4. Code Coverage

學習目標

練習 code coverage 工具

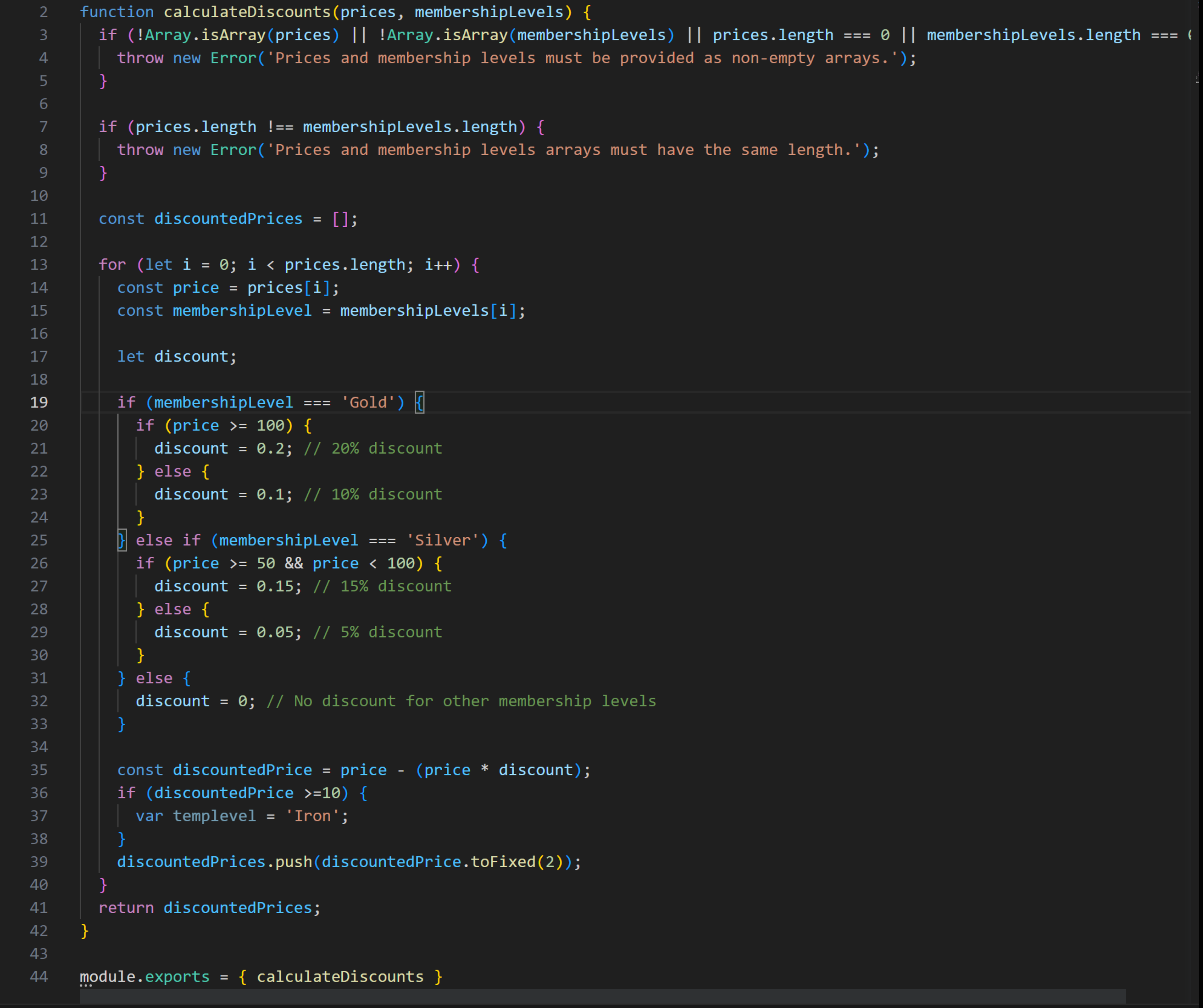
理解 statement/branch coverage criteria

學習測試理論 partition testing 來達成 coverage

Lab 4.1 : 在這個例子中:

1. 我們提供一個 CalculateDiscount.js

2. 我們也提供了一個測試CalculateDiscount.test.js



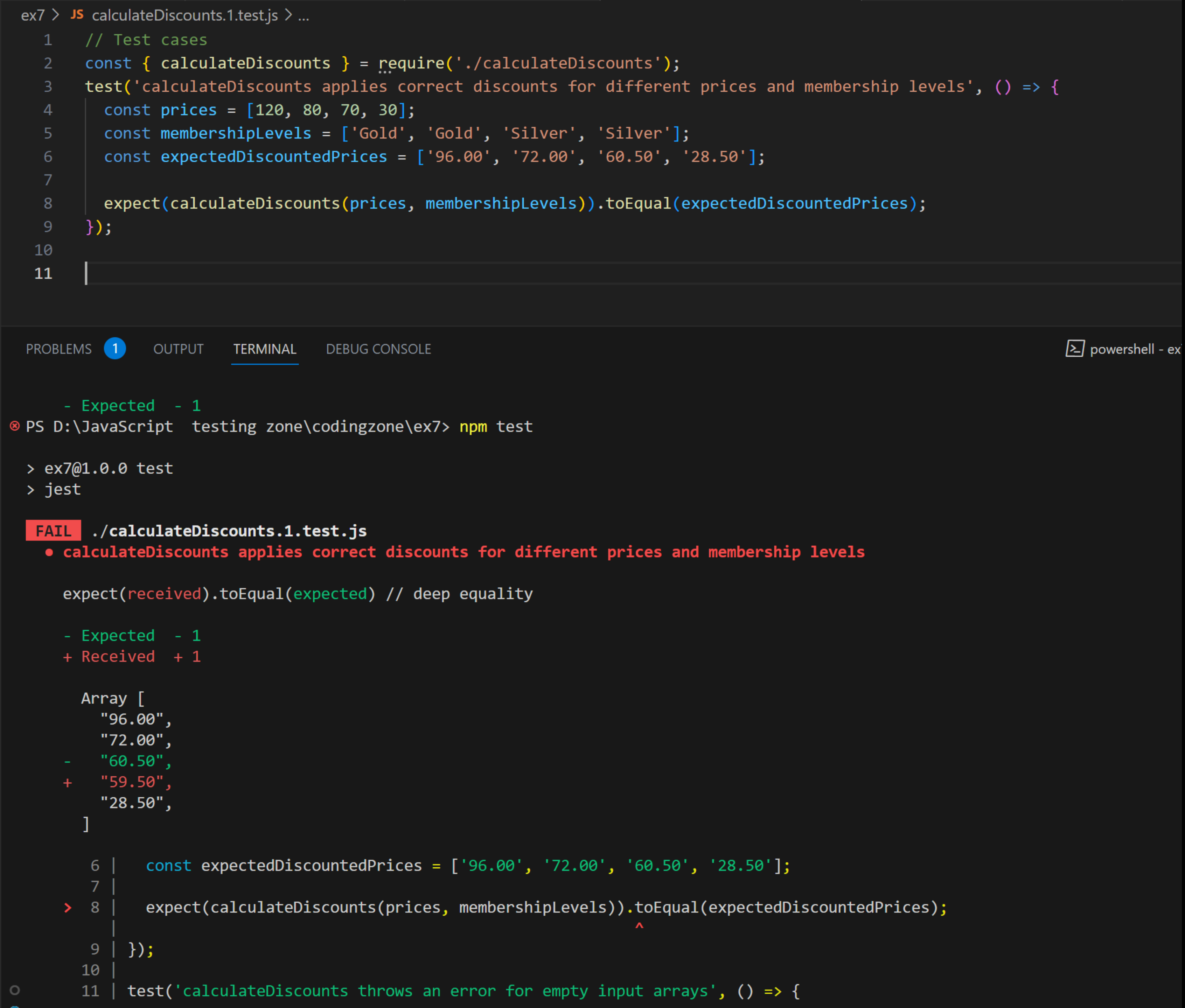
```
function calculateDiscounts(prices, membershipLevels) {
  if (!Array.isArray(prices) || !Array.isArray(membershipLevels) || prices.length === 0 || membershipLevels.length === 0
    throw new Error('Prices and membership levels must be provided as non-empty arrays.');
  }

  if (prices.length !== membershipLevels.length) {
    throw new Error('Prices and membership levels arrays must have the same length.');
  }

  const discountedPrices = [];

  for (let i = 0; i < prices.length; i++) {
    const price = prices[i];
    const membershipLevel = membershipLevels[i];

    let discount;

    if (membershipLevel === 'Gold') {
      if (price >= 100) {
        discount = 0.2; // 20% discount
      } else {
        discount = 0.1; // 10% discount
      }
    } else if (membershipLevel === 'Silver') {
      if (price >= 50 && price < 100) {
        discount = 0.15; // 15% discount
      } else {
        discount = 0.05; // 5% discount
      }
    } else {
      discount = 0; // No discount for other membership levels
    }

    const discountedPrice = price - (price * discount);
    if (discountedPrice >=10) {
      var templevel = 'Iron';
    }
    discountedPrices.push(discountedPrice.toFixed(2));
  }
  return discountedPrices;
}

module.exports = { calculateDiscounts }
```

說明

1. Prices 是一個整數的 array
2. membershipLevels 是一個會員等級(字串)的 array。會員等級可以是 ‘Gold’, ‘Silver’, ‘Bronze’
3. 會員等級越高，折扣越高
4. 最後回傳折扣後的價錢

