Segundo control de programación en Python de 2022

Crea un arquivo de texto `datos1.txt` con letras na primeira columna e o valor de cada letra na segunda columna. Por exemplo o arquivo seguinte:

a 3
b 7
c 10
e 4
i 2
o 1

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

1. Lee o arquivo de texto `datos1.txt`, comprobando se hai erros, e almacene as letras nunha lista e os valores de cada letra noutra lista. Imprime por pantalla as dúas listas.
2. Define unha función co nome `valorTexto()`, tes que decidir os argumentos, que calcule o valor dunha cadea de caracteres e o número de espacios en branco, usando as listas do apartado anterior. O valor dunha cadea de caracteres calcúlase sumando os valores de todas as súas letras no arquivo `datos1.txt`, considerando que as letras que non están no arquivo teñen o valor 0.5 e os espacios en branco 1.
3. O programa ten que pedir cadeas de caracteres por teclado ate que se introduce unha cadea baleira (de lonxitude 0) e calcular para cada cadea o valor e o número de espacios.
4. Representa nun gráfico de barras o valor da cadea e o o número de espacios (poñendo no eixo x o número de cadea de cada dato). Pon títulos os eixos e lendas ó gráfico. Garda o gráfico no arquivo `analisis.png`.

```
from numpy import *
from matplotlib.pyplot import *
from sys import *
try:
    f=open('datos1.txt', 'r')
    letras=[]; puntos=[]
    for linha in f:
        aux=linha.rsplit()
        for i in aux:
            letras.append(aux[0])
            puntos.append(float(aux[1]))
    f.close()
except IOError:
    exit('Erro abrindo arquivo datos1.txt')

def valorTexto(l, p, texto):
    valor=0; ne=0
    for le in texto:
        if le in l:
            i=l.index(le)
            valor = valor + p[i]
        elif le == ' ':
            valor = valor + 1; ne = ne+1
        else:
            valor = valor + 0.5
    return [valor, ne]
texto = input('Texto para analizar: ')
vp=[]; np=[]
```

```
while len(texto)>0:
    [v, n]=valorTexto(letras, puntos, texto)
    vp.append(v); np.append(n)
    texto = input('Texto para analizar: ')
n=len(vp)
figure(1); clf()
t=range(n)
bar(t, vp, label='Valor total')
bar(t, np, label='Numero espacios')
legend(); xticks(list(range(n)))
xlabel('textos'); ylabel('valor'); grid(True); show(False)
savefig('analisis.png')
```

Segundo control de programación en Python de 2022

Crea un arquivo de texto `datos2.txt` no que a fila i-ésima ten i números enteiros, como por exemplo:

2
2 3
1 2 3
4 3 5 6

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

1. Pida polo teclado o nome do arquivo e lea o arquivo ó vector **x**. Sea n a lonxitude de **x**. Define outro vector **y** de dimensión n con elementos aleatorios entre o máximo e o mínimo do vector **x**.
2. Define unha función `calculaMatriz()`, tes que decidir os argumentos, que dado un vector **z** de dimensión n, calcule unha matriz cadrada de orde n. Para calcular cada elemento a_{ij} desta matriz, debes sumar os elementos de **z** ate que a suma supere o valor ij^2 ou chegues ó final do vector **z**. O valor a_{ij} será o número de elementos sumados.
3. Desde o programa principal, calcula as matrices **a** e **b** aplicando a función `calculaMatriz()` os vectores **x** e **y** respectivamente. Visualiza na pantalla a matriz que teña o determinante maior.
4. Representa no mesmo gráfico de liñas o vector **x** con asteriscos azuis e o vector **y** con asteriscos vermellos, marcando no gráfico con asteriscos verdes os puntos nos que o vector **x** e maior que o vector **y**. Pon títulos e lendas ó gráfico.

```
from numpy import *
from sys import *
from matplotlib.pyplot import *
from numpy.random import randint
from numpy.linalg import det
try:
    f=open('datos2.txt', 'r')
    x=[];
    for linha in f:
        aux=linha.rsplit()
        for i in aux:
            x.append(int(i))
    f.close()
except IOError:
    exit('Erro abrindo arquivo datos2.txt')

def calculaMatriz(z):
    n=len(z); a=zeros([n,n])
    for i in range(n):
        for j in range(n):
            umbral=(i+1)*(j+1)
            k=0; suma=0; m=0
            while suma < umbral:
                suma = suma + z[k]
                m=m+1; k=k+1
                if k == n:
                    break
            a[i,j]=m
    return a
a=calculaMatriz(x)
n=len(x); y=randint(min(x), max(x), n)
b=calculaMatriz(y)
```

```
if det(a) > det(b):
    print('a= ', a)
else:
    print('b= ', b)
figure(1); clf()
t=arange(n); x=array(x)
plot(t, x, 'b*', label='x')
plot(t, y, 'r*', label='y')
pos=where(x > y)[0]
plot(t[pos], x[pos], 'go', label= 'x>y')
title('Grafico de puntos')
xlabel('Indice'); ylabel('Magnitude')
legend(); grid(True); show(False)
```

