

PARTE 1:

Dentro de la función main, creamos la lista que se evaluará

```
1 import pdb
2 from turtle import pd
3
4 pdb.set_trace()
5
6 def main():
7     # PARTE 1
8     # Definiendo la lista
9     lista = [[2, 4, 1, 7], [1, 2, 56, 7, 11, 8], [102, 224, 43, 121]]
10
```

En la nueva lista que definiremos como "arr_max", utilizaremos la función max, dentro de cada lista contenida en esta.

```
10 lista = [[2, 4, 1, 7], [1, 2, 56, 7, 11, 8], [102, 224, 43, 121]]
11
12 # Utilizando la función max definimos el valor máximo de cada lista contenida
13 # en la lista inicial
14 arr_max = [max(elemento) for elemento in lista]
```

Ahora utilizando Pdb, definimos un breakpoint en la línea 13, donde se define la lista comprimida, y uno en la línea 16, donde se realiza la impresión del resultado.

Al dirigirnos a la línea 16 con el comando next, nos damos cuenta que ya cuenta con el resultado de todas las listas dentro de la variable "arr_max".

```
--Return--
> c:\users\ruben\desktop\maestria\buena practicas programacion\clase_4\main.py(13)<listcomp>()->[7, 56, 224]
-> arr_max = [max(elemento) for elemento in lista]
(Pdb) break 16
Breakpoint 2 at c:\users\ruben\desktop\maestria\buena practicas programacion\clase_4\main.py:16
(Pdb) next
> c:\users\ruben\desktop\maestria\buena practicas programacion\clase_4\main.py(16)main()
-> print(arr_max)
(Pdb) p arr_max
[7, 56, 224]
(Pdb)
```

PARTE 2:

Definimos la función es_primo, el cual evalúa a todos los números anteriores al ingresado hasta la mitad del valor del número (caso par), o el mayor valor cercano a la mitad (caso impar), si la división entre el número original con uno de estos números tiene residuo 0, no será primo

```
def es_primo(n):  
    if n % 2 == 0:  
        num_f = n/2  
    else:  
        num_f = int(n/2) + 1  
  
    num_f = int(num_f)  
  
    res = True  
    for numero in range(num_f):  
        numero_1 = numero + 1  
        if numero_1 > 1:  
            if n % numero_1 == 0:  
                res = False  
    return res
```

Usando filter, obtenemos todos los resultados que nos dan como verdadero.

```
# PARTE 2  
lista_1 = [3, 4, 8, 5, 5, 22, 13]  
arr_primo = list(filter(es_primo, lista_1))  
  
print(arr_primo)
```