4.1 TASK TO DO:

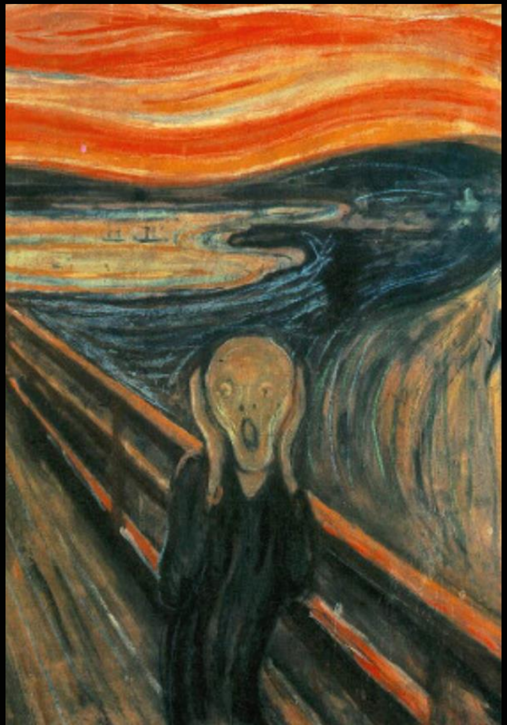
1. 請看懂 CalculateDiscount.js
2. 請執行 CalculateDiscount.test.js
3. 測試不會過，請修正讓測試可以過

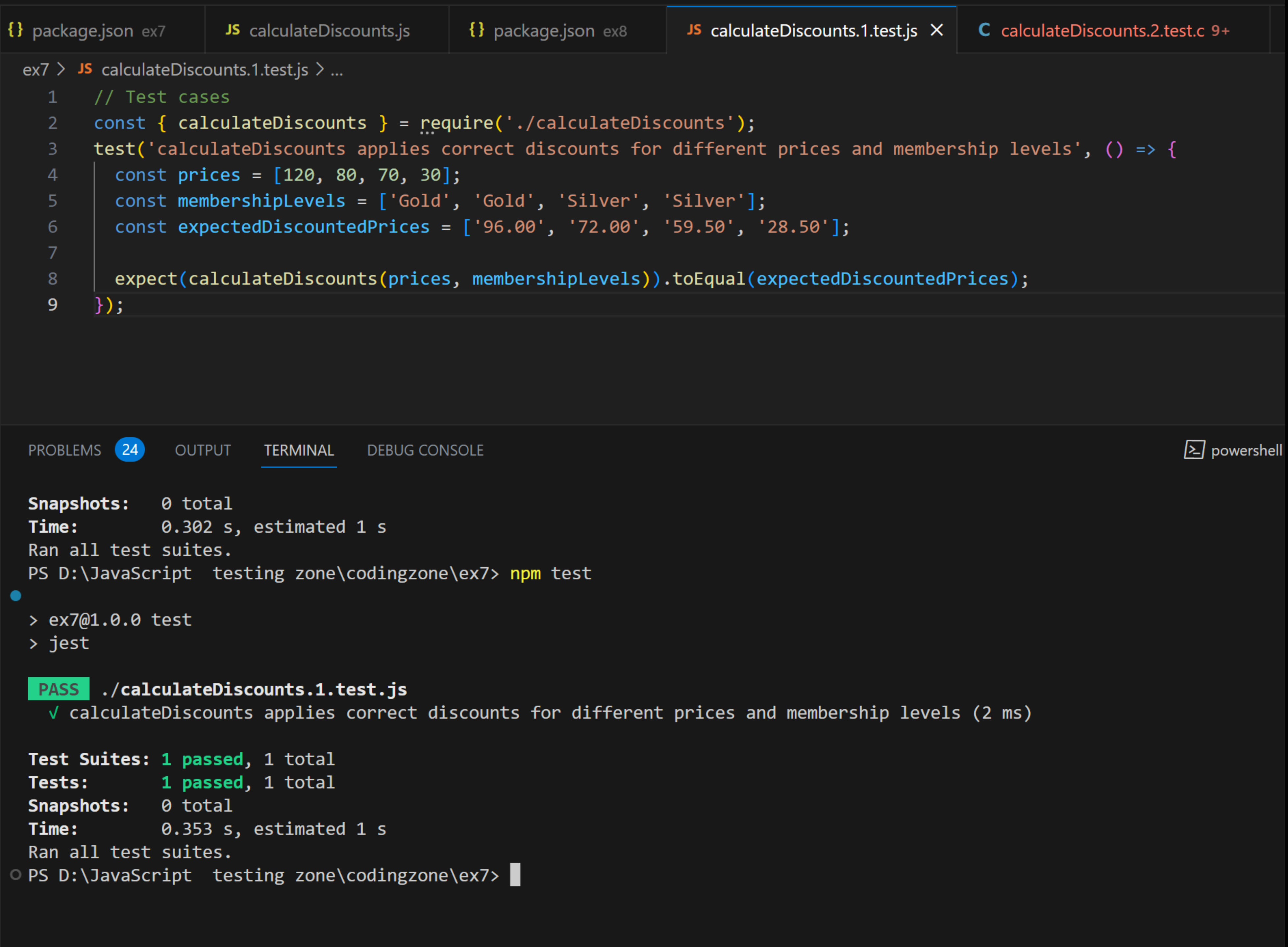


```javascript
// Test cases
const { calculateDiscounts } = require('./calculateDiscounts');
test('calculateDiscounts applies correct discounts for different prices and membership levels', () => {
  const prices = [120, 80, 70, 30];
  const membershipLevels = ['Gold', 'Gold', 'Silver', 'Silver'];
  const expectedDiscountedPrices = ['96.00', '72.00', '60.50', '28.50'];

  expect(calculateDiscounts(prices, membershipLevels)).toEqual(expectedDiscountedPrices);
});

```

```
    - Expected  - 1
⊗ PS D:\JavaScript  testing zone\codingzone\ex7> npm test

  > ex7@1.0.0 test
  > jest

  FAIL  ./calculateDiscounts.1.test.js
    ● calculateDiscounts applies correct discounts for different prices and membership levels

      expect(received).toEqual(expected) // deep equality

      - Expected  - 1
      + Received  + 1

        Array [
          "96.00",
          "72.00",
      -   "60.50",
      +   "59.50",
          "28.50",
        ]

         6 |   const expectedDiscountedPrices = ['96.00', '72.00', '60.50', '28.50'];
         7 |
      >  8 |   expect(calculateDiscounts(prices, membershipLevels)).toEqual(expectedDiscountedPrices);
           |                                                        ^
         9 | });
        10 |
        11 | test('calculateDiscounts throws an error for empty input arrays', () => {
```

有時候測試沒有過，不見得是你的程式錯了。有時候是你準備的正確答案錯了





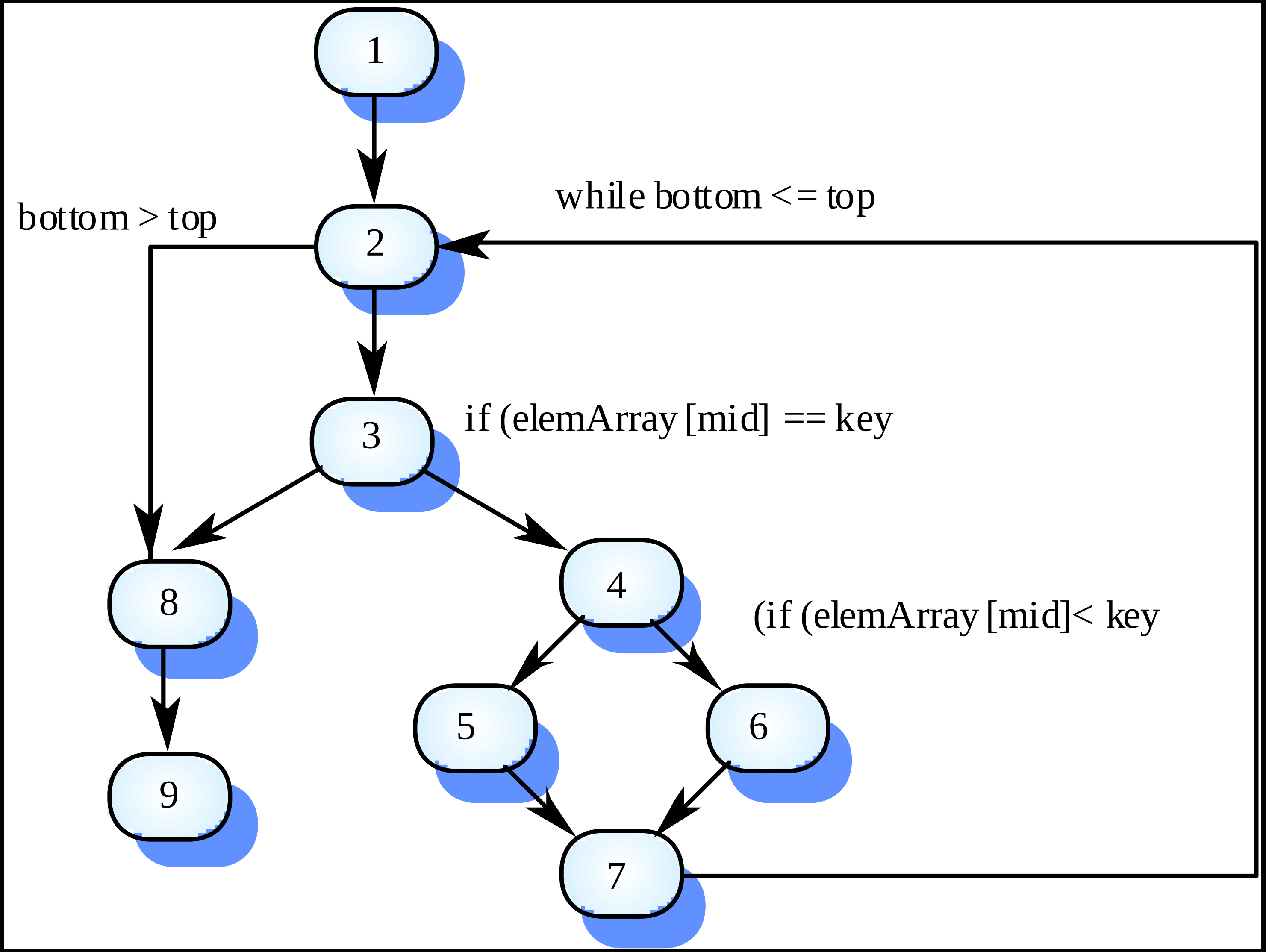
```javascript
// Test cases
const { calculateDiscounts } = require('./calculateDiscounts');
test('calculateDiscounts applies correct discounts for different prices and membership levels', () => {
  const prices = [120, 80, 70, 30];
  const membershipLevels = ['Gold', 'Gold', 'Silver', 'Silver'];
  const expectedDiscountedPrices = ['96.00', '72.00', '59.50', '28.50'];

  expect(calculateDiscounts(prices, membershipLevels)).toEqual(expectedDiscountedPrices);
});
```

```
Snapshots:   0 total
Time:        0.302 s, estimated 1 s
Ran all test suites.
PS D:\JavaScript  testing zone\codingzone\ex7> npm test

> ex7@1.0.0 test
> jest

 PASS  ./calculateDiscounts.1.test.js
  √ calculateDiscounts applies correct discounts for different prices and membership levels (2 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        0.353 s, estimated 1 s
Ran all test suites.
PS D:\JavaScript  testing zone\codingzone\ex7> 
```

