

Primeiro control de programación en Python de 2023

Escrebe un programa en Python que faga o seguinte:

1. Ler un número n enteiro por teclado no intervalo $[10,20]$ que sexa par (comproba que n é un valor válido antes de avanzar a execución do programa).
2. Crea un vector $\mathbf{x}$ de dimensión n con valores reais aleatorios no intervalo $[0, 100]$. Crea o vector $\mathbf{v}$ cos elementos $x_i \in \mathbf{x}$ nas posicións $i = 0, \dots, n$ pares e outro vector $\mathbf{w}$ cos elementos $x_i \in \mathbf{x}$ nas posicións $i = 0, \dots, n$ impares. Visualiza ambos vectores na pantalla.
3. Crea unha matriz $\mathbf{a}$ de dimensión $m = n/2$ onde cada elemento a_{ij} , $i, j = 0, \dots, m - 1$ é:

$$a_{ij} = \begin{cases} \frac{v_i w_j}{v_i} & \text{se } i > j \\ \frac{w_j + 1}{j} & \text{se } i = j \\ \sum_{l=0}^j v_l \sum_{k=0}^i w_k & \text{resto dos casos} \end{cases}$$

4. Visualiza na pantalla a matriz $\mathbf{a}$, a media dos seus elementos e o seu determinante.

```
from numpy import *
from numpy.random import *
n=0
while n<10 or n>20 or n%2==1:
    n=int(input('n= '))
x=random(n)*100
v=x[0:n:2]
w=x[1:n:2]
print('v= ', v)
print('w= ', w)
m=int(n/2)
a=zeros([m,m])
for i in range(m):
    for j in range(m):
        if i > j:
            a[i,j]=v[i]*w[j]
        elif i == j:
            a[i,j]=v[i]/(w[j]+1)
        else:
            a[i,j]=sum(v[0:j+1])*sum(w[0:i+1])
print('a= '); print(a)
print('Media: ', mean(a))
from numpy.linalg import *
print('Determinante: ', det(a))
```

Primeiro control de programación en Python de 2023

Escribe no editor de texto unha matriz de números naturais positivos (varias liñas con igual número de elementos cada unha) como por exemplo:

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

e chama o arquivo `datos2.txt`. Escribe un programa en Python que realice as seguintes operacións:

1. Lee o arquivo `datos2.txt` e almacénalo na matriz **a** de números enteiros sen comprobar os erros de lectura do arquivo. Visualiza na pantalla a matriz **a** e o seu rango.
2. Crea un vector **x** con elementos de **a** no intervalo (2,5] e un vector **y** cos elementos de **a** múltiplos de 3. Crea un vector **z** concatenando os vectores **x** e **y**. Visualiza todos os vectores na pantalla.
3. Sexa n o número de elementos do vector **z**, calcula un vector **v** de dimensión n , onde cada elemento v_i , $i = 0, \dots, n - 1$ defínese como:

$$v_i = z_i^2 z_{n-i-1}^2$$

4. Sexan sz e sv a suma dos elementos do vector **z** e **v**, respectivamente. Se $sv > sz$, calcula o número de veces que hai que dividir os elementos do vector **v** polos do vector **z** (división elemento a elemento, $\frac{v_i}{z_i}$, $i = 0, \dots, n - 1$) para que se cumpla que $sv < sz$. Mostra na pantalla sv , sz e o número de veces que se divide. Se, polo contrario, $sv \leq sz$, o programa debe escribir unha mensaxe de que non se pode realizar a operación.

```
from numpy import *
from numpy.linalg import *
a=loadtxt('datos2.txt', 'int')
print("a= "); print(a)
print('Rango de a: ', matrix_rank(a))
aux=a.flatten()
x=extract((a>2) & (a<=5), a)
y=extract(a%3==0, a)
z=concatenate((x,y))
print('x= ', x)
print('y= ', y)
print('z= ', z)
n=len(z); v=zeros(n)
for i in range(n):
    v[i]=z[i]**2*z[n-i-1]**2
sz=sum(z); sv=sum(v)
if sv > sz:
    c=0
    while sum(v) > sz:
        c = c+1
        v=v/z
    print('sz= ', sz, ' sv= ', sv, ' c= ', c)
else:
    print('Non se pode realizar a operación')
```

