

National University of Sciences and Technology
School of Electrical Engineering and Computer Science
Department of Computing

CS-250 Data Structures and Algorithms

Spring 2024

Assignment 2

Stacks and Queues

Announcement Date: 15th Mar 2024

Due Date: 24th Mar 2024 at 11:59 pm (on LMS)

Instructor: Bostan Khan

Objective: The objective of this assignment is to explore the applications of stacks and queues and gain hands-on experience with implementing and manipulating these data structures.

Please read the instructions for each task with great care and be sure to implement the requirements stated against each task.

This is a group assignment where each group can have a maximum of **2 students**. Remember to include the names and registration numbers of both students at the start of the report.

- **Tasks 1:**

Implement a stack data structure in C++ to create a simple text editor with undo and redo functionality. The stack will be used to store the history of text changes, allowing users to undo their actions and redo them if needed.

Requirements:

1. Implement a stack data structure in C++ to store text strings.
2. Develop a text editor program that allows users to perform the following operations:
 - a. Insert text: Add a new text string to the editor.
 - b. Delete text: Remove text from the editor.
 - c. Undo: Revert the last text operation (insertion or deletion).
 - d. Redo: Restore the text operation that was undone.
3. Ensure that undo and redo operations are properly managed using the stack data structure.
4. Handle edge cases such as undo/redo when there are no actions to perform.
5. Implement a user-friendly menu-driven interface to interact with the text editor.

- **Task 2:**

You are tasked with designing a printing queue management system for a busy office environment. The office has a single printer that serves multiple users. Users submit print jobs to the printer, and the printer processes the jobs in the order they were received. Each print job has a specific number of pages to be printed.

Requirements:

1. Implement a queue data structure in C++.
2. Implement functions to enqueue (add to the end of the queue) and dequeue (remove from the front of the queue).
3. Implement a function to display the current state of the queue.
4. Create a menu-driven program to simulate the printing queue management system.
5. Users should be able to submit print jobs to the queue with the number of pages to be printed.
6. The program should display the status of the printing queue, including the number of jobs in the queue and their details (e.g., job number, number of pages).
7. The printer should process print jobs in the order they were received, printing each job's pages and removing it from the queue once completed.
8. Display appropriate messages to inform users about their actions and the status of their print jobs.

Instructions:

- a) Implement the Queue data structure using either arrays or linked lists while justifying your choice.
- b) Design a menu with options for users like submit print jobs or exit the program.

- c) Use appropriate error handling to prevent queue underflow or overflow.
- d) Test your program with multiple scenarios to ensure its correctness.

- **Documentation Report PDF [Marks: 30]:** Provide a report PDF for your implementations, including explanations of the data structures and algorithms used, as well as any design decisions made for each task. Document the usage of your data structures and algorithms in each of the application scenarios, including how the data structure is utilized to solve the problem. **The marks of your coding tasks will be directly influenced by the documentation report.** Include clear instructions on how to compile/run your code and how to interact with the implemented applications.

- **Testing Results in Documentation [Marks: 20]:** Test your implementations thoroughly to ensure correctness and robustness. Provide sample input-output scenarios for each application scenario to demonstrate the functionality of your implementations.

Remember to document your work well in the documentation report. Explain the code, its design and test results with details for obtaining max marks for the whole assignment.

- **Submission:** You will have to submit a separate .cpp source file for each task. Present The documentation report including the test results in a well-prepared PDF.
- **The report PDF and the sources files should be compressed in a single zip file and submitted on LMS with the name in following format: <Student_name>_<Reg_no>.zip**
- **Additional Notes:**
 - You are encouraged to be creative and innovative in your implementations. Consider additional features or optimizations that could enhance the functionality or performance of your applications.
 - Collaboration with classmates is allowed for discussing concepts and problem-solving strategies, but each student must submit their own individual solution.
 - **Plagiarism or copying of code from other students will result in zero marks.**