

Introducción a la Computación Convocatoria de Enero – Curso 2018/2019

Normas

- No se permite ningún tipo de material de referencia (libros, apuntes, etc.) ni aparatos electrónicos (ordenador, móvil, calculadora, etc.)

1. (1 punto) Ejercicio 1.

Indicar lo que muestran por pantalla los siguientes códigos. Si hubiese algún error, modificar el código para corregirlo.

- (a)

```
cadena="hola"+6
cadena[5]=8
print(cadena)
```
- (b)

```
cadena="numero: 2 4 7 9"
em=0
for z in cadena.split():
    if z%2==0:
        em+=int(z)
print(em)
```
- (c)

```
def func(x):
    return x-1
x=4
print(func(x)*func(3*x))
print(func(a)*func(b))
```

2. (1.5 puntos) Ejercicio 2.

- (a) Indicar lo que muestra por pantalla el siguiente código. Si hubiese algún error, modificar el código para corregirlo.

```
def func(nombre):
    fp2=open(nombre[:len(fich)-4]+"2.txt","w")
    while linea:
        datos=linea.split()
        me=0
        for i in range(1,len(datos)):
            me+=int(datos[i])
        me/=len(datos)-1
        fp2.write(datos[0], " ",me,"\n")
    fp2.close()
func("alumnos.txt")
```

Supongamos que el fichero de texto "alumnos.txt" contiene cadenas del tipo "nombre nota1 nota2 nota3", por ejemplo "Oscar 5 7 4 9"

- (b) ¿Cuál sería la diferencia si se sustituye la línea `fp2=open(fich[:len(fich)-4]+"2.txt","w")` por `fp2=open(fich[:len(fich)-4]+"2.txt","a")`?

3. (3.5 puntos) Ejercicio 3.

Desarrollar funciones en código Python que permitan realizar las siguientes acciones:

- (a) Los argumentos de la función serán una lista de nombres y una lista de números de teléfono, por ejemplo `lista_nombres=["Ander", "Julen", "Ane"]` y `lista_telefonos=["654237892", "623418741", "619773322"]` y tendrá que mostrar por pantalla:

```
Ander 654237892
Julen 623418741
Ane 619773322
```

- (b) Los argumentos de la función serán los nombres de dos ficheros de texto. El primer fichero en cada línea tiene un nombre y el segundo fichero en cada línea tiene un número de teléfono. Tendrá que mostrar por pantalla lo mismo que la función anterior.
- (c) Los argumentos de la función serán una lista de nombres y una lista de números de teléfono como en el apartado a) pero el resultado será un diccionario en el que las claves son los nombres y el valor asociado a cada nombre será el teléfono.

```
D={"Ander": "654237892", "Julen": "623418741", "Ane": "619773322"}
```

- (d) El argumento de entrada será un fichero de texto en el que en cada línea aparece un nombre y un número de teléfono separados por espacio en blanco. En el fichero puede haber varias líneas que tengan el mismo nombre y un número de teléfono distinto. El resultado será un diccionario en el que las claves son los nombres y los valores asociados serán listas con todos los teléfonos asociados a cada nombre.

```
D={"Ander": ["654237892", "678456712"], "Julen": ["623418741"], "Ane": ["619773322", "624678912"]}
```

4. (2 puntos) Ejercicio 4.

- (a) Indicar lo que muestra por pantalla el siguiente código.

```
def func(lista):
    dicc={}
    for elem in lista:
        a=elem[0]
        b=elem[1]
        dicc[a]=dicc.get(a,0)+b
    return dicc

print(func([("a",3), ("b", 4), ("c", 0), ("a",2)]))
```

- (b) Dada la función anterior y la que se muestra a continuación:

```
def sacar_valores(cadena):
    lista=[]
    car=[]
    for c in cadena:
        if c not in car:
            val=cadena.count(c)
            lista.append((c,val))
            car.append(c)
    return lista
```

Escribir un programa principal que guarde en un diccionario información sobre el número de apariciones de los diferentes caracteres que hay en un fichero de texto.

5. (2 puntos) Ejercicio 5.

Diseñar una función que tomando como argumento de entrada una lista de listas M , identifique si se trata de una matriz y en ese caso devolverá dos listas correspondientes a:

- 1) la fila cuyos elementos suman el máximo
- 2) la columna cuyos elementos suman el máximo

Si no se trata de una matriz, devolverá dos listas vacías.

Por ejemplo:

$M_1 = [[2, 5, 3], [6, 1, 8], [9, 3, 2], [7, 5, 4]]$ devolverá: $L_1 = [7, 5, 4]$ y $L_2 = [2, 6, 9, 7]$

$M_2 = [[4, 7, 6], [2, 3], [1, 9, 5]]$ devolverá: $L_1 = []$ y $L_2 = []$