Primeiro control de programación en Python de 2023

Escribe un programa en Python que realice as seguintes operacións:

1. Pida por teclado unha ristra de números separados por espazos e os almacene no vector **x**. Visualiza na pantalla o vector **x**, a súa media e desviación típica.
2. Sexa n a dimensión do vector **x** e p , con $0 \leq p < n$ a posición onde se atopa o valor máximo de **x**. Crea un vector **y** cos elementos de **x** anteriores a p , é dicir, $y_i = x_i$ para $i = 0, \dots, p$, e outro vector **z** cos elementos posteriores a p , é dicir, $z_{i-p} = x_i$, tal que $i = p, \dots, n - 1$. Visualiza na pantalla ambos vectores.
3. Sexa m_1 e m_2 o número de elementos dos vectores **y** e **z** respectivamente. Se algún dos vectores non ten elementos ($m_1 = 0$ ou $m_2 = 0$) visualiza unha mensaxe de erro. En caso contrario, crea unha matriz **a**, de dimensións $m_1 \times m_2$, onde cada elemento a_{ij} calculase como:

$$a_{ij} = y_i^2 z_j^2 \quad i = 0, \dots, m_1 - 1; j = 0, \dots, m_2 - 1$$

Visualiza a matriz **a** na pantalla.

4. Calcula $s = \sum_{i=0}^{m_1-1} y_i^2 \sum_{j=0}^{m_2-1} z_j$. Logo, calcula e mostra por pantalla o número de elementos da matriz **a**, percorrendo a matriz por filas, que hai que sumar para superar o valor s . Se chegas ó final sen superar s , comeza de novo desde o principio.

```
from numpy import *
x=int_(array(input('x= ').rsplit()))
print('x= ', x, ' Media= ', mean(x), ' Desv.= ', std(x))
n=len(x)
p=argmax(x)
y=x[:p]; z=x[p:n]
print('y= ', y)
print('z= ', z)
m1=len(y); m2=len(z)
if m1==0 or m2==0:
    print('Erro')
else:
    a=zeros([m1,m2])
    for i in range(m1):
        for j in range(m2):
            a[i,j]=y[i]**2*z[j]**2
    s=sum(y)*sum(z)
    aux=a.flatten(); l=m1*m2;
    suma=0; k=0; contador=0
    while suma < s:
        suma = suma + aux[k]
        k+=1; contador +=1
        if k == l:
            k=0
    print('Numero de elementos: ', contador)
```

Primeiro control de programación en Python de 2023

Crea un arquivo de texto co editor con números enteiros separados por espazos nunha mesma liña. Por exemplo:

2 4 7 2 1 1 8 9 7 6 8 3 1 12 3

Escrebe un programa en Python que realice as seguintes operacións:

1. Lea por teclado o nome do arquivo cos datos e almacena o seu contido no vector de valores enteiros $\mathbf{x}$, comprobando se hai erro na lectura do arquivo. Visualiza na pantalla o vector $\mathbf{x}$ e o número de múltiplos de 4 que contén $\mathbf{x}$.
2. Sexa n o número de elementos de $\mathbf{x}$. Crea un vector $\mathbf{y}$ de dimensión n , onde cada elemento y_i é:

$$y_i = \frac{x_i \sin x_i}{x_{n-i-1}} \quad i = 0, \dots, n-1$$

e mostra $\mathbf{y}$ por pantalla.

3. Se o vector $\mathbf{x}$ contén números negativos ou 0, mostrar na pantalla unha mensaxe de erro ou rematar o programa. En caso contrario, sexa n , a e b o número de elementos de $\mathbf{x}$, o seu valor máximo e o seu valor mínimo respectivamente. Xera un número enteiro aleatorio p no intervalo $[a, b]$ e calcula o número de elementos k que hai que multiplicar para que

$$\prod_{i=0}^k x_i > p^3$$

Se chega o final do vector sin acadar o valor p^3 , entón k será o tamaño do vector. Mostra en pantalla a , b , p e k .

```
from numpy import *
from sys import *
from random import *
nome=input('Nome arquivo: ')
try:
    x=loadtxt(nome, int)
    if any(x<=0):
        print('x contén numeros negativos')
        exit()
except IOError:
    print('Erro abrindo arquivo ', nome)
    exit()
print('x= ', x)
print('Multiplos 4: ', sum(x%4==0))
n=len(x); y=zeros(n)
for i in range(n):
    y[i]=x[i]*sin(x[i])/x[n-i-1]
print('y= ', y)

a=max(x); b=min(x)
p=randint(b, a)
umbral = p**3
s=1
for i in range(n):
    s = s * x[i]
    if s >= umbral:
        break
print("n= %i, a=%i, b=%i, p=%i e k=%i"%(n, a, b, p, i+1))
```