好吧，我們把它
Fix 成 59.50 這樣總
會對了吧

Code Coverage

Code Coverage Theory



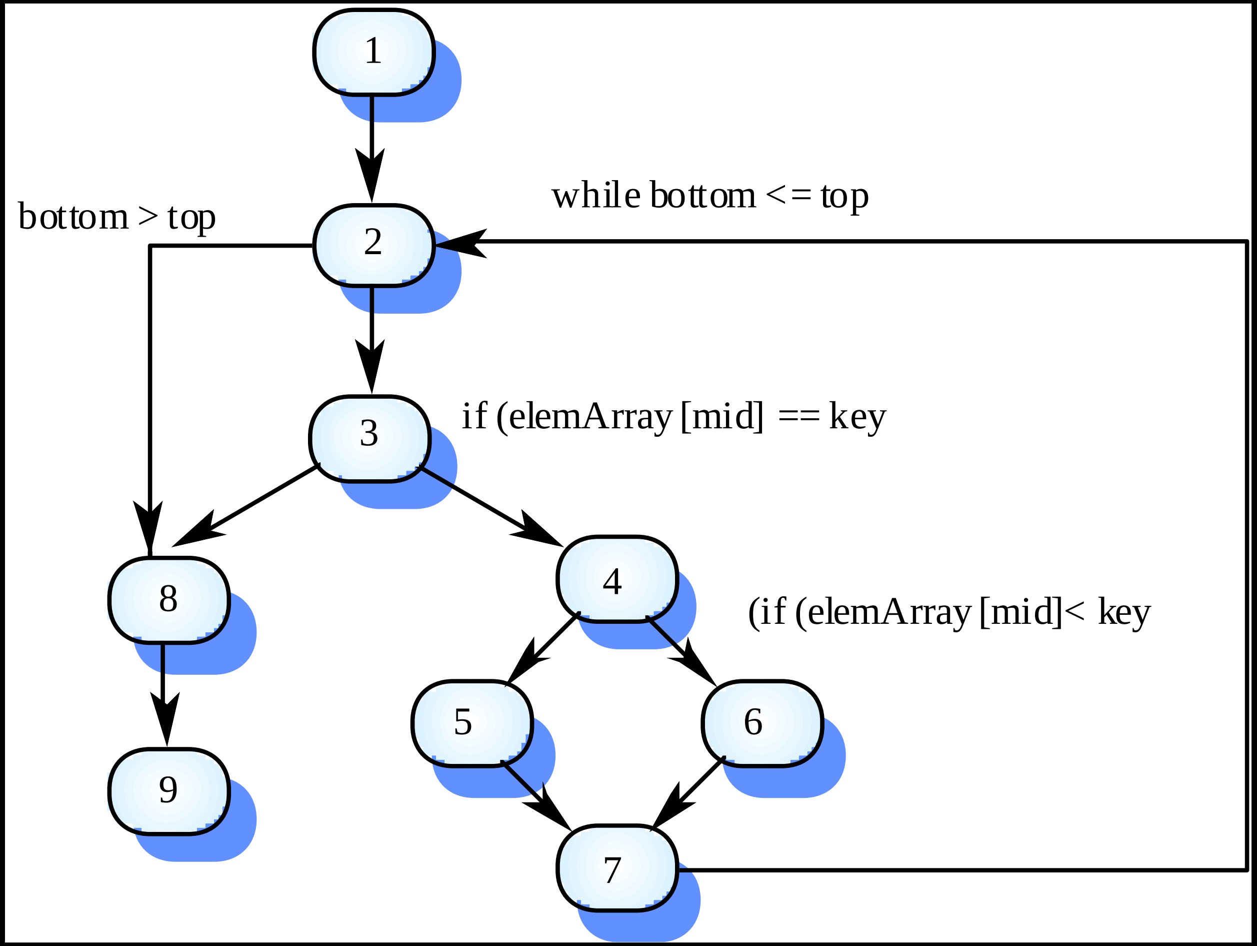
What is statement coverage?

The weakest criteria

- Statement coverage
  1 test case to run 1 2 3 8 9

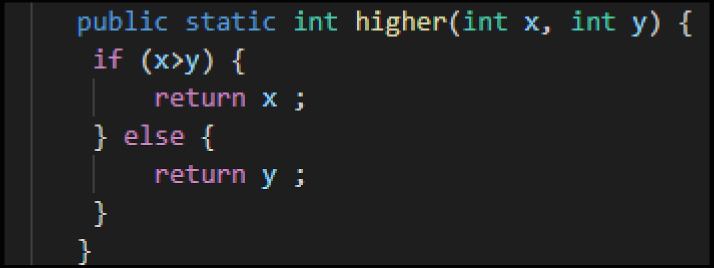
  1 test case to run 1 2 4 6 7 2

  1 test case to run 1 2 4 5 7 2 8
- You only need 3 test cases to have each statement at least executed once.

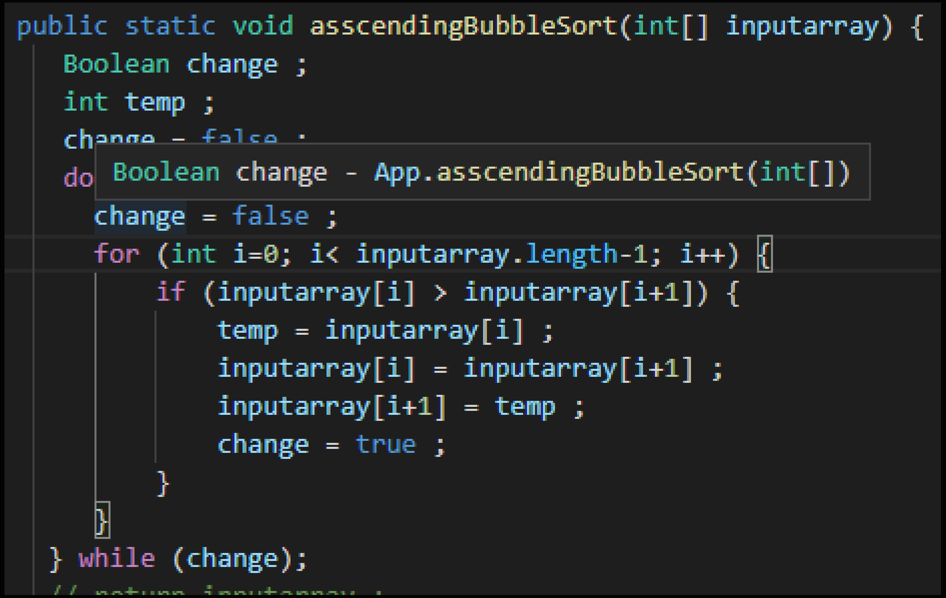


Comments about statement coverage

- Testing 的最低標
- 以 higher() 為例，你只要兩個 test cases 就能達標 (100%)
- 以 asscendingBubbleSort() 為例，你只要一個 5 4 3 2 1 測試案例就能達標 (100%)
- 所以 statement coverage 達標是個非常低的測試標準。

