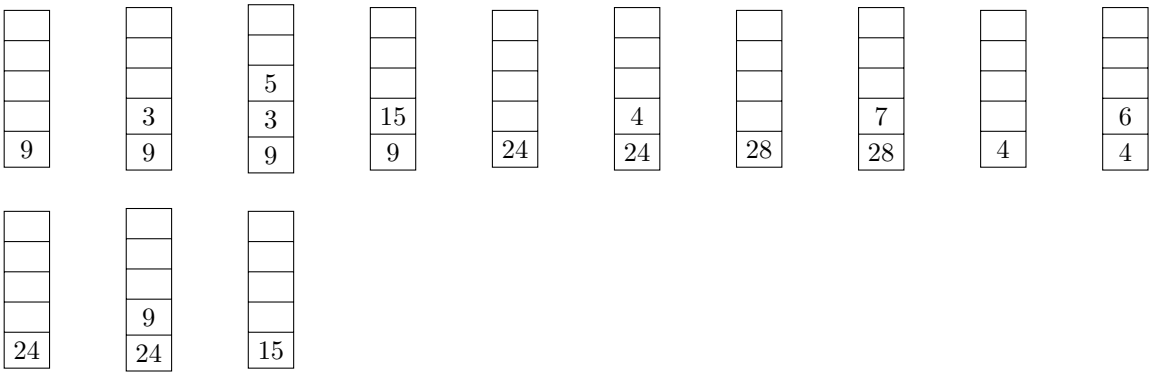
ICS 202 Assignment 2

Loay Shqair, 202365030

March 15, 2025

Question III

Question III (10 points): Evaluate the following postfix expression using a stack, showing all the intermediate steps: `9 3 5 * + 4 + 7 / 6 * 9 -`



Question VII

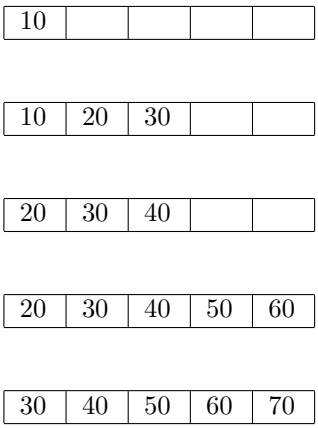
Question IV (10 points):
Consider a queue implemented using a circular array with a maximum capacity of 5 elements. Initially, the queue is empty.

a. Perform the following sequence of operations and show the state of the queue after each operation: Enqueue(10), Enqueue(20), Enqueue(30), Dequeue(), Enqueue(40), Enqueue(50), Enqueue(60), Dequeue(), Enqueue(70).
b. Explain why the last enqueue operation may or may not be successful.
c. What are the advantages of using a circular array implementation for a queue compared to a linear array implementation?

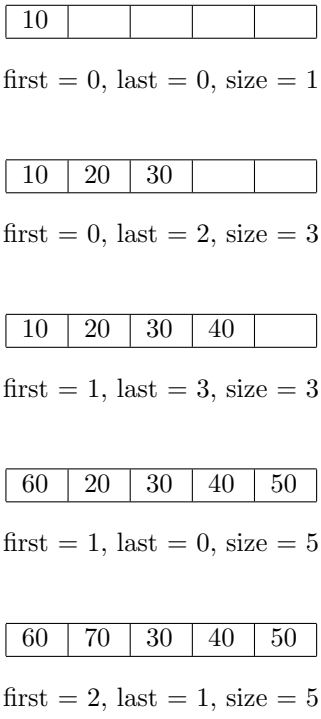
a.)

The state of the queue and the state of the circular array used to implement the queue are not exactly the same. Queues are abstract data types so they will hide/encapsulate their implementation details. In other words, its state representation would remain the same whether it was implemented as a linked list or an array.

Queue state



Circular array state



b.)

According to my research, some circular arrays implementation of queues don't store a occupied size as an attribute. In this case, the condition for checking whether an array is full or not would be to check whether (rear + 1)%capacity == front where rear = (last + 1)%capacity and front = first. If it the case, the queue is full. Otherwise, it is not. When we attempt to enqueue 70, first = 2 and last = 0, so front = 2 and rear = 1.. (rear + 1)%5 = (1 + 1)%5 = 2 = front, the queue would be considered full and 70 won't be enqueued. In such an implementation however, 60 would also not be enqueued.

c.

For a linear array implementation, we're gonna have to shift all the elements of the array whenever we dequeue which, in the worst-case, is $O(n)$ where n is the size of the array. However, for a circular array implementation, we simply modify the start and end pointers/indices of the queue using modular arithmetic which is constant time. Thus, the main advantage of using a circular array is that it is more efficient time-wise.