**Práctica 2- Python (Multiprocessing)**

En el presente documento se adjuntan NN ejercicios para poner en práctica la programación en multiproceso o paralela.

Recordad que para la entrega, los ejercicios irán dentro de una carpeta. Hacemos un zip con la siguiente estructura: **[P2-PSP]-AntonioCalabuigPuigvert.zip**. No olvidéis realizar seguimiento con un control de versiones para toda la práctica.

**Ejercicio 1:**

Basándonos en los ejemplos que tenemos disponibles en Aules sobre multiprocessing, realizad el siguiente programa.

Data esta lista de elementos: mylist = [2,4,9,16,25,36,49,64,81,100]

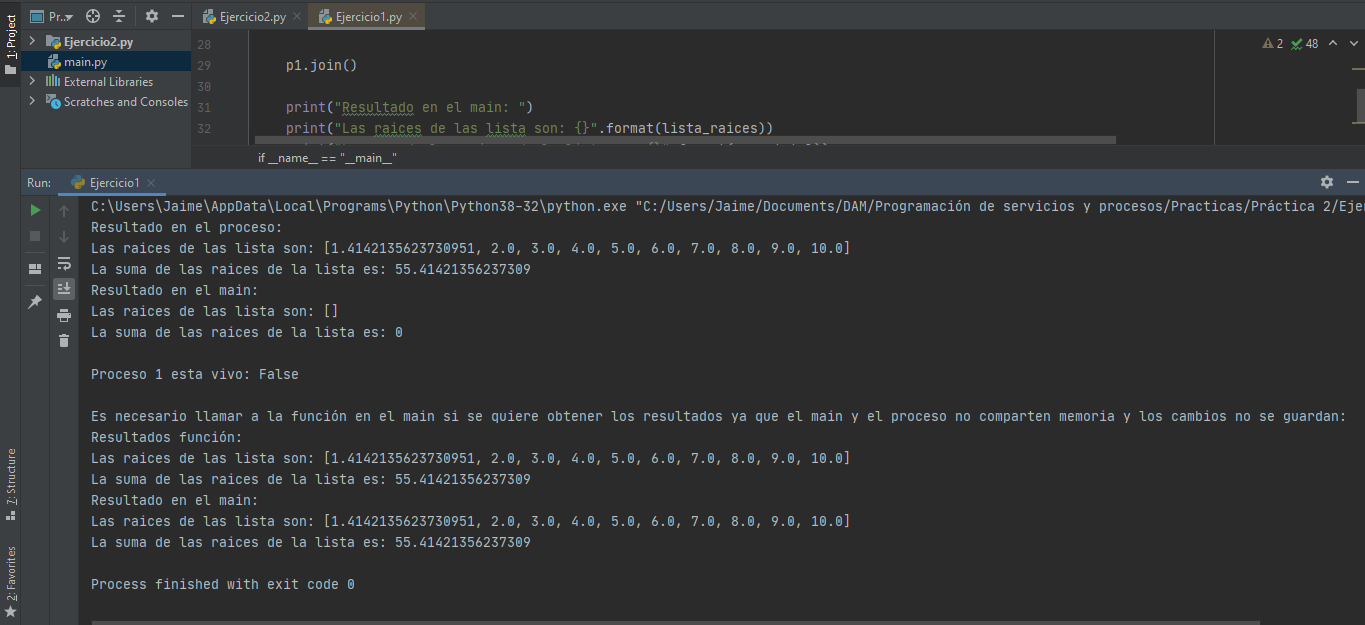
Queremos que nuestro programa obtenga la lista de todas las raíces cuadradas de cada elemento y la suma total de los valores de la lista de raíces cuadradas.

En este caso la implementación tiene que ser sin memoria compartida y usando solo un proceso a parte del proceso de ejecución principal. Es decir, el del main y uno nuevo que crearemos. Al final de la ejecución ambos procesos tiene que mostrar el resultado de su lista de raíces cuadradas y la suma.

Comprobar si el proceso secundario está vivo y mostrarlo por pantalla.

