

# Primeiro control de programación en Python

---

Crea co editor un arquivo chamado `exame11.dat` co seguinte contido:

```
3 1 0 8 2 7 9 5
```

Escribe un programa en Python chamado `exame11.py` que:

1. Lea os datos dende o arquivo anterior a un vector  $\mathbf{v}$  e calcule outro vector  $\mathbf{w}$ , da mesma lonxitude  $n$  que  $\mathbf{v}$ , de modo que o elemento  $w_i$ , con  $0 \leq i < n$  sexa, se  $i$  é par, o número de elementos de  $\mathbf{v}$  maiores que  $v_i$ ; se  $i$  é impar,  $w_i$  debe ser o número de elementos de  $\mathbf{v}$  menores ou iguais que  $v_i$ .
2. Cree un vector  $\mathbf{z}$ , tamén de lonxitude  $n$ , con elementos dados por:

$$z_i = \sum_{j < i} \max(v_j, w_j), \quad 0 \leq i < n \quad (1)$$

3. Represente gráficamente os vectores  $\mathbf{w}$  e  $\mathbf{z}$  (dúas gráficas distintas no mesmo cadrado) como cadrados unidos por liñas en cores azul e vermella respectivamente, con grid, etiquetas **índice** e **vectores** para os eixos horizontal e vertical, título de gráfica **vectores w e z**, e unha lenda con etiquetas **vector w** e **vector z** situada arriba á dereita. A figura debe gardarse con formato PNG no arquivo `exame11.png`.

**NOTA:** co arquivo anterior, tes que obter  $\mathbf{w} = [4, 2, 7, 7, 5, 6, 0, 5]$  e  $\mathbf{z} = [0, 4, 6, 13, 21, 26, 33, 42]$ .

---

```
#!/usr/bin/python
#-*- coding: utf-8 -*-
from numpy import *
from pylab import *
v=loadtxt('exame11.dat','int')
n=len(v);w=zeros(n);l=range(n)
for i in l:
    if i%2==0:
        w[i]=sum(v>v[i])
    else:
        w[i]=sum(v<=v[i])
z=zeros(n)
for i in l:
    t=0
    for j in range(i):
        t=t+max(v[j],w[j])
    z[i]=t
    #alternativa usando maximum() de numpy
    #z[i]=sum(maximum(v[:i],w[:i]))
clf()
plot(w,'sb-',label='vector w');plot(z,'sr-',label='vector z')
xlabel(u'índice');ylabel('vectores');title('vectores w e z')
legend(loc='upper right');grid(True);show(False)
savefig('exame11.png')
```

---

# Primeiro control de programación en Python

---

Crea co editor un arquivo de texto chamado `exame12.dat` co seguinte contido:

```
3 1 0 8 2 7 9 5 6 2 4 1 5 8 6 2 9 7 4
```

Escribe un programa en Python chamado `exame12.py` que:

1. Lea os números deste arquivo a un vector `v` con elementos enteiros e almacene noutro vector `x` os valores pares de `v` mentres que a súa suma sexa inferior a 40.
2. Cree un terceiro vector `y`, da mesma lonxitude `n` ca `x`, de modo que o elemento  $y_i$ , con  $0 \leq i < n$ , sexa o número de elementos de `v` inferiores a  $x_i$ .
3. Represente gráficamente os vectores `x` e `y` (dúas gráficas distintas no mesmo cadrado): `x` con cadrados e liña vermella, `y` con círculos e liña azul, con grid e unha lenda con etiquetas **vector x** e **vector y**.

**NOTA:** tes que obter `x=[0, 8, 2, 6, 2, 4, 8, 6, 2]` e `y=[ 0, 15, 3, 11, 3, 7, 15, 11, 3]`.

---

```
#!/usr/bin/python
#-*- coding: utf-8 -*-
from numpy import *
from pylab import *
v=loadtxt('exame12.dat', 'int'); n=len(v); suma=0; x=[]
for i in range(n):
    t=v[i]
    if t%2==0:
        s=suma+t
        if s>100:
            break
    x.append(t); suma=s
m=len(x); y=zeros(m, 'int')
for i in range(m):
    y[i]=sum(v<x[i])
clf()
plot(x, 'sr-', label='vector x')
plot(y, 'ob-', label='vector y')
legend(loc='upper right')
grid(True); show(False)
```

---

# Primeiro control de programación en Python

---

Crea co editor un arquivo de texto `exame13.dat` co seguinte contido:

```
1 8 3 2 5 6
7 2 9 5 1 6
5 4 6 2 7 2
6 3 5 9 1 8
2 8 4 8 5 4
5 2 9 7 3 4
```

Escribe un programa en Python chamado `exame13.py` que:

1. Lea este arquivo e o almacene na matriz **a**, mostrando por pantalla os índices e valores dos elementos de **a** múltiplos de 3, e cantos valores son.
2. Calcule e mostre por pantalla un vector **v** de lonxitude  $n$  (sendo  $n$  a orde da matriz) con elementos  $v_i$  dados por:

$$v_i = \sum_{j < i} \sum_{k < i} a_{jk}, \quad 0 \leq i < n \quad (2)$$

3. Mostre por pantalla os elementos de **v** ata que a súa suma sexa maior ca 10.
4. Represente gráficamente o vector **v** con asteriscos e liñas de cor verde, grid, título **vector v** e etiquetas **índices** e **valores** para os eixos horizontal e vertical respectivamente.

