

# Programación—C.Recuperativo (CC y CSJ) - Jueves 17 de Julio de 2014

Nombre:

Rol: -

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

```
a = 101
b = ''
while a > 0:
    d = a % 10
    a = a / 10
    b += str(d)
print int(b)
```

```
d = {1:'0',0:'1','0':1,'1':0}
for k in d.items():
    if k[1] == '1':
        d[k[1]] = 4
        d[k[0]] = 0
        d[k[0]+1] = 2
print d.values()
```

```
def f1(x,y):
    x.append(4)
    y.append(4)
    return x,y
```

```
x, y = [1,2,3], [4,5,6]
s1 = set(f1(x,y)[0])
print s1 & set(f1(x,y)[1])
```

```
x = ['IWI-131,inf,utfsm',
      'ADI-90,inf,uch',
      'ILI-101,intro,utfsm']
```

```
f = []
for k in x:
    i = k.split(',')
    if i[-1] == 'utfsm':
        f.append(i[0])
r = 'o'.join(f)
print r
```

## Pregunta de archivo

1. Considere el archivo grupo\_b.dat. Indique lo que imprime y cómo queda el archivo ganadores.dat al ejecutar el siguiente segmento de código:

```
d = []
f1 = open('grupo_b.dat')
f2 = open('ganadores.dat','w')
for linea in f1:
    p, pt = linea.strip().split(';')
    d.append((int(pt), p))
f1.close()
g = 2
for j in range(len(d)):
    p, pt = d[j]
    if p == max(d)[0] and bool(g):
        f2.write(str(pt) + '\n')
        print 'Pasa', pt
        g-=1
        d[j] = 0,0
f2.close()
```

grupo\_b.dat

```
Espana;10
Holanda;8
Chile;9
Australia;8
```

ganadores.dat

## Programación—C.Recuperativo (CC y CSJ) - Jueves 17 de Julio de 2014

Nombre:

Rol: -

2. [35 %] El profesor Pine es un reconocido científico del mundo Pukeman y tiene en su haber una cantidad considerable de Pukemanes. Como buen científico, el profesor Pine es muy ordenado y tiene a los Pukemanes almacenados en un diccionario, con sus respectivas características, de la siguiente manera:

```
Pukemanes = {  
# nombre: (puntos de salud, ataque, defensa, at. especial, def.  
    especial, velocidad, tipo),  
'Bulbasaur':(45, 49, 49, 65, 65, 45, 'Grass'),  
'Charmander':(39, 52, 43, 60, 50, 65, 'Fire'),  
'Pikachu':(35, 55, 40, 50, 50, 90, 'Electric'),  
'Jiglypuff': (115, 45, 20, 45, 25, 20, 'Normal'),  
# ...  
}
```

El profesor Pine recibe la visita de 3 amigos: Nash, Fisty y Block. El profesor les da a elegir un Pukeman a cada uno. Dado que cada uno tiene una estrategia distinta para elegir, y que la cantidad de Pukemanes es muy grande, Ud. tiene que ayudarles escribiendo una función que retorne al (los) Pukeman(es) de acuerdo a sus demandas.

- a) Nash prefiere a los Pukemanes equilibrados, por lo que debe escribir una función que retorne al Pukeman cuya varianza de las características numéricas sea la menor. La varianza la puede calcular de la siguiente manera:  $var(x) = (\frac{1}{N} \sum x^2) - (\bar{x})^2$ , donde  $\bar{x}$  es el promedio.

```
>>> mejor_Nash(Pukemanes):  
'Charmander'
```

- b) Fisty tiene una predilección por los Pukemanes de Césped (tipo Grass) y por los que la suma de su ataque especial y defensa especial sea la mayor.

```
>>> mejor_Fisty(Pukemanes)  
'Bulbasaur'
```

- c) Block es más difícil de satisfacer. El no cree mucho en las estadísticas por lo que prefiera elegir a su Pukeman de acuerdo a sus instintos. Pero claro, como todos, tiene sus preferencias. El prefiere a los Pukemanes de tipo 'Normal' y 'Electric', y además que sus puntos de salud sea mayor o igual a cierto valor (valor dado como parámetro). Ayúdele a Block filtrando a los Pukemanes, generando otro diccionario donde los Pukemanes agregados cumplan con esas características.

```
>>> filtro_Block(Pukemanes, 35)  
{'Pikachu': (35, 55, 40, 50, 50, 90, 'Electric'),  
'Jiglypuff': (115, 45, 20, 45, 25, 20, 'Normal')}
```

## Programación—C.Recuperativo (CC y CSJ) - Jueves 17 de Julio de 2014

Nombre:

Rol: -

3. [40%] La empresa de videojuegos Xmart maneja los juegos en venta y preventa en dos archivos diferentes con la siguiente estructura: `titulo;empresa`, como por ejemplo los siguientes:

venta.txt

```
starcraft ii;blizzard
the last of us;naughty dog
```

preventa.txt

```
uncharted 4;naughty dog
the witcher 3;cd projekt RED
sims 4;EA
```

Los archivos anteriores son ejemplos, pueden contener más juegos y no necesariamente se llaman `venta.txt` y `preventa.txt`.

- a) Desarrolle la función `agregar_venta(nombre_archivo, datos)`, que reciba como parámetro una variable de tipo `string` con el nombre del archivo de ventas y los datos del juego como una lista. Debe agregar al final del archivo el juego ingresado, retornando **True**. En caso de que el nombre del juego ya se encuentre en el archivo, no lo guarda y simplemente retorna **False**.

```
>>> agregar_venta('venta.txt', ['
    assasins creed iv', 'ubisoft'])
True
```

venta.txt

```
starcraft ii;blizzard
the last of us;naughty dog
assasins creed iv;ubisoft
```

- b) Desarrolle la función `preventa_a_venta(archivo_venta, archivo_preventa, titulo)` que reciba como parámetro tres variables de tipo `string`, una con el nombre del archivo con las ventas, otra con las preventas y la última con el título de un juego. La función crea un nuevo archivo de preventas (a cuyo nombre original se le antepone la palabra `nueva`), eliminando el juego y agregándolo en el archivo `venta`, retornando **True**. En caso de no existir en `preventa`, no lo guarda y simplemente retorna **False**.

```
>>> preventa_a_venta('venta.txt', 'preventa.txt', 'sims 4')
True
```

venta.txt

```
starcraft ii;blizzard
the last of us;naughty dog
assasins creed iv;ubisoft
sims 4;EA
```

nuevapreventa.txt

```
uncharted 4;naughty dog
the witcher 3;cd projekt RED
```

- c) Desarrolle la función `buscar_juegos(archivo_venta, archivo_preventa, empresa)` que recibe el nombre del archivo de ventas, de preventas y el nombre de una empresa de juegos. La función retorna una lista de tuplas, donde la tupla posee la siguiente estructura (nombre del juego, archivo donde se encuentra). Guíese por el ejemplo.

```
>>> buscar_juegos('venta.txt', 'preventa.txt', 'naughty dog')
[('the last of us', 'venta'), ('uncharted 4', 'preventa')]
```