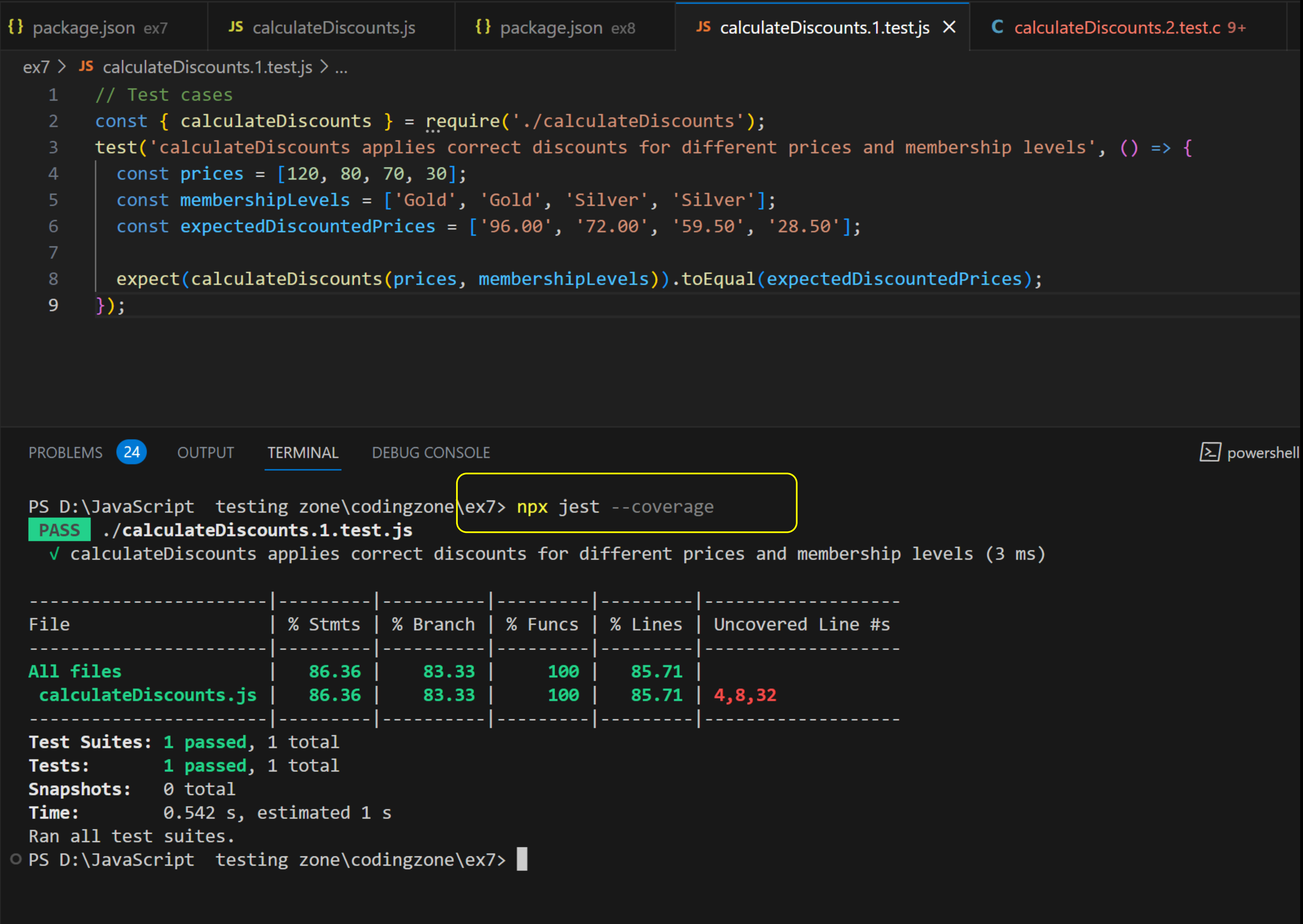


```
  public static int higher(int x, int y) {
   if (x>y) {
       return x ;
   } else {
       return y ;
   }
  }
```



```
public static void asscendingBubbleSort(int[] inputarray) {
   Boolean change ;
   int temp ;
   change = false ;
   do  Boolean change - App.asscendingBubbleSort(int[])
     change = false ;
     for (int i=0; i< inputarray.length-1; i++) {
         if (inputarray[i] > inputarray[i+1]) {
             temp = inputarray[i] ;
             inputarray[i] = inputarray[i+1] ;
             inputarray[i+1] = temp ;
             change = true ;
         }
     }
 } while (change);
 // return inputarray ;
```

Code Coverage Tool - JS

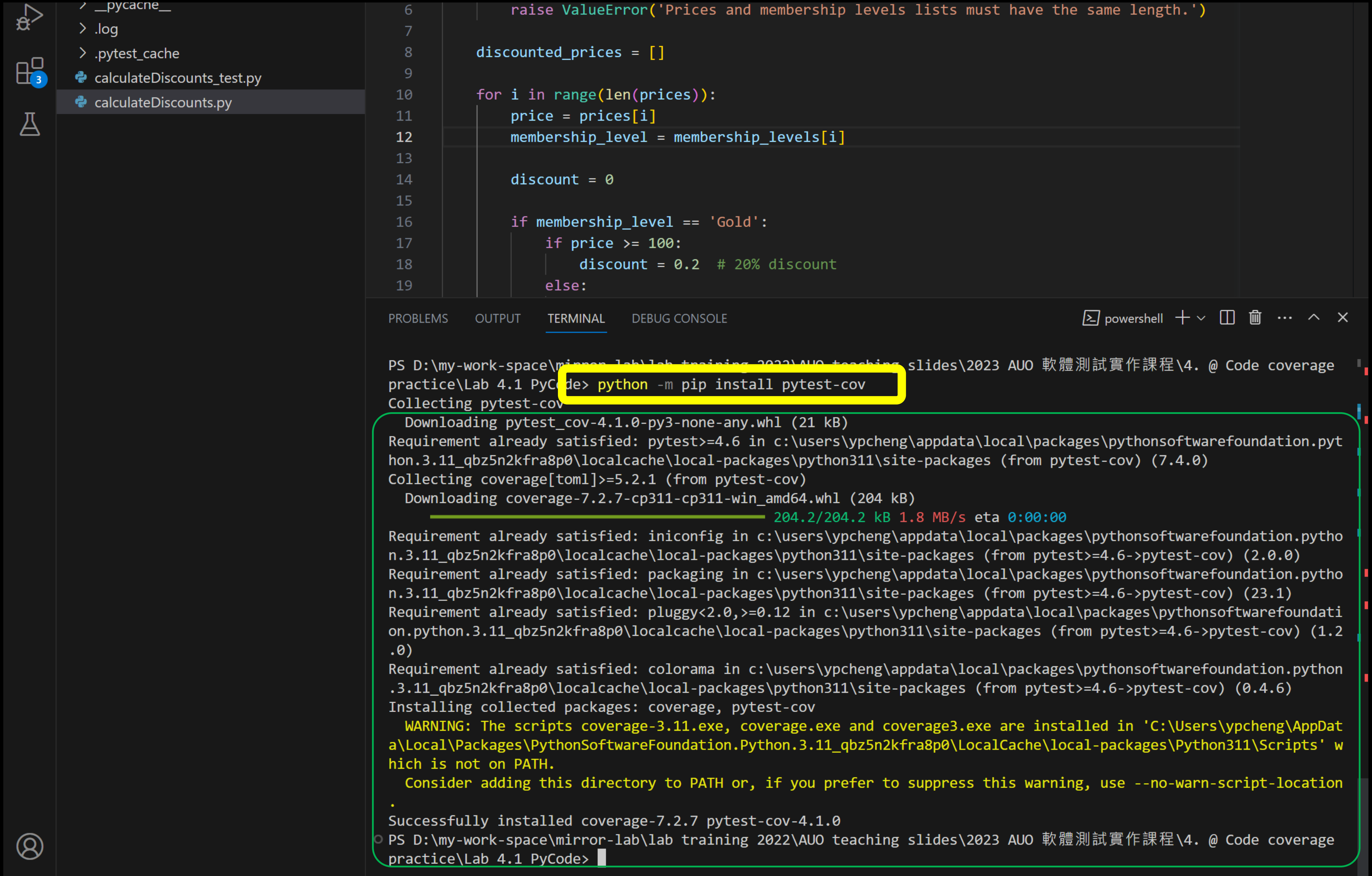


```javascript
// Test cases
const { calculateDiscounts } = require('./calculateDiscounts');
test('calculateDiscounts applies correct discounts for different prices and membership levels', () => {
  const prices = [120, 80, 70, 30];
  const membershipLevels = ['Gold', 'Gold', 'Silver', 'Silver'];
  const expectedDiscountedPrices = ['96.00', '72.00', '59.50', '28.50'];

  expect(calculateDiscounts(prices, membershipLevels)).toEqual(expectedDiscountedPrices);
});
```

```
PS D:\JavaScript  testing zone\codingzone\ex7> npx jest --coverage
 PASS  ./calculateDiscounts.1.test.js
  √ calculateDiscounts applies correct discounts for different prices and membership levels (3 ms)

-----------------------|---------|----------|---------|---------|-------------------
File                   | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
-----------------------|---------|----------|---------|---------|-------------------
All files              |   86.36 |    83.33 |     100 |   85.71 |
 calculateDiscounts.js |   86.36 |    83.33 |     100 |   85.71 | 4,8,32
-----------------------|---------|----------|---------|---------|-------------------
Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        0.542 s, estimated 1 s
Ran all test suites.
PS D:\JavaScript  testing zone\codingzone\ex7>
```

