Segundo control de programación en Python de 2023

Crea un arquivo de texto con distinto número de números enteiros en cada liña. Por exemplo o arquivo seguinte:

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

Escribe un programa chamado `exame1.py` que realice as seguintes operacións:

1. Pida por teclado o nome do arquivo de texto e lea os elementos do arquivo a dous vectores **x** e **y** da seguinte forma: o primeiro elemento para o vector **x** o segundo para o vector **y**, o terceiro para **x** o cuarto para **y**, o quinto para **x** e así sucesivamente ate acadar o final do arquivo. Comproba se hai erros na lectura. Se o arquivo ten menos de dous número remata o programa.

2. Se a dimensión de **x** e **y** é distinta, engade o final do vector que sexa menor o valor mínimo dese vector. Sea n a dimensión dos vectores **x** e **y**, representa nun gráfico en 2D os puntos de **x** como asteriscos azuis e os puntos de **y** como círculos vermellos. Marca en verde o valor máximo do vector **y**, debuxa unha liña verde co valor medio, m_x, do vector **x**, e resalta en negro os valores de **x** que son superiores a m_x. Pon títulos ós eixos e enreixado ó grafico.

3. Define unha función chamada `calcula()`, decide os argumentos axeitados, que calcule o número m de valores común a dous vectores **v** e **w** que hai que sumar para que a suma supere o valor 10 (cada valor común so se suma unha vez).

4. Aplica a función `calcula()` os vectores **x** e **y**.

```
from matplotlib.pyplot import *
from numpy import *
from sys import *
nome=input('Nome arquivo: ')
try:
    f=open(nome, 'r')
    aux=int_(array(f.read().split()))
    n=len(aux)
    if n <2:
        exit('Arquivo con menos de dous números')
    x=aux[0:n:2]
    y=aux[1:n:2]
    print('x= ', x)
    print('y= ', y)
    f.close()
except IOError:
    print('Erro lendo arquivo ', nome)
    exit()

nx=len(x); ny=len(y)
if nx != ny:
    if nx>ny:
        y=append(y, min(y))
    else:
        x=append(x, min(x))
n=len(x); t=arange(1,n+1)
plot(t, x, 'b*'); plot(t, y, 'ro')
pm=argmax(y); plot(pm+1, y[pm], 'go')
```

```
mx=mean(x); plot([1, n], [mx, mx],'g-')
ps=where(x>mx)[0]; plot(ps+1, x[ps], 'ko')
grid(True); xlabel('Punto'); ylabel('Valor'); show()

def calcula(v, w):
    m=0; s=0
    for i in v:
        if i in w:
            s = s + i
            m = m+1
            if s > 10:
                break
    return m
m=calcula(x,y)
print('m=%d'%m)
```

Segundo control de programación en Python de 2023

Escribe un programa en Python3 chamado `exame2.py` que realice o seguinte:

1. Lea por teclado un número enteiro n no intervalo [5,10], comprobando que o número é valido.
2. Defina unha función `calcula()`, cos argumentos axeitados, que calcule os elementos a_{ij} dunha matriz **a** con $i, j = 0, \ldots, m-1$. O valor a_{ij} será o número de elementos pares entre $i+1$ e $(i+j+2)^2$ que hai que sumar para superar a cantidade $2i+3j+10$.
3. Utiliza a función `calcula()` para crear unha matriz **a** cadrada de orde n e visualiza **a** na pantalla. Representa a matriz **a** nun gráfico en 3D, onde os valores da matriz son as alturas da superficie.
4. Garda no arquivo `saida2.txt` os elementos impares da matriz **a** con formato enteiro (cada fila nunha liña do arquivo).

```
from numpy import *
from matplotlib.pyplot import *
from mpl_toolkits.mplot3d import Axes3D
n=0
while n<5 or n>10:
    n=int(input('n= '))

def calcula(m):
    a=zeros([m,m])
    for i in range(m):
        for j in range(m):
            c=0; suma=0;
            umbral=2*i+3*j+10; fin=(i+j+2)**2
            for k in range(i+1, fin+1):
                if k%2==0:
                    c=c+1
                    suma=suma+k
                    if suma > umbral:
                        break
            a[i,j]=c
    return a
a=calcula(n)
fig = figure(); ax = Axes3D(fig)
x = arange(0,n); X, Y= meshgrid(x,x)
ax.plot_surface(X, Y, a); show()
try:
    f=open('saida2.txt', 'w')
    for i in range(n):
        aux=''
        for j in range(n):
           if a[i,j]%2==1:
               aux= aux+'%s '%str(int(a[i,j]))
        aux = aux + '\n'
        f.write(aux)
    f.close()
except IOError:
    print('Erro escribindo en saida2.txt')
```

