

Primeiro control de programación en Python de 2020

Crea un arquivo de texto co editor no que estea contida unha matriz. Por exemplo:

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

Escribe un programa en Python que faga o seguinte:

1. Ler un nome de arquivo do teclado que contén unha matriz. Ler a matriz do arquivo e almacénaa na variable **a**, comprobando que non houbo erros na lectura. Visualiza na pantalla a matriz **a** e as súas dimensións.
2. Calcula e visualiza a matriz **b**, das mesmas dimensións que **a**, dada pola seguinte expresión:

$$b_{ij} = \begin{cases} \log(\text{abs}(a_{ij}) + 1) & \text{se } a_{ij} < \sum_{k=0}^{i-1} a_{kj} \\ a_{ij} & \text{resto dos casos} \end{cases}$$

3. Selecciona unha posición aleatoria na matriz **a** (por exemplo, xerar un número enteiro aleatorio x entre 1 e o número de elementos de **a** e percorrendo a matriz por filas ate acadar a posición de x). Divide iterativamente todos os elementos da matriz **a** por $\text{abs}(a(x-1)) + 1$ ate que a suma da matriz **a** sexa menor que 1. Visualiza na pantalla o número de veces que se realizou a operación de división.

```
from numpy import *
nome=input('Nome arquivo: ')
try:
    a=loadtxt(nome)
    print('a= ', a)
    nf=a.shape[0]; nc=a.shape[1]
    print('No. filas: ', nf, " No. columnas: ", nc)
    b=zeros([nf, nc])
    for i in range(nf):
        for j in range(nc):
            if a[i,j] > sum(a[0:i, j]):
                b[i,j]=log10(abs(a[i,j])+1)
            else:
                b[i,j]=a[i,j]
    print('b= ', b)
    from random import *
    x=randint(1,a.size)
    print('x= ', x)
    aa=a.flatten()
    print("aa= ", aa)
    contador = 0
    while sum(aa) >= 1:
        aa=aa/(abs(aa[x-1])+1)
        contador = contador + 1
    print('No. veces division: ', contador)
except IOError:
    print('Erro lendo arquivo ', nome)
```

Primeiro control de programación en Python de 2020

Escribe un programa en Python que realice as seguintes operacións:

1. Xera un número enteiro aleatorio n no intervalo $[50, 200]$.
2. Calcula dous vectores \mathbf{x} e \mathbf{y} de dimensión n , onde cada elemento x_i e y_i , $i = 0, \dots, n - 1$ están definidos polas expresións:

$$x_i = 3 \sin\left(\frac{2\pi}{i+1}\right) + 5 \quad (1)$$

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

3. Sexa sx a suma dos elementos do vector \mathbf{x} , calcula o número de elementos do vector \mathbf{y} que hai que sumar para superar a suma sx . Visualiza na pantalla sx e o número de elementos.
4. Calcula un vector \mathbf{z} de dimensión n con $z_0 = 0$ e z_i , $i = 1, \dots, n - 1$ dados pola expresión:

$$z_i = \begin{cases} 5 & \text{se } x_i + x_{i-1} < y_i \\ 7 & \text{resto dos casos} \end{cases}$$

5. Calcula unha matriz \mathbf{a} que conteña o vector \mathbf{x} na primeira fila, o \mathbf{y} na segunda e o \mathbf{z} na terceira. Visualiza a matriz na pantalla.

```
from numpy import *
from random import *
n=randint(50, 200)
i=arange(0,n)
x=3*sin(2*pi/(i+1))+5
print('x= ', x)
y=zeros(n)
for i in range(n):
    y[i]=x[i]+x[n-i-1]
print('y= ', y)
sx=sum(x)
sy = 0; i=0
while sy < sx:
    sy = sy + y[i]
    i = i+1
print('Sx= ', sx)
print('No. elementos sumados: ', i)
z=zeros(n)
for i in range(1, n):
    if x[i]+x[i-1] < y[i]:
        z[i]=5
    else:
        z[i]=7
print('z= ', z)
a=vstack((x, y, z))
print('a= ', a)
```

Primeiro control de programación en Python de 2020

Escribe no editor de texto un vector de números e gárdao no arquivo `datos1.txt`. Por exemplo, co seguinte contido:

```
2 3 4 -5 7 2 5 9
```

