

Programación—Certamen 3, sábado 18 de junio de 2011

Nombre:

Rol: -

1. [25%] Indique qué es lo que imprimen los siguientes programas.

```
a = arange(3)
b = array([0, 1, 2])
print a == b
```

```
a = array([4, 1, 2, 6])
print 2 * a
```

```
a = ones(5) + arange(5)
b = arange(5) - 1
print any(a + b > 8)
```

```
a = array([[1, 2, 3],
           [4, 5, 6]])
b = array([[4, 5, 6],
           [1, 2, 3]])
print a + 2 * b
```

```
a = array([[1, 0, 1],
           [0, 1, 1],
           [0, 0, 1]])
b = array([2, 3, 4])
print dot(a, b)
```

```
a = arange(6)
b = a.reshape((3, 2))
b[0, 1] = 99
print a
print b
```

Dibuje la ventana que resulta al ejecutar el siguiente programa.

```
from Tkinter import *
v = Tk()
for i in range(4):
    for j in range(4):
        if i == j:
            val = '1'
        else:
            val = '-'
        l = Label(v, text=val)
        l.grid(row=i, column=j)
v.mainloop()
```

Programación—Certamen 3, sábado 18 de junio de 2011

Nombre:

Rol: -

2. [25 %] Una *serie de tiempo* es una secuencia de valores numéricos obtenidos al medir algún fenómeno cada cierto tiempo. Algunos ejemplos de series de tiempo son: el precio del dólar en cada segundo, el nivel medio mensual de concentración de CO₂ en el aire y las temperaturas máximas anuales de una ciudad. En un programa, los valores de una serie de tiempo se pueden guardar en un arreglo.

a) Las *medias móviles con retardo p* de una serie de tiempo son la secuencia de todos los promedios de p valores consecutivos de la serie.

Escriba la función `medias_moviles(serie, p)` que retorne el arreglo de las medias móviles con retardo p de la serie:

```
>>> s = array([5, 2, 2, 8, -4, -1, 2])
>>> medias_moviles(s, 3)
array([ 3,  4,  2,  1, -1])
```

En este ejemplo, las medias móviles son: $\frac{5+2+2}{3}$, $\frac{2+2+8}{3}$, $\frac{2+8-4}{3}$, etc.

b) Las *diferencias finitas* de una serie de tiempo son la secuencia de todas las diferencias entre un valor y el anterior.

Escriba la función `diferencias_finitas(serie)` que retorne el arreglo de las diferencias finitas de la serie:

```
>>> s = array([5, 2, 2, 8, -4, -1, 2])
>>> diferencias_finitas(s)
array([-3,  0,  6, -12,  3,  3])
```

Programación—Certamen 3, sábado 18 de junio de 2011

Nombre:

Rol: -

3. [25 %] Alicia y Benito usan un programa para llevar la cuenta del marcador de sus partidos de tenis de mesa. Cada vez que alguno de ellos gana un punto, hace clic en el botón con su nombre, y el marcador es actualizado.

Un partido de tenis de mesa está dividido en 7 juegos. Cuando alguien gana 11 puntos, el juego se termina y comienza el juego siguiente.



El programa define dos listas de modelos, llamadas `modelos_a` y `modelos_b`, que guardan los puntos ganados en cada juego por el jugador respectivo. Por ejemplo, `modelos_a[3]` guarda los puntos que ha ganado Alicia en el cuarto juego (recuerde que se cuenta desde cero).

Además, hay un modelo que guarda cuál es el juego actual (un número entre 0 y 6):

```
juego_actual = StringVar()  
juego_actual.set('0')
```

Los botones fueron creados de la siguiente manera:

```
b_a = Button(w, text='Alicia', width=7, command=punto_a)  
b_b = Button(w, text='Benito', width=7, command=punto_b)
```

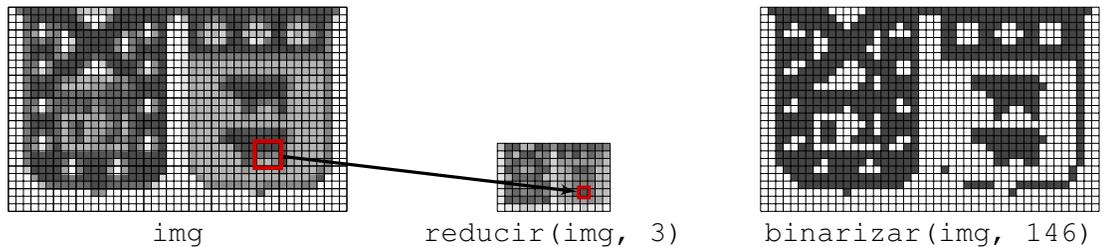
Escriba el código del controlador `punto_a`.

Programación—Certamen 3, sábado 18 de junio de 2011

Nombre:

Rol: -

4. [25 %] Las imágenes en tonos de grises (por ejemplo, una foto en blanco y negro) son matrices de valores entre 0 y 255. Cada elemento de la matriz se denomina *píxel*. El valor 0 representa el negro, el 255 el blanco, y los valores intermedios las distintas intensidades de grises.



- a) Un método para reducir f veces el tamaño de una imagen es partir la imagen en bloques de $f \times f$ y promediar los valores de los píxeles de cada bloque. Este promedio será el valor del píxel correspondiente en la nueva imagen. Si la imagen original es de $m \times n$ píxeles, la imagen reducida quedará de $(m/f) \times (n/f)$ píxeles.
Escriba la función `reducir(img, f)`, donde `img` es una imagen (es decir, una matriz), que retorne una nueva imagen que sea `img` reducida en un factor de f .
- b) La *binarización* de una imagen consiste en crear una nueva imagen cuyos píxeles sólo son negros o blancos. Para esto se escoge un valor llamado *umbral*; todos los píxeles mayores o iguales que el umbral quedan blancos, y los menores al umbral quedan negros.
Escriba la función `binarizar(img, umbral)` que retorne una nueva imagen que sea una versión binarizada de `img` de acuerdo al umbral indicado.