Guidelines:

- **Make sure to submit your files to Google Classroom before the deadline,** otherwise, your work won't be considered for grading.
- **Submit only the java files as separate files (i.e. not as zipped files).**
- **Refer to the Java files attached on Google classroom**

Question 1: ArrayList Implementation

Consider the `MUArrayList<E>` implemented in Lesson 3

1. Implement the methods `indexOf` method of the `List` interface.
2. Implement the method `addMiddle(item E)` that adds the element item to the middle of the list. Do not use any of the List interface methods in the implementation of this method.

```
public boolean addMiddle(E item)
```

Question 2: LinkedList Implementation

Consider the `MUSingleLinkedList<E>` implemented in Lesson 4

1. Override the `toString()` method such that it prints the content of the list as follows:

   Tom ==> Dick ==> Harry ==> Sam

2. Implement the methods `size`, `indexOf`, and `remove` methods of the `List` interface (**Use the private helper methods implemented in Lesson 4, but not the `List` interface methods**)
3. Write the remove method whose method heading follows.

```
/**
 * Remove the first occurrence of element item.
 *
 * @param itemToRemoveThe item to be removed
 * @return true if item is found and removed; otherwise, return
 * false.
 */
public boolean remove(E itemToRemove)
```

4. Write the following method add for the class `MUSingleLinkedList<E>` **without using any helper methods**.

```
/**
 * Insert a new item before the one at position index, starting
 * at 0 for the list head. The new item is inserted between the
 * one at position index-1 and the one formerly at position
 * index.
 *
 * @param index The index where the new item is to be inserted
 * @param item  The item to be inserted
 * @throws IndexOutOfBoundsException if the index is out of range
 */
public void add2(int index, E item)
```