Question 1

Please indicate if each statement is `true` or `false` by returning the appropriate `boolean` from the corresponding method in the `BinaryWarmUp` class!

For example, the following means the first statement is false.

```
public static boolean s1() {
    return false;
}
```

Statements

1. Circular array enables efficient implementation (amortized constant time) of all operations for Queue.
2. Implementing a doubly linked list without sentinel nodes results in much simpler methods and fewer special (edge) cases.
3. A pre-order traversal of a binary tree always visits the subtree before it processes the data in the root of that subtree.
4. A Set is a collection of unique elements, but the elements might not be Comparable.
5. The height of a node in a tree is defined as the length of the shortest path from the root of the tree to that node.

Question 2

Please indicate the correct choice by returning the appropriate `char []` array from the corresponding method in the `ToughChoices` class!

For example, the following means the answer to the first MCQ is `a` and `c`, whereas the answer to the second question is `b`.

```
public static char q1() {
    return new char[] {'a', 'c'};
}

public static char q2() {
    return new char[] {'b'};
}
```

Multiple-Choice Questions

1. How efficiently can we do an `insertBack()` operation on a List containing N items if it is implemented as a singly linked list that has both a `head` and a `tail` reference?
    - a. $O(1)$
    - b. $O(\lg N)$
    - c. $O(N)$
    - d. It cannot be done in a singly linked list
2. Suppose you are implementing a *minimum* priority queue with a tree-based heap. In which type of node will a unique largest priority be stored?
    - a. root only
    - b. External nodes (leaves)
    - c. Internal nodes (those other than the root or leaves)
    - d. It could be stored anywhere
3. The `__________` cost of an operation is calculated by summing the total cost of some number of operations and then dividing by that number of operations.
    - a. amortized
    - b. best case
    - c. worst case
    - d. expected
4. This is supposed to be an AVL tree. Which AVL tree properties, if any, are violated in this example?

```
        4
     /     \
   1        8
 /   \
0     3
       \
        5
```

- a. no violations
- b. order only
- c. balance only
- d. order and balance both

5. Given this heap based minimum priority queue, what would its ranked array representation be after a remove operation?

```
        4
     /     \
   8        20
 /   \
10    16
```

    - a. $[-, 4, 8, 20, 10]$
    - b. $[-, 8, 16, 20, 10]$
    - c. $[-, 8, 10, 20, 16]$
    - d. $[-, 16, 8, 20, 10]$

6. What is the height of a binary heap with 2^N elements?
    - a. 2^N
    - b. N
    - c. N^2
    - d. $\log N$

Question 3

Consider the following AVL Tree:

```
        20
      /    \
    18      21
   /
  16
```

And, assume we have a `toString` method which displays this tree as

```
[20, 18, 21, 16, null, null, null, null, null]
```

What does `toString` method return after `insert(17)`?

Write your answer in the `ShortAnswer.md` file and justify it by showing the AVL Tree states through the insertion process.

Feel free to check out the implementation of `BinaryTree.toString` method.

Question 4

Complete the implementation of `PriorityQueue.swim` based on the provided code, comments therein, and your general understanding of the "swim" process.

Question 5

Complete the implementation of `BinarySearchTree.isValid` based on the provided code, comments therein, and your general understanding of the binary search tree properties.

Hints:

1. The input is (for sure) a binary tree
2. The order property in BST states "all" values to the left of a node are smaller than the value stored in it (not just the left child). Likewise, all values to the right of a node are larger than the value stored in it (not just the right child).