i in range(n):
    print('a[%d, %d]=%d\n'%(fila[i],col[i], valor[i]))
v=b.flatten()
w=d.flatten('F')
p=polyfit(v, w, 1)
figure(1); clf()
plot(v, w, 'b*', label='Puntos')
plot(v, polyval(p,v), 'r-', label='Recta')
legend()
xlabel('v'); ylabel('w'); grid(True); show(False)
```

Segundo control de programación en Python de 2020

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

1. Pida polo teclado dous números enteiros n e m (por exemplo $n=2$ e $m=5$) e calcule $p = n + m$ e $q = nm$.
2. Calcula un vector \mathbf{x} de dimensión q , onde cada elemento x_i , con $i = 0, \dots, q - 1$, sea o número de múltiplos de 5 que hai entre i e $q(i + 1)$. Visualiza na pantalla o vector \mathbf{x} .
3. Define unha función `matriz(...)`, cos argumentos axeitados, que calcule unha matriz \mathbf{a} de orde p , onde cada elemento a_{ij} , con $i, j = 0, \dots, p - 1$ calcúlase como:

$$a_{ij} = \sum_{k=0}^j (i + 1)x_p \quad (1)$$

onde p é o resto da división enteira de k entre q .

4. O programa principal debe almacenar no arquivo `saida2.txt` o vector \mathbf{x} nunha única liña, e a matriz \mathbf{a} , cada fila nunha liña do arquivo con formato enteiro de ancho 5.

```
from numpy import *
from sys import *
from matplotlib.pyplot import *
n=abs(int(input('n= ')))
m=abs(int(input('m= ')))
p=n+m; q=n*m
x=zeros(q)
for i in range(q):
    v=arange(i, q*(i+1)+1)
    x[i]=sum(v%5 == 0)
# alternativa sin vectorizar
#for i in range(q):
#    #k=0; t=q*(i+1)
#    for j in range(i, t+1):
#        #if j%5==0:
#            #k=k+1
#    #x[i]=k
print('x= ', x)
def matriz(z, p):
    n=len(z)
    a=zeros([p,p])
    for i in range(p):
        for j in range(p):
            s=0
            for k in range(j+1):
                s= s + (i+1)*z[k%n]
            a[i,j]=s
    return a
a=matriz(x, p)
print('a= '); print(a)
try:
    f=open('saida2.txt', 'w')
    aux=''
    for i in x:
        aux=aux + '%5d' % i
```

```
aux=aux + '\n'
f.write(aux)
for i in range(p):
    aux=''
    for j in range(p):
        aux = aux + '%5d'% a[i,j]
    aux=aux + '\n'
    f.write(aux)
f.close()
except IOError:
    print('Erro escribindo en saida2.dat')
    exit()
```

Segundo control de programación en Python de 2020

Escribe co editor de texto un arquivo chamado `datos3.txt` co seguinte contido:

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

Escribe un programa en Python chamado `exame3.py` que realice o seguinte:

1. Lea o arquivo `datos3.txt` á matriz **a**, comprobando a posibilidade de erros na lectura. Sexan n e m o número de filas e columnas da matriz **a**. Converte a matriz **a** nun vector por filas e suma os elementos a_{ij} pares na variábel k e os múltiplos de 3 na variábel l ata que $kl \geq p^2$, sendo:

$$p = \left(\sum_{i=1}^n \sum_{j=1}^m a_{ij} \right)^2 \quad (2)$$

Cando se acade o final da matriz, o programa ten que continuar polo comezo da mesma. O programa debe mostrar, en cada iteración, os valores a_{ij} , k , l , kl e p , todos na mesma liña. Tamén debe gardar os elementos múltiplos de 3 sumados no vector **v** e os elementos pares no vector **w**.

2. Definir unha función `analizaVector(...)` que calcule a diferenza mínima e máxima en valor absoluto entre dous elementos consecutivos do vector.
3. Desde o programa principal, aplica a función `analizaVector(...)` ós vectores **v** e **w**. Garda no arquivo `saida3.txt` ambos vectores e o resultado da análise.
4. Crea unha gráfica cunha superficie en 3D na que a matriz **a** representa as alturas e os eixos x e y son os índices de acceso ós elementos da matriz. Pista: para evitar erros, intercambia os rangos de x e y na función `meshgrid()`.

