

This C++ code simulates enqueue and dequeue operations on a queue data structure. It begins by including necessary header files and defining the main function. Inside the main function, it initializes variables and a queue object. The code then generates a random number of trials to simulate queue operations. For each trial, it randomly determines the number of elements to enqueue and prints the trial number along with the number of elements to enqueue. It then enqueues random elements onto the queue and prints the current size of the queue along with its contents. Next, it randomly determines the number of elements to dequeue and prints the number of elements to dequeue. It dequeues the specified number of elements from the queue, prints the current size of the queue, and its contents. The process repeats for each trial, and the program ends after completing all trials. Overall, the code demonstrates basic operations on a queue data structure using randomization for simulation purposes.

Here are some problems I encounter during the process:

1. I forgot to consider the situation when the queue is empty when deleting nodes from it. This broke the code and I had to fix it by considering when the head is NULL. Thus, the problem is solved.
2. I was not sure about the concept of class in C++ that I misused it in my code. After some research I was able to correctly use the node and queue to complete this assignment.