

# Primeiro control de programación en Python

---

Crea co editor un arquivo de texto chamado `datos_exame11.dat` co seguinte contido:

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

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

1. Lea os datos dende o arquivo anterior a un vector  $v$  (supón que non hai erros no proceso de lectura).
2. Ler un número enteiro por teclado e almacénalo na variable  $x$ .
3. Visualice na pantalla o número de veces consecutivas máximo que se repite o número  $x$  no vector  $v$ . Se  $x$  non está contido no vector  $v$  a saída sería 0.

**NOTA:** co arquivo anterior e  $x = 8$  deberías obter 3 xa que hai 3 oitos consecutivos no vector.

---

```
#!/usr/bin/python
#-*- coding: utf-8 -*-
from numpy import *
v=loadtxt('datos_exame11.dat','int')
x=int(raw_input('Introduce x: '))
n=len(v)
veces=0 # contador do numero de veces
i=0
for i in range(n):
    if(v[i]==x):
        aux=0 # contador auxiliar
        k=i
        while(k<n and v[k]==x):
            aux = aux + 1
            k=k+1
        if(aux > veces):
            veces=aux
print 'Veces que ', x, ' se repite en ', v, ' e ', veces
```

---

# Primeiro control de programación en Python

---

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

1. Lea un número enteiro positivo  $n$  por teclado e xere un vector  $\mathbf{v}$  de dimensión  $n$  con números enteiros aleatorios no intervalo  $[5,15]$ .
2. Calcule a diferenza máxima en valor absoluto entre dous puntos consecutivos do vector  $\mathbf{v}$  e entre que números ocorre. Visualice na pantalla os resultados.
3. Se esta diferenza é superior a 5 visualiza na pantalla a mensaxe *Datos dispersos*.

**NOTA:** Nunha execución do programa poderías obter a seguinte información:

```
Introduce n: 8
v= [ 6  5  5 14  9  7 11  6]
Maxima diferencia= 9 entre numeros 14 e 5
datos dispersos
```

---

```
#!/usr/bin/python
#-*- coding: utf-8 -*-
from numpy import *
from numpy.random import *
n=abs(int(raw_input('Introduce n: ')))
v=int_(5+ 10*rand(n))
print "v= ", v
n=len(v)
d=[]
for i in range(1,n):
    d.append(abs(v[i]-v[i-1]))
i=argmax(d)
md=max(d)
print 'Maxima diferencia= ', md,
print ' entre numeros ', v[i+1], ' e ', v[i]
if (md > 5):
    print "datos dispersos"
```

---

# Primeiro control de programación en Python

---

Crea co editor un arquivo de texto chamado `datos_exame13.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 `exame13.py` que:

1. Lea os números deste arquivo a un vector  $\mathbf{v}$  (supón que non hai erros no proceso de lectura).
2. Converte os elementos de  $\mathbf{v}$  nunha matriz cadrada  $\mathbf{a}$  de dimensión  $m$ , onde  $m$  é a dimensión mínima para que colla o vector  $\mathbf{v}$  ( $m$  é o enteiro por exceso de  $\sqrt{n}$ ). O recheamento de  $\mathbf{a}$  farase por filas e ós elementos restantes de  $\mathbf{a}$  asígnaselles o valor 5.
3. Visualice na pantalla a matriz  $\mathbf{a}$ .

**NOTA:** co arquivo anterior deberías obter a matriz:

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

---

```
#!/usr/bin/python
#-*- coding: utf-8 -*-
from numpy import *
v=loadtxt('datos_exame12.dat')
n=len(v)
m=ceil(sqrt(n))
m2=m*m # numero elementos matriz
while(n < m2):
    v=append(v,5)
    n+=1
a=v.reshape([m,m])
print "a= ", a
```

---

Outra alternativa sería:

---

```
#!/usr/bin/python
#-*- coding: utf-8 -*-
from numpy import *
v=loadtxt('datos_exame13.dat')
n=len(v)
m=ceil(sqrt(n))
a=ones([m,m])*5
f=0; c=0 # indices fila e columna
for i in range(n):
    a[f, c]=v[i]
    c=c+1
    if(c==m):
        c=0
        f=f+1
print "a= ", a
```

---

Outra alternativa sería:

---

```
#!/usr/bin/python
#-*- coding: utf-8 -*-
```

---

```
from numpy import *
v=loadtxt('datos_exame12.dat')
n=len(v)
m=int(ceil(sqrt(n)))
a=ones([m,m])*5
k=0 # indices para vector
for i in range(m):
    for j in range(m):
        if k<n:
            a[i,j]=v[k]
            k=k+1
print "a= ", a
```

---

# Primeiro control de programación en Python

---

Crea co editor un arquivo de texto `datos_exame14.dat` co seguinte contido:

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

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

1. Lea este arquivo e o almacene na matriz **a**. Xenéricamente, a matriz **a** será de dimensións  $m \times n$ . Extraer (comezando polo índice (0,0)) a matriz cadrada **b** de maior grao ( $n \times n$ , se  $n < m$  e, noutro caso,  $m \times m$ ). Visualiza a matriz **b**.
2. Calcula o vector **v** da mesma dimensión que **b** (supoñamos  $n$ ), onde cada elemento de **v** ven dado por:

$$v_i = b_{ii} + b_{i(n-i-1)} \quad i = 0, \dots, n-1 \quad (1)$$

3. Visualiza o vector **v** na pantalla.

**NOTA:** co arquivo anterior deberías obter a seguinte matriz **b** e vector **v**:

```
b= [[ 1.  8.  3.]
     [ 7.  2.  9.]
     [ 5.  4.  6.]]
v= [-2.  0.  1.]
```

---

```
#!/usr/bin/python
#-*- coding: utf-8 -*-
from numpy import *
a=loadtxt('datos_exame14.dat')
n,m=a.shape
if(m>n):
    b=a[:n,:n].copy()
elif(m<n):
    b=a[:m,:m].copy()
else:
    b=a.copy()
print "b=", b
n,m=b.shape
v=zeros(n)
for i in range(n):
    v[i]=a[i,i]-a[i,n-i-1]
print 'v= ', v
```

---

# Primeiro control de programación en Python

---

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

1. Lea por teclado dous vectores **v** e **w**.
2. Calcule un vector **z** con todos os elementos contidos en **v** e en **w** (non considerar elementos repetidos). Nota: podes comprobar se un elemento *x* está contido nunha lista **z** coa sentença `x in z`.
3. Visualice na pantalla os tres vectores.

**NOTA:** se `v=[4,3,5,6,5, 4]` e `w=[3,5,0,7,3]` obteríamos un `z=[3,5]`

---

```
#!/usr/bin/python
#-*- coding: utf-8 -*-
from numpy import *
v=input('v=')
w=input('w=')
nv=len(v); nw=len(w)
z=[]
for i in range(nv):
    for j in range(nw):
        if(v[i]==w[j] and (not v[i] in z)):
            z.append(v[i])
print 'v=', v
print 'w=', w
print 'z=', z
```

---

Outra alternativa sería:

---

```
#!/usr/bin/python
#-*- coding: utf-8 -*-
from numpy import *
v=input('v=')
w=input('w=')
nv=len(v); nw=len(w)
z=[]
for i in range(nv):
    if (v[i] in w):
        if(v[i] not in z):
            z.append(v[i])
print 'z=', z
```

---