

[**Closed book and notes.**] Show all of your work clearly in the space provided or on the additional page at the end of the exam. If the additional page is used, clearly identify to which exam question it is related. Be sure to **read each problem carefully**. Note that the exam is double sided.

1. (10 points) Give an example one method that is $O(n)$ for the `ArrayList` and $O(1)$ for the `LinkedList`. Explain why (discuss the internal structure of each container that causes the specific time complexity).

2. (10 points) Give an example of one method that is $O(1)$ for the `ArrayList` and $O(n)$ for the `LinkedList`. Explain why.

3. (10 points) Consider the following partial implementation of the ExamIterator that is cleverly designed to work with either an ArrayList or a LinkedList (if it were added as a nested class to our ArrayList or LinkedList) implementation.

```
private class ExamIterator implements Iterator<E> {
    private int index;

    private ExamIterator() {
        index = -1;
    }

    public boolean hasNext() {
        boolean hasNext = true;
        try { // Try to get the next element
            get(index + 1);
        } catch (IndexOutOfBoundsException e) {
            hasNext = false; // Return false if unable to get the next element
        }
        return hasNext;
    }

    public E next() {
        E value = null;
        if (hasNext()) {
            ++index;
            value = get(index);
        } else {
            throw new NoSuchElementException("Iteration has no more elements");
        }
        return value;
    }

    public void remove() {
        throw new UnsupportedOperationException("The operation has not been implemented.");
    }
}
```

Is this implementation of the Iterator interface more appropriate for an ArrayList or a LinkedList? Justify your answer.



4. (15 points) Implement the following method that uses a list iterator to display (to the console) the second, fourth, sixth, etc... elements in the list (in order). For example, a list with

"first", "second", "third", "fourth", "fifth", "sixth", "seventh"

should produce the following result:

second fourth sixth

```
public static void display(List<String> list) {
```

```
}
```



5. Consider an implementation of an `ArrayList<E>` that contains one attribute: `data` which is an array that stores all of the `E` elements. (The size of the array is always the same as the size of the `ArrayList`.)

(a) (10 points) Implement the `size()` method.

(b) (10 points) Implement the `add(E element)` method.

6. Recall the `LinkedList` class that we have been developing in lecture had two attributes: `head` and `tail`.

(a) (10 points) Draw a diagram that illustrates what the list looks like after the following operations:

```
LinkedList<Integer> list = new LinkedList<Integer>();  
list.add(3);  
list.add(5);
```

(b) (10 points) Draw a diagram that illustrates what the list above looks like after the following additional operations:

```
list.add(1, 1);
```



(c) (15 points) Implement a method called `E removeFront()` which is a new method on the `LinkedList` class that removes the first element in the list whenever it is called. The method returns the value that was removed from the list. If the list is empty, it must throw a `NoSuchElementException`.