Segundo control de programación en Python de 2023

Escribe un programa en Python chamado `exame3.py` que realice o seguinte:

1. Cree un vector **x** con números aleatorios no intervalo [1,100) de dimensión $n = 10$ e visualízao na pantalla.

2. Sexa **v** un vector cos índices dos elementos do vector **x**, atopa a ecuación do polinomio de orde 3 que mellor axusta os puntos (v_i, x_i), $i = 0, \ldots, n-1$ e representa nun gráfico 2D os pares de puntos con círculos azuis e unha liña verde coa curva do polinomio. Mostra en vermello os tres primeiros puntos (v_i, x_i) que estan por encima da curva do polinomio. Pon lendas e enreixado ó gráfico e gárdao no arquivo `grafico.png`.

3. Define a función `calcula()`, cos argumentos axeitados, que calcule dous vectores **y** e **z**. Para calcular cada elemento y_i e z_i, $i = 0, \ldots, n-1$, suma os valores de x_j, comezando por $j = 0$, ata que se supere o valor $i^4 + 50$. Cando $j > n-1$, debes continuar en $j = 0$. O elemento y_j será o índice j do último x_j sumado e o valor de z_i será o número de elementos sumados.

4. Calcula os vectores **y** e **z** usando a función `calcula()`. Pide por teclado un nome de arquivo e garda nese arquivo todos os elementos de **x**, **y** e **z**, cada elemento x_i, y_i e z_i nunha liña distinta en formato real con dous decimais para x_i e formato enteiro para y_i e z_i usando un ancho de 10.

```
from numpy import *
from numpy.random import *
from matplotlib.pyplot import *
n=10; x=1+99*random(n); v=arange(n)
p=polyfit(v,x, 3); xp=polyval(p,v)
figure(1); clf()
plot(v, x, 'bo', label='Puntos')
plot(v, xp, 'g-', label='Polinomio')
ind=where(x >xp)[0]
plot(ind[0:3], x[ind[0:3]], 'ro', label='Puntos maiores')
grid(True); legend(); show()
def calcula(x):
    n=len(x); y=zeros(n); z=zeros(n)
    for i in range(n):
        s=0;umbral=i**4+50; j=0; k=0
        while s < umbral:
            s=s + x[j]
            y[i]=j
            k=k+1; j=j+1
            if j==n:
                j=0
        z[i]=k
    return [y,z]
[y, z]=calcula(x)
try:
    nome=input('Nome arquivo: ')
    f=open(nome, 'w')
    f.write('%10s %10s %10s\n'%('x', 'y', 'z'))
    for i, j, k in zip(x, y, z):
        f.write('%10.2f %10i %10i\n'%(i,j, k))
    f.close()
except IOError:
    print('Erro escribindo en ', nome)
```

Segundo control de programación en Python de 2023

Escribe un programa chamado `exame4.py` que realice as seguintes operacións:

