# Homework: Test Levels and Test Types

## Unit Testing in the Real Life: Testing a Battery

|  |  |
| --- | --- |
| **Test #1** | Take a **bulb 1.5V** and check if the battery is works as expected: the bulb should light up after connection properly |
| **Test #2** | The **multimeter** and check the voltage. It should be ~ 1.5V |
| **Test #3** | Take the battery and check it **visually**   * Check **length** * Check its **diameter** * Check if it has a form of **cylinder** * Check for **corrosion**, **leaks**, et. |
| **Test #4** | Check with a compatible **flashlight**. This will check two things.   * Whether battery size matches the flashlight * Whether the batteries work as expected (light the bulb) |
| **Test #5** | Check the **labels** on the battery.   * The denoted size should be “AA”. * The denoted voltage should “1.5V” |
| **Test #6** | Check if “**+**” and “**-**” are correctly positioned. Use a multimeter. |
| **Test #7** | Environmental test:   * Low temperature, e.g. 2 degrees Celsius * High temperature, e.g. 35 degree Celsius |
| **Test #8** | Check the **expiration date label**. It should be in the future |
| **Test #9** | Overheating test. |

## Unit Testing in the Real Life: Testing a Light Bulb

|  |  |
| --- | --- |
| **Test #1** | **Continuity Test**: Measure the **resistance** of the light bulb using a multimeter to check if it is intact or if it has a broken filament. |
| **Test #2** | The **multimeter** and check the voltage. It should be ~ 1.5V |
| **Test #3** | Take the battery and check it **visually**   * Check **length** * Check **height** * Check its **diameter** * Check for **physical damage** or obvious **defects**. |
| **Test #4** | Check the **labels** on the battery.   * The denoted size should be “E10”. * The denoted voltage should “1.5V” |
| **Test #5** | Environmental test:   * Low temperature, e.g. 2 degrees Celsius * High temperature, e.g. 35 degree Celsius |
| **Test #6** | Overheating test. |
| **Test #7** | Power Test: **Connect** the light bulb to a **1.5V power source** and observe if it illuminates. |
| **Test #8** | Check if “**+**” and “**-**” are correctly positioned. Use a multimeter |
| **Test #9** | Check with a compatible **flashlight**. This will check two things.   * Whether bulb size matches the flashlight * Whether the bulb work as expected (light the bulb) |
| **Test #10** | Check the **expiration date label**. It should be in the future |

## Unit Testing in the Software World: Age Checker

|  |  |
| --- | --- |
| **Test #1** | AgeChecker(0) 🡪 child |
| **Test #2** | Age Checker(5) -> child |
| **Test #3** | Age Checker(12.99) -> child |
| **Test #4** | AgeChecker(13) 🡪 teenager |
| **Test #5** | AgeChecker(19.5) 🡪 teenager |
| **Test #6** | AgeChecker(20) 🡪 adult |
| **Test #7** | AgeChecker(21) 🡪 adult |
| **Test #8** | AgeChecker(50) 🡪 adult |
| **Test #9** | AgeChecker(64.7) 🡪 adult |
| **Test #10** | AgeChecker(65) 🡪 elder |
| **Test #11** | AgeChecker(75.3) 🡪 elder |
| **Test #12** | AgeChecker(150) 🡪 elder |
| **Test #13** | AgeChecker(150.1) 🡪 error |
| **Test #14** | AgeChecker(-5) 🡪 error |
| **Test #15** | AgeChecker(-1) 🡪 error |
| **Test #16** | AgeChecker(12800) 🡪 Error |
| **Test #17** | AgeChecker(“Peter”) 🡪 Error |

## Unit Testing in the Software World: Income Checker

|  |  |
| --- | --- |
| **Test #1** | IncomeChecker(0) 🡪 low |
| **Test #2** | IncomeChecker(250) 🡪 low |
| **Test #3** | IncomeChecker(999.99) 🡪 low |
| **Test #4** | AgeChecker(1000) 🡪 mid |
| **Test #5** | AgeChecker(2300.70) 🡪 mid |
| **Test #6** | AgeChecker(2999.99) 🡪 mid |
| **Test #7** | AgeChecker(3000) 🡪 high |
| **Test #8** | AgeChecker(7000) 🡪 high |
| **Test #9** | AgeChecker(-5) 🡪 error |
| **Test #10** | AgeChecker(-1) 🡪 error |
| **Test #11** | AgeChecker(“Peter”) 🡪 error |

