Fall 2022 Recitation 3: Stacks

1. [10 minutes] Write the following push(), pop(), peek() methods to implement a stack using a doubly linked list (head is top of the stack)

```
public class intNode {
        private int number;
        private intNode next;
        private intNode prev;
        public intNode(int number) {
                this.number = number;
                next = null;
                prev = null;
        }
}
public class intStack {
        private intNode top;
        public intStack() {
                top = null;
        }
        public void push(int number) {





        }
        public int pop() {





        }
        public int peek() {




        }
}
```

2. [5 minutes] Write the order of complexity in Big-O for the following operations

    a. Searching for a value in a stack.

    b. Reversing an array using a stack.

    c. Evaluating a postfix expression using a stack.

    d. Adding an element to a stack.

    e. Retrieving the bottom-most value in a stack.

    f. Removing a single element from a stack.

3. [5 minutes] Evaluate the following postfix expression:

   2 3 – 4 + 5 6 7 * + *

4. [5 minutes] Evaluate the following prefix expression:

   + - * 2 2 / 9 8 5

5. [5 minutes] Convert the following prefix expression to postfix:

   / * A - B C * / D E F

6. [5 minutes] Write a method to reverse an array using a stack

```
public void reverse(int array[]) {
```

7. [15 minutes] Write a method that evaluates a postfix expression using a stack and returns the result. Assume a valid postfix string is given and assume you have a stack with the following methods: .push(), .pop(), .peek():

   ```
   Public static int evaluatePostfix(String str) {
   ```