# EECS 4313 – Homework 1 (Junit 8.4.1)

## Task 1:

* For normal boundary value testing, I am using the test reduction case by assuming single fault assumption.
* This is to reduce the number of test cases from 125 cases to 13 cases.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | Selected Inputs | Description of the test case | Actual Output | Expected Output | Passed? |
| isValid\_Test3Min | Side 1: 1071934583  Side 2: 1071934583  Side 3: 1 | Testing if isValid will return true if:  (Nom, Nom, Min)  Side 1 and Side 2 have a **normal value** between a minimum of 1 and maximum of Integer.MAX\_VALUE.  Side 3 has a value at its minimum value. | true | true | Yes |
| isValid\_Test3MinPlus | Side 1: 1071934583  Side 2: 1071934583  Side 3: 2 | Testing if isValid will return true if:  (Nom, Nom, Min+)  Side 1 and Side 2 have a **normal value** between a minimum of 1 and maximum of Integer.MAX\_VALUE.  Side 3 is one above the minimum value. | true | true | Yes |
| isValid\_Test3Norm | Side 1: 1071934583  Side 2: 1071934583  Side 3: 1071934583 | Testing if isValid will return true if:  (Nom, Nom, Nom)  Side 1, Side 2 and Side 3 have a **normal value** between a minimum of 1 and maximum of Integer.MAX\_VALUE. | true | true | Yes |
| isValid\_Test3MaxMinus | Side 1: 1071934583  Side 2: 1071934583  Side 3: Integer.MAX\_VALUE - 1 | Testing if isValid will return false if:  (Nom, Nom, Max-)  Side 1 and Side 2 have a **normal value** between a minimum of 1 and maximum of Integer.MAX\_VALUE.  Side 3 is one below maximum value. | false | false | Yes |
| isValid\_Test3Max | Side 1: 1071934583  Side 2: 1071934583  Side 3: Integer.MAX\_VALUE | Testing if isValid will return false if:  (Nom, Nom, Max)  Side 1 and Side 2 have a **normal value** between a minimum of 1 and maximum of Integer.MAX\_VALUE.  Side 3 has the highest possible value. | false | false | Yes |
| isValid\_Test2Min | Side 1: 1071934583  Side 2: 1  Side 3: 1071934583 | Testing if isValid will return false if:  (Nom, Min, Nom)  Side 1 and Side 3 have a **normal value** between a minimum of 1 and maximum of Integer.MAX\_VALUE.  Side 2 is at the minimum value. | true | true | Yes |
| isValid\_Test2MinPlus | Side 1: 1071934583  Side 2: 2  Side 3: 1071934583 | Testing if isValid will return false if:  (Nom, Min+, Nom)  Side 1 and Side 3 have a **normal value** between a minimum of 1 and maximum of Integer.MAX\_VALUE.  Side 2 is one above the minimum value. | true | true | Yes |
| isValid\_Test2MaxMinus | Side 1: 1071934583  Side 2: Integer.MAX\_VALUE - 1  Side 3: 1071934583 | Testing if isValid will return false if:  (Nom, Max-, Nom)  Side 1 and Side 3 have a **normal value** between a minimum of 1 and maximum of Integer.MAX\_VALUE.  Side 2 is one below the maximum value. | false | false | Yes |
| isValid\_Test2Max | Side 1: 1071934583  Side 2: Integer.MAX\_VALUE  Side 3: 1071934583 | Testing if isValid will return false if:  (Nom, Max, Nom)  Side 1 and Side 3 have a **normal value** between a minimum of 1 and maximum of Integer.MAX\_VALUE.  Side 2 is at maximum value. | false | false | Yes |
| isValid\_Test1min | Side 1: 1  Side 2: 1071934583  Side 3: 1071934583 | Checking response for isValid when the following inputs for (Side 1, Side 2, Side 3) are  (Min, Norm, Norm)  Side 2 and 3 have a **normal value** between a minimum of 1 and maximum of Integer.MAX\_VALUE.  Side 1 is the minimum value. | true | true | Yes |
| isValid\_Test1minPlus | Side 1: 2  Side 2: 1071934583  Side 3: 1071934583 | Checking response for isValid when the following inputs for (Side 1, Side 2, Side 3) are  (Min+, Norm, Norm)  Side 2 and 3 have a **normal value** between a minimum of 1 and maximum of Integer.MAX\_VALUE.  Side 1 is one above the minimum value. | true | true | Yes |
| isValid\_Test1MaxMinus | Side 1: Integer.MAX\_VALUE - 1  Side 2: 1071934583  Side 3: 1071934583 | Checking response for isValid when the following inputs for (Side 1, Side 2, Side 3) are  (Max-, Norm, Norm)  Side 2 and 3 have a **normal value** between a minimum of 1 and maximum of Integer.MAX\_VALUE.  Side 1 is one below the maximum value. | false | false | Yes |
| isValid\_Test1Max | Side 1: Integer.MAX\_VALUE  Side 2: 1071934583  Side 3: 1071934583 | Checking response for isValid when the following inputs for (Side 1, Side 2, Side 3) are  (Max, Norm, Norm)  Side 2 and 3 have a **normal value** between a minimum of 1 and maximum of Integer.MAX\_VALUE.  Side 1 is at its maximum value. | false | false | Yes |