Segundo control de programación en Python de 2022

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

1. Lea por teclado unha serie de números (todos na mesma liña) ate que se poida convertir nunha matriz cadrada **a**. Para comprobar se un número é un cadrado exacto, podes comprobar que o enteiro por exceso e defecto da súa raiz cadrada sexan iguais. Proba, por exemplo, cos números `8 1 2 3 5 6 9 7 2`. Sexa n a orde da matriz **a**.

2. Gardar no arquivo `datos3.txt` os valores pares por filas (na primeira liña do arquivo os valores pares da primeira fila da matriz, e así sucesivamente).

3. Logo o programa principal debe chamar a función `calcula(...)`, cos argumentos axeitados, que calcule:

   - Unha matriz **b** de orde n onde cada elemento b_{ij} sexa o número de valores pares de **a** que hai que sumar para superar o valor $(i+1)n+j$ para $i, j = 0, 1, \ldots, n-1$.
   - Un vector **x** de lonxitude m, sendo m o número de valores pares de **a**, tal que o elemento x_i sexa a suma dos valores pares de **a** elevado a i.
   - Un número real p igual á media dos elementos pares de **a**

4. Desde o programa principal visibiliza o valor de p con un ancho de 7 e 2 decimais, representa nun gráfico de liñas o vector **x**, resaltando dunha cor distinta o valor do máximo, e visibiliza a matriz **b** na pantalla.

```
from matplotlib.pyplot import *
from sys import *
from numpy import *
while True:
    x=int_(input('a= ').rsplit())
    k=len(x); d=sqrt(k)
    if ceil(d)==floor(d):
        n=int(d); a=x.reshape([n,n]); break
try:
    f=open('datos3.txt', 'w')
    for i in range(n):
        y=extract(a[i]%2==0, a[i])
        ny=len(y)
        cad=''
        for j in y:
            cad = cad + '%i '%j
        cad = cad +'\n'
        f.write(cad)
    f.close()
except IOError:
    print('Erro escribindo en datos3.txt'); exit()

def calcula(a):
    n=a.shape[0]
    b=zeros([n, n])
    y=extract(a%2==0, a); m=len(y)
    for i in range(n):
        for j in range(n):
            umbral=(i+1)*n+j
            s=0; nc=0
            for k in y:
```

```
                s= s + k
                nc = nc + 1
                if s > umbral:
                    break
            b[i,j]=nc
    x=zeros(m); sp=sum(y)
    for i in range(m):
        x[i]=sp**i
    p=mean(y)
    return [b, x, p]

[b, x, p]=calcula(a)
print('b= ', b)
figure(1); clf()
plot(arange(len(x)), x, 'b*')
plot(argmax(x), max(x), 'ro')
grid(True); show(False)
print('p=%7.2f'%p)
```

Segundo control de programación en Python de 2022

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

1. Crea un vector **x** de dimensión $n = 8$ con valores enteiros aleatorios entre 1 e 10.
2. Define unha función `calcula(...)`, cos argumentos axeitados, que calcule unha matriz cadrada **a** de orde n con elementos a_{ij} dados por:

$$a_{ij} = \begin{cases} \dfrac{x_i + x_{i+1}}{2} & i = 0, j < n-1 \\ \dfrac{x_{n-1} + x_0}{2} & i = 0, j = n-1 \\ \dfrac{a_{(i-1)j} + a_{(i-1)(j+1)}}{2} & i > 0, j < n-1 \\ \dfrac{a_{(i-1)(n-1)} + a_{(i-1)0}}{2} & i > 0, j = n-1 \end{cases}$$

