

Primeiro control de programación en Python de 2024

Escribe un programa en Python que faga o seguinte:

1. Lea por teclado dous vectores \mathbf{v} e \mathbf{w} con números enteiros. Sea n_v e n_w o número de elementos de \mathbf{v} e \mathbf{w} respectivamente. Se $n_v \neq n_w$, engade números enteiros aleatorios no intervalo $[a, b]$, considerando que a e b son, respectivamente, os valores máximo e mínimo do vector \mathbf{v} , ao vector con menos elementos ata formar dous vectores da mesma dimensión.
2. Sexa n a dimensión de ambos vectores \mathbf{v} e \mathbf{w} . Calcula o valor de s dado pola seguinte expresión:

$$s = \sum_{i=0}^{n-1} v_{n-i} w_i$$

e visualízao na pantalla.

3. Calcula o número de elementos, $p + 1$, que hai que sumar para que se verifique a expresión (se chegamos ao final dos vectores sen que se cumpra a condición, o valor calculado debe ser n):

$$\sum_{i=0}^p \frac{v_i}{w_i} > 1$$

e visualiza para cada i o valor da suma.

```
from numpy import *
from random import *
v=int_(array(input('v= ').rsplit()))
w=int_(array(input('w= ').rsplit()))
nw=len(w)
nv=len(v)
if nv != nw:
    maxv=max(v); minv=min(v)
    if nv > nw:
        ne=nv-nw
        for i in range(ne):
            w=append(w, randint(minv, maxv))
    else:
        ne=nw-nv
        for i in range(ne):
            v=append(v,randint(minv, maxv))
print('v= ', v)
print('w= ', w)

n=len(v); s=0
for i in range(n):
    s= s + v[n-i-1]*w[i]
s=0; p=0
for i, j in zip(v, w):
    s = s + float(i)/j
    print('suma = ', s, " na iteración ", i)
    p = p + 1
    if s > 1:
        break
print('p= ', p)
```

Primeiro control de programación en Python de 2024

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

```
2 4 7 3
8 1 0 4
9 3 2 4
1 2 3 4
```

e almacena esta matriz no 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. Se a matriz non é cadrada visualiza unha mensaxe na pantalla, e se é cadrada visualiza a matriz **a** e o seu determinante.
2. Se a matriz é cadrada, sexa n a orde da matriz **a**. Calcula b_1 e b_2 dadas pola seguintes expresións:

$$b_1 = \sum_{i=0, \text{par}}^{n-1} \sum_{j=0}^{n-1} a_{ij} \quad b_2 = \sum_{i=1, \text{impar}}^{n-1} \sum_{j=0}^{n-1} a_{ij}$$

se $b_1 > b_2$, convirte a matriz **a** no vector **x** por columnas e calcula o número mínimo k de elementos para que se cumpra:

$$\sum_{i=0}^k x_{n-1-i} > b_1$$

visualizando na pantalla o valor de k e os elementos de **x** sumados.

```
from numpy import *
from numpy.linalg import *
a=loadtxt('datos2.txt')
[nf, nc]=a.shape
if nf != nc:
    print('Matriz non cuadrada')
else:
    print('a= ', a)
    print('Determinante de a= ', det(a))
    n=nf; b1=0; b2=0
    for i in range(0, n,2):
        b1 = b1 + sum(a[i])
    for j in range(1, n, 2):
        b2 = b2 + sum(a[j])
    print('b1= ', b1, ' b2= ', b2)
    if b1>b2:
        x=a.flatten('F')
        k=0; suma=0; y=[ ]
        while suma < b1:
            suma = suma + x[n-1-k]
            y.append(x[n-1-k])
            k = k+1
        print('No. elementos: ', k)
        print('Numeros sumados : ', y)
```

Primeiro control de programación en Python de 2024

Escribe un programa en Python que realice as seguintes operacións:

