COMPUTER SCIENCE A
SECTION II
Time— 22.5 **minutes**

Directions: SHOW ALL YOUR WORK, REMEMBER THAT PROGRAM SEGMENTS ARE TO BE WRITTEN IN Java.

Notes:

- Assume that the classes listed in the Quick Reference found in the Appendix have been imported where appropriate.
- Unless otherwise noted in the question, assume that parameters in method calls are not `null` and that methods are called only when their preconditions are satisfied.

The following class `WordList` is designed to store and manipulate a list of words. The incomplete class declaration is shown below. You will be asked to implement two methods.

```
public class WordList
{
  private ArrayList<String> myList; // contains Strings made up of letters


  // postcondition: returns the number of words in this WordList that
  //                are exactly len letters long
  public int numWordsOfLength(int len)
  {  /* to be implemented in part (a) */  }


  // postcondition: all words that are exactly len letters long
  //                have been removed from this WordList, with the
  //                order of the remaining words unchanged
  public void removeWordsOfLength(int len)
  {  /* to be implemented in part (b) */  }


  // ... constructor and other methods not shown
}
```

(a) Write the `WordList` method `numWordsOfLength`. Method `numWordsOfLength` returns the number of words in the `WordList` that are exactly `len` letters long. For example, assume that the instance variable `myList` of the `WordList animals` contains the following.

```
["cat", "mouse", "frog", "dog", "dog"]
```

The table below shows several sample calls to `numWordsOfLength`.

Call	Result returned by call
`animals.numWordsOfLength(4)`	1
`animals.numWordsOfLength(3)`	3
`animals.numWordsOfLength(2)`	0

Complete method `numWordsOfLength` below.

```
// postcondition: returns the number of words in this WordList that
//                are exactly len letters long
public int numWordsOfLength(int len)
```

(b) Write the `WordList` method `removeWordsOfLength`. Method `removeWordsOfLength` removes all words from the `WordList` that are exactly `len` letters long, leaving the order of the remaining words unchanged. For example, assume that the instance variable `myList` of the `WordList` `animals` contains the following.

```
["cat", "mouse", "frog", "dog", "dog"]
```

The table below shows a sequence of calls to the `removeWordsOfLength` method.

Call	`myList` after the call
`animals.removeWordsOfLength(4);`	`["cat", "mouse", "dog", "dog"]`
`animals.removeWordsOfLength(3);`	`["mouse"]`
`animals.removeWordsOfLength(2);`	`["mouse"]`

Complete method `removeWordsOfLength` below.

```
// postcondition: all words that are exactly len letters long
//                have been removed from this WordList, with the
//                order of the remaining words unchanged
public void removeWordsOfLength(int len)
```