

# Functional Testing

**Interface: UnitTestingHomework**

**Method: double sqrt (int n)**

**Functionality:**

This method calculates the square root of an integer and returns a double value.

**Possible input values:**

An integer is given as input. There are few possible values for an integer input, namely positive which can be either even or odd, negative which can be either even or odd. There are also special input cases namely 1 and 0.

**Expected Output:**

- **Positive even integer:** Square root of a positive even integer should return a positive even number.
- **Positive odd integer:** Square root of a positive odd integer should return a double positive number.
- **Negative even/odd integer:** Square root of a negative even/odd integer returns an imaginary number, which is not a double number. Hence it can be converted into a long value which will return 0.0.
- **Input value 1:** Square root of 1 should return 1.0.
- **Input value 0:** Square root of 0 should return 0.0.

**Conclusion:**

Since all the above cases are handled, this implementation can be accepted.

**Method: int sqr(int n)**

**Functionality:**

This method calculates the square of an integer and returns an integer.

**Possible input values:**

An integer is given as input. There are few possible values for an integer input, namely positive integer, negative integer. There are also special input cases namely 1, 0, Maximum value and Minimum value.

**Expected Output:**

- **Positive integer:** Square of a positive integer should return a positive integer.

- **Maximum Positive integer:** Square of a maximum positive integer will round off.
- **Negative integer:** Square of a negative integer should return a positive integer.
- **Maximum Negative integer:** Square of a maximum negative integer will round off.
- **Input value 1:** Square of 1 should return 1.
- **Input value 0:** Square of 0 should return 0.

**Conclusion:**

Since all the above cases are handled, this implementation can be accepted.

**Method: int factorial (int n)**

**Functionality:**

This method calculates the factorial of an integer and returns an integer.

**Possible input values:**

An integer is given as input. Since factorial is defined for only non-negative integers the possible values for an integer input are positive integers. There are also special input cases namely 1, 0 and maximum value.

**Expected Output:**

- **Positive integer:** Factorial of a positive integer should return a positive integer.
- **Maximum Positive integer:** Factorial of a maximum positive integer should round off.
- **Input value 1:** Factorial of 1 should return 1.
- **Input value 0:** Square of 0 should return 1.

**Conclusion:**

Since all the above cases are handled, this implementation can be accepted.

**Method: int sumUp(int n)**

**Functionality:**

This method calculates the sum from 0 to n, where n is the largest addend.

**Possible input values:**

An integer is given as input. Since n should be the largest addend the possible values for an integer input are positive integers. There are also special input cases namely 1, 0 and maximum value.

### Expected Output:

- **Positive integer:** Sum from 0 to n, where n is a positive integer should return a positive integer.
- **Maximum Positive integer:** Sum from 0 to n, where n is a positive integer such that sum of 0 to n will return a value greater than MAX\_VALUE. Then the result will be rounded off.
- **Input value 1:** Sum from 0 to n, where n is 1 should return 1.
- **Input value 0:** Sum from 0 to n, where n is 0 should return 0.

### Conclusion:

Since all the above cases are handled, this implementation can be accepted.

**Method:** `int simpleFunctionXplusY(int x, int y)`

### Functionality:

This method calculates the sum of two integers and returns an integer.

### Possible input values:

An integer is given as input. The possible values for an integer input are positive integers, negative integers. There are also special input cases namely 1, 0, maximum value and minimum value.

### Expected Output:

- **Two positive integers:** Sum of 2 positive integer should return a positive integer.
- **One maximum value and a positive integer:** Sum of a positive integer and a maximum value will return a rounded off number.
- **Two maximum value:** Sum of 2 maximum value will return a rounded off number.
- **Two negative integers:** Sum of 2 negative integer should return a negative integer.
- **One maximum value and a positive integer:** Sum of a negative integer and a minimum value will return a rounded off number.
- **Two maximum value:** Sum of 2 minimum value will return a rounded off number.
- **One positive and one negative integers:** Sum of one positive and one negative integer should return a positive integer if positive integer is greater than negative integer. If negative integer is greater than positive integer, then a negative integer will be returned.
- **Input values 1 and 0:** Sum of 1 and 0 should return 1.
- **Input values 0 and 0:** Sum from two 0's should return 0.

### Conclusion:

Since all the above cases are handled, this implementation can be accepted.

### **Method: String despacer(String inputText)**

#### **Functionality:**

This method replaces multiple contiguous spaces in a text string with a single space.

#### **Possible input values:**

A string value is given as input. The possible input values are multiple contiguous spaces in the beginning of the string, at the end, in between. A string with no multiple contiguous spaces, string with no spaces and an empty string can also be given as an input.

#### **Expected Output:**

- **Multiple contiguous spaces in the beginning of the string:** Should remove the multiple contiguous spaces from the beginning of the string and return a normal string.
- **Multiple contiguous spaces at the end of the string:** Should remove the multiple contiguous spaces from the end of the string and return a normal string.
- **Multiple contiguous spaces in between the string:** Should remove the multiple contiguous spaces from the string and return a normal string.
- **String with no multiple contiguous spaces:** Should return the string as is.
- **String with no spaces:** Should return the string as is.
- **An empty string with multiple contiguous spaces:** Should return an empty string with one space.

#### **Conclusion:**

Since all the above cases are handled, this implementation can be accepted.