Problem Statement

1. Implement the serial version of Quick and merge sort for sorting arrays of the following sizes taken as input. , 2^{11}, 2^{13}, 2^{15}, 2^{17}
    a. Generate one array of each of the above-given sizes and save it into a file.
    b. Whenever a program is executed, it takes command line parameters to identify the input and saves the sorted file into the corresponding output file.
    c. Analyze the performance of both algorithms in terms of GFlops for each of the sizes in the form of a plot.

2. Implement the parallel version of quick and merge sort using MPI basic and collective communication functions. The program should divide the input array into small parts. Each processor should compute a sorting operation on a portion of its assigned work and reply to the master processor using the appropriate collective MPI function.
    a. The size of the arrays will be. 2^{11}, 2^{13}, 2^{15}, 2^{17}
    b. Each leaf process should be allocated an array of the following sizes.
        i. $2^{\text{lastdigitofyourRollNo+1}}$
        ii. $2^{\text{2ndlastdigitofyourRollNo+1}}$
        iii. $2^{\text{3rdlastdigitofyourRollNo+1}}$

Note: In case you have 2 or more same digits in your roll number then add 1 to make them different.

    c. Parallel implementation must use a cluster setup of 2 machines for dividing a task among parallel machines.
    d. Analyze the performance of both algorithms in terms of GFlops for each of the sizes in the form of a plot, considering
        i. 2 machines in the cluster
        ii. 1 machine in the cluster