

Programación—Certamen 3, sábado 24 de marzo de 2012 (80 mins.)

Nombre:

Rol:

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

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

```
a = arange(3)
b = arange(6)
print (b[3:] + a)[len(a) - 1]
```

```
z = zeros(5) > 1
print ((arange(5) % 3) + z) > 1
```

```
a = arange(5)[-1]
b = ones(5) * a
print all(b >= 4)
```

```
a = array([arange(1,4),
           arange(4,7)])
b = array([arange(7,4,-1),
           arange(2,5)])
print len(a + 1 + b)
```

```
a = arange(6) + ones(6)
b = a.reshape((3, 2))
b[2,1] = 10
print sum(b[2:] * a[3])
```

Indique en el recuadro lo que queda almacenado en el archivo `xyz.txt` luego de ejecutar el siguiente segmento de código:

```
a = open('xyz.txt', 'w')
n = 'ue-colo-uchile-uc'
corte = n.split('-')
p = '{0}: {1}\n'

for i in range(1, len(corte)):
    e = corte[i]
    linea = p.format(i, e)
    a.write(linea)
a.close()
```

Programación—Certamen 3, sábado 24 de marzo de 2012 (80 mins.)

Nombre:

Rol:

2. [35 %]

Los horarios y salas de la asignatura de programación se encuentran en un archivo llamado `horarios.txt` cuyo formato es el siguiente:

`paralelo:nombre_profesor: dia1 bloque1/sala1:dia2 bloque2/sala2:`

```
1:Albert Einstein:Lunes 3-4/P307:Jueves 3-4/M201
2:Justino Vivar:Lunes 1-2/M202:Martes 1-2/M201
3:Ivan Rosa:Lunes 3-4/C203:Martes 5-6/B101
4:Salomon Fuentes:Lunes 1-2/B102:Miercoles 1-2/P307
5:Diego Campusano:Martes 1-2/C201:Jueves 3-4/P114
```

- a) Implemente la función `disponibilidad_de_sala(horario, dia_bloque, sala)` que reciba como parámetro el nombre del archivo con el horario, el día-bloque, y la sala. La función debe retornar **True** si la sala está desocupada en ese día-bloque, de lo contrario debe retornar **False**.

```
>>> disponibilidad_de_sala('horarios.txt', 'Martes 1-2', 'C201')
False
```

- b) Implemente la función `profesores_por_dia(horario)` que reciba como parámetro el nombre del archivo donde se encuentran los horarios y a partir de este generar otro archivo llamado `profesores_por_dia.txt` en el que aparezcan los profesores que hacen clases en cada día-bloque que aparecen en `horarios.txt`. El archivo debe tener el siguiente formato:

```
Martes 5-6: Ivan Rosa/
Jueves 3-4: Albert Einstein/Diego Campusano/
Miercoles 1-2: Salomon Fuentes/
Lunes 1-2: Justino Vivar/Salomon Fuentes/
Lunes 3-4: Albert Einstein/Ivan Rosa/
Martes 1-2: Justino Vivar/Diego Campusano/
```

La función no debe retornar nada, sólo generar el archivo.

No es necesario que el archivo esté ordenado por día.

Programación—Certamen 3, sábado 24 de marzo de 2012 (80 mins.)

Nombre: Rol: -

3. [40%] La meteoróloga Jenny Relámpagos registra los milímetros de lluvia caída durante cada minuto del día. Para preparar sus reportes, al final del día ella carga estos datos en el arreglo `lluvia_por_minuto`, que tiene tamaño $24 \cdot 60 = 1440$.

El registro histórico de lluvia caída *por hora* desde el año 2000 en adelante está almacenado en un arreglo de cuatro dimensiones ($13 \times 12 \times 31 \times 24$) llamado `historico`.

Cuando Jenny quiere saber cuánta lluvia cayó el 14 de mayo de 2007 desde las 22:00 hasta las 22:59, ella tipea en la consola:

```
>>> print historico[7, 4, 13, 22]
```

- Escriba la función `tiempo_lluvia(lluvia_por_minuto)` que retorne la cantidad de minutos del día en que estuvo lloviendo.
- Escriba la función `mayor_diferencia(lluvia_por_minuto)` que retorne cuál fue la mayor diferencia de lluvia entre un minuto y el anterior.
- Escriba la función `obtener_lluvia_por_hora(lluvia_por_minuto)` que, a partir del registro de lluvias por minuto, retorne un arreglo de tamaño 24 con los totales de lluvia por hora.
- Hoy de madrugada, Jenny almacenó todos datos del día en el arreglo `lluvia_por_minuto`. ¿Qué debería tipear Jenny en la consola para almacenar los registros por hora del día de ayer (23 de marzo de 2012) en el arreglo `historico`?
- Escriba la función `llovio(historico, anno, mes, dia)` que retorne **True** si llovió ese día, y **False** si no llovió.