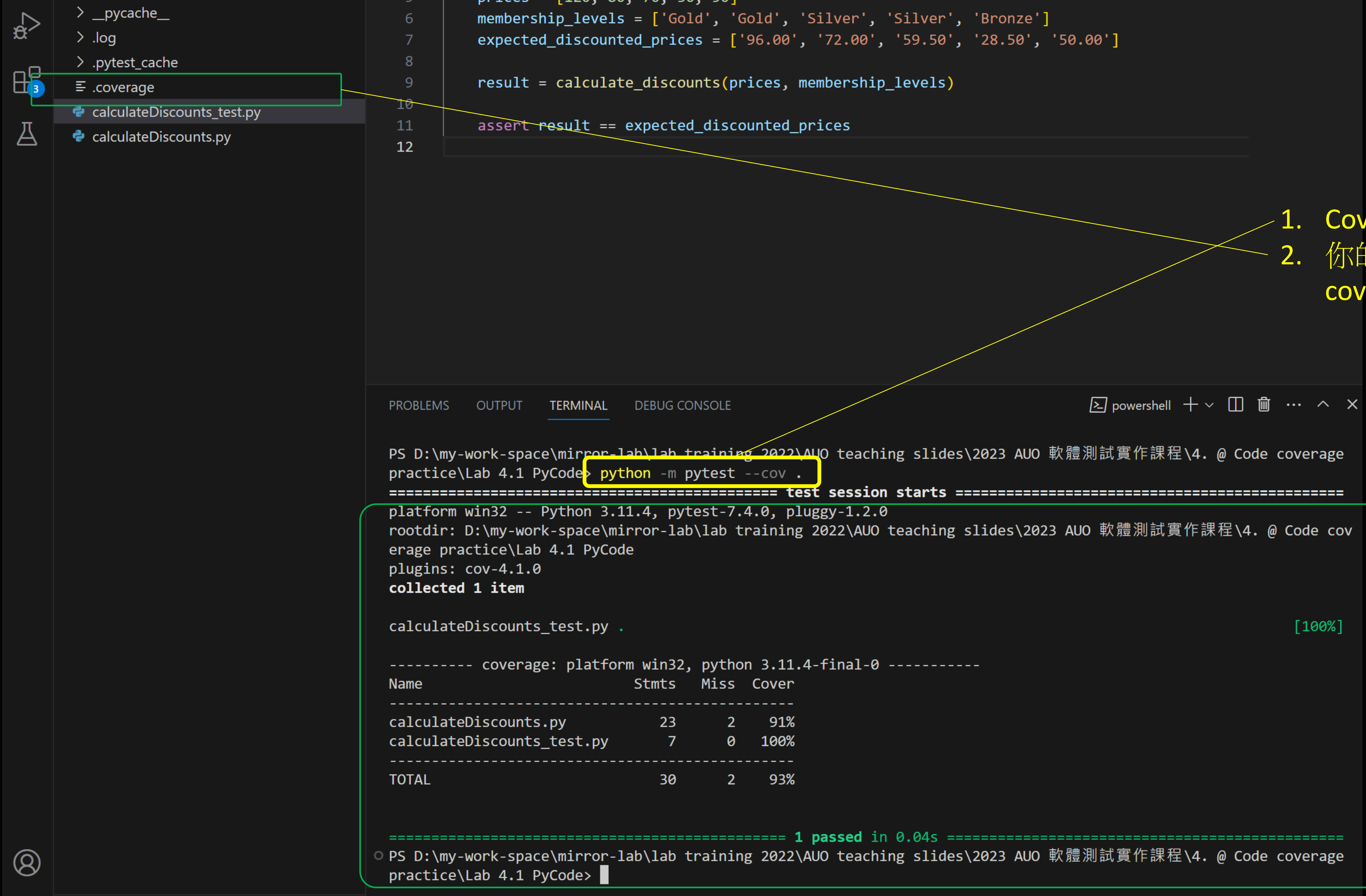
Code Coverage Tool - Python

安裝一下 pytest-cov



```python
        raise ValueError('Prices and membership levels lists must have the same length.')

    discounted_prices = []

    for i in range(len(prices)):
        price = prices[i]
        membership_level = membership_levels[i]

        discount = 0

        if membership_level == 'Gold':
            if price >= 100:
                discount = 0.2  # 20% discount
            else:
```

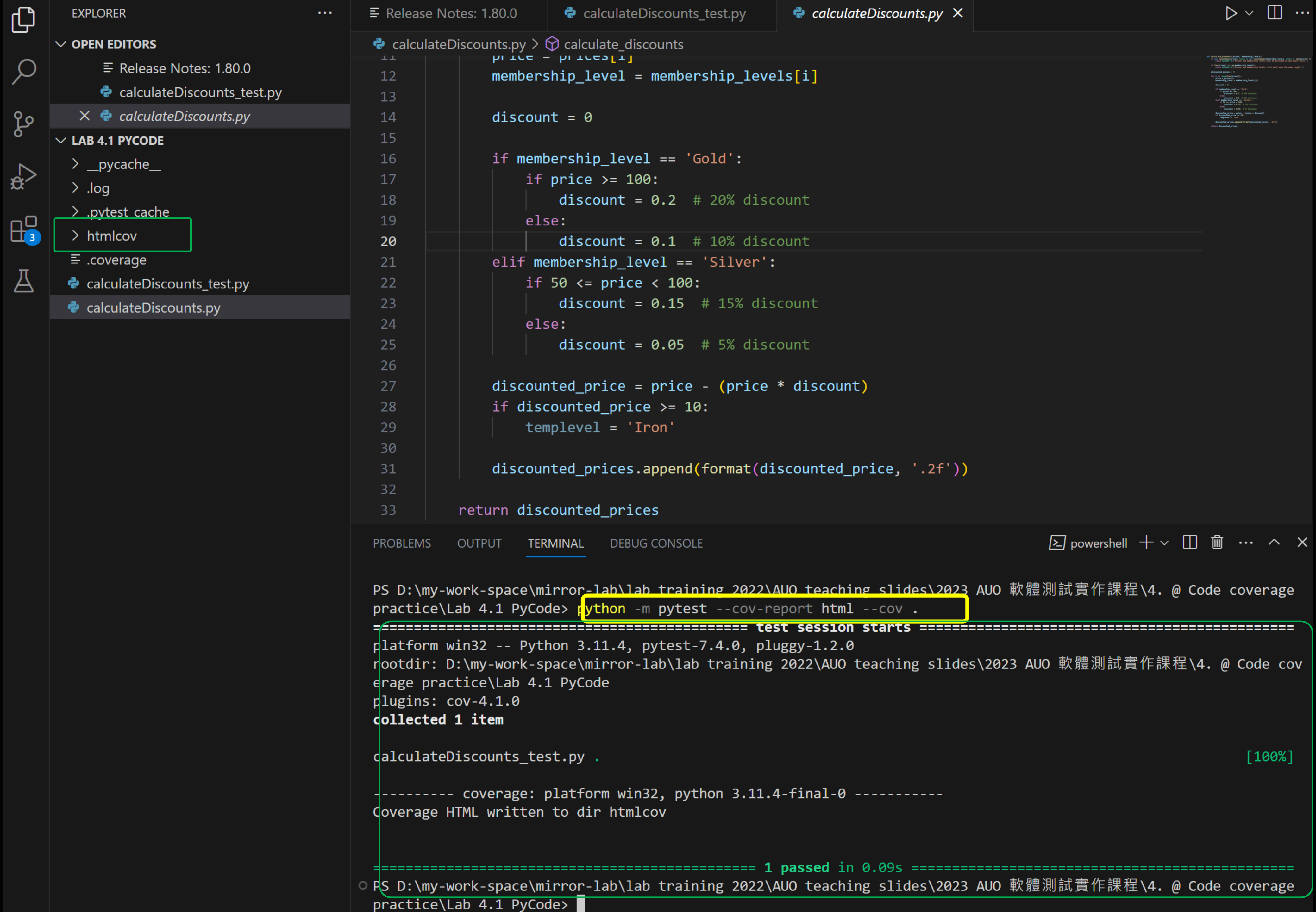
```
PS D:\my-work-space\mirror-lab\lab training 2022\AUO teaching slides\2023 AUO 軟體測試實作課程\4. @ Code coverage
practice\Lab 4.1 PyCode> python -m pip install pytest-cov
Collecting pytest-cov
  Downloading pytest_cov-4.1.0-py3-none-any.whl (21 kB)
Requirement already satisfied: pytest>=4.6 in c:\users\ypcheng\appdata\local\packages\pythonsoftwarefoundation.pyt
hon.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from pytest-cov) (7.4.0)
Collecting coverage[toml]>=5.2.1 (from pytest-cov)
  Downloading coverage-7.2.7-cp311-cp311-win_amd64.whl (204 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 204.2/204.2 kB 1.8 MB/s eta 0:00:00
Requirement already satisfied: iniconfig in c:\users\ypcheng\appdata\local\packages\pythonsoftwarefoundation.pytho
n.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from pytest>=4.6->pytest-cov) (2.0.0)
Requirement already satisfied: packaging in c:\users\ypcheng\appdata\local\packages\pythonsoftwarefoundation.pytho
n.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from pytest>=4.6->pytest-cov) (23.1)
Requirement already satisfied: pluggy<2.0,>=0.12 in c:\users\ypcheng\appdata\local\packages\pythonsoftwarefoundati
on.python.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from pytest>=4.6->pytest-cov) (1.2
.0)
Requirement already satisfied: colorama in c:\users\ypcheng\appdata\local\packages\pythonsoftwarefoundation.python
.3.11_qbz5n2kfra8p0\localcache\local-packages\python311\site-packages (from pytest>=4.6->pytest-cov) (0.4.6)
Installing collected packages: coverage, pytest-cov
  WARNING: The scripts coverage-3.11.exe, coverage.exe and coverage3.exe are installed in 'C:\Users\ypcheng\AppDat
a\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\Scripts' w
hich is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location
.
Successfully installed coverage-7.2.7 pytest-cov-4.1.0
PS D:\my-work-space\mirror-lab\lab training 2022\AUO teaching slides\2023 AUO 軟體測試實作課程\4. @ Code coverage
practice\Lab 4.1 PyCode> 
```



```python
    membership_levels = ['Gold', 'Gold', 'Silver', 'Silver', 'Bronze']
    expected_discounted_prices = ['96.00', '72.00', '59.50', '28.50', '50.00']

    result = calculate_discounts(prices, membership_levels)

    assert result == expected_discounted_prices

