

# Exame de xaneiro de programación en Python de 2018

---

Escribe un programa en Python chamado `xaneiro.py` que xere un número enteiro aleatorio  $n$  no intervalo  $[4,20]$  e realiza as seguintes operacións:

1. Calcule un vector  $\mathbf{p}$  de dimensión  $n$ , onde cada elemento  $p_i$  ven dado pola expresión:

$$p_i = \sum_{m=0}^{d-1} \frac{m}{5d} \cos\left(\frac{i\pi m}{20d}\right) \quad i = 0, \dots, n-1$$

considerando  $d = 100$ .

2. Calcula o vector  $\mathbf{q}$  de dimensión  $k = 50$  realizando unha interpolación lineal dos puntos do vector  $\mathbf{p}$  (a coordenada horizontal é o índice de cada compoñente do vector  $\mathbf{p}$ ). Representa gráficamente os valores de  $\mathbf{p}$  como asteriscos en azul e os valores interpolados como unha liña verde. Pon etiquetas ós eixos, título á gráfica e lendas para os puntos de  $\mathbf{p}$  e os puntos interpolados. Marca no gráfico cun círculo vermello o valor máximo.
3. Define unha función chamada `calculos()` cos argumentos axeitados, que calcule unha matriz  $\mathbf{b}$  e un vector  $\mathbf{y}$  da seguinte maneira:

- A matriz  $\mathbf{b}$  constrúese da seguinte maneira: se a suma do vector  $\mathbf{p}$  xa fose maior que 100, a matriz  $\mathbf{b}$  será un vector fila de dimensión  $n$  de ceros. En caso contrario, a matriz  $\mathbf{b}$  constrúese engadindo verticalmente o vector  $\mathbf{p}$  mentres que a suma dos elementos da matriz sexa menor que 100.
- Cada elemento  $y_j$ ,  $j = 0, \dots, n-1$  do vector  $\mathbf{y}$  defínese como:

$$y_j = \sum_{i=0}^{m-1} b_{ij}$$

onde  $m$  é o número de filas da matriz  $\mathbf{b}$ .

4. Finalmente, o programa debe chamar á función `calculos()` e gardar no arquivo `resultado.txt` os vectores  $\mathbf{p}$  e  $\mathbf{y}$  e a matriz  $\mathbf{b}$  utilizando reais con ancho 7 con dous decimais.

```
#!/usr/bin/env python
#-*-coding:utf-8-*-
from numpy import *
from numpy.random import *
from pylab import *

n=randint(4,20)
p=zeros(n)
d=100;m=arange(d)
for i in range(n):
    #t=0 # sen vectorizar
    #for m in range(d):
        #t=t+m*cos(i*pi*m/20/d)/5/d
    #p[i]=t
    p[i]=sum(m*cos(i*pi*m/20/d)/5/d)

k=50;x=range(n);x2=linspace(0,n-1,k)
q=interp(x2,x,p)
imax=argmax(p)

clf()
plot(x,p,'*b',label='Puntos')
```

```

plot(x2,q,'g-',label=u'Interpolante')
plot(imax,max(p),'or')
xlabel('Eixo OX');ylabel('Eixo OY')
title(u'Interpolación linear')
legend(loc='upper right');grid(True)
show(False)

def calculos(p):
    n=len(p)
    if sum(p)>=100:
        b=zeros(n)
        y=b.copy()
    else:
        b=p.copy();sp=sum(p);s=sp
        while s<100:
            b=vstack([b,p])
            s=s+sp
        y=sum(b,0)
    return b,y

b,y=calculos(p)
savetxt('resultado.txt',vstack([p,b,y]),'%7.2f')

```