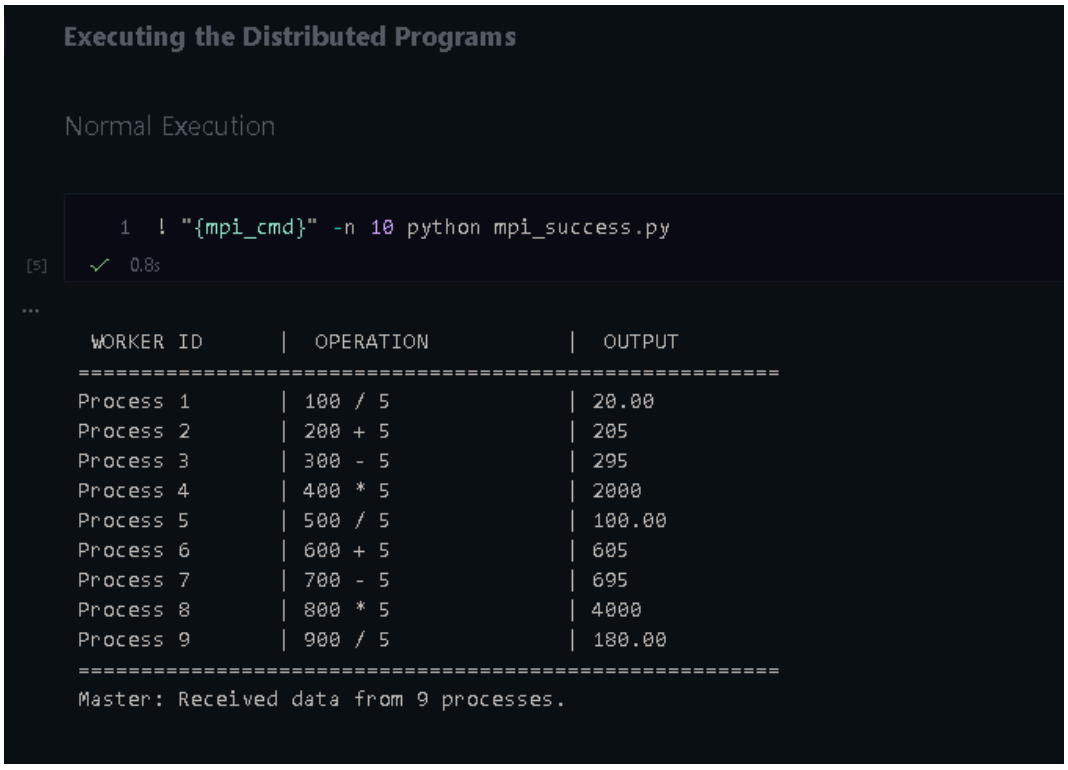
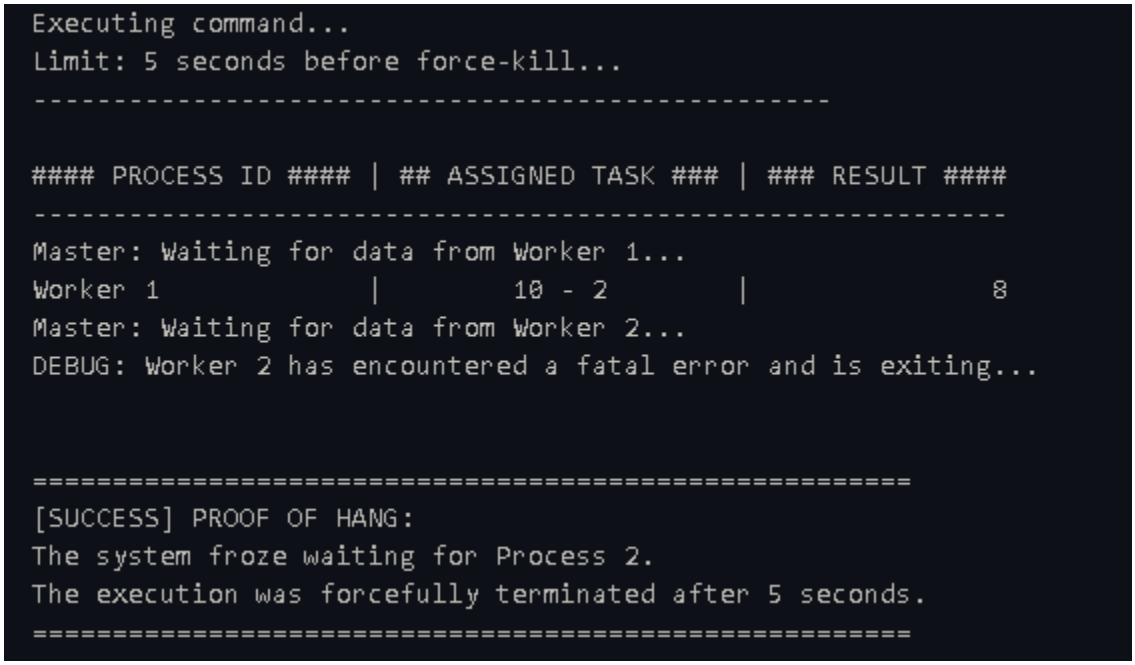
France Raphael R. Rivera

Screenshots:

Success



Fail



PART III – DISCUSSION & REFLECTION

1. **Why is message passing required in distributed systems?**

- In a distributed system, every process operates in its own isolated memory space, meaning one processor cannot look at or change the variables of another. Message passing is required because it is the only way to move data between these separate spaces. Without it, the processes would be completely independent and could not work together to solve a single problem.

2. **What happens if one process fails?**

- If a process crashes before it sends its assigned data, the system will likely freeze. This happens because the receiving process uses a "blocking" command that pauses everything until a message arrives. Since the crashed process never sends the message, the receiver waits forever, causing the entire application to hang.

3. **How does this model differ from shared-memory programming?**

- In shared-memory programming, all threads can read and write to the same variables instantly, which is faster but can lead to errors if two threads change data at the same time. The message passing model is different because processes cannot see each other's memory. They must strictly use send and receive commands to communicate, which is safer for data but requires more lines of code to manage.