1.      If you have 3 bulbs outside the room and 3 switches inside the room. Initially all bulbs are off. You have to decide which switch is for which bulbs and you can go out side the room only once.

Turn [light switch](http://everything2.com/title/light+switch) 1 to the ON [position](http://everything2.com/title/position) for a few [minutes](http://everything2.com/title/minutes), then turn it off. Then turn on [switch](http://everything2.com/title/switch) 2, and go [upstairs](http://everything2.com/title/upstairs). The [light bulb](http://everything2.com/title/light+bulb) that is [off](http://everything2.com/title/off) and [hot](http://everything2.com/title/hot) is[attach](http://everything2.com/title/attach)ed to switch 1. The one that is [on](http://everything2.com/title/on) is attached to switch 2, and the one that is off and [cold](http://everything2.com/title/cold) is attached to switch 3.

There are several ways to solve this problem. An elegant solution is shown below. Note again: in advance, you **do not know** whether the coin with different weight is heavier or lighter!

Number the coins from 1 up to 12. Perform the following three weighings:

|  |  |  |
| --- | --- | --- |
|  | **Left side:** | **Right side:** |
| **Weighing 1:** | 1  2  3  10 | 4  5  6  11 |
| **Weighing 2:** | 1  2  3  11 | 7  8  9  10 |
| **Weighing 3:** | 1  4  7  10 | 3  6  9  12 |

Call the outcome of a weighing "L" if the left side is most heavy, call the outcome "R" if the right side is most heavy, and call the outcome "B" if the left and right sides have the same weight. Then the following outcomes are possible:

|  |  |  |  |
| --- | --- | --- | --- |
| **Weighing 1:** | **Weighing 2:** | **Weighing 3:** | **Different coin:** |
| L | L | L | 1 heavier |
| L | L | R | 3 heavier |
| L | L | B | 2 heavier |
| L | R | L | 10 heavier |
| L | R | B | 11 lighter |
| L | B | L | 6 lighter |
| L | B | R | 4 lighter |
| L | B | B | 5 lighter |
| R | L | R | 10 lighter |
| R | L | B | 11 heavier |
| R | R | L | 3 lighter |
| R | R | R | 1 lighter |
| R | R | B | 2 lighter |
| R | B | L | 4 heavier |
| R | B | R | 6 heavier |
| R | B | B | 5 heavier |
| B | L | L | 9 lighter |
| B | L | R | 7 lighter |
| B | L | B | 8 lighter |
| B | R | L | 7 heavier |
| B | R | R | 9 heavier |
| B | R | B | 8 heavier |
| B | B | L | 12 lighter |
| B | B | R | 12 heavier |

Note: outcomes LRR, RLL, and BBB are not possible.

**Writing JUnit tests for equals method in Java**

Its good practice to write JUnit test cases to test your equals method. Here is my approach for writing JUnit test case for equals method in Java. I will write test cases to check equals behavior on different circumstances:

          **testReflexive**() this method will test reflexive nature of equals() method in Java.

          **testSymmeteric**() this method will verify symmetric nature of equals() method in Java.

          **testNull**() this method will verify null comparison and will pass if equals method returns false.

          **testConsistent**() will verify consistent nature of equals method in Java.

          **testNotEquals**() will verify if two object which are not supposed to equals is actually not equal

          **testHashcode**() will verify that if two objects are equal by equals() method in Java then there hashcode must be same.