3. Desde o programa principal, garda nun arquivo (pide o seu nome por teclado) o vector **x** na primeira liña como números enteiros que ocupan 6 espacios. Nas seguintes liñas a matriz **a** do xeito seguinte: se $a_{ij} \in (4, 8)$, os datos almacénanse con ancho 6 e 2 decimais e, en caso contrario, almacénase unha cadea de caracteres `------` (6 signos `-`), de modo que as columnas da matriz **a** queden aliñadas.
4. Representa nun gráfico tridimensional os valores da matriz **a**, considerando que os valores nos eixos `x` e `y` son os índices de fila e columna da matriz **a**. Pon título e etiquetas nos eixos ó gráfico e gardao no arquivo `figura4.png`.

```
from numpy import *
from matplotlib.pyplot import *
from sys import exit
from numpy.random import *
from mpl_toolkits.mplot3d import Axes3D
n=8; x=randint(1,10, n)

def calcula(x):
    n=len(x); a=zeros([n,n])
    a[0, n-1]=(x[n-1]+x[0])/2
    for j in range(n-1):
        a[0, j]=(x[j]+x[j+1])/2
    for i in range(1, n):
        for j in range(n-1):
            a[i,j]=(a[i-1, j]+a[i-1,j+1])/2
        a[i,n-1]=(a[i-1,n-1]+a[i-1,0])/2
    return a
a=calcula(x)
nome=input('Nome arquivo: ')
try:
    savetxt(nome, x, '%6i', newline='')
    fout=open(nome, 'a')
    fout.write('\n')
    for i in range(n):
        aux=''
        for j in range(n):
            y=a[i,j]
            if y > 4 and y<8:
                aux= aux + '%6.2f'%y
            else:
```

```
                aux= aux + ' -----'
        aux=aux+'\n'
        fout.write(aux)
    fout.close()
except IOError:
    exit('Erro escribindo en %s'% nome)
fig = figure(); clf()
ax = Axes3D(fig)
x=arange(n); y=arange(n)
X, Y= meshgrid(x, y)
ax.plot_surface(X, Y, a, rstride=1, cstride=1, cmap='hot')
ax.set_xlabel('x'); ax.set_ylabel('y')
ax.set_title('Grafico 3D'); savefig('grafico3d.png')
grid(True); legend(); show(False)
```

Segundo control de programación en Python de 2022

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

1. Lea por teclado unha cadea de caracteres s cunha función matemática. Por exemplo, podes usar $x^2 - 3x + 5$. Transforma a cadea de caracteres nunha función f. Constrúe o vector $\mathbf{t}$ con $n = 50$ puntos no intervalo [a, b]=[-2,4] e constrúe o vector $\mathbf{y}$ co valor da función f sobre o vector $\mathbf{t}$. Calcula simbólicamente a integral indefinida da función f e avalíaa no mesmo intervalo.

2. Calcula numéricamente a derivada da función f, df, dada pola expresión:

$$df_i = \frac{y_{i+1} - y_i}{h} \quad i = 0, \ldots, n-2 \quad h = \frac{b-a}{n} \tag{1}$$

3. Representa gráficamente a función f, a súa integral indefinida no intervalo [a,b] con liñas azul e verde, respectivamente, poñendo lendas, etiquetas nos eixos e un título no gráfico.

4. Calcula a integral definida, d, que podes realizar simbólicamente, ou numéricamente utilizando a expresión:

$$d = h \sum_{i=0}^{n-1} y_i \quad h = \frac{b-a}{n} \tag{2}$$

   e garda no arquivo `saida5.txt` os vectores $\mathbf{t}$ e $\mathbf{y}$ (un en cada liña) usando números reais con 2 decimais. Finalmente almacena o valor da integral definida d noutra liña.

5. Define unha función `calcula(...)`, cos argumentos axeitados, que calcule o número de elementos dun vector que hai que sumar para alcanzar o umbral p comezando polo inicio do vector (se non se acada p devolve o número de elementos do vector).