```

```
PS D:\my-work-space\mirror-lab\lab training 2022\AUO teaching slides\2023 AUO 軟體測試實作課程\4. @ Code coverage
practice\Lab 4.1 PyCode> python -m pytest --cov .
================================================ test session starts ================================================
platform win32 -- Python 3.11.4, pytest-7.4.0, pluggy-1.2.0
rootdir: D:\my-work-space\mirror-lab\lab training 2022\AUO teaching slides\2023 AUO 軟體測試實作課程\4. @ Code cov
erage practice\Lab 4.1 PyCode
plugins: cov-4.1.0
collected 1 item

calculateDiscounts_test.py .                                                                                  [100%]

---------- coverage: platform win32, python 3.11.4-final-0 -----------
Name                         Stmts   Miss  Cover
------------------------------------------------
calculateDiscounts.py           23      2    91%
calculateDiscounts_test.py       7      0   100%
------------------------------------------------
TOTAL                           30      2    93%


================================================= 1 passed in 0.04s =================================================
PS D:\my-work-space\mirror-lab\lab training 2022\AUO teaching slides\2023 AUO 軟體測試實作課程\4. @ Code coverage
practice\Lab 4.1 PyCode> 
```

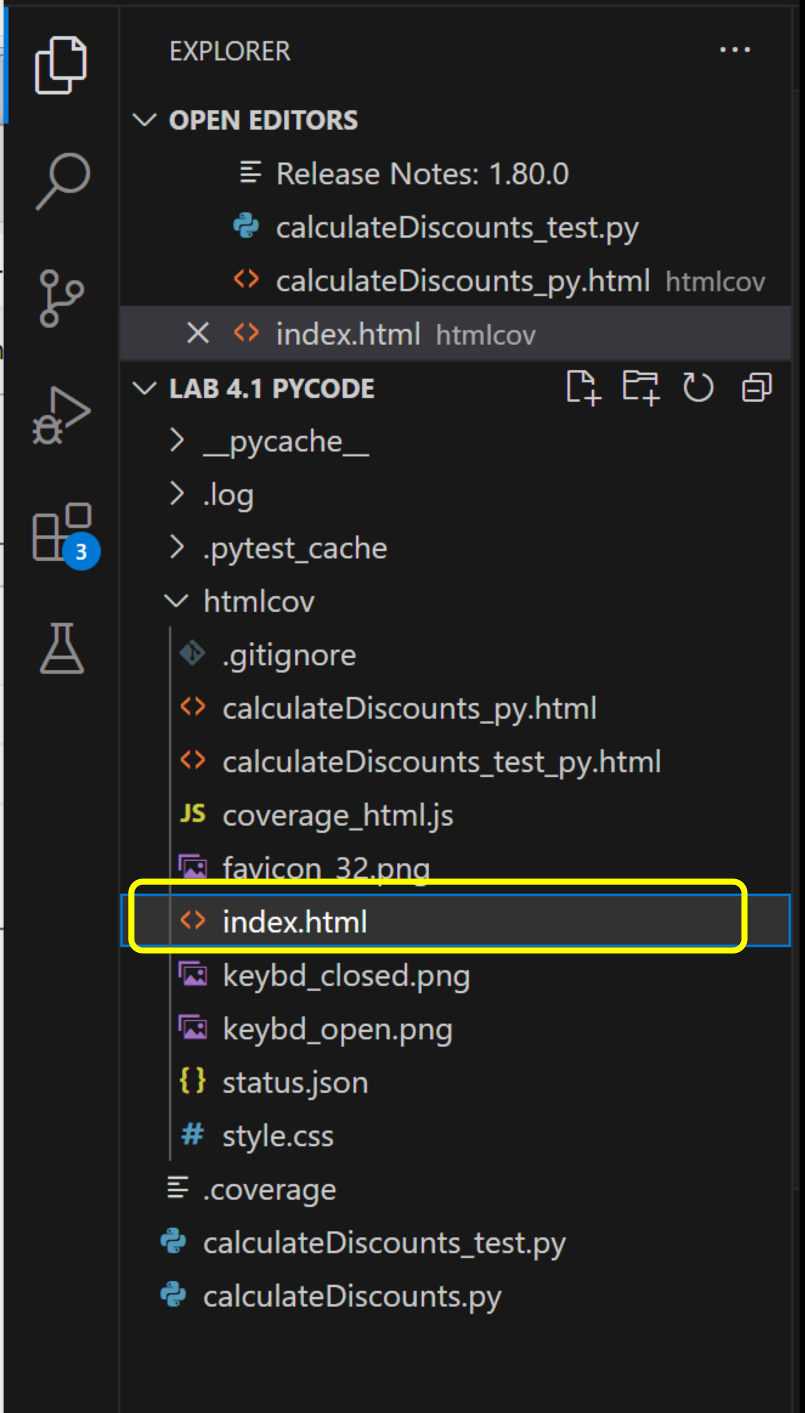
1. Coverage 這樣執行
2. 你的目錄會有一個 coverage file



```python
        price = prices[i]
        membership_level = membership_levels[i]

        discount = 0

        if membership_level == 'Gold':
            if price >= 100:
                discount = 0.2  # 20% discount
            else:
                discount = 0.1  # 10% discount
        elif membership_level == 'Silver':
            if 50 <= price < 100:
                discount = 0.15  # 15% discount
            else:
                discount = 0.05  # 5% discount

        discounted_price = price - (price * discount)
        if discounted_price >= 10:
            templevel = 'Iron'

        discounted_prices.append(format(discounted_price, '.2f'))

    return discounted_prices
```

```
PS D:\my-work-space\mirror-lab\lab training 2022\AUO teaching slides\2023 AUO 軟體測試實作課程\4. @ Code coverage practice\Lab 4.1 PyCode> python -m pytest --cov-report html --cov .
============================================ test session starts ============================================
platform win32 -- Python 3.11.4, pytest-7.4.0, pluggy-1.2.0
rootdir: D:\my-work-space\mirror-lab\lab training 2022\AUO teaching slides\2023 AUO 軟體測試實作課程\4. @ Code coverage practice\Lab 4.1 PyCode
plugins: cov-4.1.0
collected 1 item

calculateDiscounts_test.py .                                                                          [100%]

---------- coverage: platform win32, python 3.11.4-final-0 -----------
Coverage HTML written to dir htmlcov


============================================= 1 passed in 0.09s =============================================
PS D:\my-work-space\mirror-lab\lab training 2022\AUO teaching slides\2023 AUO 軟體測試實作課程\4. @ Code coverage practice\Lab 4.1 PyCode>
```

