

Primeiro control de programación en Python de 2022

Escribe un programa en Python que haga o seguinte:

1. Definir unha variable n co valor 3. Ler por teclado números enteros positivos e introducilos por columnas nunha matriz \mathbf{a} de orde n (o programa debe introducir en \mathbf{a} os números positivos que vai lendo, e descartar os números negativos). Visualizar a matriz \mathbf{a} na pantalla.
2. Calcular e visualizar na pantalla o rango da matriz \mathbf{a} , o seu determinante e os seus autovalores.
3. A partir de a_{ij} , con $0 \leq i, j < n$, calcular o vector \mathbf{v} , de dimensión n^2 , onde cada elemento v_{in+j} está dado pola expresión:

$$v_{in+j} = \begin{cases} \sum_{k=0}^{n-1} a_{ik} & \text{se } i \geq j \\ \sum_{k=0}^{n-1} a_{kk} & \text{se } i = j \\ \sum_{k=0}^{n-1} a_{kj} & \text{resto dos casos} \end{cases}$$

4. Visualiza na pantalla o vector \mathbf{v} e o número de elementos pares e impares en \mathbf{v} .

```
from numpy import *
from numpy.linalg import *
n=3; n2=n*n
a=[]; i=0
while i<n2:
    x=float(input('Introduce número= '))
    if x > 0:
        a.append(x)
        i=i+1
a=array(a)
a=a.reshape([n,n]).T
print('a= ', a)
print('Rango= ', matrix_rank(a))
print('Determinante= ', det(a))
print('Autovalores= ', eig(a))
v=zeros(n2)
for i in range(n):
    for j in range(n):
        t=i*n+j
        if i > j:
            v[t]=sum(a[i])
        elif i == j:
            v[t]=trace(a)
        else:
            v[t]=sum(a[:,j])
print('v= ', v)
print('No. de pares: ', sum(v%2 == 0))
print('No. de impares: ', sum(v%2 == 1))
```

Primeiro control de programación en Python de 2022

Escribe un programa en Python que realice as seguintes operacións:

1. Lea por teclado un número enteiro, n , no intervalo $[5,15]$ que sexa impar (o programa non pode avanzar ate que n sexa correcto).
2. Crea unha matriz **a** cadrada de orde n con números aleatorios enteiros no intervalo $[0,100]$, e calcula un vector **v** de dimensión $m = \lceil \frac{n}{2} \rceil$, onde cada elemento v_k , con $0 \leq k < m$, ven dado pola expresión:

$$v_k = \sum_{p=m-k}^{m+k} \sum_{q=m-k}^{m+k} a_{pq} \quad (1)$$

3. Calcula a suma s_s dos elementos do triángulo superior de **a**, e a suma dos elementos do triángulo inferior, s_i . Se $s_s > s_i$, transforma a matriz **a** nunha matriz **b** das mesmas dimensións que **a**, onde cada elemento b_{ij} , con $0 \leq i, j < n$, está dado por $b_{ij} = a_{(n-i-1)j} a_{i(n-j-1)}$, e garda as dúas matrices **a** e **b** no arquivo `datos2.txt`. En caso contrario, comproba se a matriz **a** é simétrica e visualiza o resultado na pantalla.

