

Programación—Certamen 3, jueves 12 de julio de 2012

Nombre:

Rol: -

1. [30 %] Indique qué es lo que imprimen los siguientes programas.

```
tres = array([1, 4, 2])
seis = 2 * tres
print seis
```

```
a = 'papanatas'
a = a.replace('p', 'n')
a = a.replace('n', 'p')
print a
```

```
a = array([[ 0, 99, 33, 11],
          [55, 33,  0, 33],
          [88, 44, 22, 11]])
b = a[1:, 2:]
print b.sum()
```

```
w = ['Hoy', 'hay', 'pan']
p = '\n'
print p.join(w)
```

```
a = zeros(900).astype(int)
a[300:600:200] += 1
print a.sum()
```

```
a = 'abeja:becerro:doberman'
b = a.split('be')
print b[1]
```

Indique cómo queda el archivo resultado.txt después de ejecutar el siguiente programa.

```
arch = open('palabras.txt')
arch2 = open('resultado.txt', 'w')
s = ''
i = 0
for linea in arch:
    s += linea.strip().split()[0][i]
    i += 1
arch.close()
arch2.write(s)
arch2.close()
```

palabras.txt

```
Antes de rendirse
Apenas tengan la opcion
Dormir no podran
Todo sea porque
Ganabamos con progra
Aprobar el semestre
```

Programación—Certamen 3, jueves 12 de julio de 2012

Nombre:

Rol: -

2. [35 %] Los profesores de la asignatura de Programología están hartos de calcular manualmente las notas de cada certamen. Para apurar un poco más el proceso decidieron implementar un programa que los ayude en esta tarea.

Los certámenes consisten de varias preguntas. Cada pregunta tiene un puntaje máximo posible y una ponderación. La ponderación es el porcentaje de la nota final que aporta cada pregunta. La nota del certamen será un entero ente 0 y 100.

Para cada certamen, se poblará una matriz en que cada columna está asociada a una pregunta y cada fila contiene los puntajes de un alumno. Además, se creará arreglos con las ponderaciones y los puntajes máximos de cada pregunta:

```
ponderaciones = array([30, 35, 35])
totales       = array([ 8, 14, 14])
```

```
puntajes = array([[ 8, 12, 11],
                  [ 8,  9,  9],
                  [ 7, 12,  9],
                  [ 6,  9,  9],
                  [ 8, 14, 14],
                  [ 6,  8, 10],
                  [ 5,  0,  1],
                  [ 6, 14,  9]])
```

- a) Escriba la función `calcular_notas(ponderaciones, totales, puntajes)` que retorne el arreglo con las notas de los estudiantes:

```
>>> calcular_notas(ponderaciones, totales, puntajes)
array([ 88,  75,  79,  68, 100,  68,  21,  80])
```

Para redondear las notas use el método `round`:

```
>>> a = array([54.8, 91.1, 54.2])
>>> a.round()
array([ 55.,  91.,  54.])
```

- b) Cuando hay resultados muy malos en alguna de las preguntas, a veces los profesores deciden «inflar» los puntajes usando un factor f . Si p es el puntaje obtenido en la pregunta por un estudiante y n es el puntaje máximo de la pregunta, entonces su puntaje «inflado» será:

$$f \cdot p + (1 - f) \cdot n.$$

Escriba la función `inflar(puntajes, totales, pregunta, f)` que modifique la matriz de puntajes inflando los puntajes de la pregunta en el factor indicado.

```
>>> inflar(puntajes, totales, 3, 0.8)
>>> puntajes
array([[ 8, 12, 11],
       [ 8,  9, 10],
       [ 7, 12, 10],
       [ 6,  9, 10],
       [ 8, 14, 14],
       [ 6,  8, 10],
       [ 5,  0,  3],
       [ 6, 14, 10]])
```

- c) Para evaluar la dificultad de las preguntas del certamen, los profesores suelen contar cuántos estudiantes obtuvieron puntaje completo en cada pregunta.

Escriba la función `contar_puntajes_completos(ponderaciones, totales, puntajes)` que haga precisamente esto.

```
>>> contar_puntajes_completos(ponderaciones, totales, puntajes)
array([3, 2, 1])
```

Programación—Certamen 3, jueves 12 de julio de 2012

Nombre:

Rol: -

3. [35 %] Una *página web* es simplemente un archivo de texto escrito en un lenguaje llamado HTML. Este lenguaje consiste en etiquetas que van intercaladas en el texto para describir las distintas partes de un documento.

Las *tablas* son uno de los componentes que pueden ser parte de una página web. Por ejemplo, la siguiente es una tabla de tres filas, cada una de las cuales tiene tres celdas:

Manzana	7	\$615
Uva	10	\$399
Kiwi	2	\$471

En un archivo HTML, el inicio de una tabla se marca con la etiqueta `<table>`, y el final con `</table>`. Cada fila comienza con la etiqueta `<tr>` y termina con `</tr>`. El contenido de cada celda va encerrado entre las etiquetas `<td>` y `</td>`.

Nuestra tabla de frutas se escribiría en HTML como se muestra en el ejemplo de la derecha. Si uno abre este archivo en un navegador web, la tabla aparecería tal como se ve más arriba.

Evidentemente, escribir una tabla directamente en HTML resulta muy tedioso. Una estrategia más sencilla es crear la tabla en una planilla de cálculo (como MS Excel) y guardarla en formato CSV (valores separados por comas). Al hacer esto, la tabla queda guardada en un archivo como el mostrado abajo a la derecha.

Aunque no tengan la extensión `.txt`, los archivos HTML y CSV sí son de texto plano, y por lo tanto es sencillo y conveniente escribir programas que lean y escriban en estos formatos.

Archivo `frutas.html`:

```
<table>
<tr>
<td>Manzana</td>
<td>7</td>
<td>$615</td>
</tr>
<tr>
<td>Uva</td>
<td>10</td>
<td>$399</td>
</tr>
<tr>
<td>Kiwi</td>
<td>2</td>
<td>$471</td>
</tr>
</table>
```

Archivo `frutas.csv`:

```
Manzana,7,$615
Uva,10,$399
Kiwi,2,$471
```

- a) Escriba la función `convertir_csv_a_html(tabla)` que convierta un archivo CSV a otro HTML. El parámetro es el nombre del archivo sin la extensión. Por ejemplo, la siguiente llamada crea el archivo `frutas.html` a partir de los datos del archivo `frutas.csv`:

```
>>> convertir_csv_a_html("frutas")
```

La función no debe retornar nada; sólo debe crear el archivo.

- b) Escriba la función `obtener_tamano(archivo_html)` que retorne una tupla (m, n) con el número de filas y el número de columnas que tiene una tabla HTML.

```
>>> obtener_tamano("frutas.html")
(3, 3)
```

Suponga que, al igual que en el archivo de ejemplo, las etiquetas `<table>`, `</table>`, `<tr>` y `</tr>` siempre ocupan una línea completa, y que las etiquetas `<td>` y `</td>` están en la misma línea del valor al que rodean. Además, suponga que todas las filas tienen la misma cantidad de celdas.

- c) Escriba la función `obtener_precio(fruta)` que retorne el precio de la fruta pasada como parámetro:

```
>>> obtener_precio("Kiwi")
471
```

La función debe buscar los datos en el archivo `frutas.csv`. Si la fruta no está en el archivo, la función debe retornar **None**.