Project 1 Sorting Problem

This is a **group** project. Each group should have 3-4 members. If you have difficulty in finding a group, please let me know as soon as possible.

For this project, you should develop both C and Python programs to solve the sorting problems. In you submission, you will have two separate folders:

- C
- Python

The project is due on **Oct 30 at 23:55**. (Note: Start working on this project earlier; do not wait until the last week.)

Objectives

- Practice using the algorithms we have learned to solve the sorting problem.
- Compare the running time of different algorithms.
- Learn to write C programs which will be widely used in upper-level CS courses.
- Compare the performance of C and Python in solving the same problem.

Procedure

For both programming languages, you should complete the following:

1. Write a **function** number_generator(int size) to generate a group of random numbers and write the numbers into a file.
   a. The numbers been generated could be integers or **floating**-point numbers.
   b. The function accepts one parameter which specifies the count of numbers.
   c. The generated numbers should be stored in a file named “XXX.txt”, where XXX is the count of numbers. (E.g., 10.txt, 100.txt, 1000.txt, etc.)
2. Implement at least three algorithms (**MergeSort**, **Insertion Sort**, **QuickSort, or others**) to sort the numbers you generated in the first step.
   a. Input for each algorithm is a list of numbers.
   b. Output for each algorithm is a sorted list with the elements in the original list.
3. Write a **driver program**, which invokes the function for generating a group of numbers and uses the three sorting algorithms to sort the list of numbers.
   a. **Start** from 10, call function number_generator to generate a file named “10.txt”.
   b. **Increase** the input size for number_generator by multiplying it with 10 each time until 100,000. After this step, you will have 5 original files: 10.txt, 100.txt, 1000.txt and 10000.txt.
   c. For each sorting algorithm, read the numbers from a file into a list (it is your own choice to use an array-based list or a linked list). Call the sorting algorithm to sort the list, and **print** out the **number** of items in the list need to be sorted and the

time cost for sorting the list. A suggested format for the output of the program is as following:

```
QuickSort

Input size (N):  (# of numbers)           Time cost:

10                                        X seconds/milliseconds

100                                       X seconds/milliseconds

1,000                                     X seconds/milliseconds

10,000                                    X seconds/milliseconds

100,000                                   X seconds/milliseconds
```

d. Write the sorted list items into a file named “XXX_YYY_sorted.txt” (YYY is the name of the algorithm you used to sort the list). E.g., 1000_quicksort.txt is the file that stores the sorted list of numbers read from 1000.txt and use quicksort for sorting.
e. Prompt a question asking if the user would like to delete all the generated files or not.

Submission:

Place all the following files in a single folder named Project1, tar and compress it as Project1.tar.gz.

1. Well-documented source code files and executable files in two separate folders:
    a. Folder C: C source files, executable files
    b. Folder Python: Python source files
2. A PDF file report.pdf:
    a. Explaining how to run you programs
    b. Pseudocode for the three sorting algorithms, and which data structures are used in your program.
    c. Running time for different input sizes with different algorithms. Remember to provide the results for C version and Python version. Compare the performance of C and Python in solving the same problem.
    d. Responsibility for each group member.
    e. Problems or questions you have towards this project. Did you solve the problems before submission? If yes, how did you solve the problems? If no, what’s your plan for solving these problems?
    f. What you have learned from this project? Any comments or other things you would like to tell the instructor?