

University of Wisconsin - La Crosse

Date: April 29

- 1

3. What is displayed to the console after running the code below? What does the stack looks like over time?

```
1 LinkedList<Integer> stack = new LinkedList<>();
2 stack.push(3);
3 for (int i = 1; i <= 5; ++i) {
4     if (peek() % 2 == 0) {
5         stack.push(i);
6     } else {
7         int r = stack.pop();
8         stack.push(i + r);
9     }
10 }
11 while (!stack.isEmpty()) {
12     System.out.print(stack.pop() + " ");
13 }
```

4. What is displayed to the console after running the code below? What does the queue looks like over time?

```
1 LinkedList<Integer> queue = new LinkedList<>();
2 queue.add(3);
3 for (int i = 1; i <= 5; ++i) {
4     if ((i + queue.peek()) % 2 == 0) {
5         queue.add(i);
6     } else {
7         int r = queue.poll();
8         queue.add(i + r);
9     }
10 }
11 while (!queue.isEmpty()) {
12     System.out.print(queue.poll() + " ");
13 }
```

5. Explain how you could use a stack and a queue to determine if the characters in a **String** form a palindrome. Note: This solution is not the most efficient way to solve this problem. The problem is an exercise in understanding stacks and queues.