## Integration Testing in the Real Life: Lighting the Bulb

|  |  |
| --- | --- |
| **Test #1** | Implement the following circuit, using the provided components.  A picture containing shape  Description automatically generated  The bulb should light. |
| **Test #2** | Implement the following circuit, using the provided components.  Diagram  Description automatically generated  Switch on the switch button -> The bulb should light |
| **Test #3** | Implement the following circuit, using the provided components.  Diagram  Description automatically generated  Switch off the switch button -> The bulb should not light. |
| **Test #4** | Implement the following circuit, using the provided components.  But switch the “+” and “-“ of the battery.  A picture containing shape  Description automatically generated  The bulb should light. |
| **Test #5** | Implement the following circuit, using the provided components.  But turn the switch button to 180 degrees.  Diagram  Description automatically generated  Switch on the switch button --> The bulb should light |

## \* Integration Testing in the Software World: Ads

|  |  |
| --- | --- |
| **Test #1** | **Check** if user can browse **ads by categories**. Steps:   1. Choose first category. 2. Check if the ads are from the relevant category. 3. Repeat with other categories   Expect ads from the relevant category. This check if the main section collaborate with category filter section. |
| **Test #2** | **Check** if user can browse **ads by towns**.   1. Choose first town. 2. Check if the ads are from the relevant town. 3. Repeat with other towns.   Expect ads from the relevant town. This check if the main section collaborate with filter town section |
| **Test #3** | **Check** if user can browse **ads by towns and by categories**.   1. Choose first town. 2. Choose first category 3. Check if the ads are from the relevant town. 4. Repeat with other towns and categories.   Expect ads from the relevant town and category. This check if the main section collaborate with both filter section. |
| **Test #4** | Navigation **from Homepage to Login**. The [Login] button should navigate to Login form page. |
| **Test #5** | Navigation **from Login to Homepage**. [Home] button should navigate to Home Page unlogged. The user can`t publish an ads. |
| **Test #6** | Navigation **from Homepage to Register form**. The [Register] button should navigate to Register form page. |
| **Test #7** | **Successful Login**   1. Home page 🡪 Login 2. Provide correct username and password 3. Navigate to User Home Page   The user can publish a new add, to view own published ads and to edit own profile. |
| **Test #8** | **Successful Logout** – after successful login.  Click [Logout] button 🡪 Home Page (proper behavior – unlogged, without options for publish a new add) |
| **Test #9** | **Unsuccessful Login 🡪 need registration**   1. Home page 🡪 Login 2. Provide incorrect username and password 3. Navigate to Registration form.   The user should register to use the app. |

## \* Integration Testing in the Software World: Credit Risk

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test** | CreditRisk(age: 5, income: 700) 🡪 100%  CreditRisk(age: 6, income: 2000) 🡪 100%  CreditRisk(age: 7, income: 6000) 🡪 100%  CreditRisk(age: 15, income: 700) 🡪 80%  CreditRisk(age: 16, income: 2000) 🡪 72%  CreditRisk(age: 17, income: 6000) 🡪 64%  CreditRisk(age: 21, income: 700) 🡪 55%  CreditRisk(age: 32, income: 2000) 🡪 37%  CreditRisk(age: 43, income: 6000) 🡪 19%  CreditRisk(age: 67, income: 700) 🡪 60%  CreditRisk(age: 78, income: 2000) 🡪 44%  CreditRisk(age: 89, income: 6000) 🡪 28%  Additional tests for invalid input:   * CreditRisk(age: -9, income: 9000) 🡪 error * CreditRisk(age: 19, income: -1000) 🡪 error   Regression test:   * CreditRisk(age: 17, income: 0) 🡪 80%      * CreditRisk(age: 0, income: 1000) 🡪 80% | | | | |
|  | child | teenager | adult | elder | negative |
| low | 100% | 80% | 55% | 60% | Error |
| mid | 100% | 72% | 37% | 44% | Error |
| high | 100% | 64% | 19% | 28% | Error |
| negative | Error | Error | Error | Error | Error |

