

Práctica 1: Estructuras de Datos y Algoritmos II UPE 2015

1. Escriba tres funciones que reciban por parámetro dos números enteros a y b e impriman todos los números enteros comprendidos entre a ; b (inclusive), uno por cada línea en la salida estándar. Para ello, dentro de una nueva función escriba una función por cada uno de los siguientes incisos:

- Que realice lo pedido con un `for`.
- Que realice lo pedido con un `while`.
- Que realice lo pedido sin utilizar estructuras de control iterativas (`for`, `while`, `do while`).

Tener en cuenta que no necesariamente $a < b$.

Por último, escriba una función `main` llamando a cada uno de las funciones creadas, con valores de ejemplo. En una computadora, ejecute el programa y verifique que se cumple con lo pedido.

2. Escriba una función que dado un número n devuelva una nueva lista de tamaño n con los n primeros múltiplos enteros de n mayores o iguales que 1.

Ejemplo: $f(5) = [5; 10; 15; 20; 25]$; $f(k) = \{nk/k : 1..k\}$

Agregue al programa la posibilidad de probar con distintos valores de n ingresándolos por teclado, mediante el uso de `raw_input()`.

3.¿Qué imprime el siguiente programa al ejecutarlo?

- Intente averiguarlo sin ejecutar el programa en su computadora.
- Ejecute el ejercicio en su computadora, y compare su resultado con lo esperado en el inciso anterior.

```
def swap1(x,y):
```

```
    if x < y:
        tmp = x
        x = y
        y = tmp
```

```
def swap2(lista):
```

```
    if lista[0]< lista[1]:
        newList = [0,0]
        lista = newList
```

```
a = 1
```

```
b = 2
```

```
l = [a,b]
```

```
swap1(a,b)
```

```
swap2(l)
```

```
print a
print b
print l
```

4. Dado un lista de valores tipo int se desea calcular el valor máximo, mínimo, y promedio en una única función. Escriba dos funciones, donde respectivamente:

- a. Devuelva lo pedido por el mecanismo de retorno ("return").
- b. Devuelva lo pedido interactuando con algún parámetro.

5. Realice a través de funciones el comportamiento de una pila, las operaciones necesarias son:

pila_sacar(lista) : saca un elemento al tope de la pila y lo retorna
pila_poner(lista,elemento): pone un elemento en la pila
pila_esVacia(lista): devuelve true si es vacía, false caso contrario
pila_tope(lista): devuelve el elemento que está al tope de la pila

6. Realice las mismas operaciones pero para el comportamiento de una cola.

7. Realice una función que dada una lista recibida como parámetro y un número entero, inserte este mismo en ella y la devuelva ordenada.

8. Realice la clase Persona, que tenga como atributos nombre, apellido, teléfono, dni, edad con sus setters y getters.

9. Realice un script que construya una lista de personas e imprima sus datos en pantalla.

10. Realice un script que con los mismos datos del punto anterior construya una lista de personas y las ordene según su dni.

Nota: tenga en cuenta el método `__cmp__` y la funcion `cmp()`

11. Realice la clase Celular que tenga como atributos numero, marca, compañía, con sus setters y getters y haga lo mismo que en punto 9 pero con Celulares, y el punto 10 pero que los ordene por número.

12. Modifique la clase Persona, para que ahora tenga un atributo celular, luego construya una lista de Personas con objetos celulares y relacionelos a través del setter.