

Assignment 1 (Due: Nov 11)

1. The idea is to develop an information system of your choice (e.g., for a mobile shop, boutique, library, groceries store, beauty parlor, or any business of your choice). The main functionality of the system would be to maintain records of the business you choose. [CLO 2]
 - a. When the program starts, the main menu of the program will open which will allow the user to interact with the system (e.g., adding new items or updating/deleting/searching existing items, etc.). Each record should have a unique ID, duplicate records should be avoided.
 - b. Whenever the user gives some input to the system, make sure the input is validated. Your program should be able to handle unexpected inputs.
 - c. All the records must be stored in a binary file. If the file for storing data does not exist, it should be created. Whenever a record is changed (created, updated, or deleted), the file should be updated accordingly.
 - d. There should be an option of importing and exporting the records from a CSV file. CSV files should only be used for importing and exporting data. Actual data used by the system should be stored in binary files.
 - e. There should be an option of showing help for using the program.
 - f. Write a manual (in MS Word) which explains how the system should be used. Do not use LLMS (gemini, ChatGPT, etc. for generating/editing the documentation or code)
 - g. Prepare a presentation in which you will explain your system and how it can be used.
2. This part of assignment will help you in understanding the relation between execution time and time complexity. We will measure the execution times of algorithms having different best case time complexities. And also measure the execution times of algorithms having same worst-case time complexities, but different approaches (like using algorithms which do less or more swaps). You need to implement the following algorithms: [CLO 2]
 - a. Correct selection sort (in which at most $n-1$ swaps are performed)
 - b. Wrong selection sort (in which $O(n^2)$ swaps are performed)
 - c. Bubble sort (stop sorting, if there is no swap in an iteration)
 - d. Insertion sort
 - e. You have to create three arrays A, B and C.
 - Array A should contain random values.
 - You can use *rand* function for generating random values:
<https://en.cppreference.com/w/c/numeric/random/rand>
 - Array B should contain values sorted in ascending order.
 - Array C should contain values sorted in descending order.
 - f. You also need to learn how to measure execution time, this can be done using *chrono*:
<https://en.cppreference.com/w/cpp/chrono>
 - g. Now do the following tasks:
 - i. Measure how much time each of the three algorithms takes to sort the random array (A). Make sure, you generate a new random array before calling the sort function. Only measure the time for sorting the array (not for the whole program or generating arrays).
 - ii. Measure how much time each of the three algorithms takes to sort an already sorted (ascending) array (B).
 - iii. Measure how much time each of the three algorithms takes to sort an already sorted (descending) array (C).

- iv. Instead of getting results for a single run, repeat the experiments 10 times (each time with different numbers in the arrays) and take the average time of 10 runs.
- v. There should be a single program to do all the tasks. There should be no input taken from the user.
- vi. Generate a CSV file for results, which should be openable in MS Excel. The CSV file should have the columns. (Algorithm, Array Size, Array Type (random/sorted-asc/sorted-desc), Average Time Taken).
- vii. Write a detailed report with appropriate tables and charts showing the performance of different algorithms. Discuss in detail which algorithms perform the best under what conditions? And why? Which are worst? And why?

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. 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., 04071512007-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 (you may use the standard getchar() function instead).
- 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 (Chat GPT, Gemini, Meta, etc.) 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.