

Primeiro control de programación en Python de 2021

Crea un arquivo de texto co editor con números reais de xeito que todas as liñas teñen o mesmo número de elementos. Por exemplo:

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

Escrebe un programa en Python que faga o seguinte:

1. Ler un nome de arquivo do teclado. Se o número de filas é par, ler o arquivo e almacena na variable **a** a primeira metade das liñas e na variable **b** a segunda metade das liñas, comprobando que non houbo erros na lectura. Visualiza na pantalla as matrices **a** e **b**, así como as súas dimensións.
2. Calcula o produto da matriz **a** pola transposta da matriz **b** e visualiza na pantalla o determinante do resultado.
3. Sexan sa e sb a suma dos elementos da matriz **a** e **b** respectivamente. Se $sa \geq sb$, calcula unha matriz **d**, das mesmas dimensións de **a** e **b**, onde cada elemento d_{ij} ven dado pola expresión;

$$d_{ij} = \frac{a_{ij}b_{(nf-i-1)(nc-j-1)}}{1 + a_{ij}^2} \quad i = 0, \dots, nf - 1 \quad j = 0, \dots, nc - 1 \quad (1)$$

e visualiza na pantalla o resultado.

4. Se $sa < sb$, introduce nun vector **x** os elementos da matriz **a** (percorrida por filas) que son superiores os seus correspondentes na matriz **b** mentres que a suma dos elementos de **x** sexa inferior a $sa/4$ ou se acade o final da matriz. Visualiza o vector **x**, o número de elementos sumados e o valor da súa suma.

```
from numpy import *
try:
    nf=3
    while nf%2 != 0:
        nome=input('Nome arquivo: ')
        aux=loadtxt(nome)
        nf=aux.shape[0]
    n=int(nf/2)
    a=aux[:n].copy()
    b=aux[n:].copy()
    print('No. filas: ', n, " No. columnas: ", a.shape[1])
    c=dot(a, b.T)
    print("Multiplicación= ", c)
    d=zeros(a.shape)
    nf=d.shape[0]; nc=d.shape[1]
    sa=sum(a); sb=sum(b)
    if sa >= sb:
        for i in range(nf):
            for j in range(nc):
                d[i,j]=(a[i,j]-b[nf-i-1, nc-j-1])/(1+a[i,j]*a[i,j])
        print('d= ', d)
    else:
        x=[]
        ax=a.flatten()
        bx=b.flatten()
```

```
n=len(ax)
sa4=sa/4
for i in range(n):
    if ax[i] > bx[i]:
        x.append(ax[i])
    if sum(x) > sa4:
        break
    print('x= ', x, ' n= ', len(x), ' suma= ', sum(x))
except IOError:
    print('Erro lendo arquivo ', nome)
```

Primeiro control de programación en Python de 2021

Escribe un programa en Python que realice as seguintes operacións:

1. Ler do teclado un vector \mathbf{x} , e sexa n a súa dimensión. Calcula un vector \mathbf{y} de dimensión n , onde $y_0 = 0$ e y_i , con $i = 1, \dots, n - 1$, está definido por:

$$y_i = \frac{x_i + y_{i-1}}{x_{n-i-1}^2} \quad (2)$$

2. Crea un vector \mathbf{z} con números aleatorios no intervalo $[1, 20]$ e o vector \mathbf{v} como $v_i = x_i + z_i$, con $i = 0, \dots, n$. Apila verticalmente todos os vectores $(\mathbf{x}, \mathbf{y}, \mathbf{z}$ e $\mathbf{v})$ para formar a matriz \mathbf{a} e garda a matriz no arquivo `datosaida.txt`. Calcula o rango da matriz \mathbf{a} e visualízao na pantalla.
3. Calcula o vector \mathbf{w} , de lonxitude n , onde w_i , con $i = 0, \dots, n - 1$, calcúlase, para i par, como o número p de elementos que hai que sumar para que se cumpra a condición:

$$\sum_{j=0}^p (x_j + z_j) > \sum_{k=0}^i v_k \quad (3)$$

Se i é impar, w_i será o produto escalar dos vectores \mathbf{x} e \mathbf{z} ate a posición i , definido por:

$$w_i = \sum_{j=0}^i x_j z_j \quad (4)$$

Visualiza na pantalla o vector \mathbf{w} .