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

n=0
while n<5 or n>15 or n%2==0:
    n=int(input('n= '))
a=int_(100*rand(n,n))
m=int(ceil(n/2))
v=zeros(m)
for k in range(m):
    v[k]=sum(a[m-k:m+k+1, m-k:m+k+1])
print('v= ', v)
sa=sum(triu(a)); si=sum(tril(a))
print('sa= ', sa, ' si= ', si)
if sa > si:
    b=zeros([n,n])
    for i in range(n):
        for j in range(n):
            b[i,j]=a[n-i-1, j]*a[i, n-j-1]
    ab=vstack((a, b))
    savetxt('datos2.txt', ab, '%i')
else:
    if all(a == a.T):
        print('Matriz simétrica')
    else:
        print('Matriz non simétrica')
```

Primeiro control de programación en Python de 2022

Escribe un programa en Python que realice as seguintes operacións:

1. Lea un vector \mathbf{x} por teclado con números enteros de dimensión maior que 3 (comprobando que a dimensión sexa correcta). Sexa n a lonxitude de \mathbf{x} .
2. Calcula $n_3 = \sum_{i=0}^{n-1} n_i$, onde n_i é o número de veces que o elemento x_i é divisible por 3, e garda no vector \mathbf{v} , de lonxitude n , os restos da división. Visualiza na pantalla n_3 e o vector \mathbf{v} . Se probas co vector $\mathbf{x}=[9, 16, 54, 15]$ terías que obter $n_3 = 6$ e $\mathbf{v}=[1, 16, 2, 5]$.
3. Constrúe unha matriz \mathbf{a} co vector \mathbf{x} na diagonal principal, que sexa simétrica ($a_{ij} = a_{ji}, \forall j > i$) e co vector \mathbf{v} no triángulo superior da matriz \mathbf{a} .

```
from numpy import *
from numpy.linalg import *
n=0
while n<3:
    x=int_(input('x= ').rsplit())
    n=len(x)
n3=0; v=[]
for m in x:
    while m%3 == 0:
        n3= n3 +1
        m=m/3
    v.append(m)
print('n3= ', n3)
print('v= ', v)
a=diag(x); k=0
for i in range(n):
    for j in range(i+1, n):
        a[i,j]=v[k]
        a[j,i]=v[k]
        k = k+1
        if k == n:
            break
    if k==n:
        break
print('a= ', a)
```

Primeiro control de programación en Python de 2022

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

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

Escrebe un programa en Python que realice as seguintes operacións:

1. Lea por teclado o nome do arquivo e almacene o seu contido na matriz **a**, comprobando que non haia erros na lectura. Visualiza a matriz e as súas dimensións: **nf** e **nc** para o número de filas e columnas respectivamente.
2. Se hai algún número negativo na matriz **a**, conta o número de elementos que son divisibles por 3, e visualiza o resultado na pantalla. En caso contrario, conta o número de elementos da matriz **a** que é necesario sumar (percorrendo a matriz por columnas) para acadar a metade da suma de tódolos elementos da matriz.
3. Calcula un vector **v** de dimensión *m*, sendo *m* o valor mínimo entre o número de filas e columnas da matriz **a**, onde cada elemento v_i ven dado pola expresión:

$$v_i = a_{ii} \left(\sum_{k=0}^{nc-1} a_{ik} \right) \left(\sum_{k=0}^{nf-1} a_{ki} \right) \quad (2)$$

e visualiza o vector **v** na pantalla, xunto coa súa media e desviación típica.

```
from numpy import *
from sys import exit
nome=input('Nome arquivo a ler: ')
try:
    a=loadtxt(nome, int)
    (nf, nc)=a.shape
    print('a= ', a)
    print('Filas: ', nf, ' Columnas: ', nc)
except IOError:
    print('Erro lendo o arquivo:', nome)
    exit()
if any(a < 0):
    n3=sum(a % 3 == 0)
    print('No. elementos multiples de 3: ', n3)
else:
    sa=sum(a)/2
    x=a.flatten('F')
    suma=0; k=0
    while suma<sa:
        suma = suma + x[k]
        k=k+1
    print('Suma %i con %i elementos' % (suma, k))
m=min(nf, nc)
v=zeros(m)
for i in range(m):
    v[i]=a[i,i]*(sum(a[i])+sum(a[:,i]))
print('v= ', v)
print('media = %.2f desviación típica= %.2f' % (mean(v), std(v)))
```

Primeiro control de programación en Python de 2022

Escribe un programa en Python que realice as seguintes operacións:

1. Lea por teclado dous vectores \mathbf{v} e \mathbf{w} con números enteiros. Sexa nv o número de elementos de \mathbf{v} e nw o número de elementos de \mathbf{w} , e sexa n a dimensión do vector máis pequeno. Mostra por pantalla \mathbf{v} e \mathbf{w} , así coma nv e nw .
2. Chamemos \mathbf{x} o vector con menos elementos (entre \mathbf{v} e \mathbf{w}) e \mathbf{y} ó vector con máis elementos. Elimina no vector \mathbf{x} os elementos repetidos e conta o número de veces que os elementos de \mathbf{x} se repiten en \mathbf{y} . Usa a función `unique(x)` para obter os valores de \mathbf{x} non repetidos.
3. Sexa s_x a suma dos elementos do vector \mathbf{x} , calcula o número de elementos do vector \mathbf{y} que hai que sumar para superar o valor s_x ou para que se acade o final do vector \mathbf{y} . Almacena no vector \mathbf{z} os elementos sumados. Se probas cos valores 3 3 4 6 3 4 para o vector \mathbf{v} e 3 -5 6 -7 3 3 6 -5 -7 6 para \mathbf{w} , deberías obter $s_x = 13$, $s_z = 3$ e súmanse todos os elementos de $\mathbf{y}=\mathbf{w}$. Se \mathbf{w} fose 3 5 6 7 3 3 6 6 5 7, entón $s_z = 14$ e considéranse so os tres primeiros valores.

