FAST- National university of Computer and Emerging Sciences

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Parallel and Distributed Computing**

**MPI Programming - Matrix Multiplication**

**Project Report**

**Supervised:**

**Dr. Hassan Jamil Syed**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Group Members:**

Mahad Khalid Tarar , 18K-0187  
Abdullah Raheel , 18K-0170  
M.Ammar bin Nasir , 18K-1037

**Introduction:**

The main purpose of this project is to perform various operations on multiprocessing, and record the time taken for computation with serial computation on the same operations. For this work, we choose matrix multiplication as it requires NxN operations to compute if N is the matrix size.

**Results:**



**Working:**

In the code, first we randomly fill the arrays, then we break the array a evenly according to number of processes, that is row = arraySize/numberOfProcess. We also set an offset variable which will tell from which index the values of matrix a will be sent. We then send offset, rows, array a and the entire array b among all the processes. Then we compute the multiplication and store it in array c. We then again perform the send function, which will send back the resultant matrix C to get the final result.

**Conclusion:**

In the beginning serial execution was quicker than mpi but as the size of matrix increases the performance of serial execution decreases while mpi performance increases. Hence for greater data and computation multiprocessing becomes faster than serial computation.