This exam has 3 additional pages. Please answer each question on the page on which it is asked. You may write on the back of the facing page if you need to.
1. (a) For each of the following operations, state whether ArrayList is much better, LinkedList is much better, or whether both are basically the same.

- add( 0, x )
- get( idx )
- remove( x )
- add( x )

(b) For each of the following, state whether it is true for Lists only, Sets only, both Lists and Sets, or neither Lists nor Sets.

- has an efficient contains operation
- allows duplicates
- supports add and remove of a single object
2. Class `Multiset`, shown below is used to maintain a collection of `Strings`, **in which duplicates are allowed**. It stores a map in which the key is a `String`, and the value is the number of occurrences of the `String`.

```java
class MultiSet {
    // Constructor and most methods not shown

    private Map<String, Integer> counts;

    // Return size of the MultiSet
    // If counts = [ hello=2, world=1, zebra=1 ]
    // This routine returns 4
    public int size() {
        /* You provide implementation */
    }

    // Return true if x is in this Multiset
    public boolean contains(String x) {
        /* You provide implementation */
    }

    // Add a new item
    public void add(String x) {
        /* You provide implementation */
    }

    // Remove an item.
    // If x is not found, return false.
    // If x is present with count 1, remove it from the map.
    // If x is present with count > 1, lower the count by 1.
    // If x was present return true.
    public boolean remove(String x) {
        /* You provide implementation */
    }
}
```

(a) Implement `contains`.
(b) Implement `add`.
(c) Implement `remove`.
(d) Implement `size`.
3. Suppose a database contains a table named Schedule, with columns Name, Number, Section, AvailableSeats. An example of five rows in this table could be

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>COP</td>
<td>2250</td>
<td>01</td>
<td>0</td>
</tr>
<tr>
<td>COP</td>
<td>2250</td>
<td>02</td>
<td>5</td>
</tr>
<tr>
<td>COP</td>
<td>2250</td>
<td>03</td>
<td>0</td>
</tr>
<tr>
<td>COP</td>
<td>2250</td>
<td>04</td>
<td>7</td>
</tr>
<tr>
<td>COP</td>
<td>3804</td>
<td>01</td>
<td>3</td>
</tr>
</tbody>
</table>

This database might model the Spring 2010 schedule, and there can be several sections of a given course.

Method closedSections prints all sections of the specified course that have no available seats. It takes a Connection object, so you do not need to worry about opening the database. As an example invoking closedSeats( c, "COP", "2250" ) with an appropriate Connection c will print 01 03.

Implement closedSections below. You do not have to use any import directives, and you can propagate SQLExceptions.

```java
public static void closedSections( Connection c, String name, String number )
throws SQLException
{
}```