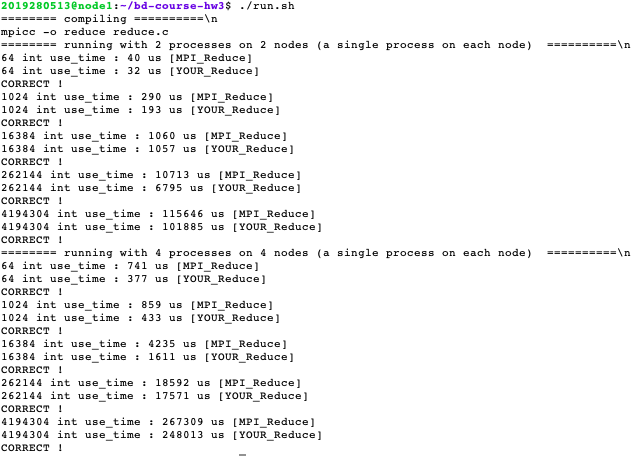
BIG DATA COURSEWORK 3

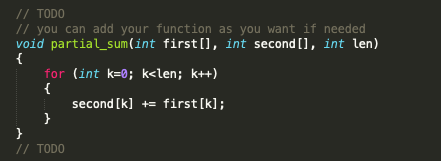
MPI PROGRAMMING



To reduce the time complexity of the program, parallel execution of sub-arrays is being used by parallel programming MPI to calculate their partial sums and then finally, the master (root) which in our case is Array 0 calculates the sum of these partial sums and returns the total sum of the array.

In order to use partial sum in our code first we had to initialize variables and create a function that handles the partial summation for us.





In my MPI implementation. I have used MPI\_Send and MPI\_Recv functions. I also used 4 rank numbers which one of them was used for partial summation I tried to implement the code in such a way that to make sure that the IDs matched between the sender and the receiver, used status to make sure that the receiver successfully received the package the reason I have used size is >2 in rank 0 is that to make sure it was the last step to calculate the summation from other ranks and gather it in our root process. The for loop at the end of the screenshot also made sure that the master process can add its own array so that the program doesn’t crash. Maybe there was a more efficient way to develop a better program by sending b from one process to another by multiplying by its size but I couldn’t manage to optimize it any further but still my MPI program worked and performed slightly better. So, I am satisfied with the result.

