In this offline, you have to implement three sorting algorithms based on Divide and Conquer and evaluate their performance against your implemented insertion sort and other library-provided sorting algorithms for different inputs. Overall, you have to implement the following four sorting algorithms.

1. Merge Sort
2. Quicksort (taking the last element as the *pivot*)
3. Randomized Quicksort (taking a random element as *pivot*)
4. Insertion Sort

For different values of `n` from the set `[5,10,100,1000,10000,100000]`

a) Generate an array of `n` random numbers in random order. Then use your implemented four sorting algorithms to sort these `n` numbers. You have to use a fixed seed to generate the random numbers so that the results can be reproduced. You can use `srand(some constant number)` to generate your random numbers.

b) Use your sorted array of `n` numbers as input to run your implemented Quicksort and Randomized Quicksort algorithms.

c) Use the `sort()` function provided by C++ Standard Template Library (STL) to sort the same array of `n` randomly ordered numbers from (a).
[Yes! You have to write your code in C++ for this assignment].

d) Repeat each step for 20 times and report the average execution time required for each of the sorting algorithms you ran. You have to submit a formatted .csv file with your submission. A sample formatted .csv file is provided in this link. The output file should look like this if opened with an excel viewer.

Submission Guideline:

Create a folder and name it as your Student ID. Put all the source codes inside this folder along with the excel file. Make a zip file of this folder and submit it on moodle. Submission deadline is **January 29 (Saturday) at 10.55AM**.