UnitTestingHomework Testing

The following explanation gives you an idea of what I believe should be checked and why it makes the implementation should be accepted if it meets the conditions given in the test cases.

double sqrt (int n)

To check the implementation of this function we need to check for the following
Boundary conditions -

0 ,1

Max_integer

We do not consider neg numbers or an input greater than max_integer and also the max digits allowed after decimal is two. For which we use an error allowed value supported by junit.

When We are checking for this.
there are two possibilites

-> there can be an exception

-> there is some standard code which is returned for all neg integers.

We check all this in our test suits. Incase both these don't happen then the implementation is invalid. Either the exception should pass or a standard value should be returned else the implementation is not valid.

int sqr (int n)

To check the implementation of this function we need to check for the following
Boundary conditions -

0 ,1, -1

46340 , -46340 as after these you will get a overflow

We also check for condition when Neg and Pos of the same number is given as input. It should return the same value.

And value that exceed the boundary max and min will over flow

When We are checking for this.
there are two possibilites

-> there can be an exception

-> there is some standard code which is returned for all invalid integers.

We check all this in our test suits. Incase both these don't happen then the implementation is invalid.Either the exception should pass or a standard value should be returned else the implementation is not valid.

int factorial (int n);

Boundary

0,1,2

12 -> after which it will overflow
There shouldn't be any neg value and the there shouldn't be any value which exceeds the boundary of 12.

When We are checking for this.
there are two possibilites

-> there can be an exception

-> there is some standard code which is returned for all invalid integers.

We check all this in our test suits. Incase both these don't happen then the implementation is invalid.Either the exception should pass or a standard value should be returned else the implementation is not valid.

int sumUp(int n)

Boundary

0

65535 after which there will be an overflow.

There shouldn't be any neg value and the there shouldn't be any value which exceeds the boundary of 65535 .

When We are checking for this.
there are two possibilites

-> there can be an exception

-> there is some standard code which is returned for all invalid integers.

We check all this in our test suits. Incase both these don't happen then the implementation is invalid.Either the exception should pass or a standard value should be returned else the implementation is not valid.

int simpleFunctionXplusY(int x, int y)

Boundary

MAX_INTEGER

MIN_INTEGER

Anyvalue of x and y which will add upto Greater than MAX_INTERGER or MIN_INTEGER

As there is no standard code that can be possibly returned. We only check for exceptions when there is an invalid input given (which cause overflow).

We check all this in our test suits. Incase these don’t happen then the implementation is invalid.

String despacer(String inputText)

We pass various string inputs to check if the returned value is what is expected.
We check what happens when there is an empty string

We check when there are only spaces given as an input

What happens when there is a null given.

In this case of null it can be handled in multiple ways. And exception/ a standard value or just return we check for both. One of the test should pass . Else it will not be excepted.

We also check for tabs.