```
from numpy import *
from matplotlib.pyplot import *
from mpl_toolkits.mplot3d import Axes3D
from sys import *
try:
    a=loadtxt('datos3.txt')
except IOError:
    print('Erro lendo datos3.txt'); exit()
[n, m]=a.shape
l=0; k=0; v=[]; w=[]
p=sum(a)**2
x=a.flatten(); nx=len(x)
while k*l < p:
    for i in range(nx):
        z=x[i]
        if z%2==0:
            k = k+z; w.append(z)
        if z%3 == 0:
            l= l+z; v.append(z)
        print('a[%d]=%g, k=%g, l=%g, kl=%g, p=%g'%(i,z, k, l, k*l, p))
        if k*l >=p:
            break
#####
def analizaVector(z):
```

```

z=array(z); n=len(z)
ma=-Inf; mi=Inf
for i in range(n-1):
    dif= abs(z[i+1]-z[i])
    if dif > ma:
        ma=dif
    if dif < mi:
        mi=dif
return [ma, mi]
*****
[mav, miv]=analizaVector(v)
[maw, miw]=analizaVector(w)
*****
def escribeArchivo(fin, x, ma, mi):
    n=len(x); aux=''
    for i in range(n):
        aux= aux + '%g %x[i]'
    aux=aux+'\n'
    fin.write(aux)
    fin.write('ma=%g, mi=%g\n'%(ma, mi))

try:
    f=open('saida3.txt', 'w')
    f.write('Informacion de v: \n')
    escribeArchivo(f, v, mav, miv)
    escribeArchivo(f, w, maw, miw)
    f.close()
except IOError:
    print('Erro escribindo en saida3.txt'); exit()
*****
fig=figure(); clf(); ax=Axes3D(fig)
x=arange(n); y=arange(m)
X,Y=meshgrid(y,x); ax.plot_surface(X, Y,a)
title('Superficie en 3D'); show(False)

```

Segundo control de programación en Python de 2020

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

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

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

1. Pida o nome do arquivo por teclado e lea o arquivo, de xeito que almacene no vector **x** os elementos pares e no vector **y** os múltiplos de 3.
2. Define unha función `matriz(...)` cos argumentos axeitados, que calcule unha matriz cadrada **a** de orde m a partir dun vector **z** de dimensión n . A orde da matriz m é o enteiro por exceso de $\frac{1+\sqrt{1+8n}}{2}$. Trátase de meter o vector **z** nos triángulo inferior e superior definidos pola diagonal secundaria de xeito que a matriz sexa simétrica respecto a diagonal secundaria, é dicir, $a_{00} = a_{(m-1)(m-1)} = z_0$, $a_{10} = a_{(m-1)(m-2)} = z_1$, $a_{20} = a_{(m-1)(m-3)} = z_2$, etc.
3. Calcula a matriz **b** aplicando a función `matriz(...)` ó vector **x**, e a matriz **d** utilizando a mesma función co vector **y**. Visualiza ambas matrices na pantalla.
4. Realiza unha figura con dous gráficos: 1) un gráfico de barras co vector **x**, 2) un histograma co vector **y**. En ambos casos pon enreixado e título ó gráfico.

```
from numpy import *
from matplotlib.pyplot import *
from sys import *
nome=input('Nome do arquivo: ')
try:
    a=loadtxt(nome);
except IOError:
    exit('Erro lendo arquivo %s'%nome)
x=extract(a%2==0,a)
y=extract(a%3==0,a)
print('x= ', x); print('y= ', y)
#x=[]; y=[]
#try:
#    #f=open(nome, 'r')
#    #for linha in f:
#        #aux=linha.rsplit()
#        #for i in aux:
#            #l=float(i)
#            #if l%2==0: x.append(l)
#            #if l%3==0: y.append(l)
#    #f.close()
#except IOError:
#    #print('Erro lendo arquivo %s' % nome)
#    #exit()
#*****
def matriz(z):
    n=len(z)
    m=int(ceil((1.0+sqrt(1.0+8*n))/2))
    a=zeros([m,m]); k=0
    for i in range(m-1):
```