## System Testing in the Real Life: Flashlight

|  |  |
| --- | --- |
| **Test #1** | **Test switch on / switch off the light**  We take the flashlight. Put new batteries correctly. Switch on the flashlight 🡪 the bulb should light. Switch off the light 🡪 the bulb should light off |
| **Test #2** | **Test battery replacement**   1. Take the flashlight. 2. Open the battery compartment. 3. We take out the old batteries. 4. Put brand new batteries. 5. Close the battery compartment.   Check if light on. |
| **Test #3** | **Test bulb replacement**. Change the bulb. Check if the flashlight flash on. |
| **Test #4** | **Test battery duration**. At least 1 hour of lighting with new batteries. |
| **Test #5** | **Test the illumination distance**. It should illuminate cleanly at distance of 30 meters or less(with new batteries). |
| **Test #6** | **Shock resistance** test |
| **Test #7** | **Operation under high / low temperature** |
| **Test #8** | **Water resistant**  If the flashlight is advertised as waterproof or water-resistant, conduct a water resistance test by submerging the flashlight in water for a specific amount of time. |
| **Test #9** | **Visual Inspection Tests**  Inspect the device for any physical damages like scratches, dents, or broken parts. Check if the lenses or bulbs are in good condition. |
| **Test #10** | **Impact Resistance Test**  Drop the flashlight from a height of about 1-meter and check if it still functions properly. Test the flashlight in a rough environment, such as a construction site or a hiking trail, to check its impact resistance. |

## System Testing in the Real Life: Digital Scale

|  |  |
| --- | --- |
| **Test #1** | **Accuracy test**: Place a known weight on the scale and compare the reading with the actual weight. Repeat this test with different known weights of varying values to ensure the scale is accurately measuring weight. |
| **Test #2** | **Precision test**: Weigh an object multiple times, ideally within a short period, and ensure the scale gives the same reading every time. |
| **Test #3** | **Zero reading test**: Check if the scale correctly reads zero when no weight is placed on it. |
| **Test #4** | **Load capacity test**: Test the scale's maximum weight capacity to ensure it can handle the maximum load specified by the manufacturer. |
| **Test #5** | **Auto-shutoff test**: Test if the scale automatically turns off after a set period of inactivity, as specified by the manufacturer. |
| **Test #6** | **Display test**: Test the display to ensure that it is clear and easy to read, and that all digits and indicators are functioning properly. |
| **Test #7** | **Durability test**: Drop the scale from a low height to check if it still functions properly. |
| **Test #8** | **Temperature test:** Test the scale in different temperatures to see if it affects its accuracy or performance. |
| **Test #9** | **Environmental test:** Test the scale in different environmental conditions to ensure it can withstand normal usage in various environments. |

