

A. How many mutants are there in total? (5%)

106 Mutants

B. What mutation score were you able to achieve in Lab1' s test suite? (5%)

a. You don' t need to reimplement your test in lab1. It' s okay to just paste screenshot of StrykerJS's result.

Ran 1.00 tests per mutant on average.						
File	% score	# killed	# timeout	# survived	# no cov	# errors
All files	88.24	30	0	4	0	0
calculator.js	88.24	30	0	4	0	0

(Note: 方便起見直接把 HW4 的兩個檔案改成 lab1 的內容，所以上面檔名才會是 calculator.js)

C. How do you kill those mutants? (please paste all of your test code here) (30%)

```
const assert = require('assert');

const { test } = require('node:test');

const Calculator = require('../src/calculator');

// TODO: write your test cases here to kill mutants

test("Test Calculator's cal", () => {

    assert.strictEqual(62, Calculator.main(1, 2, 3, 4, 2012));

    assert.strictEqual(0, Calculator.main(1, 1, 1, 1, 2012));

    assert.strictEqual(0, Calculator.main(12, 1, 12, 1, 10000));

    assert.strictEqual(88, Calculator.main(1, 2, 3, 31, 2023));

    assert.strictEqual(60, Calculator.main(1, 31, 3, 31, 2000));

    assert.strictEqual(57, Calculator.main(1, 4, 3, 2, 1));

    assert.strictEqual(28, Calculator.main(2, 1, 3, 1, 100));

});
```

```

const testcases = [

    { param: [1, 31, 3, 31, -1], expected: 'invalid year'},

    { param: [1, 31, 3, 31, 10001], expected: 'invalid year'},

    { param: [13, 31, 13, 31, 2012], expected: 'invalid month1'},

    { param: [0, 31, 13, 31, 2012], expected: 'invalid month1'},

    { param: [1, 31, 13, 31, 2012], expected: 'invalid month2'},

    { param: [1, 0, 0, 31, 2012], expected: 'invalid month2'},

    { param: [1, 32, 1, 3, 2012], expected: 'invalid day1'},

    { param: [1, 0, 1, 3, 2012], expected: 'invalid day1'},

    { param: [1, 31, 1, 32, 2012], expected: 'invalid day2'},

    { param: [1, 31, 2, 0, 2012], expected: 'invalid day2'},

    { param: [5, 2, 3, 4, 2012], expected: 'month1 must be less than
month2'},

    { param: [1, 31, 1, 32, 2012], expected: 'invalid day2'},

    { param: [1, 31, 1, 29, 2012], expected: 'day1 must be less than day2 if
month1 is equal to month2'},

]

for (const tc of testcases) {

    assert.throws(() => {

        Calculator.main.apply(this, tc.param)

    }, {

```

```
        name : 'Error',

        message: tc.expected

    });

}

});
```

D. What mutation score were you able to achieve? Please paste screenshot of StrykerJS's result (40%)

a. $\text{score} = 40 * (S - L) / (H - L)$

```
All tests
✓ All tests (killed 105)

Ran 0.99 tests per mutant on average.

File      % score  # killed  # timeout  # survived  # no cov  # errors
-----
All files  100.00   105      1          0           0         0
calculator.js 100.00   105      1          0           0         0

21:57:18 (15924) INFO HtmlReporter Your report can be found at: file:///C:/Users/Cherry/Desktop/Course/SoftwareTesting/12-spring-software-testing/hw4/reports/mutation/mutation.html
21:57:18 (15924) INFO MutationTestExecutor Done in 36 seconds.
```

E. Can your test achieve 100% mutation score? Why not if you can't? (20%)

我的方法是先觀察每次執行完的結果中，所列出 survived 的 mutation，接著開始想 test case 要怎麼設計，寫完之後重新執行。不斷重複這樣的動作，一直到 mutation score 達到 100% 為止。有時候一個 test case 剛好能 kill 好幾個 mutation，所以不必針對這 106 個 mutation 都各自撰寫 test case。