Write test cases to the following code.

```
Demo (int a) {
If (a> 5)
a=a*3
Print (a)}
```

1. Positive Number Greater Than 5:
   - Input: `a = 7`
   - Expected Output: `a` should be multiplied by 3 and printed, resulting in `21`.

2. Positive Number Equal to 5:
   - Input: `a = 5`
   - Expected Output: Since `a` is not greater than 5, it should not be multiplied, and the output should be `5`.

3. Positive Number Less Than 5:
   - Input: `a = 3`
   - Expected Output: Since `a` is not greater than 5, it should not be multiplied, and the output should be `3`.

4. Negative Number:
   - Input: `a = -7`
   - Expected Output: Since `a` is negative and not greater than 5, it should not be modified, and the output should be `-7`.

5. Zero:
   - Input: `a = 0`
   - Expected Output: Since `a` is not greater than 5, it should not be modified, and the output should be `0`.

6. Maximum Integer Value:
   - Input: `a = Integer.MAX_VALUE`
   - Expected Output: Since `a` is greater than 5, it should be multiplied by 3. However, be cautious about overflow issues with the integer value.

7. Minimum Integer Value:
   - Input: `a = Integer.MIN_VALUE`
   - Expected Output: Since `a` is not greater than 5, it should not be modified, and the output should remain as `Integer.MIN_VALUE`.

8. Floating Point Number Greater Than 5:
   - Input: `a = 6.5`

   - Expected Output: Since the code appears to be written for integer inputs only, this test case may result in a compilation error or unexpected behavior.

9. Character Input:
   - Input: `a = 'A'`
   - Expected Output: Since the code is designed to work with integer inputs, passing a character should result in a compilation error or unexpected behavior.

10. String Input:
    - Input: `a = "10"`
    - Expected Output: Since the code is designed to work with integer inputs, passing a string should result in a compilation error or unexpected behavior.