```

    for j in range(m-i-1):
        print('m=%d, n=%d, i=%d, j=%d, k=%d'%(m,n,i,j, k))
        a[i,j]=z[k]
        a[m-j-1,m-i-1]=z[k]
        k=k+1
        if k==n: break
    if k==n: break
return a
*****
b=matriz(x); print('b= ', b)
c=matriz(y); print('c= ', c)
figure(1); clf()
subplot(121); bar(arange(len(x)), x); grid(True); title('Vector z')
subplot(122); hist(y); grid(True); title('Vector y')
show(False)

```


Segundo control de programación en Python de 2020

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

1. Pida por teclado un número enteiro n maior que 10, non deixando avanzar o programa ate que o número sexa correcto. Crea un vector \mathbf{z} de dimensión n con números aleatorios no intervalo $[-10,10]$. Crea outro vector \mathbf{y} de dimensión n definido como: $y_0 = 0$ e $y_i = 2z_i - z_{i-1}$, para $i = 1, \dots, n - 1$.
2. Guarda os vectores \mathbf{z} e \mathbf{y} no arquivo `saida6.txt` con un ancho de 6 e dúas cifras decimais (un vector en cada liña).
3. Representa nun gráfico os valores do vector \mathbf{z} con asteriscos azuis, os valores do vector \mathbf{y} como puntos verdes, unha liña horizontal azul co valor medio dos valores de \mathbf{z} , unha liña verde co valor medio dos valores de \mathbf{y} , e representa en vermello os valores de \mathbf{y} que son maiores ós valores de \mathbf{z} na mesma posición. Pon un título e lendas o gráfico.
4. Define unha función `matriz(...)` cos argumentos axeitados, que calcule: unha matriz \mathbf{a} de orde m e un escalar c . A orde m calcúlase como a parte enteira por exceso de $\sqrt{2n}$. Crea o vector \mathbf{x} concatenando os vectores \mathbf{z} e \mathbf{y} . A matriz \mathbf{a} debe conter os elementos do vector \mathbf{x} por columnas, isto é, $a_{00} = x_0$, $a_{10} = x_1$, $a_{20} = x_2$, etc. O escalar c é o número de elementos do vector \mathbf{z} que hai que sumar (se chega ó final do vector, volve a comezar polo principio) para superar o valor sy , onde sy é a suma dos elementos do vector \mathbf{y} .
5. Desde o programa principal chama a función `matriz(...)` e engade ó final do arquivo `saida6.txt` a matriz \mathbf{a} (unha fila en cada liña con ancho de campo 6 e dous decimais) e o valor de c .

```
from numpy import *
from pylab import *
from numpy.random import *
n=0
while n<10:
    n=int(input('n= '))
z=20*rand(n)-10
y=zeros(n)
for i in range(1,n):
    y[i]=2*z[i]-z[i-1]
print('z= ', z); print('y= ', y)
b=vstack((z,y))
savetxt('saida6.txt', b, '%5.2f')
#####
figure(1); clf()
x=arange(n)
mz=mean(z); my=mean(y)
plot(x, z, 'b*', label='Ptos z')
plot([0, n-1], [mz, mz], 'b-', label='Media z')
plot(x, y, 'go', label='Ptos y');
plot([0, n-1], [my, my], 'g-', label='Media y')
ind=where(y > z)[0]
plot(ind, y[ind], 'r*', label='y>z')
legend(); title('Grafico'); show(False)
#####
def matriz(z,y):
    n=len(z); x=hstack((z,y))
    m=int(ceil(sqrt(2*n)))
    a=zeros([m,m]);
    k=0; nx=len(x)
    for j in range(m):
```

```

    for i in range(m):
        a[i,j]=x[k]
        k=k+1
        if k==nx: break
    if k==nx: break
sy=sum(y); suma=0; c=0; k=0
while suma < sy:
    suma = suma + y[k]
    c = c+1; k += 1
    if k==n: k=0
return [a,c]
*****
[a, c]=matriz(z,y)
[nf, nc]=a.shape
try:
    f=open('saida6.txt', 'a')
    for i in range(nf):
        aux=''
        for j in range(nc):
            aux = aux + '%5.2f'%a[i,j]
        aux= aux + '\n'
        f.write(aux)
    f.write('Valor de c= %d\n'% c)
    f.close()
except IOError:
    print('Erro escribindo en saida6.txt')

```