

# Exame de Python, xaneiro de 2022

Escrebe un programa en Python chamado `xaneiro.py` que realice o seguinte:

1. Lea do teclado unha cadea de caracteres cunha función  $f(x)$ , sendo  $x$  unha variable simbólica, por exemplo, `x**2`. Sexa  $\mathbf{xi}$  un vector de 49 números equidistantes no intervalo  $[0, 2\pi]$ , calcula o valor de  $f(\mathbf{xi})$ , da derivada de  $f$  en  $\mathbf{xi}$ , e e da integral de  $f$  en  $\mathbf{xi}$ . Representa nunha figura (en tres gráficas distintas) os valores  $f(x)$ ,  $f'(x)$  e  $\int f(x)dx$  en tódolos puntos do vector  $\mathbf{xi}$ . Pon títulos ás gráficas.
2. Sexan  $\mathbf{y} = f(\mathbf{xi})$  e  $\mathbf{z} = f'(\mathbf{xi})$  os valores da función e da súa derivada nos puntos do vector  $\mathbf{xi}$ . Converte os vectores  $\mathbf{y}$  e  $\mathbf{z}$  (ambos de lonxitude 49) nas matrices  $\mathbf{a}$  e  $\mathbf{b}$  cadradas de orde 7, por filas. Define a función `operaMatrices(...)`, cos argumentos axeitados, que calcule o vector  $\mathbf{v}$  como os elementos da matriz  $\mathbf{a}$  que verifican  $a_{ij} \leq b_{ij}$  e o vector  $\mathbf{w}$  como os elementos de  $\mathbf{a}$  tal que  $a_{ij} > b_{ij}$ . Calcula un vector  $\mathbf{p}$  da dimensión do vector  $\mathbf{w}$  (denotada por  $m$ ), onde cada elemento  $p_i$ , con  $i = 0, \dots, m - 1$ , sexa o número de elementos do vector  $\mathbf{v}$  que hai que sumar para que a suma sexa maior que  $w_i$  (se chegas ó final do vector  $\mathbf{v}$ , volve a comezar desde o principio).
3. Desde o programa principal chama a función `operaMatrices()` e garda os vectores  $\mathbf{v}$ ,  $\mathbf{w}$  e  $\mathbf{p}$  no arquivo `saida.txt` (un vector en cada liña con dúas cifras decimais).

```
from matplotlib.pyplot import *
from sys import *
import sympy as sy
from numpy import *
# apartado 1-----
x=sy.symbols('x')
expr=input('f(x)= ')
f=sy.lambdify(x, expr)
df=sy.lambdify(x, sy.diff(expr, x))
ifx = sy.lambdify(x, sy.integrate(expr, x))
xi=linspace(0, 2*pi, 49, endpoint=True)
y=f(xi)
z=df(xi)
iz=ifx(xi)
clf();subplots_adjust(hspace=0.4)
subplot(311); plot(xi,y); title('Funcion %s'%expr); grid(True)
subplot(312); plot(xi,z); title('Derivada %s'%sy.diff(expr, x)); grid(True)
subplot(313); plot(xi,iz); title('Integral %s'%sy.integrate(expr, x)); grid(True)
show()
# apartado 2-----
a=y.reshape([7,7])
b=z.reshape([7,7])
def operaMatrices(a, b):
    v=extract(a <= b, a)
    w=extract(a > b, a)
    m=len(w); n=len(v)
    p=zeros(m)
    for i in range(m):
        suma = 0; k=0; nc=0
        while suma < w[i]:
            suma = suma + v[k]
            k = k+1; nc = nc + 1
            if k==n:
                k=0
        p[i] = nc
    return [v, w, p]
```

```

[v, w, p] = operaMatrices(a,b)
# apartado 3-----
def escribeVector(fin, nome, q):
    aux='%s: '%nome
    for i in q:
        aux+='%.2f '%i
    fin.write(aux+'\n')
    return
try:
    f=open('saida.txt', 'w')
    f.write('v: ');f.write(' '.join('%.2f'%i for i in v)+'\n')
    f.write('w: ');f.write(' '.join('%.2f'%i for i in w)+'\n')
    f.write('p: ');f.write(' '.join('%.2f'%i for i in p)+'\n')
    #escribeVector(f, 'v', v)
    #escribeVector(f, 'w', w)
    #escribeVector(f, 'p', p)
    f.close()
except IOError:
    print('Erro escribindo en saida3.txt'); exit()

```