6. Aplica a función `calcula(...)` os vectores `y` e `df` utilizando como umbral o valor d da integral definida calculado anteriormente e visualiza os resultados.

```
from numpy import *
from matplotlib.pyplot import *
from sys import exit
import sympy as sp

expr=input('f(x)= ')
x=sp.symbols('x')
f=sp.lambdify(x, expr)
g=sp.lambdify(x, sp.integrate(expr, x))
n=50; a=-2; b=4; t=linspace(a, b, n, endpoint=True)
y=f(t); iy=g(t)
m=n-1; df=zeros(m)
for i in range(m):
    df[i]=y[i+1]-y[i]
figure(1); clf()
plot(t, y, 'b-', label='f(x)')
plot(t, iy, 'g-', label='intf(x)')
title('f(x) e intf(x)'); xlabel('x'); ylabel('f')
legend(); grid(True); show(False)
#integral definida simbolicamente
ds=sp.integrate(expr, (x,a,b))
#integral definida numericamente
dn=0; h=(b-a)/n
for i in range(n):
    dn=dn+h*y[i]
```

```
# vectorizado a integral numerica
dnv = h*sum(y)
try:
    savetxt('saida5.txt', vstack((t, y)), '%.2f')
    f=open('saida5.txt', 'a')
    f.write('Integral definida= %.2f\n'%dn)
    f.close()
except IOError:
    exit('Erro escribindo en saida5.txt')
def calcula(y,p):
    n=len(y); suma=0; ni=0
    for i in range(n):
        suma=suma + y[i]; ni=ni+1
        if suma > p:
            break
    return ni
print('y: ni=%i\n' % calcula(y, dn))
print('df: ni=%i\n' % calcula(df, dn))
```

Segundo control de programación en Python de 2022

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

1. Define a variable $n = 5$ e lee por teclado un número enteiro x no intervalo [2,7], sen deixar avanzar o programa ate que x estea neste intervalo.

2. Crea un vector **v** con n elementos definido como $v_i = x + 2ix + i^2$, para $i = 0, \ldots, n-1$.

3. Define unha función `calcula(...)`, cos argumentos axeitados, que calcule a matriz **a** de orde n con elementos $a_{00} = x$ e $a_{0j} = v_j$ para $j = 1, \ldots, n-1$. Para calcular os elementos a_{ij} con $i = 1, \ldots, n-1$ e $j = 0, \ldots, n-1$, percorre o vector **v** comezando por v_0 e rematando cando o número de elementos percorridos sexa $i(j+2)$. Se acadas o final do vector **v**, volve o comezo. O valor de a_{ij} debe ser o último elemento percorrido.

4. O programa principal ten que chamar a función `calcula()` e almacenar no arquivo `saida6.txt` o vector **v** nunha liña e os valores impares da matriz **a** na liña seguinte.

5. Representa nun gráfico os elementos do vector **v** como puntos azuis e unha liña horizontal de cor verde coa media dos valores de **v**. Resalta en vermello os elementos de **v** superiores a media. Pon lendas, título e etiquetas nos eixos ó gráfico.

```
from numpy import *
from matplotlib.pyplot import *
from sys import exit
n=5; x=0
while x<2 or x>7:
    x=int(input('x= '))
i=arange(n)
v=x+2*i*x+i**2
def calcula(v, x):
    n=len(v); a=zeros([n,n])
    a[0,0]=x
    a[0,1:n]=v[1:n]
    for i in range(1,n):
        for j in range(n):
            umbral=i*(j+2); k=0; l=0
            while k < umbral:
                k=k+1; l=l+1
                if l==n:
                    l=0
            a[i,j]=v[l]
    return a
a=calcula(v,x)
print('a= ', a)
def escribeVector(y, farquivo):
    n=len(y); aux=''
    for i in y:
        aux = aux + '%i '%i
    aux=aux+'\n'
    farquivo.write(aux)
    return

try:
    f=open('saida6.txt', 'w')
    escribeVector(v, f)
    imp=extract(a%2==1, a)
```

```
    escribeVector(imp,f)
    f.close()
except IOError:
    exit('Erro escribindo saida6.txt')
figure(1); clf()
plot(i, v, 'b*', label='Vector v')
media=mean(v)
plot([0, n-1], [media, media], 'g-', label='Media')
ind=where(v > media)[0]
plot(ind, v[ind], 'r*', label='v > media')
title('Vector v'); xlabel('i');ylabel('v(i)')
grid(True); legend(); show(False)
```