1. Lee do teclado un número enteiro par n no intervalo (2, 10), verificando que é un número válido. Crea unha matriz **a** de orde n con números reais aleatorios no intervalo [0,10].
2. Define $m = 0$ e percorre os elementos a_{ij} da matriz **a** escribindo, para cada a_{ij}, a seguinte información no arquivo `saida4.txt`. Se $j < i - 1$ escribe 6 espacios en branco. Cando $i - 1 \leq j \leq i + 1$ escribe a_{ij} con ancho 6 e un decimal, e incrementa a variable m nunha unidade. No resto dos casos, escribe 6 símbolos `'x'`. So podes cambiar a liña seguinte cando se acade o final da fila. Finalmente, escribe o valor da variable m nunha liña, con formato enteiro.
3. Define unha función `calcula()`, cos argumentos axeitados, que redondee a valores enteiros a matriz **a**, calculando: o número k de valores enteiros de **a** que son impares ou maiores que 5, e o número de elementos, c, percorrendo a matriz por columnas, que hai que sumar para superar o valor $2k$.
4. Chama a función `calcula()` e visualiza na pantalla os valores de k e c.
5. Representa a matriz **a** como un mapa de calor e ponlle título.

```
from numpy import *
from numpy.random import *
from sys import *
from matplotlib.pyplot import *
n=0
while n<3 or n>9 or n%2==1:
    n=int(input('n= '))
a=10*random([n,n])
try:
    f=open('saida4.txt', 'w'); m=0
    for i in range(n):
        for j in range(n):
            if j<i-1:
                f.write('      ')
            elif i-1 <= j and j <= i+1:
                f.write('%6.1f'%a[i,j]); m=m+1
            else:
                f.write('xxxxxx')
        f.write('\n')
    f.write('m= %d\n'%m)
    f.close()
except IOError:
    print('Erro escribindo en saida4.txt'); exit()
def calcula(a):
    b=int_(a)
    x=b.flatten('F')
    k=len(x[logical_or(x%2==1,x>5)])
    s=0; umbral=2*k; c=0; n=len(x)
    while s < umbral:
        s = s + x[c]; c=c+1
        if c == n:
            break
    return [k,c]
[k,c]=calcula(a)
print('k=%d, c=%d\n'%(k,c))
figure(1);clf(); imshow(a)
title('Mapa de calor');show()
```

Segundo control de programación en Python de 2023

Escribe un programa chamado `exame5.py` que realice as seguintes operacións:

1. Lea por teclado unha expresión matemática dunha función $f(x)$, sendo x unha variable simbólica, como unha cadea de caracteres (utiliza, por exemplo, $f(x) = 3x^2 + 6$). Convírtea a función lambda e calcula o valor da función en 100 puntos equidistantes no intervalo [0,10]. Calcula tamén o valor da derivada e integral indefinida de $f(x)$ nos puntos dese intervalo. Sexan **y**, **z** e **v** os vectores co valor da función $f(x)$, a súa derivada e a súa integral indefinida respectivamente. Representa gráficamente **y**, **z** e **v** nun gráfico 2D, poñendo títulos os eixos, lenda e enreixado no gráfico.

2. Define unha función `calcula()`, cos argumentos axeitados, que dado un vector **w** de dimensión n substitúa cada elemento $w_i > n$ por $w_i - n$, e cada $w_i < 50$ por $w_i + n$ con $i = 0, \ldots, n-1$. A función ten que calcular o número de valores, k, no primeiro caso ($w_i > n$) e o número de elementos de **w** que hai que sumar, m, comezando por w_0 para superar $20k$ (se chegas ó final de **w**, debes de comezar de novo polo principio).