## System Testing in the Software World: Number Calculator

|  |  |
| --- | --- |
| Test scenario: Sum two numbers | |
| **Test #1** | Calc(5 + 3) 🡪 8  Test Passed |
| **Test #2** | Calc(+5 + 3) 🡪 8  Test Passed |
| **Test #3** | Calc(5 + +3) 🡪 8  Test Passed |
| **Test #4** | Calc(+5 + +3) 🡪 8  Test Passed |
| **Test #5** | Calc(+5 + ++3) 🡪 *invalid input*  Test Passed |
| **Test #6** | Calc(++5 + +3) 🡪 *invalid input*  Test Passed |
| **Test #7** | Calc(++5 + ++3) 🡪 *invalid input*  Test Passed |
| **Test #8** | Calc(5 + -3) 🡪 2  Test Passed |
| **Test #9** | Calc(-5 + -3) 🡪 8  Test Passed |
| **Test #10** | Calc(-5 + 3) 🡪 -2  Test Passed |
| **Test #11** | Calc(-5 + -3) 🡪 2  Test Passed |
| **Test #12** | Calc(5 + \*3) 🡪 *invalid input*  Test Passed |
| **Test #13** | Calc(5 + /3) 🡪 *invalid input*  Test Passed |
| **Test #14** | Calc(5banana + 3) 🡪 8  Test Failed **🡪 Expected result:** *invalid input* |
| **Test #15** | Calc(5 + 3banana) 🡪 8  Test Failed **🡪 Expected result:** *invalid input* |
| **Test #16** | Calc(5 5 + 3) 🡪 8  Test Failed **🡪 Expected result:** *invalid input* |
| **Test #17** | Calc(5.5 + 3) 🡪 8.5  Test Passed |
| **Test #18** | Calc(5,5 + 3) 🡪 8  Test Failed **🡪 Expected result:** *invalid input* |
| **Test #19** | Calc(5 + 3.5) 🡪 8.5  Test Passed |
| **Test #20** | Calc(5 + 3,5) 🡪 8  Test Failed **🡪 Expected result:** *invalid input* |
| **Test #21** | Calc(1e10 + 5) 🡪 10000000005  Test Passed |
| **Test #22** | Calc(10000000000000000000000 + 5) 🡪 1e+22  Test Failed **🡪 Expected result:** 10000000000000000000005  If the number is too long, it been round up incorrect. |
| **Test #23** | Calc(5 + Infinity) 🡪 Infinity  Test Passed |
| **Test #24** | Calc(Infinity + 5) 🡪 Infinity  Test Passed |
| **Test #25** | Calc(Infinity + Infinity) 🡪 Infinity  Test Passed |
| **Test #26** | Calc(Infinity + -Infinity) 🡪 *invalid calculation*  Test Passed |
| **Test #27** | Calc(-Infinity + Infinity) 🡪 *invalid calculation*  Test Passed |
| **Test #28** | Calc(-Infinity + -Infinity) 🡪 -Infinity  Test Passed |
| **Test #29** | Calc(rado + 3) 🡪 *invalid input*  Test Passed |
| **Test #30** | Calc(3 + rado) 🡪 *invalid input*  Test Passed |
| **Test #31** | Calc(rado + rado) 🡪 *invalid input*  Test Passed |
| **Test #32** | Calc(5 + 0) 🡪 5  Test Passed |
| **Test #33** | Calc(0 + 5) 🡪 5  Test Passed |
| **Test #34** | Calc(-0 + 5) 🡪 5  Test Passed |
| **Test #35** | Calc(5 + -0) 🡪 5  Test Passed |
| **Test #36** | Calc(0 + 0) 🡪 0  Test Passed |
| **Test #37** | Calc(-0 + -0) 🡪 0  Test Passed |
| **Test #38** | Calc(0 + -0) 🡪 0  Test Passed |
| **Test #39** | Calc(-0 + 0) 🡪 0  Test Passed |
| Test case: Subtract two numbers | |
| **Test #1** | Calc(5 - 3) 🡪 2  Test Passed |
| **Test #2** | Calc(+5 - 3) 🡪 2  Test Passed |
| **Test #3** | Calc(5 - +3) 🡪 2  Test Passed |
| **Test #4** | Calc(+5 - +3) 🡪 2  Test Passed |
| **Test #5** | Calc(+5 - ++3) 🡪 *invalid input*  Test Passed |
| **Test #6** | Calc(++5 - +3) 🡪 *invalid input*  Test Passed |
| **Test #7** | Calc(++5 - ++3) 🡪 *invalid input*  Test Passed |
| **Test #8** | Calc(5 - -3) 🡪 8  Test Passed |
| **Test #9** | Calc(-5 - -3) 🡪 -2  Test Passed |
| **Test #10** | Calc(-5 - 3) 🡪 -8  Test Passed |
| **Test #11** | Calc(-5 - -3) 🡪 -8  Test Passed |
| **Test #12** | Calc(5 - \*3) 🡪 *invalid input*  Test Passed |
| **Test #13** | Calc(5 - /3) 🡪 *invalid input*  Test Passed |
| **Test #14** | Calc(5banana - 3) 🡪 2  Test Failed **🡪 Expected result:** *invalid input* |
| **Test #15** | Calc(5 - 3banana) 🡪 2  Test Failed **🡪 Expected result:** *invalid input* |
