

- 1) Implement the map class (binary search tree) using linked structures and a dummy head node. Each node should contain pointers to left, right, and parent nodes. The ADT should contain the following functions: constructor, copy constructor, operator=, destructor, at, operator[], begin, end, empty, size, clear, insert, erase, count, find, and contains. Also implement the iterator and reverse_iterator classes and their corresponding functions. Implement at least one overload mentioned in the C++ documentation: <https://en.cppreference.com/w/cpp/container/map>
- 2) Use the an STL ADT to solve the following problem:
 - a) Given an input text file, display the frequency of the words in that file.
 - i) The file name/path should be passed as a command line argument, or taken as input from user if no command line argument is passed.
 - ii) Remove the punctuation marks from the words.
 - iii) Ignore the stop words (<https://gist.github.com/larsyencken/1440509>).
 - iv) The word count should be case-insensitive (e.g., Science and science should be considered as one word).
 - v) Implement an efficient solution by keeping time complexity in mind.
 - b) Bonus task 1
 - i) Instead of opening a file, open a webpage when its URL is provided. You may use libcurl for downloading a webpage (<https://curl.se/libcurl/>).
 - ii) Remove the HTML tags from the downloaded webpage (they are enclosed in < and >).
 - c) Bonus task 2
 - i) Instead of displaying the frequencies of the words, generate an HTML file which shows a word cloud. You may use frequency as a measure of font-size and display frequent words in larger font.
 - d) Bonus task 3
 - i) Maintain the variations of the words, e.g., if science appears 2 times, Science 3 times, and SCIENCE 1 times, you should count the word science as 6 times in total and keep track of its variations (science 2, Science 3, SCIENCE 1). This should be done using minimum time complexity.

Instructions:

- Start from day 1. Submit to MS Teams before due time. Do not delay submission for the last moment. Late submissions will not be accepted.
- 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). Delete the .vs hidden folder before submission.
- Select .cpp and .h files and compress them using your full registration number and name, (e.g., 04072312007-Ali-Ahmad.zip).
- Avoid using conio.h, as it is not part of standard C++. Don't use clear screen function. Don't use getch function.
- The source code should be properly indented and commented.
- Any genuine efforts in each part, would result in at least 50% marks (for that part). Make sure you put your best efforts to solve every part. Each part carries its own marks.

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 or giving help in any form including but not limited to code, concept or idea for the solution, algorithm, or pseudocode. Taking help from any source including but not limited to classmates, seniors, internet, or LLMs is strictly 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 may result in getting failed in the course.