```
from numpy import *

v=int_(input('v= ').rsplit())
w=int_(input('w= ').rsplit())
nv=len(v); nw=len(w)
n=min(nv,nw)
if nv < nw:
    x=v; y=w
else:
    x=w; y=v
cont=0
x=unique(x)
for i in x:
    cont = cont + sum(y == i)
print('x= ', x)
print('y= ', y)
print('No. de repeticións: ',cont)
z=[]; sx=sum(x);
sz=0;k=0; ny=len(y)
while sz < sx:
    sz = sz + y[k]
    z.append(y[k])
    k = k + 1
    if k == ny:
        break
print('sx= ', sx, ' sz= ', sz)
print('Elementos sumados: ', k, ' z= ', z)
```

Primeiro control de programación en Python de 2022

Escribe un programa en Python que realice as seguintes operacións:

1. Xere dous números enteiros aleatorios n_1 e n_2 no intervalo $[10,25]$. Sexa n o mínimo de n_1 e n_2 e m o valor máximo. Constrúe dous vectores \mathbf{x} e \mathbf{y} , de dimensións n e m respectivamente, con valores enteiros aleatorios no intervalo $[0,10]$. Utiliza a función `unique(x)` para que o vector \mathbf{x} non teña elementos repetidos. Visualiza ambos vectores na pantalla.
2. Calcula a suma dos elementos do vector \mathbf{x} que se repiten no vector \mathbf{y} e visualiza esta suma na pantalla. Constrúe un vector \mathbf{z} cos elementos de \mathbf{x} que non se atopan no vector \mathbf{y} , e mostra \mathbf{z} na pantalla.
3. Sexa s_y a suma dos elementos do vector \mathbf{y} . Suma os elementos das posicións pares do vector \mathbf{x} (posicións 0, 2, 4, ...) ate que se supere o valor s_y . Se se chega ó final do vector sen acadar a suma, volve a comezar desde o principio. Visualiza na pantalla o número de elementos sumados e a súa suma.

```
from numpy import *
from numpy.random import *
import random
n1=random.randint(10,25)
n2=random.randint(10,25)
n=min(n1,n2); m=max(n1,n2)
print('n1= ', n1, ' n2= ', n2)
x=int_(rand(n)*10); x=unique(x)
y=int_(rand(m)*10)
print('x= ', x)
print('y= ', y)
suma = 0; z=[]
for i in x:
    if i in y:
        suma = suma + i
    else:
        z.append(i)
print('Suma repetidos: ', suma)
print('Non contidos: ', z)
sy=sum(y); sx=0; k=0;
nx=len(x); ne = 0
while sx < sy:
    sx = sx + x[k]
    k = k+2
    ne = ne+1
    if k >= nx:
        k=0
print('No. elementos: ', ne)
print('Suma: ', sx)
```