3. O programa principal debe aplicar a función `calcula()` ós vectores **y**, **z** e **v** e almacenar no arquivo `saida5.txt` o valor de k e m (éste con ancho 5) para cada vector (cada par de valores nunha liña).

```
from numpy import *
import sympy as sp
from matplotlib.pyplot import *
expr=input('f(x)= ')
x=sp.symbols('x')
f=sp.lambdify(x, expr)
df=sp.lambdify(x, sp.diff(expr, x))
g=sp.lambdify(x, sp.integrate(expr,x))
t=linspace(0,10, 100)
y=f(t); z=df(t); v=g(t)
figure(1); clf()
plot(t,y,'b-', label='f(x)')
plot(t,z, 'g-', label='df(x)')
plot(t,v, 'r-', label='intf(x)')
xlabel('x'); ylabel('Valor')
grid(True); legend(); show()
def calcula(w):
    n=len(w); k=0
    for i in range(n):
        if w[i]>n:
            w[i]=w[i]-n; k=k+1
        if w[i]<50:
            w[i]=w[i]+n
    s=0; m=0; u=20*k; i=0
    while s<u:
       s=s+w[i]
       i=i+1; m=m+1
       if i==n:
           i=0
    return [w, k, m]
[y, ky, my]=calcula(y)
[z, kz, mz]=calcula(z)
[v, kv, mv]=calcula(v)
try:
    f=open('saida5.txt', 'w')
    f.write('y: k=%d m=%5d\n'%(ky, my))
```

```
    f.write('z: k=%d m=%5d\n'%(kz, mz))
    f.write('v: k=%d m=%5d\n'%(kv, mv))
    f.close()
except IOError:
    print('Erro abrindo saida5.txt')
```

Segundo control de programación en Python de 2023

Crea co editor de texto o seguinte arquivo `datos6.txt`:

```
12  2  3
7  5
4  6  8
5
```

Escribe un programa chamado `exame6.py` que realice as seguintes operacións:

1. Le tódolos elementos do arquivo `datos6.txt` ó vector fila **x**. Percorre o vector ate que a suma dos elementos supere 20 ou chegues o seu final. Se existen elementos restantes, deben eliminarse de **x** e almacenarse no vector fila **y**. En caso contrario, o vector **y** debe conter os elementos pares de **x**.

2. Representa nun gráfico en 2D os puntos do vector **x** en azul e os do vector **y** en vermello. Pon enreixado e lenda.

3. Define a función `calcula()`, cos argumentos axeitados, que calcule as matrices **a** e **b** de ordes $n \times m$ e $m \times n$ respectivamente, onde n e m son as lonxitudes dos vectores **x** e **y** respectivamente. A matriz **a** debe ser $\mathbf{x}^T\mathbf{y}$, onde $\mathbf{x}^T$ é un vector columna, de modo que $a_{ji} = x_j y_i$, mentres que a matriz **b** debe ter o elemento $b_{ij} = \sum_{k=0}^{j} x_k \sum_{l=0}^{i} y_l a_{kl}$, en ambos casos con $i = 0, \ldots, m-1$ e $j = 0, \ldots, n-1$.

4. O programa ten que chamar a función `calcula()` e visualizar as matrices **a** e **b** na pantalla.

```
from numpy import *
from sys import *
from matplotlib.pyplot import *
try:
    f=open('datos6.txt', 'r')
    x=int_(f.read().rsplit())
    f.close()
except IOError:
    print('Erro lendo datos6.txt'); exit()
print('x= '); print(x)
n=len(x); suma=0
for i in range(n):
    suma = suma + x[i]
    if suma > 20:
        break
if i == n-1:
    y=extract(x%2==0, x)
else:
    y=x[i+1:n].copy()
    x=x[0:i+1].copy()
figure(1); clf()
plot(x, 'b*', label='Vector x')
plot(y, 'ro', label='Vector y')
grid(True); legend(); show()
def calcula(x,y):
    n=len(x); m=len(y)
    a=x.reshape([n,1])*y
    print('m= ', m, ' n= ', n, 'a = ', a)
    b=zeros([m,n])
    for i in range(m):
        for j in range(n):
```

```
            s=0
            for k in range(j+1):
                s = s+x[k]*sum(y[0:i+1]*a[k, 0:i+1])
            b[i, j]=s
    return [a, b]
[a, b]=calcula(x,y)
print('a= '); print(a)
print('b= '); print(b)
```