

Exame de xullo de programación en Python de 2020

Crea co editor un arquivo de texto `datos.dat` con dúas filas de n números cada unha. Por exemplo, co seguinte contido (neste caso $n = 6$):

```
1 2 3 4 5 6
5 4 3 8 6 13.2
```

Escribe un programa en Python chamado `xullo.py` que realice as seguintes operacións:

1. Pide por teclado o nome do arquivo e almacena os números da primeira liña no vector \mathbf{v} e os da segunda liña no vector \mathbf{w} , abortando o programa se hai erros na lectura do arquivo ou o arquivo non ten dúas liñas.
2. Crea unha matriz \mathbf{a} cadrada de orde n con elementos $a_{ij} = v_i + 2w_j$, con $i, j = 0, \dots, n - 1$ e visualiza o seu contido na pantalla.
3. Define a función `calcula()`, cos argumentos axeitados, que calcule: 1) un vector \mathbf{x} de lonxitude n , de modo que x_i , con $i = 0, \dots, n - 1$ sexa a suma das $i + 1$ primeiras filas e columnas da matriz \mathbf{a} ; 2) a suma s dos valores $v_i a_{ii} x_i$ con $i = 0, 1, 2, \dots$, rematando cando $s > \sum_{i=0}^{n-1} x_i^2$, de modo que cando $i = n$ pase a ser $i = 0$; e 3) o número m de valores sumados.
4. Chama a función `calcula()` e garda no arquivo `datossaida.txt` o vector \mathbf{x} nunha liña con valores con dous decimais, o valor de s noutra liña, e o valor de m na terceira liña como un número enteiro.
5. Xera un número enteiro m no intervalo $[10, 20]$ e crea un vector \mathbf{t} con m números equidistantes no intervalo $[\min(v), \max(v)]$. Dados os puntos (v_i, w_i) , con $i = 0, \dots, n - 1$, calcula os puntos interpolados y_i , $i = 0, \dots, m - 1$ dos elementos do vector \mathbf{t} utilizando interpolación lineal. Representa nun gráfico os valores do vector \mathbf{w} con puntos azuis (considera o vector \mathbf{v} no eixo x) e os valores do vector \mathbf{y} con puntos verdes. Marca cun punto vermello o valor máximo do vector \mathbf{w} e cunha liña vermella o valor medio. Pon lendas, etiquetas nos eixos e enreixado no gráfico.

```
from numpy import *
from sys import exit
from matplotlib.pyplot import *
import random as ra

nome=input('Nome arquivo: ')
try:
    b=loadtxt(nome)
    nf=size(b,0)
    if(nf != 2):
        exit('O arquivo non ten duas linhas')
except IOError:
    exit('Erro lendo arquivo %s' % nome)
v=b[0]; w=b[1]; n=len(v)
print('v= ', v); print('w= ', w)
a=zeros([n,n])
for i in range(n):
    for j in range(n):
        a[i,j]=v[i]+2*w[j]
print('a= ', a);
def calcula(v, b):
    n=len(v); x=zeros(n)
    for i in range(n):
        x[i]=sum(b[:i+1,:i+1])
```

```

s=0; umbral=sum(x**2); i=0; m=0
while s<=umbral:
    s = s + v[i]*b[i,i]*x[i]
    i=i+1
    if i == n:
        i=0
        m +=1
    return [x, s, m]
[x,s, m]=calcula(v,a)
print('x= ', x); print('s=%.2f m=%i'% (s, m))
try:
    f=open('datossaida.txt', 'w')
    cad=''
    for i in range(n):
        cad= cad + '%.2f ' % x[i]
    cad= cad + '\n'
    f.write(cad)
    f.write('s=%.2f m=%i\n'% (s, m))
    f.close()
except IOError:
    exit('Erro escribindo en datossaida.txt')
m=random.randint(10,20); minv=min(v); maxv=max(v)
t=linspace(minv, maxv, m)
y=interp(t, v, w)
clf(); plot(v, w, 'b*', label='w')
plot(t, y, 'g*', label='y')
plot(v[argmax(w)], max(w), 'rs', label='max(w)')
mu=mean(w)
plot([minv, maxv], [mu, mu], 'r-', label='media(w)')
xlabel('Eixo x'); ylabel('Eixo y'); legend(); grid(True); show(False)

```