Test cases for testing the calculator input field

1. Enter -1 in the input field. ([ Min-1]: Invalid Boundary Check)
2. Enter 0 in the input field. ([ Min]: Valid Boundary Check)
3. Enter 1 in the input field. ([ Min+1]: Valid Boundary Check)
4. Enter 99999 in the input field. ([ Max]: Valid Boundary Check)
5. Enter 99998 in the input field. ([ Max-1]: Valid Boundary Check)
6. Enter 100000 in the input field. ([ Max+1]: Invalid Boundary Check)

|  |  |  |  |
| --- | --- | --- | --- |
| Invalid (min.-1) | Valid (min, min+, max, max-) | Invalid (max.+1) | Boundary Testing |
| -1 | 0, 1, 99999,99998 | 100000 |

|  |  |  |  |
| --- | --- | --- | --- |
| Invalid | Valid | Invalid | Equivalence Partitioning Testing |
| -1- -**∞** | 0-99999 | 100000 - +**∞** |

The technique of equivalence partitioning is used to reduce the number of tests, while maintaining an acceptable test coverage. It is very important to check the boundary values, because quite often errors occur precisely at the boundaries of equivalence classes. At each limit of the range, needs to be checked 3 values: the limit value, the value before and after the limit.

General Test Cases

1. Add two positive integers.
2. Add two negative integers.
3. Subtract two positive integers.
4. Subtract two negative integers.
5. Add one negative integer and one positive integer.
6. Check that the sum of the digits is displayed correctly.
7. Enter a decimal.
8. Check that the amount changes while entering digits.
9. Check by pressing C button all the digits are cleared.
10. Check that the user is able to clear a single digit by backspace.
11. Check the input field by simultaneously pressing 2 digits.