

# Programación—Certamen 3 (CV), jueves 15 de noviembre de 2012

Nombre:

Rol: -

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

```
a = [ 3.4, 4.7, 8.5, 5]
s = map(round, a)
b = str(s)
print len(b)
```

```
a = 'el gato esta en la mesa'
b = a.replace ('g', 'p')
b = b.replace ('m', 'p')
print b
```

```
d = ['patio', 'raton', 'pala', 'coro']
b = ';'.join(d)
c = b[12:16].replace('a', 'i', 1)
print c
```

```
s = 'Pablito clavo un clavito'
print s.split('a')[2]
```

## Preguntas de archivos

1. Considere el archivo `datos.txt`. Indique en el cuadro de más abajo qué es lo que se muestra al ejecutar el siguiente segmento de código:

```
archivo = open('datos.txt')
for linea in archivo:
    l = linea.strip().split(':')
    y, z = l[1].split(' ')
    if y[1:3] == '45':
        print l[0], 'es de ', z
archivo.close()
```

datos.txt

```
Miguel:032-2454529 Valparaiso
Andrea:045-2349580 Araucania
Paula:067-4564532 Aysen
Martin:058-3456673 Arica
```

2. Indique en el cuadro de más abajo cómo queda el archivo `resultado.txt` después de ejecutar el siguiente segmento de código:

```
archivo = open('ramos.txt')
arch = open('resultado.txt', 'w')
p = '{0}: {1}\n'
for linea in archivo:
    j = 0
    l = linea.strip().split('=')
    q = l[1].split(' ')
    t = map(int, q)
    for i in t:
        j = j + i
    s = p.format(l[0], round(j/3.0))
    arch.write(s)
archivo.close()
arch.close()
```

ramos.txt

```
Calculo=10 50 6
Programacion=45 100 35
Fisica=60 60 30
Introduccion=90 80 40
```

## Programación—Certamen 3 (CV), jueves 15 de noviembre de 2012

Nombre:  Rol: -

2. [35 %] El edificio *Progra Paraíso* tiene el registro de cada departamento en la lista `deptos`. Cada elemento de la lista tiene el formato:

número depto:cantidad de habitaciones:tamaño en  $m^2$ :valor en UF

Por ejemplo,

```
>>> deptos
['100:2:35:1200', '101:3:45:1400', '1304:3:42:1800', '1408:3:45:2300']
```

También existe la lista `ventas`, que contiene el registro de los departamentos vendidos. El formato de cada elemento es el siguiente:

número del departamento: rut del comprador: fecha de la venta (día-mes-año)

Por ejemplo,

```
>>> ventas
['101:11.222.333-0:11-11-2012', '1304:12.222.444-K:8-11-2012',
'100:12.774.233-K:21-05-2012']
```

Considere que las listas `ventas` y `deptos` tienen muchos datos. Lo de arriba son sólo ejemplos.

- a) Escriba la función `ingreso_total(deptos, ventas)` que reciba las listas `deptos` y `ventas`, y retorne una tupla con la cantidad de departamentos vendidos y el valor total en UF que se ha recaudado por estas ventas.

```
>>> ingreso_total(deptos, ventas)
(3, 4400)
```

- b) Con el fin de hacer un análisis de los meses en que ha habido mayor venta, escriba la función `vendidos(mes, ventas)` que reciba como parámetro el `mes` (numerado del 1 al 12) y la lista `ventas`. La función debe retornar una lista con los departamentos que se vendieron durante ese mes (independiente del año).

```
>>> vendidos(11, ventas)
['101', '1304']
```

- c) Todos los departamentos deben pagar los gastos básicos del edificio (llamados gastos comunes). Sin embargo, la administración ha decidido fijar en \$50000 el precio a los departamentos NO vendidos y en \$2000 por  $m^2$  a los departamentos ya vendidos. Escriba la función `gastos_comunes(deptos, ventas)` que retorne un diccionario cuya llave sea el número del departamento y el valor, la cantidad a pagar por dicho departamento.

```
>>> gastos_comunes(deptos, ventas)
{'1304': 84000, '100': 70000, '101': 90000, '1408': 50000}
```

## Programación—Certamen 3 (CV), jueves 15 de noviembre de 2012

Nombre:  Rol: -

3. [35 %] Un banco dispone de 2 archivos para el control de las cuentas corrientes de sus clientes: `clientes.txt` y `movimientos.txt`. En ambos archivos los campos se encuentran separados por coma y tienen la siguiente estructura de campos:

El archivo `clientes.txt` tiene el formato: `Rut, Nombre, Estado(Activo/Inactivo)`

```
10043499-7, Juan Perez, Activo
7655511-7, Esteban Perez, Inactivo
```

El archivo `movimientos.txt` tiene el formato:

`Rut, Movimiento(G=Giro, D=Deposito), Anio, mes, dia, monto`

```
10043499-7, D, 2012/8/28, 500000
10043499-7, G, 2012/8/28, 300000
10043499-7, D, 2012/8/29, 20000
```

Considere que ambos archivos pueden tener muchos datos. Lo anterior es sólo un ejemplo.

- a) Escriba la función `agregar_movimiento(rut, movimiento, fecha, monto)` que permita agregar un nuevo movimiento al archivo `movimientos.txt` con el formato que corresponde. La función debe recibir como parámetro el `rut`, el tipo de movimiento, la fecha en formato (año, mes, día) y el monto de la transacción. La función no debe retornar nada

```
>>> agregar_movimiento('10043499-7', 'D', (2012, 8, 28), 50000)
```

- b) Escriba la función `giros_depositos(rut, mm, aa)` que retorne una tupla con la cantidad total de giros y depósitos a partir del `rut` del cliente.

```
>>> giros_depositos('10043499-7', 8, 2012)
(300000, 520000)
```

- c) Escriba la función `generar_saldos()`: que a partir del archivo de movimientos genere el archivo `saldos.txt` con el formato `rut, saldo`

```
10043499-7, 220000
```

Los campos del archivo deben estar separados por coma y el saldo corresponde a la suma de todos los depósitos menos los giros de cada cliente **Activo**. La función no debe retornar nada, simplemente realizar la operación anteriormente descrita.