Escribe un programa en Python que realice as seguintes operacións:

1. Lea o arquivo `datos1.txt` a un vector \mathbf{x} (non tes que comprobar os erros na lectura do arquivo), e visualízao na pantalla. Sexa n o número de elementos de \mathbf{x} .
2. Xera un vector \mathbf{y} de dimensión n con valores aleatorios no intervalo $[0, 2\mu)$, sendo μ o valor medio do vector \mathbf{x} . Visualiza \mathbf{y} na pantalla.
3. Sexa sx e sy a suma dos elementos do vector \mathbf{x} e \mathbf{y} respectivamente, calcula o valor p dado pola expresión:

$$p = \begin{cases} \sum_{i=0}^{n-1} x_i y_i & \text{se } sx > sy \\ \sum_{i=0}^{n-1} 2x_{n-i-1} y_i & \text{resto dos casos} \end{cases}$$

4. Sexa m a mediana do vector \mathbf{x} , calcula o número r de elementos do vector \mathbf{x} que hai que sumar para superar o valor $10m$ (se chega o final do vector, debe volver a comezar polo principio). Visualiza o número r e mais os números sumados na pantalla.

```
from numpy import *
from numpy.random import *
x=loadtxt('datos1.txt')
print("x= ", x)
n=len(x) ; mu=mean(x)
y=2*mu*rand(n)
print("y= ", y)
sx=sum(x); sy=sum(y)
print('sx= ', sx, " sy= ", sy)
if sx >sy:
    p=dot(x,y)
else:
    p=0
    for i in range(n):
        p = p + 2*x[n-i-1]*y[i]
print("p= ", p)
m=10*median(x); s=0; i=0
z=[]
while s < m:
    s = s + x[i]
    i = (i+1) % n
    z.append(x[i])
print('No. elementos sumados: ', len(z))
print('Elementos: ', z)
```

Primeiro control de programación en Python de 2020

Escribe un programa en Python que realice as seguintes operacións:

1. Ler polo teclado dous números enteiros, n e m , no intervalo $[5, 10]$. O programa non pode avanzar ate que n e m cumpran a condición.
2. Crea unha matriz \mathbf{a} de dimensión $n \times m$ con números aleatorios enteiros no intervalo $[5, 30]$ e visualízaa na pantalla.
3. Sexan sf e sc as suma dos elementos da primeira fila e columna da matriz \mathbf{a} respectivamente. O programa ten que calcular o rango da matriz \mathbf{a} se $sf > sc$ e a suma dos elementos da diagonal principal en caso contrario.
4. Calcula unha matriz \mathbf{b} das mesmas dimensións que \mathbf{a} , onde cada elemento b_{ij} , $i = 0, \dots, n$ e $j = 0, \dots, m - 1$:

$$b_{ij} = \sum_{k=0}^i \sum_{l=j}^{m-1} a_{kl} \quad (3)$$

e visualiza na pantalla a matriz transposta de \mathbf{b} .

```
from numpy import *
from numpy.random import *
from numpy.linalg import *
n=-1
while n<5 or n>10:
    n=int(input('n= '))
m=-1
while m<5 or m>10:
    m=int(input('m= '))
print('n= ', n, " m= ", m)
a=int_(5+25*random((n,m)))
print('a= ', a)
sf=sum(a[0]); sc=sum(a[:,0])
if sf > sc:
    print('Determinante de a: ', matrix_rank(a))
else:
    print('Suma diagonal principal: ', trace(a))
b=zeros([n,m])
for i in range(n):
    for j in range(m):
        b[i,j]=sum(a[0:i+1, j:m])
print('Transposta b: ', b.T)
#b=zeros([n,m]) # version iterativa
#for i in range(n):
#    for j in range(m):
#        #s=0
#        for k in range(i+1):
#            for l in range(j, m):
#                #s = s + a[k,l]
#        #b[i,j]=s
#print('Transposta b: ', b.T)
```

Primeiro control de programación en Python de 2020

Escribe un programa en Python que realice as seguintes operacións:

1. Lea do teclado un número enteiro n que sexa par e maior que 0.
2. Crea un vector \mathbf{x} con n elementos aleatorios enteiros no intervalo $[50, 60)$ e visualízao na pantalla.
3. Crea un vector \mathbf{y} con dimensión 10, onde cada elemento y_i , $i = 0, \dots, 9$ sexa o número de veces que o valor $i + 50$ se repite no vector \mathbf{x} . Visualiza na pantalla o vector \mathbf{y} .
4. Calcula o elemento do vector que se repite mais veces e gárdao na variable k . Visualiza na pantalla o elemento que se repite e o número de veces k que se repite. No caso de que haxa varios elementos que se repiten k veces, visualizaos tamén. Por exemplo: se \mathbf{x} contén os números $[54\ 51\ 56\ 52\ 58\ 57\ 54\ 53\ 52\ 54\ 52\ 51\ 56\ 57\ 56\ 50\ 51\ 53\ 50\ 53]$, terá que visualizar os números $51\ 52\ 53\ 54\ 56$ que se repiten 3 veces.
5. Converte o vector \mathbf{x} nunha matriz \mathbf{a} con dúas filas e as columnas que necesites. Se a suma dos elementos da primeira fila e maior que a suma dos elementos da segunda fila, calcula e visualiza o produto escalar da primeira fila pola segunda fila de \mathbf{a} . En caso contrario, calcula e visualiza a suma dos elementos da matriz \mathbf{a} .

```
from numpy import *
from numpy.random import *

n=-1
while n<0 or n%2==1:
    n=int(input('n= '))
x=randint(50,60, n)
# tamen x=int_(50 + 9*random(n))
print('x= ', x)
m=10; y=zeros(m)
for i in range(m):
    y[i]=sum(i+50 == x)
    # tamen con y[i]=len(where(i+10 == x)[0])
print("y= ", y)
z=[] # elementos mais repetidos
ind=argmax(y)
k=y[argmax(y)]
z.append(ind+50)
i=ind +1
while i < m:
    if y[i]==k:
        z.append(i+50)
    i= i+1
print('Elemento mais repetido: ', z)
print('No repeticions: ', k)
a=x.reshape([2, int(n/2)])
if sum(a[0]) > sum(a[1]):
    print('Producto escalar: ', dot(a[0],a[1]))
else:
    print('Suma matriz: ', sum(a))
```

Primeiro control de programación en Python de 2020

Escribe no editor de texto dúas liñas de números con igual número de elementos e gárdao no arquivo `datos2.txt`. Por exemplo, co seguinte contido:

```
2 7 3 4 5 9 12.3
4 2 6 7 4 2 1
```

Escribe un programa en Python que realice as seguintes operacións:

1. Lea o contido do arquivo `datos2.txt` na matriz **a** sen comprobar se hai erro na lectura. Calcula o valor máximo e mínimo de **a** e visualizaos na pantalla.
2. Crea dous vectores **x** e **y** coa primeira e segunda fila de **a** respectivamente. Sexa n a lonxitude de **x** e **y**, calcula o produto escalar p de **x** e **y** e unha matriz cadrada **b** de orde n , onde p e cada elemento b_{ij} , con $i, j = 0, \dots, n - 1$ veñen dados polas expresións:

$$p = \sum_{i=0}^{n-1} x_i y_i \quad b_{ij} = x_i y_j \quad (4)$$

3. Visualiza na pantalla p e **b**. Percorre a matriz **b** por filas sumando os seus elementos ata que a suma sexa superior a p . Visualiza na pantalla a suma e a fila e columna da matriz na que se cumpre a condición.
4. Calcula o determinante da matriz **b** e visualízao na pantalla.

```
from numpy import *
from numpy.linalg import *
a=loadtxt('datos2.txt')
print('Maximo: ', a.max(), " Minimo: ", a.min())
x=a[0]; y=a[1]
p=dot(x,y)
n=len(x); b=zeros([n,n])
for i in range(n):
    for j in range(n):
        b[i,j]=x[i]*y[j]
print('Producto escalar: ', p)
print('b= ', b)
s=0; fila=0; columna=0; c=0
for i in range(n):
    for j in range(n):
        c = c + 1
        s=s+b[i,j]
        if s > p:
            fila = i
            columna = j
            break
    if s > p:
        break
print('No. elementos: ', c)
print('Fila= ', fila, ' columna= ', columna)
print('Determinante: ', det(b))
```