All questions will be done with a runnable code

1. What is the difference between multiprocessing and multithreading?
2. Write a simple program of multiprocessing.
3. How does a particular python code utilize shared memory using the multiprocessing module to create a process that modifies shared data and what values are printed for the shared variables “number” and “array” after the process is executed?
4. How does the Python code utilize the “multiprocessing.Pool” to execute the square function in parallel for the provided list of numbers, and what is the output of the result variable printed at the end?
5. Explain how the Python code utilizes a shared value and a lock from the multiprocessing module to ensure synchronized access to the shared_value. Additionally, what is the final value of the shared_value printed after all processes have completed their tasks?
6. Write down a sample code which computes the difference between the speed of numpy and jax numpy ?
7. Explain how the code utilizes JAX NumPy to define and apply a custom function custom_function to the input array x, and what is the result of the custom function for the provided values in the array x? where,

   x = jnp.array([0.0, jnp.pi / 2, jnp.pi])

   and return the following operation :

   jnp.sin(x) + jnp.cos(x)

   Explain the code briefly.
8. There is an array: [1.0, -1.0, 0.0, and 1.5]. Compute this array for all values in such a way if the value is greater than 0 then it will return **sin(x)** or else will display **cos(x)**.
9. Explain the utilization of “multiprocessing.Pipe” for inter-process communication, where one process (sender) sends a series of messages through the pipe, and another process (receiver) retrieves and prints these messages. How the communication is managed between the parent and child processes, and how are the messages exchanged and processed using the pipe? Explain with a suitable code.
10. Create one numpy array (a, b) and one jax_numpy array (c, d) of 100X100 size. Perform dot(.) product between the numpy array (a, b), and jax_numpy araay (c, d). Compute the time between these two arrays.