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

x=float_(input('x= ').rsplit())
n=len(x)
y=zeros(n)
for i in range(1,n):
    y[i]=(x[i]+y[i-1])/(x[n-i-1]**2)
z=19*random(n)+1
v=x+z
a=vstack((x,y, z, v))
savetxt('datosaida.txt', a)
print('Rango de a: ', matrix_rank(a))
w=zeros(n)
for i in range(n):
    umbral=sum(v[:i+1])
    suma=0; cont=0
    if i%2 == 0:
        for j in range(n):
            suma=suma+x[j]+z[j]
            cont = cont+1
            if suma > umbral:
                break
        w[i]=cont
    else:
        w[i]=dot(x[:i+1], z[:i+1])
print('w= ', w)
```

Primeiro control de programación en Python de 2021

Escribe no editor dúas liñas con números (as dúas liñas teñen igual número de elementos) e gárdao no arquivo `datos3.txt`. Por exemplo, co seguinte contido:

```
2 3 4 -5 7 2 5 9
2 4 1 1 10 6 4 -5
```

Escribe un programa en Python que realice as seguintes operacións:

1. Lea o arquivo `datos3.txt` comprobando os erros na lectura do arquivo e garda no vector \mathbf{x} a primeira liña e no vector \mathbf{y} a segunda liña. Visualiza na pantalla os dous vectores. Sexa n o número de elementos de \mathbf{x} e \mathbf{y} .
2. Calcula un vector \mathbf{z} cos elementos comúns de \mathbf{x} e \mathbf{y} , non incluíndo elementos repetidos. Visualiza na pantalla o vector \mathbf{z} , o número de elementos repetidos, o elemento máximo de \mathbf{z} e o seu índice.
3. Sexa \mathbf{v} o vector formado apilando horizontalmente os vectores \mathbf{x} , \mathbf{y} e \mathbf{z} . Calcula o número de elementos que hai que sumar para que esta suma sexa maior que a metade da suma de todos os elementos de \mathbf{v} , é dicir, calcula p , tal que:

$$\sum_{i=0}^p v_i > \frac{1}{2} \sum_j v_j \quad (5)$$

```
from numpy import *
try:
    a=loadtxt('datos3.txt')
    x=a[0]; y=a[1]
    n=len(x)
    print('x= ', x)
    print('y= ', y)
    z=[]
    for i in range(n):
        comun=False
        for j in range(n):
            if x[i] == y[j]:
                comun=True
                break
        if comun and x[i] not in z:
            z.append(x[i])
    print('z= ', z)
    print('No. elementos: ', len(z))
    z2=array(z)
    print('Maximo: ', max(z), ' posicion: ', argmax(z2))
    v=concatenate((x,y, z2))
    print('v= ', v)
    umbral=sum(v)/2
    suma=0; i=0
    while suma < umbral:
        suma = suma + v[i]
        i=i+1
    print('Elementos sumados: ', i)
except IOError:
    print('Erro lendo datos3.txt')
```

Primeiro control de programación en Python de 2021

Escribe no editor unha matriz de números enteiros e gárdao no arquivo `datos4.txt`. Por exemplo, co seguinte contido:

```
2 3 4 -5 7 2 5 9
2 4 1 1 10 6 4 -5
8 6 5 3 9 24 5 14
```

Escribe un programa en Python que realice as seguintes operacións:

1. Ler o arquivo `datos4.txt`, sen comprobar os erros na lectura, e almacena o seu contido na matriz **a**. Visualiza a matriz e as súas dimensións.
2. Almacena no vector **x** os elementos de **a** que son primos (non debes almacenar números repetidos). Visualiza **x** na pantalla.
3. Calcula unha matriz **b**, das mesmas dimensións que **a**, onde cada elemento b_{ij} está definido por:

$$b_{ij} = \begin{cases} \sum_{l=i-1}^i \sum_{k=j-1}^j a_{lk} & \text{se } i, j \geq 1 \\ a_{ij} & \text{resto dos casos} \end{cases}$$

e visualiza na pantalla a matriz **b**.

4. Sexa **d** a matriz transposta de **b**. Elimina da matriz **d** a primeira e a última fila e calcula: 1) a súa traza (suma dos elementos da diagonal principal); e 2) calcula os elementos maiores que 10. Visualiza os resultados na pantalla.

