

Programación—Certamen Rec. (CV), viernes 23 de noviembre de 2012

Nombre:

Rol: -

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

```
c = (True and not (7.0 == 7))
b = False or c
print b
```

```
c = float(13 / 5)
d = 2.5
print c + d
```

```
n = 'Sed ut perspiciatis unde'
b = n.replace('i', 'o', 2)
print b.split()[2][3:9]
```

```
refran = 'Al mal tiempo, buena cara.'
c = refran.split()[::3]
print '\n'.join(c)
```

```
d = {
    0: {1,2,3},
    1: {2,3,4},
    2: {2,3,1}
}
print (d[0] | d[2]) - d[1]
```

```
c1 = {3, 25, 20, 3, 25}
print len(c1)
```

```
n = 483
while n / 10.0 != 0:
    d = n % 10
    n /= 10
print d
```

```
def f3(a):
    return a[1]

a = [0, 1, 2]
b = {0:1, 1:2}
print f3(b)
```

Programación—Certamen Rec. (CV), viernes 23 de noviembre de 2012

Nombre: Rol: -

2. [35 %] La asociación de turismo *Centrochile* mantiene en la lista de tuplas `centros`, los centros de recreación que pertenecen a esta asociación. Cada tupla contiene el nombre del centro, un código interno, el número de la región donde se encuentra y el número de estrellas con que se clasifican sus servicios.

```
centros = [  
    ('Hotel Neruda', 1020, 5, 4), ('Casa Detodos', 1030, 6, 3),  
    ('Centro Lago', 1040, 10, 5), ('Centro San Jose', 1010, 6, 4),  
    ('La Herradura', 1050, 4, 5), ('La Huayca', 1060, 2, 3)]
```

Por ejemplo, el centro *La Herradura* tiene código 1050, se encuentra en la región 4 (cuarta región) y está en la categoría de 5 estrellas. Considere que la lista puede tener muchos centros, no sólo los del ejemplo.

Por otro lado, existe la lista `region` que contiene los nombres de las 15 regiones. El índice de la lista representa el número de la región. Por ejemplo, la primera región (`region[0]`) es Tarapaca.

```
regiones = ['Tarapaca', 'Antofagasta', 'Atacama', 'Coquimbo', 'Valparaiso',  
            'Ohiggins', 'Maule', 'Biobio', 'Araucania', 'Los Lagos', 'Aysen',  
            'Magallanes', 'Metropolitana', 'Los Rios', 'Arica y Parinacota']
```

Además existe la lista `habitaciones` que es un diccionario que asocia el código del centro con la cantidad de habitaciones que posee.

```
habitaciones = {1020:25, 1030:15, 1040:30, 1050:20, 1060:12, 1010:28}
```

- a) Escriba la función `lista_top(centros, estrellas)` que retorne una lista con los centros cuyo número de estrellas sea igual o mayor a `estrellas`.

```
>>> lista_top(centros, 4)  
['Hotel Neruda', 'Centro Lago', 'Centro San Jose', 'La Herradura']
```

- b) Escriba la función `centros_por_region(centros, regiones)` que retorne un diccionario que asocie el nombre de la región y la cantidad de centros disponibles en esa región. Si en la región no hay centros, no la incluya en el diccionario.

```
>>> centros_por_region(centros, regiones)  
{ 'Ohiggins': 2, 'Coquimbo': 1, 'Valparaiso': 1, 'Los Lagos': 1,  
  'Antofagasta': 1}
```

- c) Escriba la función `cantidad_de_habitaciones(centros, habitaciones, regiones, r)` que retorne el número total de habitaciones en los centros de la región `r`.

```
>>> cantidad_de_habitaciones(centros, habitaciones, regiones, 'Valparaiso')  
25
```

Programación—Certamen Rec. (CV), viernes 23 de noviembre de 2012

Nombre: Rol: -

3. [35 %] La red social `birthbook` requiere desarrollar una aplicación para que un usuario pueda enviar un saludo de cumpleaños de manera automática a todos sus amigos que se encuentren de cumpleaños. La información de los amigos se encuentra en el archivo `amigos.txt` con la siguiente estructura: `nombre apellido; sexo; edad; grado de amistad: fecha_cumpleanios`. El sexo puede ser M o F. El grado de amistad va de 1 a 5 (1 es poco amigo y 5 es muy amigo). La fecha del cumpleaños tiene siempre el formato que se muestra en el siguiente ejemplo.

`amigos.txt`

```
Juan Perez;M;5;23 de noviembre del año 1990
Marcela Soto;F;1;21 de octubre del año 1992
Andrea Olivares;F;3;13 de enero del año 2000
Pedro Gonzalez;M;2;23 de noviembre del año 2010
```

Asuma que el archivo tiene muchos registros, no sólo los del ejemplo. Además los usuarios del archivo tienen registrado sólo un nombre y un apellido. Por otro lado, existe un archivo `saludos.txt` que tiene 5 saludos dependiendo del grado de amistad. La estructura es:

`grado de amistad;mensaje`

`saludos.txt`

```
1;Feliz Cumpleanios {0}, suerte!
2;Feliz Cumpleanios {0} que lo pases bien!
3;Feliz Cumpleanios {0} Que lo pases excelente en tu día.
4;Amigo {0}, felicidades en tus {1} años, espero sea un día especial.
5;Amigazo {0}, celebraremos tus {1} años como corresponde.
```

Donde `{0}` es el nombre del amigo y `{1}` es el número de años (presente sólo en mensajes 4 y 5).

- a) Escriba la función `leer_amigos(archivo_amigos)` que retorne una lista de tuplas de cada uno de los datos del archivo `archivo_amigos` con la siguiente estructura:

```
>>> leer_archivo('amigos.txt')
[('Juan Perez/M/5', '23-nov-1990'), ('Marcela Soto/F/1', '21-oct-1992'),
 ('Andrea Olivares/F/3', '13-ene-2000'), ('Pedro Gonzalez/M/2', '23-nov-2010')]
```

- b) Escriba la función `cumpleaneros(archivo_amigos, archivo_saludos, fecha)` que genere un archivo llamado `saludos-FECHA.txt`, donde `FECHA` corresponde a la fecha con el formato `dia_mes` (por ejemplo si `fecha` es `'10 de enero 2012'`, el archivo debería llamarse `saludos-10_ene.txt`). El archivo debe contener el saludo, dependiendo del nivel de amistad, de los amigos que están de cumpleaños en la fecha `fecha`. Note el formato que tiene el parámetro `fecha` en el ejemplo:

```
>>> cumpleaneros('amigos.txt', 'saludos.txt', '23 de noviembre 2012')
```

Esto debería generar un archivo como el siguiente:

`saludos-23_nov.txt`

```
Amigazo Juan Perez, tus 22 años lo celebraremos como corresponde.
Feliz Cumpleanios Pedro Gonzalez que se cumplan todos tus deseos!
```

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