COP 3330 Quiz #4 Sample

1) (10 pts) Determine the run-times of each of these Java API method calls. Assume that n represents the number of items in the appropriate data structure. Each answer will be either O(1), O(lg n) or O(n).

Class	Method	Run-time
ArrayList	E remove(Object o)	_____
ArrayList	get(int index)	_____
ArrayDeque	pollLast()	_____
HashSet	boolean contains(Object o)	_____
HashSet	int size()	_____
HashMap	containsKey(E e)	_____
HashMap	containsValue(E e)	_____
TreeMap	V get(Object key)	_____
TreeMap	Set<K> keySet()	_____
PriorityQueue	add(E e)	_____

The rest of the questions on this sample quiz are to solve a particular problem. The problem will be broken up into three pieces. An EscapeRoom business has multiple rooms. Several people are assigned to each room. No person can be assigned to more than one room. Our program will read in a list of ordered pairs: (person, room). You will write a constructor for the EscapeRoom class that will take in the name of an input file storing this information and build the corresponding EscapeRoom object. Once this object is built, you'll have to handle one type of query: given a room, return an array storing each person in that room. You'll write an instance method in the EscapeRoom class to handle this query. Finally, you'll write a main method that will allow a user to run this functionality (prompts the user to enter a file, then, if it exists, asks the user for a room to query and prints out the people in that room, if that room is a part of the EscapeRoom object in question.)

The class will have the following two instance variables:

```
private ArrayList<String> rooms;
private HashMap<String, HashSet<String>> assignments;
```

The first will store a list of each unique room in the EscapeRoom object. The second will map each room to a set of people who are assigned to that room.

The constructor will be given a String storing a file name that contains the room assignments. Since this file might not exist, the constructor will simply throw an IOException, instead of attempting to handle the exception. (We'll handle the exception from the main method of the EscapeRoom class.)

The file format is as follows: The first line contains a single positive integer, n, representing the number of people assigned to rooms. Then n lines follow, each with a person's name (all letters) followed by a space, followed by a room's name (all letters).

2) (29 pts) Write the constructor for this class. Since this constructor opens the input file first, the last thing the constructor should do close the file. Full credit will only be given if the constructor has an efficient run-time. (Namely, adding a single person to the object should always take expected O(1) time.)

```
public EscapeRoom(String filename) throws IOException {




}
```

3) (15 pts) Write the instance method getAllPeople, which takes in a String, room, and returns a String[] storing a list of all of the people who are mapped to room. If no such room exists in the EscapeRoom object, then null should be returned.

```
public String[] getAllPeople(String room) {









}
```

4) (20 pts) Write the main method of the EscapeRoom class. In this method, prompt the user to enter a file name with the escape room connections. In a try-catch structure, attempt to create an EscapeRoom object given the user's input file name. In the same try clause, after creating the object, ask the user to enter a name of a room. Use this to call the getAllPeople method and print out all of the people in the room. If there is no such room, then print this error message: "not a room." When you catch the IOException, simply print out, "file not found."

```
public static void main(String[] args) {

}
```

4) (1 pt) Name one video game character in the new movie, "Super Mario Galaxy Movie."

_____