**Parallel Processing: Assignment 4**

**Algorithm:**

For this assignment, we are given an N x N matrix D0 containing 0s in the diagonal. Each process can only handle their own sub-matrix D with dimensions , where np is the number of processes, from the original matrix D0. They cannot handle the data that is contained in the sub-matrix of other processes. With that in mind, every process will eventually need information that is handled by other processes, thus requiring communication. Each process only communicates with the processes that are in the same range of rows or columns as itself, therefore individual processes do not need to send their information to *np – 1* processes (only to the ones that require their data).

To achieve this, every process partitions the original matrix D0 into the sub-matrix D that they will be handling, so that they do not go through other process’ data. After every process know its local data (their sub-matrix), we apply the communication pattern. For every iteration of the third loop with j-index, we compute *pidComm1* and *pidComm2*, which corresponds to the processes that the current process will receive data from in this iteration. If the pids that are obtained do not equal the current process id, then we request data from both these processes. On the other hand, if *pidComm1* and *pidComm2* both equal the current process id, it is time for us to send the data from our sub-matrix to all the other process that need it instead. Therefore, on each iteration the current process communicates with either two processes, or all the other processes that need its data.

After all the communication and computation is done, every process will have their updated D sub-matrix, which it is lastly used to update the original D0 matrix.