**# Algorithm to Generate Random Numbers, Perform Bubble Sort, and Calculate Swaps:**

*B.Tech. 3rd Semester Assignment*

**DATA STRUCTURES**

**1.1** Generate n Pseudo-random numbers (+ve and -ve) between a range [a:b] using c rand() function and store them in a file. Implement Bubble sort. Take the randomly generated inputs in previous step to run the Bubble sort. In Bubble sort, count the number of times swaps happen. Now take 100, 500, 1000, 10000, 100000, 500000 and more numbers of random numbers as input to Bubble sort and report each time the number of swaps happening. Now take 10 sets of input for each of 100, 500, 1000, 10000, 100000, 500000 randomly generated numbers and run bubble sort on it. For each of 100, 500, 1000, 10000, 100000, 500000 numbers, report the average number of swaps.

Start

Include the necessary header files.

Declare the functions used in program.

Initialize variables and data structures.

Seeding the random number generator with the current time.

User input for the number 'n' (the size of the set).

Check if files with the same set size and run number already exist in the folder and if found, delete the existing files for the each of the 10 runs.

**Step-1:**

**Step-2:**

**Step-3:**

**Step-4:**

**Step-5:**

**Step-6:**

**Step-7:**

**Step-8:**

Random Number Generation and Bubble Sorting:

**Step-8.1:**  Create a new folder to store output files.

**Step-8.2:** Generate an array of size ‘n’ to store random numbers within the specified range.

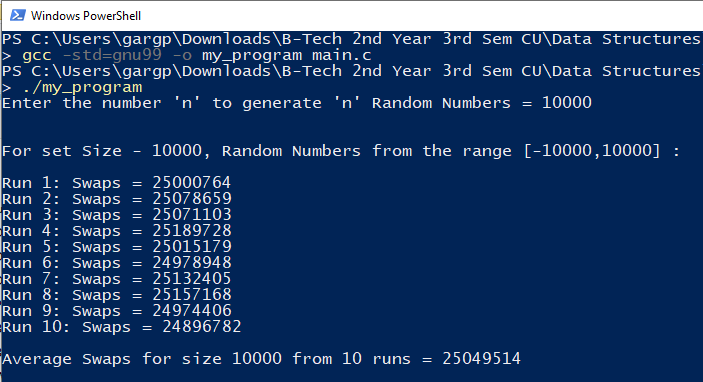
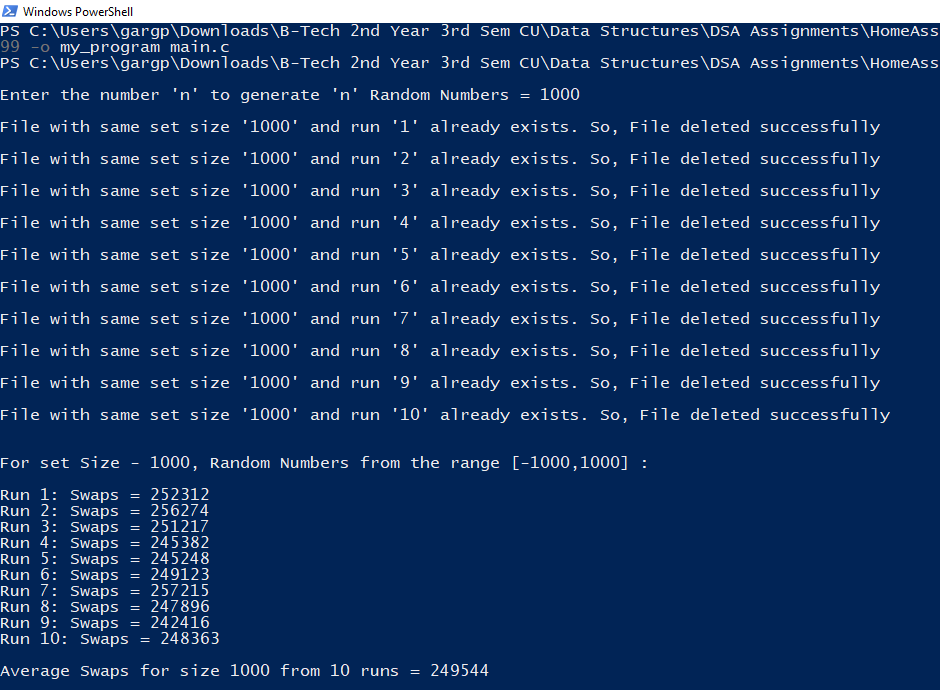
**Step-8.3:** Create a file path for the output file based on the folder name, set size, and run number.

**Step-8.4:** Open the file for writing and print the unsorted random numbers to the file.

**Step-8.5:** Sort the array using Bubble Sort and count the total number of swaps.

**Step-8.6:** Write the sorted numbers and the number of swaps to the output file.

**Step-8.7:** Display the number of swaps for the current run and store it in the structure.



**2. Case - II**

**1. Case - I**

**# Output:**

Repeat Step-8 for 10 times and 10 output files will be generated for 10 set of ‘n’ random numbers with unsorted, sorted and number of swaps written.

End

Calculate the total swaps by adding the number of swaps for each run by retrieving them for the structure.

Then, Calculate the average swap using the formula (total swaps / number of runs) and display it for the given set size.

**Step-9:**

**Step-10:**

**Step-11:**

**Step-12:**

**Yes**

**No**

**?**

**?**

**Check if all 10 runs are completed.**

**For each of the 10 runs, generate random numbers, sort them using Bubble Sort, and count swaps and print result to user. Also, store the number of swaps for each run.**

**Else continue**

**Delete them if found**

**Check if files with the same set size and run number already exist in the folder.**

**Prompt the user and read the input ‘n’ and store it.**

**Start**

**# Flow-Chart:**

**Seed the random number generator with the current time, also create a folder for storing output files.**

**Calculate the average swap from total swaps by summing swaps from all 10 runs and dividing it by number of runs.**

**End**

**Display the average swaps for the given set. size.**