Debido a que el proceso creado y el proceso de ejecución principal no comparten memoria los cambios que se originan el proceso secundario no se muestran en las variables del proceso principal. Como muestra la consola en el proceso secundario el array se ha rellenado y se ha hecho el sumatorio, sin embargo, en el proceso principal no. Si se quiere rellenar la lista con los datos es necesario que el main llame a la función por su cuenta.

**Escribe la respuesta obtenida en consola:**

****

**Ejercicio 2:**

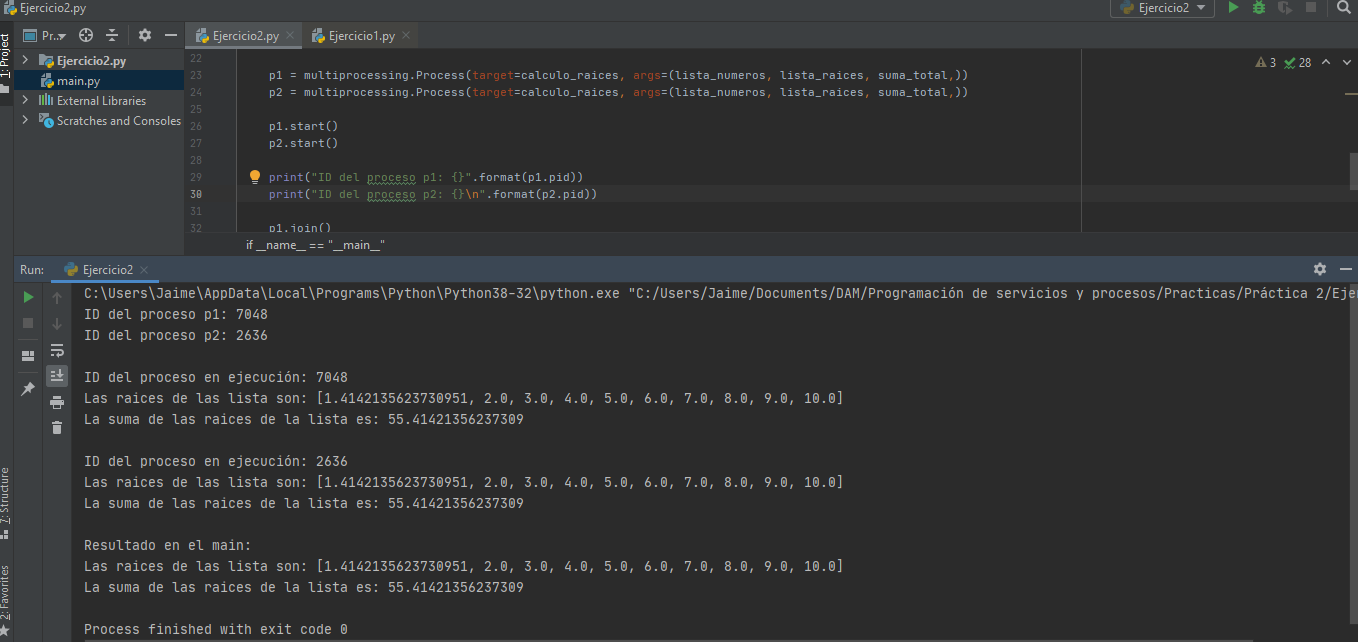
Repetimos el mismo ejercicio que en el caso anterior pero ahora lo que queremos es trabajar con memoria compartida. Aparte del proceso principal main, ahora queremos crear dos procesos y que ellos trabajen los dos sobre la misma función que calcula las raíces cuadradas de los elementos de la lista.

Mostrar para cada proceso nuevo creado, su pid y no esperar a hacer el join para ver el orden en el que se ejecutan.

Mostrar también el resultado del valor de la lista y la suma de sus elementos para cada uno de los procesos ejecutados.

En este caso las variables sí que se encuentran en una memoria compartida entre los dos procesos adicionales y el proceso principal. Como muestra la consola los cambios en las variables se observan tanto en los procesos adicionales como en el principal. La consola también muestra como el primer proceso en ejecutarse es el 1 y, hasta que no termina, el proceso 2 no entra en la función.

**Escribe la respuesta obtenida en consola:**

****