Name: Wilmar G. Lipata
Section: J3A

Parallel and Distributed Computing **Assignment2**

TESTING OUTPUTS:

```
%%writefile asgmnt2_pdc_lipatawilmar.py
from mpi4py import MPI

# Initialize MPI environment
comm = MPI.COMM_WORLD
rank = comm.Get_rank()
size = comm.Get_size()

if rank == 0:
    # MASTER PROCESS
    print(f"Master process (Rank {rank}) started. Total processes: {size}\n")

    for i in range(1, size):
        # Master receives data from each worker
        result = comm.recv(source=i)
        print(f"Received from Worker {result['rank']}:")
        print(f"  - Task assigned: {result['task']}")
        print(f"  - Calculated Sum: {result['sum']}")
        print("-" * 30)
else:
    # WORKER PROCESSES
    # 1. Assign a unique task name
    task_name = f"Data Chunk {rank}"

    # 2. Perform a simple computation (summing a range of numbers)
    # Each worker gets a different range based on its rank
    start_num = rank * 10
    end_num = start_num + 10
    total_sum = sum(range(start_num, end_num))

    # 3. Pack data into a message and send it to the Master (dest=0)
    message = {
        "rank": rank,
        "task": task_name,
        "sum": total_sum
    }
    comm.send(message, dest=0)
```

```
Overwriting asgmnt2_pdc_lipatawilmar.py
```

```
# Install OpenMPI if not already installed (common for Linux environments like Colab)
!apt-get update -qq > /dev/null
!apt-get install -y openmpi-bin > /dev/null

# Install mpi4py Python library
!pip install mpi4py

# Run the MPI script with 4 processes using mpirun, allowing root execution and oversubscribing
!mpirun --allow-run-as-root --oversubscribe -n 4 python asgmnt2_pdc_lipatawilmar.py
```

```
Received from Worker 1:
  - Task assigned: Data Chunk 1
  - Calculated Sum: 145
------------------------------
Received from Worker 2:
  - Task assigned: Data Chunk 2
  - Calculated Sum: 245
------------------------------
Received from Worker 3:
  - Task assigned: Data Chunk 3
  - Calculated Sum: 345
------------------------------
```

SHORT REFLECTION:

Working on this assignment provided a clear look into the complexities of distributed computing. The most significant challenge was navigating the environment setup. I initially encountered several "Access is denied" and "command not found" errors because I was attempting to use Windows file paths within a Linux-based Google Colab environment. Resolving this required a deeper understanding of how to install the OpenMPI engine and use specific execution flags to allow the processes to run correctly in a cloud-based container.

On the programming side, moving beyond the basic starter code was an insightful process. I had to structure the logic so that each worker performed a unique calculation on its own data chunk before packaging that information into a dictionary. Seeing the master process successfully receive and display those individual results was a great way to visualize how message passing actually coordinates isolated processes. This project really highlighted why synchronization is so important, especially when you are managing multiple workers that are all trying to report back to a single master.