| **Test #16** | Calc(5 5 - 3) 🡪 2  Test Failed **🡪 Expected result:** *invalid input* |
| **Test #17** | Calc(5.5 - 3) 🡪 2.5  Test Passed |
| **Test #18** | Calc(5,5 - 3) 🡪 2  Test Failed **🡪 Expected result:** *invalid input* |
| **Test #19** | Calc(5 - 3.5) 🡪 1.5  Test Passed |
| **Test #20** | Calc(5 - 3,5) 🡪 8  Test Failed **🡪 Expected result:** *invalid input* |
| **Test #21** | Calc(1e10 - 5) 🡪 9999999995  Test Passed |
| **Test #22** | Calc(10000000000000000000000 - 5) 🡪 1e+22  Test Failed **🡪 Expected result:** 10000000000000000000005  If the number is too long, it been round up incorrect. |
| **Test #23** | Calc(5 - Infinity) 🡪 -Infinity  Test Passed |
| **Test #24** | Calc(Infinity - 5) 🡪 Infinity  Test Passed |
| **Test #25** | Calc(Infinity - Infinity) 🡪 *invalid calculation*  Test Passed |
| **Test #26** | Calc(-Infinity - Infinity) 🡪 -Infinity  Test Passed |
| **Test #27** | Calc(-Infinity - Infinity) 🡪 -Infinity  Test Passed |
| **Test #28** | Calc(-Infinity - -Infinity) 🡪 *invalid calculation*  Test Passed |
| **Test #29** | Calc(rado - 3) 🡪 *invalid input*  Test Passed |
| **Test #30** | Calc(3 - rado) 🡪 *invalid input*  Test Passed |
| **Test #31** | Calc(rado - rado) 🡪 *invalid input*  Test Passed |
| **Test #32** | Calc(5 - 0) 🡪 5  Test Passed |
| **Test #33** | Calc(0 - 5) 🡪 -5  Test Passed |
| **Test #34** | Calc(-0 - 5) 🡪 -5  Test Passed |
| **Test #35** | Calc(5 - -0) 🡪 5  Test Passed |
| **Test #36** | Calc(0 - 0) 🡪 0  Test Passed |
| **Test #37** | Calc(-0 - -0) 🡪 0  Test Passed |
| **Test #38** | Calc(0 - -0) 🡪 0  Test Passed |
| **Test #39** | Calc(-0 - 0) 🡪 0  Test Passed |
| Test case: Multiply of two numbers | |
| **Test #1** | Calc(5 \* 3) 🡪 15  Test Passed |
| **Test #2** | Calc(+5 \* 3) 🡪 15  Test Passed |
| **Test #3** | Calc(5 \* +3) 🡪 15  Test Passed |
| **Test #4** | Calc(+5 \* +3) 🡪 15  Test Passed |
| **Test #5** | Calc(+5 \* ++3) 🡪 *invalid input*  Test Passed |
| **Test #6** | Calc(++5 \* +3) 🡪 *invalid input*  Test Passed |
| **Test #7** | Calc(++5 \* ++3) 🡪 *invalid input*  Test Passed |
| **Test #8** | Calc(5 \* -3) 🡪 -15  Test Passed |
| **Test #9** | Calc(-5 \* -3) 🡪 15  Test Passed |
| **Test #10** | Calc(-5 \* 3) 🡪 -15  Test Passed |
| **Test #11** | Calc(-5 \* -3) 🡪 15  Test Passed |
| **Test #12** | Calc(5 \* \*3) 🡪 *invalid input*  Test Passed |
| **Test #13** | Calc(5 \* /3) 🡪 *invalid input*  Test Passed |
| **Test #14** | Calc(5banana \* 3) 🡪 8  Test Failed **🡪 Expected result:** *invalid input* |
| **Test #15** | Calc(5 \* 3banana) 🡪 8  Test Failed **🡪 Expected result:** *invalid input* |
| **Test #16** | Calc(5 5 \* 3) 🡪 8  Test Failed **🡪 Expected result:** *invalid input* |
| **Test #17** | Calc(5.5 \* 3) 🡪 16.5  Test Passed |
| **Test #18** | Calc(5,5 \* 3) 🡪 15  Test Failed **🡪 Expected result:** *invalid input* |
| **Test #19** | Calc(5 \* 3.5) 🡪 15.5  Test Passed |
| **Test #20** | Calc(5 \* 3,5) 🡪 8  Test Failed **🡪 Expected result:** *invalid input* |
| **Test #21** | Calc(1e10 \* 5) 🡪 50000000000  Test Passed |
| **Test #22** | Calc(10000000000000000000000 \* 5) 🡪 5e+22  Test Passed |
| **Test #23** | Calc(5 \* Infinity) 🡪 Infinity  Test Passed |
| **Test #24** | Calc(Infinity \* 5) 🡪 Infinity  Test Passed |
| **Test #25** | Calc(Infinity \* Infinity) 🡪 Infinity  Test Passed |
| **Test #26** | Calc(Infinity \* -Infinity) 🡪 -Infinity  Test Passed |
| **Test #27** | Calc(-Infinity \* Infinity) 🡪 *-*Infinity  Test Passed |
| **Test #28** | Calc(-Infinity \* -Infinity) 🡪 Infinity  Test Passed |
