

Programación—Certamen 2, sábado 14 de mayo de 2011

Nombre:

Rol: -

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

```
s = {78, 15, 91, 15}
print len(s)
```

```
d = {78: 15, 91: 15}
print len(d)
```

```
n = (17, 3, 1993)
h = (14, 5, 2011)
print n < h
```

```
x, y = ((27, 3), 9)
z, w = x
print y + w
```

```
a = 'acabase'
b = set(a)
c = list(b)
c.sort()
print c[2]
```

```
t = 'papagayo'
w = t.split('a')
print w[3]
```

```
def f(a, b):
    return a + 2 * b
```

```
a = 5
b = 2
print f(b, a)
```

```
def f(a):
    return x + a
```

```
def g(x):
    return x + a
```

```
x = 5
a = 7
print f(x) + g(x)
```

Programación—Certamen 2, sábado 14 de mayo de 2011

Nombre:

Rol: -

2. [25%] Las temperaturas mínimas y máximas de algunas ciudades de la región están guardadas en un diccionario cuyas llaves son las ciudades y cuyos valores son tuplas (*minima*, *maxima*).

Se desea generar un archivo cuyo contenido sea un reporte como el del ejemplo de más abajo. Los nombres de las ciudades en las que hubo más de 25 grados deben aparecer en mayúsculas. El nombre del archivo debe incluir la fecha. El orden en que aparecen las ciudades dentro del archivo no importa.

Escriba la función `crear_reporte(fecha, temperaturas)`, cuyos parámetros son la fecha (una tupla (año, mes, día)) y el diccionario de temperaturas, y que genere el archivo de texto con el formato del ejemplo.

La función `crear_reporte` no debe retornar nada. Recuerde que `s.upper()` entrega el string `s` en mayúsculas.

```
temp = {
    'Vina del Mar': ( 9, 26),
    'Valparaiso': (10, 24),
    'Quilpue' : ( 7, 30),
    'Olmue': ( 5, 29),
    'Limache': ( 9, 23),
    'Villa Alemana': ( 9, 22),
}
crear_reporte((2011, 5, 14), temp)
```

Archivo `reporte-2011-5-14.txt`:

```
QUILPUE: max 30, min 7
Valparaiso: max 24, min 10
VINA DEL MAR: max 26, min 9
Villa Alemana: max 22, min 9
Limache: max 23, min 9
OLMUE: max 29, min 5
```

Programación—Certamen 2, sábado 14 de mayo de 2011

Nombre:

Rol: -

3. [25%] La red social Fookbace almacena la información de sus usuarios en un diccionario. Las llaves son un código numérico entero asignado a cada usuario, y los valores son tuplas con el nombre, la ciudad y la fecha de nacimiento del usuario. La fecha de nacimiento es una tupla (año, mes, día):

```
usuarios = {
    522514: ('Jean Dupont',      'Marseille', (1989, 11, 21)),
    587125: ('Perico Los Palotes', 'Valparaiso', (1990, 4, 12)),
    189471: ('Jan Kowalski',     'Krakow',    (1994, 4, 22)),
    914210: ('Antonio Nobel',    'Valparaiso', (1983, 7, 1)),
    # ...
}
```

- a) Escriba la función `misma_ciudad(u1, u2)`, que indique si los usuarios con códigos `u1` y `u2` viven en la misma ciudad.

```
>>> misma_ciudad(914210, 587125)
```

```
True
```

```
>>> misma_ciudad(522514, 189471)
```

```
False
```

- b) Escriba la función `diferencia_edad(u1, u2)`, que retorne la diferencia de edad entre los usuarios cuyos códigos son `u1` y `u2`. (Utilice sólo el año de nacimiento de los usuarios para calcular la diferencia, no el mes ni el día).

```
>>> diferencia_edad(914210, 587125)
```

```
7
```

Programación—Certamen 2, sábado 14 de mayo de 2011

Nombre:

Rol: -

4. [25 %] (Este ejercicio es una continuación del anterior).

Para guardar la información sobre cuáles de sus usuarios son amigos entre sí, Fookbase utiliza el conjunto `amistades`, que contiene tuplas con los códigos de dos usuarios. Si la tupla (a, b) está dentro del conjunto, significa que los usuarios con códigos a y b son amigos. En todas las tuplas se cumple que $a < b$.

```
amistades = {  
    (198471, 289142), (138555, 429900), (349123, 781118), # ...  
}
```

- a) Escriba la función `obtener_amigos(u)`, que retorne el conjunto de los códigos de los amigos de u .
- b) Escriba la función `recomendar_amigos(u)`, que retorne el conjunto de los códigos de los usuarios que cumplen todas estas condiciones a la vez:
 - son amigos de un amigo de u ,
 - no son amigos de u ,
 - viven en la misma ciudad que u , y
 - tienen menos de diez años de diferencia con u .

En ambas funciones, el parámetro u es el código de un usuario, y el valor de retorno es un conjunto de códigos de usuarios. Recuerde que `c.add(x)` agrega el valor x al conjunto c .