Assuming that our ArrayList is an ArrayList of Strings, we would have the following instance variables:

/* An array to hold the values */
String[] elements;

/* Number of values in the array */
int numElements;

/* Number of elements that will fit */
int capacity;

How would you write the remove method that is given a position in the array list and removes the element at that position? This will need to shuffle the elements down in the array so that the used part of the array list is contiguous and starts in position 0. This method should return the String that is removed.

public String remove (int position) {

}

If the contract for this method says that it will throw IndexOutOfBoundsException if the position passed in is invalid, how would you change the code above to do that?

What is the O() cost of this method?
We have seen this implementation for the method that adds an element to the end of an ArrayList:

```java
public boolean add (int value) {
    if (numElements == capacity) {
        grow();
    }

    elements[numElements] = value;
    numElements++;
    return true;
}
```

Now consider an add method that is given a value and a position and inserts the value into that position. How would you write this method?

```java
public void add (int position, int value) {
```

What is the $O()$ cost of this method?