Bonus Assignment (Given: Jan 12, Due: Jan 21)

In this assignment you will empirically evaluate the performance of different ADTs which you have implemented during the course. You will use the following 11 ADTs.

1. Linked list (unsorted)
2. Linked list (sorted)
3. List (list) in STL
4. Binary Search Tree
5. Binary Search Tree/Red-Black-Tree (map) in STL
6. Hash table with chaining and multiplication method
7. Hash table with chaining and division method
8. Hash table with open addressing and linear probing
9. Hash table with open addressing and quadratic probing
10. Hash table with open addressing and double hashing probing
11. Hash table (unordered_map) in STL

To store elements in the above mentioned ADTs, generate a random array/vector of size 100,000 using the std::shuffle function (http://www.cplusplus.com/reference/algorithm/shuffle/).

Perform the following functions on each of the 11 ADTs and measure the time taken. You can use chrono (http://www.cplusplus.com/reference/chrono/steady_clock/) for measuring the execution time. Make sure that same numbers are used for all the ADTs:

1. Insert
2. Delete (random 10,000 elements which exist in the array)
3. Search (random 10,000 elements which do not existing in the array).
4. Search (random 10,000 elements which exist in the array)

Write a detailed report with appropriate tables, and charts about the performance of different ADTs. Discuss which ADT performs better under what conditions and why?

For analyzing the results

1. Generate a CSV file for results, which can be opened in MS Excel
2. Repeat the experiments 10 times (each time with different numbers in the array) and take the average of 10 runs. This can be done programatically.

Instructions:

Following these instructions carries marks.

- Start early.
- Before submission, remove all the debugging and temporary files (in visual studio select menu *Build → Clean Solution*). Only submit the .cpp and .h files (no visual studio or other files). Select .cpp and .h files and compress them using your full registration number and name, (e.g., `04071512007-Ali-Ahmad.zip`). Submit via email (rabbasi@qau.edu.pk) within due time (no extensions).
- The source code should be properly commented and indented.
- Try to avoid using conio.h, as it is not part of standard C++. Don't use clear screen function. Don't use getch function, instead use the standard getchar function (if required).
- Any genuine efforts in each part, would result in at least 50% marks (in that part). Make sure you put your best efforts to solve every part. Each part carries its own marks. Different parts are highlighted in the assignment description (see the bold words). You are getting 50% marks for any genuine efforts in all the parts to encourage you to learn, even if your program does not compile and is full of bugs). Therefore, please do not plagiarize! Plagiarism includes taking help in any form including but not limited to code, concept or idea for the solution, algorithm, and pseudocode. Taking help from any source including but not limited to classmates, seniors, or internet is prohibited. In case your code is plagiarized, you'll get -50% absolute marks of the whole assignment. For example, if the assignment is of 50 marks, you will get -25 marks. **Even a single plagiarized statement will count as plagiarism for the whole assignment**. Plagiarism in two assignments would result in getting failed in the course.