## Task 2:

* Write all possible sets of test cases that tests isSideValid using Robust Boundary Value testing.
* “Keep min and max unchanged and apply RBVT on the first input only. (e.g., min=10, max = 20).”
  + My assumption based on the instructions above is to keep the min = 2 and max = 19 since those two values are the overall minimums and maximum possible values that any side of the triangle can take.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | Selected Inputs | Description of the test case | Actual Output | Expected Output | Passed? |
| isSideValid\_TestMinMinus | Side: 0  Min: 1  Max: 2071934583 | Testing if isSideValue will return “false” boolean value if the side value is below the minimum value allowed. | false | false | Yes |
| isSideValid\_TestMin | Side: 1  Min: 1  Max: 2071934583 | Testing if isSideValue will return “true” boolean value if the side value is the minimum value allowed. | true | true | Yes |
| isSideValid\_TestMinPlus | Side: 2  Min: 1  Max: 2071934583 | Testing if isSideValue will return “true” boolean value if the side value is the one above minimum value allowed. | true | true | Yes |
| isSideValid\_TestNorm | Side: 1071934583  Min: 1  Max: 2071934583 | Testing if isSideValue will return “true” boolean value if the side value is in between the minimum and maximum value allowed. | true | true | Yes |
| isSideValid\_TestMaxMinus | Side: 2071934582  Min: 1  Max: 2071934583 | Testing if isSideValue will return “true” boolean value if the side value is in one less than the maximum value allowed for side 1 (9). | true | true | Yes |
| isSideValid\_TestMax | Side: 2071934583  Min: 1  Max: 2071934583 | Testing if isSideValue will return “true” boolean value if the side value is the maximum value allowed. | true | true | Yes |
| isSideValid\_TestMaxPlus | Side: 2071934584  Min: 1  Max: 2071934583 | Testing if isSideValue will return “true” boolean value if the side value is above the maximum value allowed. | false | false | Yes |

## Task 2:

* “Check if the code works as expected if min>max, min<Integer.MIN\_VALUE, or max>Integer.MAX\_VALUE.”

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Selected Inputs | Description of the test case | Actual Output | Expected Output | Passed? |
| isSideValid\_SideMinOverMax | Side: 5  Min: 10  Max: 4 | Testing if the isSideValue will return “true” boolean value if the side value is in between the minimum and maximum value but minimum’s value is greater than maximum’s value. | false | false | Yes  Reason why this will always return false is because of how isSideValue is coded where it only compares if side >=min and side <= max without checking for the scenario if min > max or even the case where max < min. |
| isSideValid\_SideMinUnderInteger | Side: -1  Min: Integer.MIN\_VALUE – 1  Max: 10 | Testing if isSideValue will return “true” if the side value is in between the minimum and maximum value, when the min < Integer.MIN\_VALUE. | false | false | Yes.  Integer.MIN\_VALUE – 1 causes an overflow in the memory and thus Integer.MIN\_VALUE – 1 is equivalent to Integer.MAX\_VALUE.  Thus, results in a false return no matter what the value of the side. |
| isSideValid\_SideMaxOverInteger | Side: 1000  Min: 1  Max: Integer.MAX\_VALUE + 1 | Testing if isSideValue will return “true” if the side value is in between the minimum and maximum value, when the max > Integer.MIN\_VALUE | false | false | Yes.  Integer.MAX\_VALUE + 1 causes an overflow in the memory and thus Integer.MAX\_VALUE + 1 is equaivalent to Integer.MIN\_VALUE.  Thus, results in a false return no matter what the value of the side. |