| **Test #29** | Calc(rado \* 3) 🡪 *invalid input*  Test Passed |
| **Test #30** | Calc(3 \* rado) 🡪 *invalid input*  Test Passed |
| **Test #31** | Calc(rado \* rado) 🡪 *invalid input*  Test Passed |
| **Test #32** | Calc(5 \* 0) 🡪 0  Test Passed |
| **Test #33** | Calc(0 \* 5) 🡪 0  Test Passed |
| **Test #34** | Calc(-0 \* 5) 🡪 0  Test Passed |
| **Test #35** | Calc(5 \* -0) 🡪 0  Test Passed |
| **Test #36** | Calc(0 \* 0) 🡪 0  Test Passed |
| **Test #37** | Calc(-0 \* -0) 🡪 0  Test Passed |
| **Test #38** | Calc(0 \* -0) 🡪 0  Test Passed |
| **Test #39** | Calc(-0 \* 0) 🡪 0  Test Passed |
| Test scenario: Divide of two numbers | |
| **Test #1** | Calc(5 / 3) 🡪 1.66666666667  Test Passed |
| **Test #2** | Calc(+5 / 3) 🡪 1.66666666667  Test Passed |
| **Test #3** | Calc(5 / +3) 🡪 1.66666666  Test Passed |
| **Test #4** | Calc(+5 / +3) 🡪 1.66666666  Test Passed |
| **Test #5** | Calc(+5 / ++3) 🡪 *invalid input*  Test Passed |
| **Test #6** | Calc(++5 / +3) 🡪 *invalid input*  Test Passed |
| **Test #7** | Calc(++5 / ++3) 🡪 *invalid input*  Test Passed |
| **Test #8** | Calc(5 / -3) 🡪 -1.66666666  Test Passed |
| **Test #9** | Calc(-5 / -3) 🡪 1.66666666  Test Passed |
| **Test #10** | Calc(-5 / 3) 🡪 -1.66666666  Test Passed |
| **Test #11** | Calc(-5 / -3) 🡪 1.66666666  Test Passed |
| **Test #12** | Calc(5 / \*3) 🡪 *invalid input*  Test Passed |
| **Test #13** | Calc(5 / /3) 🡪 *invalid input*  Test Passed |
| **Test #14** | Calc(5banana / 3) 🡪 8  Test Failed **🡪 Expected result:** *invalid input* |
| **Test #15** | Calc(5 / 3banana) 🡪 8  Test Failed **🡪 Expected result:** *invalid input* |
| **Test #16** | Calc(5 5 / 3) 🡪 8  Test Failed **🡪 Expected result:** *invalid input* |
| **Test #17** | Calc(5.5 / 3) 🡪 1.83333333333  Test Passed |
| **Test #18** | Calc(5,5 / 3) 🡪1.66666666  Test Failed **🡪 Expected result:** *invalid input* |
| **Test #19** | Calc(5 / 3.5) 🡪 1.42857142857  Test Passed |
| **Test #20** | Calc(5 + 3,5) 🡪 1.42857142857  Test Failed **🡪 Expected result:** *invalid input* |
| **Test #21** | Calc(1e10 / 5) 🡪 2000000000  Test Passed |
| **Test #22** | Calc(10000000000000000000000 / 5) 🡪 1e+21  Test Failed **🡪 Expected result:** 10000000000000000000005  If the number is too long, it been round up incorrect. |
| **Test #23** | Calc(5 / Infinity) 🡪 0  Test Passed |
| **Test #24** | Calc(Infinity / 5) 🡪 Infinity  Test Passed |
| **Test #25** | Calc(Infinity / Infinity) 🡪 *invalid calculation*  Test Passed |
| **Test #26** | Calc(Infinity / -Infinity) 🡪 *invalid calculation*  Test Passed |
| **Test #27** | Calc(-Infinity / Infinity) 🡪 *invalid calculation*  Test Passed |
| **Test #28** | Calc(-Infinity / -Infinity) 🡪 *invalid calculation*  Test Passed |
| **Test #29** | Calc(rado / 3) 🡪 *invalid input*  Test Passed |
| **Test #30** | Calc(3 / rado) 🡪 *invalid input*  Test Passed |
| **Test #31** | Calc(rado / rado) 🡪 *invalid input*  Test Passed |
| **Test #32** | Calc(5 / 0) 🡪 Infinity  Test Passed |
| **Test #33** | Calc(0 / 5) 🡪 0  Test Passed |
| **Test #34** | Calc(-0 / 5) 🡪 0  Test Passed |
| **Test #35** | Calc(5 / -0) 🡪 Infinity  Test Passed |
| **Test #36** | Calc(0 / 0) 🡪 *invalid input*  Test Passed |
| **Test #37** | Calc(-0 / -0) 🡪 *invalid input*  Test Passed |
| **Test #38** | Calc(0 / -0) 🡪 *invalid input*  Test Passed |
| **Test #39** | Calc(-0 / 0) 🡪 *invalid input*  Test Passed |
| Test scenario: UX/UI design | |
| **Test #1** | Not responsive. |
| **Test #2** | Bad ui/ux. |
| Test scenario: Other tests | |
| **Test #1** | Calc without set operation. 🡪 *invalid input*  Test Passed |

