



# Listas, tuplas, for, ejercicios y malas noticias

MATERIAL DE AYUDANTÍA EN [HTTPS://GIT.IO/FAXRZ](https://git.io/faxrz)

MATERIAL DE UN AMIGO EN [HTTPS://GIT.IO/VDID3](https://git.io/vdid3)


HAGAN ESTOS EJERCICIOS [HTTP://PROGRA.USM.CL/APUNTE/EJERCICIOS/](http://progra.usm.cl/apunte/ejercicios/)



¿Como les fue en el certamen?





- 
- ▶ Tengo buenas noticias [TE]
  - ▶ Tengo malas noticias [L]
  - ▶ Tengo buenas noticias [FUTURO]
  - ▶ <https://git.io/fAxrZ>

# Problema de las dos listas desordenadas

Se necesita que implemente una función **ORDEN(A,B)** que tome como parámetros dos listas cualesquiera y retorne una tercera lista ordenada con los elementos de la primera y segunda lista

```
A = [4,6,2,4,6,-1,2,3,-4]
```

```
B = [-500,-40,123,3,2,0]
```

```
ORDEN(A,B)
```

```
>>> [-500, -40, -1, -1, 0, 2, 2, 2, 3, 4, 4, 6, 6, 123]
```



# Asesino del numero 3

Un asesino serial ha recopilado el numero de casa de muchas posibles victimas, pero el asesino sigue un patrón el cual consiste en que solo mata a las personas que su numero de casa contiene el **numero 3**. El asesino es muy metódico y le encanta la programación así que ha guardado todos los números de casa en una lista como la siguiente [num1,num2,num3,...,numN]. Se le amenaza de muerte a que haga un programa que tome una lista con números de casa y muestre todas las casas que no sobrevivirán al asesino. Pero el asesino le indica que también odia los **whiles** y que debe implementarlo con un **for**

```
A =[123,4235,56,78]  
>>> [123,4235]
```



# Mal conjunto implementado :'(

Un mal programador estaba intentando implementar un conjuntos con listas, pero no se preocupo que cuando le insertaban un valor a la lista este no se repitiese, por ende podían existir listas de la forma [1,2,3,4,5] y se podía agregar los números nuevamente dando como resultado [1,1,2,2,3,3...]

Haga una función que tome una lista, **elimine los repetidos** y retorne una lista nueva sin elementos repetidos

```
A = [1,2,3,4,4,5]
>>> = [1,2,3,4,5]
A = [5,5,5,5,5,5]
>>> = [5]
```





```
lista = [5,5,5,6,6,6,7,8,8]
def delrepetidos(a):
    b = []
    for k in a:
        if b.count(k)<1:
            b.append(k)
    return b
print delrepetidos(lista)
```



# Problema de la distancia máxima

Se tendrá un conjunto de puntos representados por tuplas dentro de una lista, y se necesita conocer los dos puntos mas lejanos uno del otro y mostrar su distancia

```
def distancia(p1,p2): #Asuman esta function o haganla ustedes mismos
    return ((p1[0]-p2[0])**2+(p1[1]-p2[1])**2)**0.5
```

```
a = [(-1,3), (5,7), (-13,5), (-3,-1), (3,-7), (6,2), (1,3), (4,5), (-3,-3)]
#Aqui va el código de ustedes...
#La distancia maxima es (-13, 5) (3, -7) y es de 20.0
```

Mas casos de prueba


```
a = [(-1,3),(5,7),(-13,5),(-3,-1)]
```

```
#La distancia maxima es (5, 7) (-13, 5) y es de 18.1107702763
```

```
a = [(-3,-4),(-1,-1),(-0,-0),(-2,7),(-7,23),(15,-15)]
```

```
#La distancia maxima es (-7, 23) (15, -15) y es de 43.9089968002
```





```
a = [(-3,-4),(-1,-1),(-0,-0),(-2,7),(-7,23),(15,-15)]  
def distancia(p1,p2):  
    return ((p1[0]-p2[0])**2+(p1[1]-p2[1])**2)**0.5
```

```
max = 0;  
puntos = [0,0]  
for x in a:  
    for y in a:  
        #print "Distancia entre",x,y,"igual a",distancia(x,y)  
        if max<distancia(x,y):  
            max = distancia(x,y)  
            puntos[0] = x  
            puntos[1] = y
```

```
print "La distancia maxima es",puntos[0],puntos[1],"y es de",max
```



# Revolver cartas

- ▶ Imagine que tiene **dos conjuntos de cartas**
- ▶ `C1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 'J', 'Q', 'K']`
- ▶ `C2 = ['K', 'Q', 'J', 10, 9, 8, 7, 6, 5, 4, 3, 2, 1]`
- ▶ Debe **intercalarlas** una por una quedando como lo que viene **abajo**
- ▶ `[1, 'K', 2, 'Q', 3, 'J', 4, 10, 5, 9, 6, 8, 7, 7, 8, 6, 9, 5, 10, 4, 'J', 3, 'Q', 2, 'K', 1]`

**Cosa útil:** `range(len(C1))`  
**Adivinen que hace c:<**