

1. Assume you have an empty stack, a sequence of operations are performed on it as shown below. Show all intermediate steps and the final stack

a. push(a), pop(), push(b), push(c), pop(), push(d), pop, push(e), pop, pop

Push a: a

Pop:

Push b: b

Push c: b,c

Pop: b

Push d: b,d

Pop: b

Push e: b,e

Pop: b

Pop:

End result is there's nothing left. Since there's five push's and five pop's, every element gets removed.

2. Assume you have an empty queue, a sequence of operations are performed on it as shown below. Show all intermediate steps and the final queue.

a. enqueue(a), enqueue(b), dequeue, enqueue(c), enqueue(d), enqueue(e), dequeue, dequeue, enqueue(f)

Enqueue a: a

Enqueue b: a,b

Dequeue: b

Enqueue c: b,c

Enqueue d: b,c,d

Enqueue e: b,c,d,e

Dequeue: c,d,e

Dequeue: d,e

Enqueue f: d,e,f

End: d,e,f

3. Answer the following questions about your C++ program:

a. Write a short summary explaining your solution for question-3

I found this one to be a bit tricky but I think I was able to find an efficient solution to it. To start, I took input from the users for both the number of lockers and passes. I then filled an array of lockers and set each one equal to 0. I set 0 as closed and 1 as open. I then made a nested for-loop where I said if the current pass index was divisible by the current locker index, then I'd switch the status. Then, I ran through it again once it was done and counted all the open lockers, and printed out the status of each one and how many were open.

- b. Did you encounter any challenges in implementing the solution? How did you overcome them?

I think understanding the puzzle was a bit complex. It would've been nice if the wording was a bit better or if a few test cases were given because I initially set up the problem wrong as I wasn't entirely clear on when the status should be changed. To fix this, I reached out to the professor and was directed to the correct method.

- c. Any special instructions to compile your code

No, it should work with the given compilation instructions and worked in all of my test cases.