```
from numpy import *
a=loadtxt('datos4.txt', int)
(nf, nc)=a.shape
print('a= ', a)
print('Filas: ', nf, ' Columnas: ', nc)
x=[]
for i in range(nf):
    for j in range(nc):
        n=a[i,j]
        primo = True
        for k in range(2, n):
            if n%k == 0:
                primo=False
                break
        if primo and n not in x:
            x.append(n)
print('x= ', x)
b=a.copy()
for i in range(1,nf):
    for j in range(1, nc):
        b[i,j]=sum(a[i-1:i+1, j-1:j+1])
print('b= ', b)
d=b.T
nfd=d.shape[0]
if nfd > 2:
    c=d[1:nfd-1].copy()
```

```
print('c= ', c)
print('Traza: ', trace(c))
print('Elementos > 10: ', extract(c > 10, c))
else:
    print('Matriz con menos de duas filas')
```

Primeiro control de programación en Python de 2021

Escribe un programa en Python que realice as seguintes operacións:

1. Lea do teclado un conxunto de números enteiros e os almacene no vector **x**. Sexa n a súa dimensión (comproba que $n \geq 4$ e, se non, volve a ler os números). Introduce os elementos de **x** nunha matriz **a** de dimensión $m \times m$, onde $m = \lfloor \sqrt{n} \rfloor$ (os últimos elementos de **x** deben quedar fóra da matriz **a**). Visualiza **x** e **a** na pantalla.
2. Se a matriz **a** ten determinante non nulo, calcula a súa inversa e visualiza na pantalla o determinante e a matriz inversa.
3. Calcula o vector **y** de dimensión n , onde y_0 é a mediana do vector **x** e y_i , con $i = 1, \dots, n - 1$, está definido por $y_i = x_i - x_{i-1}$.
4. Visualiza na pantalla os elementos de **x** que están repetidos e o número de veces que se repite cada un deles. Por exemplo se **x** contén os elementos 2 5 7 9 2 10 7 2 3 1, a resposta sería que 2 está repetido 3 veces e 7 dúas veces.

```
from numpy import *
from numpy.linalg import *
n=0
while n <=4:
    x=int_(input('x= ').rsplit())
    n=len(x)
m=int(floor(sqrt(n)))
a=x[:m*m].reshape([m,m])
print('x= ', x)
print('a= ', a)
da=det(a)
print('Determinante de a: ', da)
if da != 0:
    print('Inversa de a: ', inv(a))
y=zeros(n)
y[0]=median(x)
for i in range(1,n):
    y[i]=x[i]-x[i-1]
print('y= ', y)
r=[]
for i in range(n):
    l=x[i]
    if l not in r:
        repetido=0
        for j in range(i+1,n):
            if l == x[j]:
                repetido += 1
        if repetido > 0:
            print('Numero: ', l, ' repetido ', repetido+1, ' veces')
            r.append(l)
```

Primeiro control de programación en Python de 2021

Escribe un programa en Python que realice as seguintes operacións:

1. Xere un número enteiro aleatorio n no intervalo $[5,15]$. Crea unha matriz \mathbf{a} de dimensións $n \times n$ definida por:

$$a_{ij} = \begin{cases} e^{-j} \sin\left(\frac{\pi i}{10}\right) & \text{se } i > j \\ (i+1)(j+1)\sqrt{i+j} & \text{se } i < j \\ 10ij & \text{se } i = j \end{cases}$$

2. Pide por teclado un nome de arquivo e garda a matriz \mathbf{a} nese arquivo.
3. Suma os elementos da diagonal secundaria da matriz \mathbf{a} dados pola expresión:

$$s = \sum_{i=n-1}^0 a_{i(n-1-i)} \quad (6)$$

e calcula o determinante de \mathbf{a} . Visualiza na pantalla ambos datos.

4. Suma os elementos da matriz \mathbf{a} , percorrida por columnas ate que a súa suma sexa superior a s . Visualiza na pantalla o número de elementos sumados e o índice de fila e columna na que se parou de sumar.

```
from numpy import *
from random import *
from sys import exit
from numpy.linalg import det
n=randint(5,15)
a=zeros([n,n])
for i in range(n):
    for j in range(n):
        if i>j:
            a[i,j]=sin(pi*i/10)*exp(-j)
        elif i<j:
            a[i,j]=sqrt(i+j)*(i+1)*(j+1)
        else:
            a[i,j]=10*i*j
nome=input('Nome arquivo: ')
try:
    savetxt(nome, a, '%10.2f')
except IOError:
    print('Erro escribindo en ', nome)
    exit()
s=0
for i in range(n):
    s = s + a[i, n-i-1]
print('Suma= ', s, ' Det= ', det(a))
suma= 0; f=0; c=0; nelem=0
while suma < s:
    suma = suma + a[f, c]
    nelem = nelem + 1
    f +=1
    if f == n:
        f=0
```



```
        c +=1
    if c==n:
        break
print('No. elementos: ', nelem, ' Fila: ', f, ' Col: ', c)
```