



Seminario de Lenguajes
Opción Python
Práctica 2

Estructuras de control

- 1.- a) Dadas las instrucciones **if**, **elif** y **else** vistas en teoría programe los ejemplos ahí vistos e indique qué realizan (haga varias corridas modificando los valores en las variables).
- b) Escriba el código necesario para que dada una variable con un valor entero, si este es más chico que 4 imprima "DESAPROBADO"; si el valor está entre 4 y 10 imprima "APROBADO"; y ante cualquier otro valor imprima "El valor ingresado no corresponde a una nota".
- c) Modifique b) para que el valor de la nota sea ingresado por la entrada estándar
- 2.- a) Explique la semántica de los bucles **while** y **for..in** . Transcriba los ejemplos vistos en teoría en un módulo Python y ejecútelos.
- b) Explique para qué se utiliza la cláusula **else** dentro de los bucles.
- 3.- Modifique el ejercicio 11.- c) de la Práctica 1 para que se lean los alumnos y las notas de teclado, informándolas hasta que se encuentre el número -1, y al finalizar deberá imprimir *"Fin de la carga de notas"*.
- 4.- a) Describir qué hacen las siguientes sentencias:
- ```
for i in range(1, 10):
 print (i)
else:
 print ('Termino el bucle')
```
- b) Modificar las sentencias de a) para que informe dentro del rango del 2 al 30 (inclusive) qué números son pares y cuáles impares.

c) Ahora imprima la suma de los primeros N números impares, siendo N un entero ingresado por la entrada estándar. Resuélvalo utilizando tanto **for..in** como **while**.

d) Modifique c) para que además del número N, se obtenga a partir de qué número se quiere empezar a buscar los impares, es decir si  $N = 10$  y  $base = 5$  buscaremos los 10 primeros números impares a partir del 5 y retornaremos la suma de ellos. Elija el bucle que crea más conveniente.

Nota: Controle que  $N \geq 0$ . El programa no debe continuar mientras N no cumple esa condición (utilizar bucle).

e) Modifique d) para que si en algún momento determinado la suma de los números impares da mayor que 50 corte la ejecución del bucle. Nuevamente resuélvalo utilizando **while** y **for..in**.

5.- a) Dada una lista con N números, modifique dicha lista multiplicando por 2 cada uno de los elementos. Es decir, si tengo [2, 3, 4, 5, 6] el resultado sería [4, 6, 8, 10, 12].

b) Modifique el ejercicio anterior para que a los números impares los eleve al cuadrado y a los pares al cubo.

c) Modifique el código anterior para que los números sean cargados por la entrada estándar hasta recibir el -100.

6.- a) Sea un diccionario como el visto en 4.- b), escriba un script que recorra la estructura y para aquellas personas que tienen más de 18 años cree una tupla con su nombre y la frase "**Mayor de edad**" y para aquellos que no lo sean su nombre y "**Menor de edad**", y por último su edad y la ubique en la misma posición del diccionario.

Por ejemplo, se espera que se generen tuplas con el siguiente formato:

('Alberto', 'Mayor de edad', 40)

b. Ahora recorra nuevamente la estructura y copie los menores de edad en una lista y los elimine del diccionario original

### Funciones

7.- a) Implemente las funciones vistas de operaciones sobre conjuntos (unión, intersección y diferencia) de manera tal que cada una reciba los dos conjuntos y devuelva el resultante.

conjunto1 = set([1,2,3,4,5,6])

conjunto2 = set([3,4,5,10,15])

b) Desarrolle mediante funciones y listas el funcionamiento de las colas FIFO (el primero en entrar es el primero en salir) con sus operaciones **pop**, **push** y **length** recibiendo la lista como parámetro. En la función **push** si no se envía como parámetro ningún elemento por defecto el valor será 0 (cero).

c) Implemente una función que reciba una lista variable de números y un diccionario, y para cada elemento de la lista imprima si el mismo es una clave del diccionario.

### Módulos

8.- Escriba cada uno de los incisos del ejercicio 7.- (a, b y c) en módulos diferentes (***conjuntos.py***, ***cola.py***, ***parametros\_variables.py***), impórtelos en un nuevo módulo (***principal.py***) y utilice sus funciones en éste último.

### Funciones lambda, map y filter

9.- Resuelva los incisos del ejercicio 5 con la combinación de estas funciones.