---

```
#!/usr/bin/python
#-*- coding: utf-8 -*-
from numpy import *
from pylab import *
a=loadtxt('exame13.dat','int');n=size(a,1);ind=find(a%3==0)
print 'hai',len(ind),'multiplos de 3: indices=',ind,'valores=',a[a%3==0]
v=zeros(n)
for i in range(n):
    t=range(i);v[i]=sum(a[t,t])
print 'v=',v
print 'suma<20:',
suma=0;i=0
while suma<20:
    print v[i],
    suma+=v[i];i+=1
clf()
plot(v,'*g-')
xlabel(u'indices');ylabel('valores');title('vector v')
grid(True);show(False)
```

---

# Primeiro control de programación en Python

---

Escribe un programa en Python chamado `exame14.py` que:

1. Lea por teclado números e os almacene nun vector  $v$  ata que a súa suma sexa maior que 30 (introduce por teclado a secuencia de números [3,2,7,6,5,9]).
2. Almacene nun vector  $y$  os elementos de  $v$  con valores entre 5 e 10, ambos incluídos, e mostre  $y$  por pantalla.
3. Se  $n$  é a lonxitude de  $y$ , axuste a un polinomio de grao 3 os  $n$  primeiros elementos de  $v$  (coordenadas horizontais dos puntos) e o vector  $y$  (coordenadas verticais).
4. Represente gráficamente os puntos axustados como cadrados azuis e o polinomio (100 puntos) cunha liña vermella, con lendas **puntos** e **polinomio** respectivamente, etiquetas **coord. x** e **coord. y** para os eixos horizontal e vertical respectivamente, e título **axuste polinómico**.

---

```
#!/usr/bin/python
#-*- coding: utf-8 -*-
from numpy import *
from pylab import *
suma=0;v=array([], 'int')
while suma<30:
    n=input('n(<10)? ') # secuencia 3 2 7 6 5 9
    suma+=n;v=append(v,n)
print 'v=',v, 'suma=',suma

y=v[logical_and(v>=5,v<=10)]
print 'y=',y

n=len(y);x=v[range(n)]
p=polyfit(x,y,3)
tx=linspace(min(x),max(x),100);ty=polyval(p,tx)

clf()
plot(x,y,'sb',label='puntos')
plot(tx,ty,'r-',label='polinomio')
xlabel('coord. x');ylabel('coord. y')
title(u'axuste polinómico');legend(loc='upper right')
grid(True);show(False)
```

---

# Primeiro control de programación en Python

---

Escribe un programa en Python chamado `exame15.py` que:

1. Lea por teclado un número  $n$  enteiro (proba con  $n = 5$ ) e cree unha matriz cadrada  $\mathbf{a}$  de orde  $n$  con valores enteiros aleatorios entre 0 e 100 usando o comando `int_(100*random([n,n]))`.
2. Almacene nun vector  $\mathbf{x}$  tódolos elementos da matriz  $\mathbf{a}$  ata o primeiro elemento repetido (que non se debe incluír en  $\mathbf{x}$ ), e mostre  $\mathbf{x}$  por pantalla
3. Mostre por pantalla a suma de tódolos elementos pares de  $\mathbf{a}$ .
4. Convirta a matriz  $\mathbf{a}$  nun vector  $\mathbf{y}$ , e represente gráficamente  $\mathbf{y}$  ordeado por orde crecente, con puntos en forma de asteriscos vermellos, título **vector  $\mathbf{y}$** , grid e exporte a figura ao arquivo `exame15.jpg`.

---

```
#!/usr/bin/python
#-*- coding: utf-8 -*-
from numpy import *
from numpy.random import *
from matplotlib.pyplot import *
n=int(input('n? '))
a=int_(100*rand(n,n))
x=[];fin=False
for i in range(n):
    for j in range(n):
        t=a[i,j]
        if any(t==x):
            fin=True
            break
        x.append(t)
    if fin:
        break
print 'x=',x
print 'suma de elementos pares de a=',sum(a[a%2==0])
y=ravel(a)
clf();plot(sort(y),'*r')
title('vector y');grid(True);show(False)
savefig('exame15.jpg')
```

---