Primeiro control de programación en Python de 2023

Escribe un programa en Python que realice as seguintes operacións:

1. Xerar un número enteiro aleatorio n no intervalo $[5,15]$. Crear un vector $\mathbf{x}$ de dimensión n dado pola expresión:

$$x_i = (i + 1) \sin\left(\frac{3\pi i}{4}\right) \quad i = 0, \dots, n - 1$$

e un vector $\mathbf{y}$ dado por:

$$y_0 = 0, \quad y_i = x_i - x_{i-1} \quad i = 1, \dots, n - 1$$

e visualiza ambos vectores na pantalla.

2. Crear unha matriz $\mathbf{a}$ de orde n con valores:

$$a_{ij} = \begin{cases} x_i & \text{se } i = j \\ y_i & \text{se } j = n - i - 1 \\ 10 & \text{resto dos casos} \end{cases}$$

3. Sexa sx a suma dos elementos do vector $\mathbf{x}$, é dicir, $sx = \sum_{i=0}^{n-1} x_i$, calcula o número de elementos da matriz $\mathbf{a}$, percorrida por columnas, que hai que sumar para superar o valor sx (se chegas ó final da matriz, volve a comezar desde o principio).

```
from numpy import *
from random import *
n=randint(5,16)
i=arange(n)
x=(i+1)*sin(3*pi*i/4)
print('x= ', x)
y=zeros(n)
for i in range(1, n):
    y[i]=x[i]-x[i-1]
print('y= ', y)
a=ones([n,n])*10
for i in range(n):
    a[i,i]=x[i]
    a[i,n-i-1]=y[i]
print('a= '); print(a)
sx=sum(x)
av=a.flatten('F')
s=0; k=0; c=0; n=len(av)
while s<sx:
    s= s + av[k]
    k=k+1; c += 1
    if k == n:
        k=0
print('n= ', n, 'sx= ', sx)
print('Numero sumandos: ', c)
```

Primeiro control de programación en Python de 2023

Escribe un programa en Python que realice as seguintes operacións:

1. Dado $n = 10$, crea un vector $\mathbf{x}$ de dimensión n con números enteiros aleatorios no intervalo $[0,20]$. Intercambia no vector $\mathbf{x}$ o valor máximo por o valor mínimo do vector. Visualiza $\mathbf{x}$ na pantalla antes e despois do intercambio.
2. Calcula unha matriz $\mathbf{a}$ de orde n onde:

$$a_{ij} = x_i e^{x_i} \sum_{j=0}^i \frac{x_j}{x_i + 1} \quad i = 0, \dots, n - 1$$

3. Lee números enteiros por teclado e almacénalos no vector $\mathbf{y}$ sempre que o valor lido z estea contido no vector $\mathbf{x}$ ($z \in \mathbf{x}$) e non sexa un número repetido, é dicir, non esta contido en $\mathbf{y}$ ($z \notin \mathbf{y}$). Visualiza $\mathbf{y}$ na pantalla e o seu número de elementos.

```
from numpy import *
from numpy.random import *
n=10
x=int_(rand(n)*20)
print('x= ', x)
imax=argmax(x)
imin=argmin(x)
aux=x[imax]
x[imax]=x[imin]
x[imin]=aux
print('x= ', x)
a=zeros([n,n])
s=0
for i in range(n):
    a[i]=x[i]*exp(x[i])*sum(x[0:i+1]/(x[i]+1))
print('a= '); print(a)
y=[]
while len(y) < n:
    z=int(input('Introduce numero: '))
    if z in x:
        if z in y:
            break
        else:
            y.append(z)
print('y= ', y)
```