Implementing List using an array

(1) Imagine you are implementing a `List` of `String`s using an array. You decide to call it `CarlStringList`. It has the instance variables shown below. Write a method to get an item at a specific index in the list. Think about any edge cases you have to handle; you might `throw` a `RuntimeException` for these cases. *[Hint: This method is very simple in terms of number of lines of code.]*

```
public class CarlStringList implements List<String>
{
   private String[] array;
   private int count;

   public String get(int index)
   {









   }
}
```

(2) Write an `add(int index, String item)` method for `CarlStringList` implementation you started above, assuming there is sufficient space in the array to add an additional item. *[Hint: The tricky thing here isn't the Java syntax but ordering the steps of your algorithm.]*

```
public void add(int index, String item)
{











}
```

(over)

(3) Imagine you're trying to test your `add` method. Brainstorm some test cases: things you want to try to make sure that the method works correctly with all inputs.

(4) Write a method to "resize" the array and do any necessary copying. This can be used *e.g.* in the `add` method when there is not sufficient space to add another item. *[Hint: Read the next question.]*

```
private void resizeArray()
{



}
```

(5) Consider two ways of resizing the array: adding 10 to the size of the array each time or doubling the size of the array each time. If you add 100 items to the `List` and the array starts with a capacity of 1, how many times will each scheme resize the array? How many total times will each scheme copy `String`s when resizing?

If you have extra time, use `resizeArray` in `add`, write `remove`, `contains`, and `equals` methods.