## Acceptance Testing in the Real Life: Flashlight

|  |  |
| --- | --- |
| **Test #1** | The customer takes the flashlight, **switch on / off** the light, and assure it works. |
| **Test #2** | The customer checks the flash **illumination**. |
| **Test #3** | The customer checks how easy it is to **replace the batteries**. |
| **Test #4** | The customer checks the **brightness**. |
| **Test #5** | The customer checks if the drop the flashlight from a low height to ensure it is **durable** and can withstand normal usage. |
| **Test #6** | The customer checks the if the flashlight is **water resistance.** |

## Acceptance Testing in the Real Life: Digital Scale

|  |  |
| --- | --- |
| **Test #1** | The customer checks how easy it is to **replace the batteries**. |
| **Test #2** | The customer ensure the scale is easy to use and that the **display is easy to read**. |
| **Test #3** | The customer checks his weigh multiple times and compare the readings to ensure the scale is **accurately measuring** his weight. |
| **Test #4** | The customer check if the scale correctly reads zero when no weight is placed on it. |
| **Test #5** | The customer test the scale's maximum weight capacity to ensure it can handle your weight. |
| **Test #6** | The customer check **auto-shutoff test:** Test if the scale automatically turns off after a set period of inactivity, as specified by the manufacturer. |

## Acceptance Testing in the Software World: Number Calculator

|  |  |
| --- | --- |
| **Test #1** | The customer checks how easy it is to **use the calculator**. |
| **Test #2** | The customer checks the **usability** of the calculator |
| **Test #3** | The customer checks the buttons. |

## Functional and Non-Functional Tests: Flashlight

|  |  |
| --- | --- |
| **Functional Tests** | **Non-Functional Tests** |
| On/Off Switch Test: Test the flashlight's on/off switch to ensure that it is functional and that the flashlight turns on and off as expected. | Size and Weight Test: Test the flashlight's size and weight to ensure that it is comfortable to hold and that it is suitable for your needs. |
| Brightness Test: Test the flashlight's brightness settings to ensure that they are functional and that the flashlight can be adjusted to different levels of brightness. | Water Resistance Test: Test the flashlight's water resistance to ensure that it functions properly in wet conditions |
| Beam Focus Test: Test the flashlight's beam focus to ensure that it can be adjusted to different levels and that it provides a clear and focused beam. | Shock Resistance Test: Test the flashlight's shock resistance to ensure that it can withstand accidental drops or impacts. |
| Battery Test: Test the flashlight's battery life and performance to ensure that it lasts as long as advertised and that the flashlight is functional when the batteries are low. | Battery Compartment Test: Test the battery compartment to ensure that it is easy to open and that the batteries fit securely |
| Durability Test: Test the flashlight's durability by dropping it from a low height to ensure that it can withstand normal usage. | Material Quality Test: Test the quality of the flashlight's materials to ensure that they are durable and can withstand normal usage |