

Question 1. Using Collections**[14 marks]**

This question is about a program to keep track of the athletes in a sports competition.

The program has a `relayTeams` field containing a `List` of teams for the relay races. Each team is a `List` of `Athletes`, listed in the order that they will be running:

```
private List< List<Athlete> > relayTeams;
```

(a) **[4 marks]** Complete the following `rotateTeams()` method which will move the first athlete in every team to the last position of their team.

```
public void rotateTeams(){
    for ( List<Athlete> team : relayTeams){
        team.add(team.remove(0));
    }
}
```

(b) **[5 marks]** The program also has a `marathonRunners` field containing a `Set` of `Athletes` running in the quarter marathon.

```
private Set<Athlete> marathonRunners;
```

Complete the following `findMultiAthletes()` method that will return a `Set` of all the `Athletes` who are in a relay team and are also running in the marathon.

Assume that the `relayTeams` and `marathonRunners` fields have been initialised and filled correctly.

```
public Set<Athlete> findMultiAthletes(){
    Set<Athlete> answer = new HashSet<Athlete>();
    for ( List<Athlete> team : relayTeams){
        for ( Athlete ath : team){
            if (marathonRunners.contains(ath)){
                answer.add(ath);
            }
        }
    }
    return answer;
//OR
    Set<Athlete> answer = new HashSet<Athlete>();
    for ( List<Athlete> team : relayTeams){
        answer.addAll(team);
    }
    answer.retainAll (marathonRunners);
    return answer;
}
```

(c) [5 marks] Complete the following startingOrder method that will return a List of the Athletes in the marathonRunners Set, sorted by age, with the youngest athletes first.

Assume the Athlete class contains a int getAge() method that returns the age as an int. The Athlete class is **not** Comparable.

Hint: note that marathonRunners is a Set, not a List.

```
public List<Athlete> startingOrder(){
    List<Athlete> answer = new ArrayList<Athlete>(marathonRunners);
    Collections.sort(answer, (Athlete a1, Athlete a2) ->
        { int age1 = a1.getAge();
          int age2 = a2.getAge();
          if (age1 < age2) {return -1;}
          if (age1 > age2) {return 1;}
          return 0;
        }
    // OR (instead of the three if's)
    return (a1.getAge() - a2.getAge());
    });
    return answer;
}
```