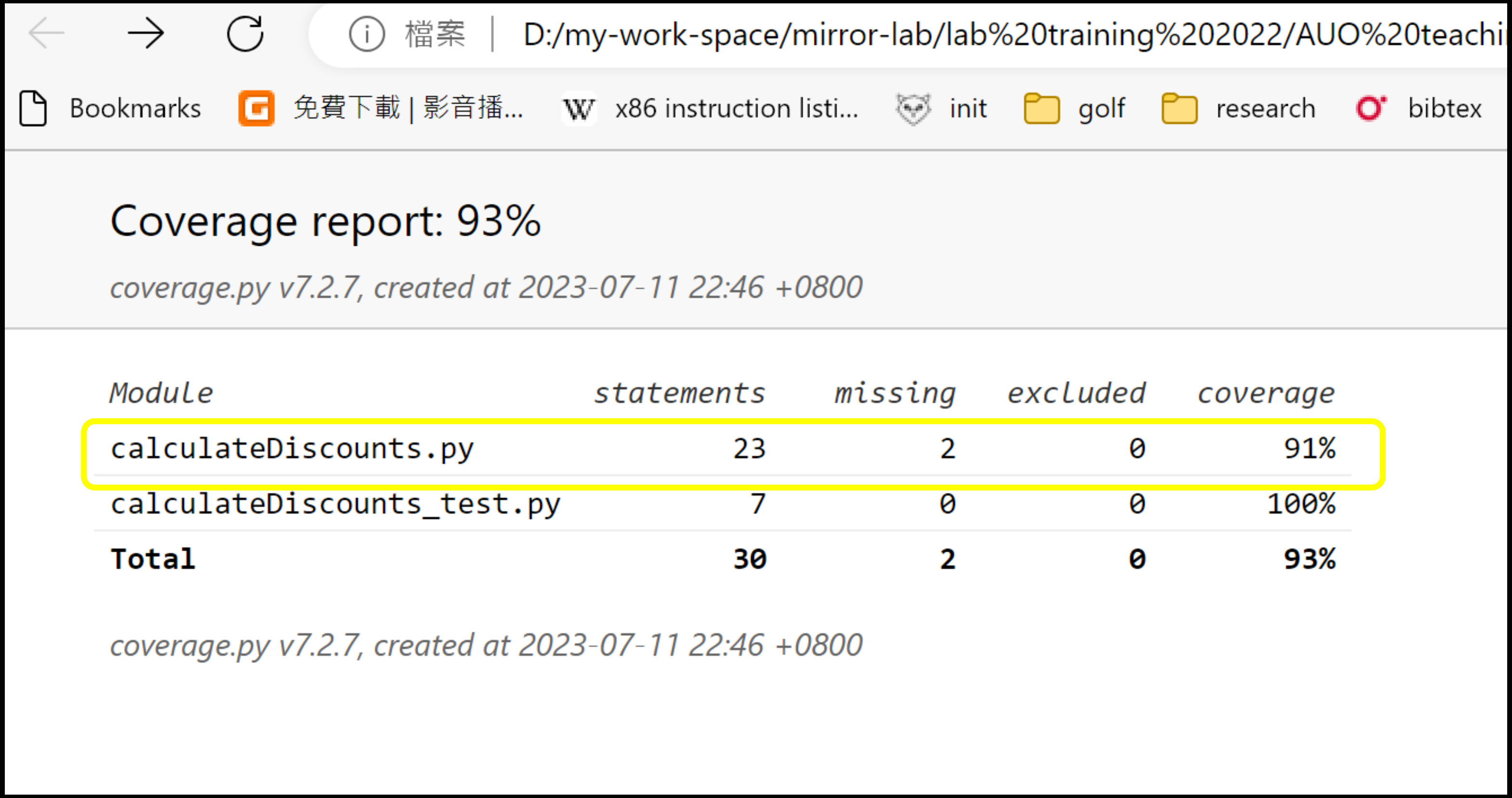
打開 coverage report



EXPLORER ...

OPEN EDITORS
- Release Notes: 1.80.0
- calculateDiscounts_test.py
- calculateDiscounts_py.html htmlcov
- index.html htmlcov

LAB 4.1 PYCODE
- _pycache_
- .log
- .pytest_cache
- htmlcov
  - .gitignore
  - calculateDiscounts_py.html
  - calculateDiscounts_test_py.html
  - coverage_html.js
  - favicon_32.png
  - index.html
  - keybd_closed.png
  - keybd_open.png
  - status.json
  - style.css
- .coverage
- calculateDiscounts_test.py
- calculateDiscounts.py

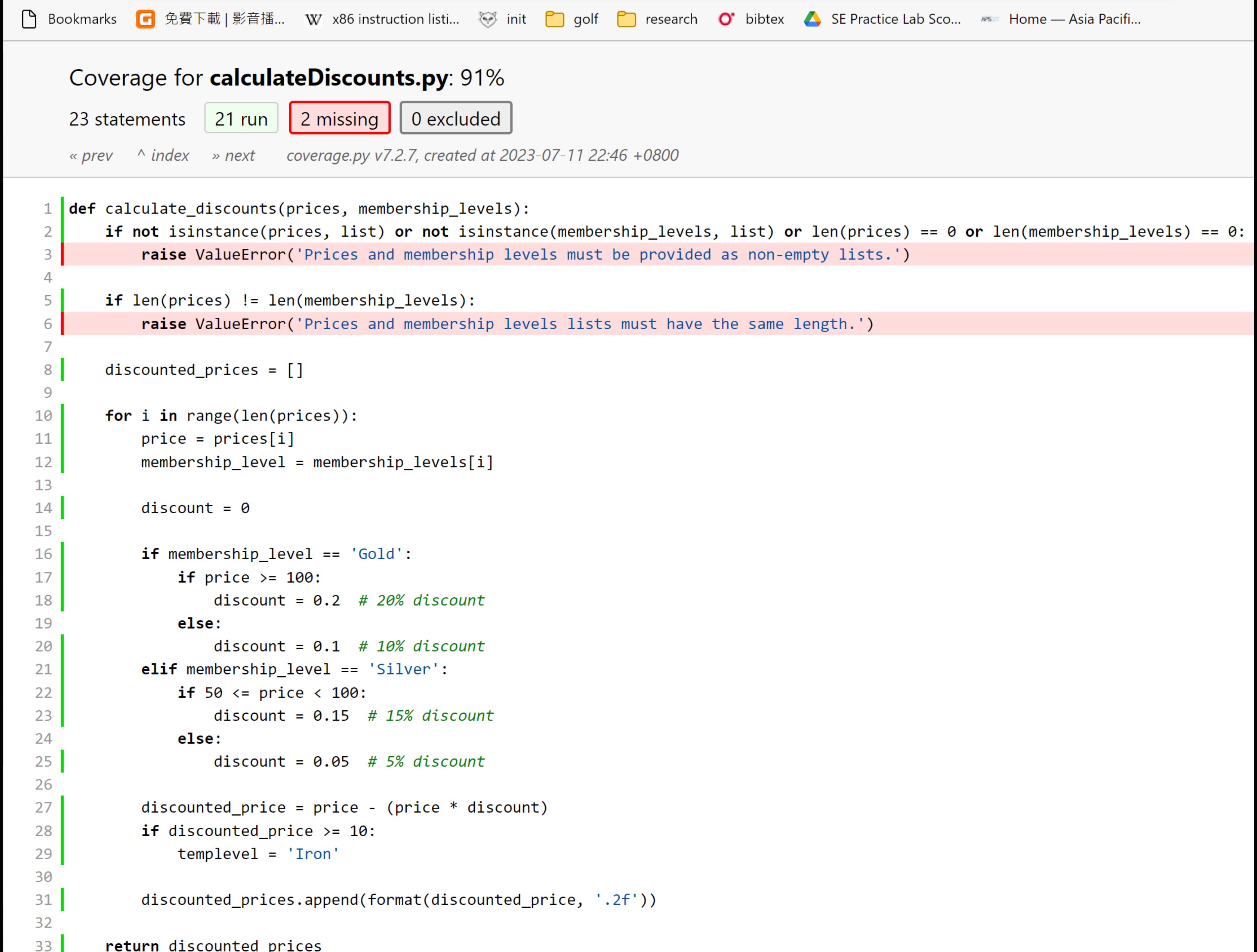


Coverage report: 93%

coverage.py v7.2.7, created at 2023-07-11 22:46 +0800

Module	*statements*	*missing*	*excluded*	*coverage*
calculateDiscounts.py	23	2	0	91%
calculateDiscounts_test.py	7	0	0	100%
Total	**30**	**2**	**0**	**93%**

coverage.py v7.2.7, created at 2023-07-11 22:46 +0800



Coverage for **calculateDiscounts.py**: 91%

23 statements 21 run 2 missing 0 excluded

coverage.py v7.2.7, created at 2023-07-11 22:46 +0800

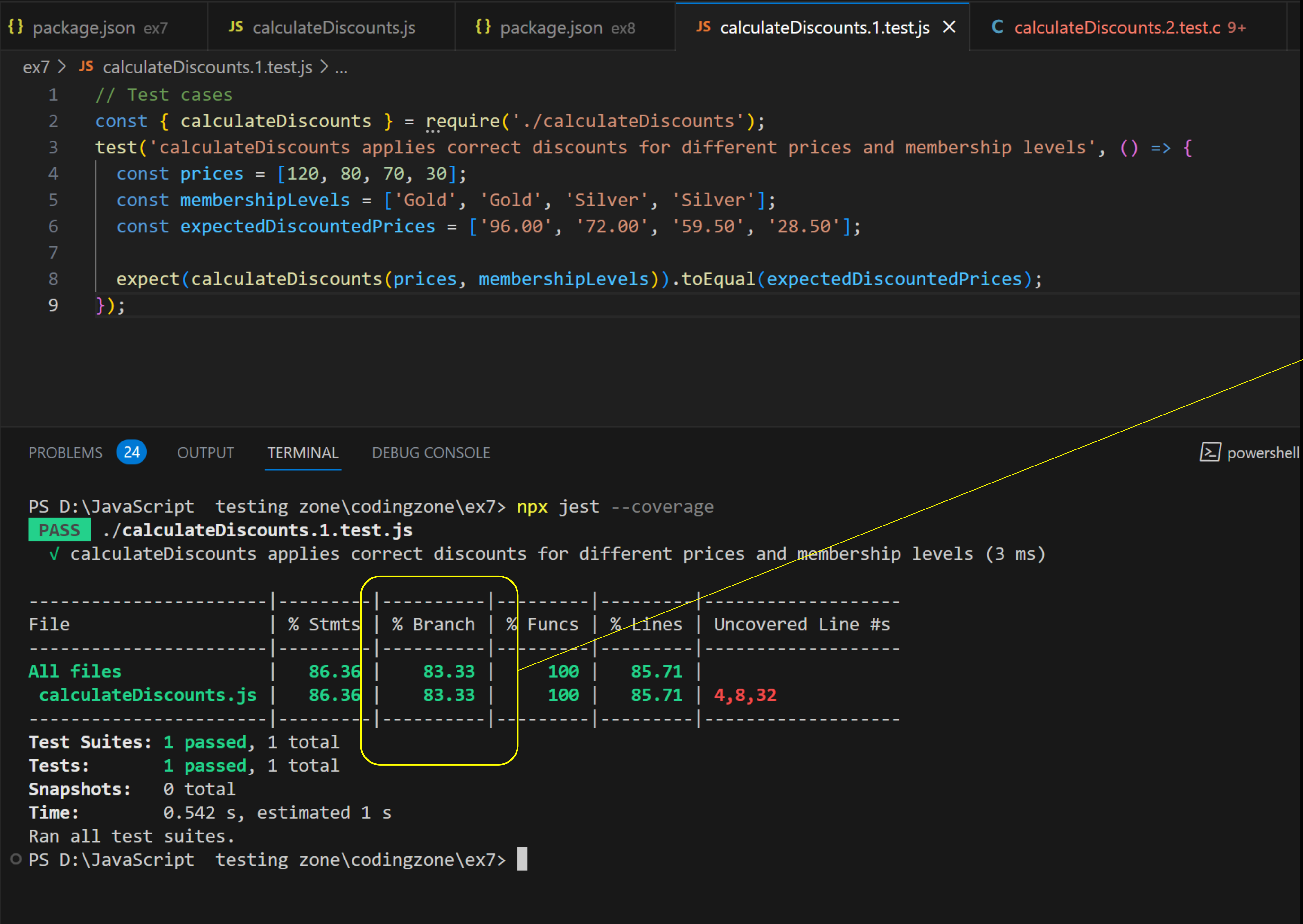
```
def calculate_discounts(prices, membership_levels):
    if not isinstance(prices, list) or not isinstance(membership_levels, list) or len(prices) == 0 or len(membership_levels) == 0:
        raise ValueError('Prices and membership levels must be provided as non-empty lists.')

    if len(prices) != len(membership_levels):
        raise ValueError('Prices and membership levels lists must have the same length.')

    discounted_prices = []

    for i in range(len(prices)):
        price = prices[i]
        membership_level = membership_levels[i]

        discount = 0

        if membership_level == 'Gold':
            if price >= 100:
                discount = 0.2  # 20% discount
            else:
                discount = 0.1  # 10% discount
        elif membership_level == 'Silver':
            if 50 <= price < 100:
                discount = 0.15  # 15% discount
            else:
                discount = 0.05  # 5% discount

        discounted_price = price - (price * discount)
        if discounted_price >= 10:
            templevel = 'Iron'

        discounted_prices.append(format(discounted_price, '.2f'))

    return discounted_prices
```

