



NOMBRE Y APELLIDOS:

1. [15%] Escribe una función `productos` que tenga como argumento una lista de números L y devuelva la lista M , donde $M[i]$ es el producto de todos los elementos de L excepto $L[i]$

```
>>> productos([1,2,3])
[6, 3, 2]
>>> productos([1,0,3])
[0, 3, 0]
>>> productos([2,3,5,1])
[15, 10, 6, 30]
```

2. [10%] ¿Qué se muestra por pantalla después de ejecutar estos dos programas?

```
##PROGRAMA 1
j = -1
for i in range(1,4,2):
    while j < 2:
        print('{}, {}'.format(i, j))
        j +=1
```

SOLUCION

```
##PROGRAMA 2
j = -2
for i in range(1,4,1):
    while j <=2:
        print('{1}, {0}'.format(i+1, j))
        j +=2
```

SOLUCION

3. [5 %] ¿Qué resultará de ejecutar las siguientes sentencias?

```
>>> sol = []
>>> sub = [1,2]
>>> sol.append(sub)
>>> sub.append(2)
>>> sol.append(sub)
>>> sub.remove(sub[-1])
>>> sol
```

<----- Solucion

4. [5 %] ¿Qué resultará de ejecutar las siguientes sentencias?

```
>>> A = [[0]*3]
>>> B = A*2
>>> B[0][1]=10
>>> B
```

<----- Solucion

5. [30 %] **Búsqueda binaria.** Supongamos que tenemos que buscar el número de teléfono de una persona con la ayuda de una guía telefónica. Abrimos la guía por la mitad y mirando el primer nombre de la página, inmediatamente sabemos en qué mitad de la guía está el número. Podemos hacer el mismo procedimiento con esa parte y, así sucesivamente. Con solamente comprobar dos nombres, hemos eliminado $\frac{3}{4}$ de los números de la guía.

Esta técnica se denomina búsqueda binaria, porque en cada etapa divide la lista (=la guía de teléfonos) en dos partes iguales: los valores que están antes y los que están después del valor que hemos comprobado.

EL PROBLEMA DE LA BÚSQUEDA BINARIA

Entrada: Un número a y una sucesión de n números ordenados $L = [a_1, a_2, \dots, a_n]$.

Salida: El primer índice i tal que $a_i = a$ o -1 si a no es un elemento de L .

- Dada una entrada como (22, [1, 1, 2, 22, 22]), un algoritmo de búsqueda debería retornar 4 .
- Dada una entrada como (7, [1, 1, 2, 22, 22]), un algoritmo de búsqueda debería retornar -1 .

Un primer intento de programar la estrategia de arriba es:

```
1 def busqueda_binaria_recursivo(a, lista):
2     if len(lista)==0:
3         return -1
4     else:
5         mitad=len(lista)//2
6         if lista[mitad]==a:
7             return mitad+1
8         else:
9             if a<lista[mitad]:
10                return busqueda_binaria_recursivo(a,lista[:mitad])
11                return busqueda_binaria_recursivo(a,lista[mitad+1:])
```

busqueda_binaria_recursivo(3, [1,2,3,4]) ----->

busqueda_binaria_recursivo(3, [1,2,3,3,4]) ----->

busqueda_binaria_recursivo(3, [1,1,3,3,4]) ----->

busqueda_binaria_recursivo(1, [1,1,2,3]). ----->

busqueda_binaria_recursivo(1, [1,1,2,2]) ----->

Pero el programa no es correcto, como se ha comprobado con algunos de los ejemplos de arriba. El siguiente programa es una modificación del anterior, intentando implementar la estrategia. Completar los puntos suspensivos

```
1 def b_binaria_rec(a, L):
2
3     mitad = len(L) // 2
4
5     if len(L) == 1 and ..... :
6
7         return mitad+1
8
9     elif ..... and L[mitad] != a:
10
11         return -1
12
13     elif L[mitad] == a and L[mitad-1] != a:
14
15         return .....
16
17     else:
18         if L[mitad] < a:
19
20             if b_binaria_rec(a, L[mitad:]) != -1:
21
22                 return mitad + .....
23
24             else:
25
26                 return .....
27
28         else:
29
30             return b_binaria_rec(a,L[:mitad])
31
```

6. [10%] ¿Qué se muestra por pantalla tras ejecutar el siguiente programa?

```
B, C, D = 0, 1, 4
for A in [ 0, 1, 3, 5, 7,0,2,9 ]:
    if A :
        if A%2 != C+ A:
            B -= 1
            continue
        else:
            C *= 2
        if A%3 == 0:
            D //= 3
            break
if not B:
    C =10

print(A) ----->

print(B) ----->

print(C) ----->

print(D) ----->
```

7. [10 %] Marcar o responder a las siguientes cuestiones:

- SQLite es una librería de C que permite acceder a bases de datos con Python usando una variante no estándar del lenguaje de consulta SQL.
- El tiempo de ciclo de un computador cuyo reloj es de 500 MHz es:
- La memoria virtual permite ejecutar programas desde disco, sin necesidad de cargar el código máquina en la memoria principal.
- El siguiente fragmento de código provocará un error al ejecutar el programa: `if (False and False): print(5 / 0)`.
- Dado el número M en binario: 100101011011010. ¿Cuál es la representación binaria de $4M$ y $2M + 1$?

4M ----->

2M + 1 ----->

8. [15 %] Escribe un programa para calcular en qué cuadrante está un punto del plano bidimensional, leído por teclado. El programa debe decir si el punto está en el primer, segundo, tercer, cuarto cuadrante o si está en el eje X , eje Y o si es el origen de coordenadas. NO se puede utilizar la sentencia `elif` ni tampoco, la sentencia `else`.