1. Pida por teclado un número real x que cumpra que $x > 10$ (se $x < 10$, visualiza unha mensaxe na pantalla e remata o programa). Calcula n , que é o número de veces que hai que dividir o número x por 2 para que o número x sexa menor que 1 e visualiza n na pantalla.
2. Crea unha matriz \mathbf{a} de orde n con números reais aleatorios no intervalo $[0, 10)$ e visualiza \mathbf{a} na pantalla.
3. Sexa t a traza de \mathbf{a} (suma dos elementos da diagonal principal) e b a suma dos elementos da primeira columna de \mathbf{a} , se $t > b$, calcula e visualiza na pantalla un vector \mathbf{v} de dimensión n , onde v_i , con $i = 0, \dots, n - 1$, ven dado pola expresión:

$$v_i = \sum_{j=0}^{n-1} a_{ij} a_{(n-i-1)j}$$

en caso contrario ($t \leq b$), calcula e visualiza o valor s dado pola expresión:

$$s = \sum_{i=0}^{n-1} a_{ii} \sum_{j=0}^i a_{ji}$$

```
from numpy import *
from numpy.random import *
from numpy.linalg import *
x=float(input('x>10 = '))

if x < 10:
    print('Numero non valido')
else:
    n=0
    while x >= 1:
        n = n + 1
        x=x/2
    print('n= ', n)
    a=rand(n,n)*10
    print('a= ', a)
    t=trace(a); b=sum(a[:,0])
    print('t= ', t, ' b= ', b)
    if t>b:
        v=zeros(n)
        for i in range(n):
            suma=0
            for j in range(n):
                suma = suma + a[i,j]*a[n-i-1, j]
            v[i]=suma
        print('v= ', v)
    else:
        s=0
        for i in range(n):
            s = s + a[i,i]*sum(a[0:i+1, i])
        print('s= ', s)
```

Primeiro control de programación en Python de 2024

Escribe un programa en Python que realice as seguintes operacións:

1. Crea un vector \mathbf{x} cos números $x_0 = 1$, $x_1 = 1$, e $x_i = x_{i-1} + x_{i-2}$ para $i \geq 2$. Crearemos números ata que $x_i \geq 100$. Visualiza na pantalla o vector \mathbf{x} creado e a súa dimensión, que asignamos a variable n .
2. Calcula os vectores \mathbf{y} e \mathbf{z} de dimensión n onde $y_i = \sin x_i \log 10x_i$, para $i = 0, 1, \dots, n - 1$ e $z_i = x_i + x_{i+1}$ para $i = 0, \dots, n - 2$ e $z_{n-1} = x_{n-1}$. Visualiza ambos vectores na pantalla.
3. Sexa \mathbf{a} a matriz xerada ó multiplicar matricialmente o vector columna \mathbf{x} polo vector fila \mathbf{z} (fíxate que $a_{ij} = x_i x_j$ para $i, j = 0, \dots, n - 1$). Calcula o determinante de \mathbf{a} e o seu rango. Visualiza na pantalla unha mensaxe indicando se o determinante de \mathbf{a} é “positivo”, “negativo” ou “cero”.

```
from numpy import *
from numpy.linalg import *
x=[1, 1]; num=2; i=2
while num < 100:
    x.append(num)
    i=i+1
    num = x[i-1]+x[i-2]
print('x= ', x)
n=len(x); print('n= ', n)
x=array(x); y=sin(x)*log(10*x)
z=zeros(n); z[n-1]=x[n-1]
for i in range(n-1):
    z[i]=x[i]+x[i+1]
print('y= ', y)
print('z= ', z)
a=x.reshape([n,1])*z
print(' rango: ', matrix_rank(a))
d=det(a)
if d>0:
    print('Determinante positivo')
elif d<0:
    print('Determinante negativo')
else:
    print('Determinante cero')
```

Primeiro control de programación en Python de 2024

Escribe no editor de texto unha liña de números enteiros positivos como por exemplo:

2 4 7 3 8 1 0 4 9 3 12 4 1 2 3 40

e chama ao arquivo `datos5.txt`. Escribe un programa en Python que realice as seguintes operacións:

1. Lee o arquivo `datos5.txt` (sen comprobar erros), almacénalo no vector \mathbf{x} e visualízalo na pantalla. Sexa n a dimensión de \mathbf{x} .
2. Sexan a e b o valor mínimo e máximo de \mathbf{x} , respectivamente. Xera un número enteiro aleatorio m no intervalo $[a, b]$. Se m está no vector \mathbf{x} , engade m ao vector \mathbf{y} . Repite este proceso ata que \mathbf{y} teña a mesma lonxitude que \mathbf{x} . Visualiza \mathbf{y} na pantalla.
3. Crea un vector \mathbf{z} , de dimensión n , onde cada elemento z_i , con $i = 0, \dots, n - 1$ defínese como:

$$z_i = \begin{cases} \sum_{j=0, \text{par}}^i y_j & \text{se } x_i \text{ é múltiplo de 3 e de 2} \\ \sum_{j=n-i}^i y_j x_i & \text{se } x_i \text{ é múltiplo de 2} \\ \sum_{j=0}^i x_j \sum_{k=0}^j y_k & \text{resto dos casos} \end{cases}$$

e visualiza na pantalla o vector \mathbf{z} , e o valor da súa media e mediana.

4. Constrúe a matriz \mathbf{b} poñendo cada un dos vectores \mathbf{x}, \mathbf{y} e \mathbf{z} nunha fila distinta.

```
from numpy import *
from random import *
x=loadtxt('datos5.txt', 'int')
print('x= ', x); n=len(x)
y=[]; a=min(x); b=max(x)
while len(y)<n:
    m=randint(a,b)
    if m in x:
        y.append(m)
print('y= ', y)
z=zeros(n); y=array(y)
for i in range(n):
    if x[i]%3 == 0 and x[i]%2 == 0:
        z[i]=sum(y[0:i+1:2])
    elif x[i]%2==0:
        z[i]=x[i]*sum(y[i:n-i+1])
    else:
        s=0
        for j in range(i+1):
            s = s + x[j]*sum(y[0:j+1])
        z[i]=s
print('z= ', z)
print('Media: ', mean(z), ' mediana= ', median(z))
b=vstack((x,y, z))
print('b= ', b)
```

Primeiro control de programación en Python de 2024

Escribe un programa en Python que realice as seguintes operacións:

1. Crea un vector \mathbf{x} de dimensión $n = 10$, onde $x_0 = 1$ e o elemento x_i , para $i = 1, \dots, n - 1$, será a suma dos números naturais entre 1 e $i + 1$, de modo que $x_1 = 1 + 2 = 3$, $x_2 = 1 + 2 + 3 = 6$, etc. Polo tanto, os valores de \mathbf{x} serían 1, 3, 6, 10, 15, 21, ... Visualiza \mathbf{x} na pantalla.
2. Crea outro vector \mathbf{y} de dimensión n , onde cada elemento y_i sexa un elemento de \mathbf{x} seleccionado aleatoriamente. Visualiza \mathbf{y} na pantalla.
3. Crea unha matriz \mathbf{a} dada por $\mathbf{a} = \mathbf{x}^T \mathbf{y}$, sendo \mathbf{x}^T o vector transposto (un vector columna) e \mathbf{y} un vector fila. Conta o número m de elementos da matriz \mathbf{a} que hai que sumar (percorrendo a matriz por filas) para superar o valor p dado por:

$$p = \sum_{i=0}^{n-1} \sum_{j=0}^{n/2} a_{ij}$$

e visualiza m na pantalla.

```
from numpy import *
from random import *
n=10; x=zeros(n); m=0; suma=0
for i in range(n):
    m=m+1
    suma = suma + m
    x[i] = suma
print('x= ', x)
y=zeros(n)
for i in range(n):
    m=randint(0,n-1)
    y[i]=x[m]
print('y= ', y)
a=int_(x.reshape([n,1])*y)
print('a= ', a)
fin=int(n/2+1)
p=sum(a[:, 0:fin]); print('p= ', p)
z=a.flatten()
suma=0; k=0
while suma<p:
    suma = suma + z[k]
    k=k+1
print('No. sumados: ', k)
```