Statement Coverage 只有 83%

Lab 4.2: 請把 CalculateDiscount.js 的例子變成 statement coverage 100%

請你完成之後執行 npx jest –coverage 然後將結果截圖上傳

好唄 Branch Coverage 還沒有 100%？

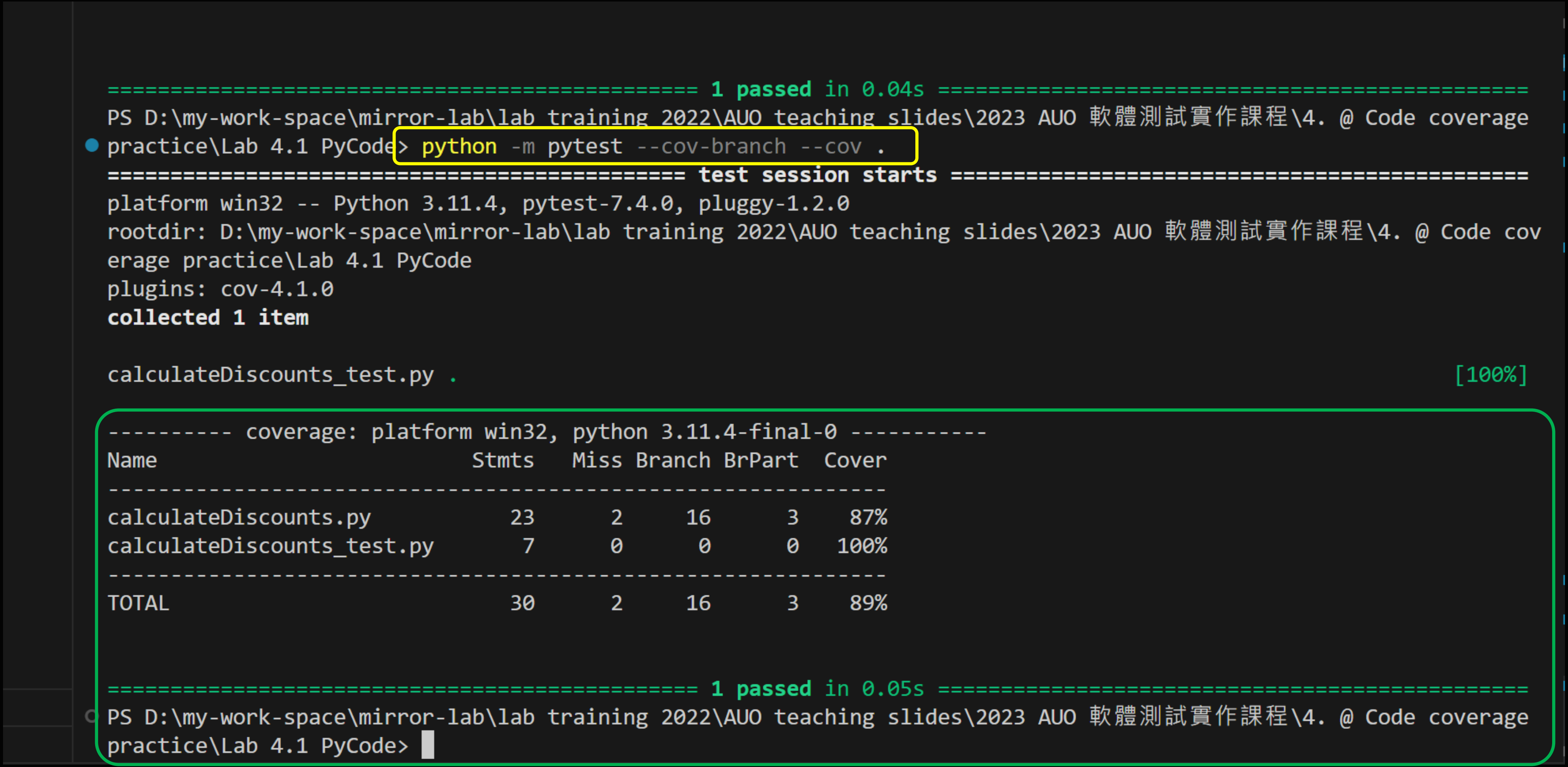


```javascript
// Test cases
const { calculateDiscounts } = require('./calculateDiscounts');
test('calculateDiscounts applies correct discounts for different prices and membership levels', () => {
  const prices = [120, 80, 70, 30];
  const membershipLevels = ['Gold', 'Gold', 'Silver', 'Silver'];
  const expectedDiscountedPrices = ['96.00', '72.00', '59.50', '28.50'];

  expect(calculateDiscounts(prices, membershipLevels)).toEqual(expectedDiscountedPrices);
});
```

```
PS D:\JavaScript  testing zone\codingzone\ex7> npx jest --coverage
 PASS  ./calculateDiscounts.1.test.js
  √ calculateDiscounts applies correct discounts for different prices and membership levels (3 ms)

-----------------------|---------|----------|---------|---------|-------------------
File                   | % Stmts | % Branch | % Funcs | % Lines | Uncovered Line #s
-----------------------|---------|----------|---------|---------|-------------------
All files              |   86.36 |    83.33 |     100 |   85.71 |
 calculateDiscounts.js |   86.36 |    83.33 |     100 |   85.71 | 4,8,32
-----------------------|---------|----------|---------|---------|-------------------
Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        0.542 s, estimated 1 s
Ran all test suites.
PS D:\JavaScript  testing zone\codingzone\ex7> 
```

???

Enable Branch Coverage in pytest



```
================================================ 1 passed in 0.04s ================================================
PS D:\my-work-space\mirror-lab\lab training 2022\AUO teaching slides\2023 AUO 軟體測試實作課程\4. @ Code coverage
practice\Lab 4.1 PyCode> python -m pytest --cov-branch --cov .
=============================================== test session starts ===============================================
platform win32 -- Python 3.11.4, pytest-7.4.0, pluggy-1.2.0
rootdir: D:\my-work-space\mirror-lab\lab training 2022\AUO teaching slides\2023 AUO 軟體測試實作課程\4. @ Code cov
erage practice\Lab 4.1 PyCode
plugins: cov-4.1.0
collected 1 item

calculateDiscounts_test.py .                                                                               [100%]

---------- coverage: platform win32, python 3.11.4-final-0 -----------
Name                          Stmts   Miss Branch BrPart  Cover
---------------------------------------------------------------
calculateDiscounts.py            23      2     16      3    87%
calculateDiscounts_test.py        7      0      0      0   100%
---------------------------------------------------------------
TOTAL                            30      2     16      3    89%


================================================ 1 passed in 0.05s ================================================
PS D:\my-work-space\mirror-lab\lab training 2022\AUO teaching slides\2023 AUO 軟體測試實作課程\4. @ Code coverage
practice\Lab 4.1 PyCode> 
```

What is branch coverage?

Branch coverage is a requirement that, for each branch in the program (e.g., if statements, loops), each branch have been executed at least once during testing. (It is sometimes also described as saying that each branch condition must have been true at least once and false at least once during testing.)

分支覆蓋是指對於程式中的每個分支（例如 if 陳述句、迴圈），在測試期間每個分支至少被執行過一次的要求。（有時也可以說每個分支條件在測試期間必須至少一次為真，至少一次為假。）

Branch coverage

- each branch can produce two choices, every choice combinations should all be exercised
- there are 3 branches, so at most 2 * 2 * 2 branching choices should be exercised if loop is not considered.
- So at least you need to find test cases to meet this criteria

1 2 8 9

1 2 3 8 9

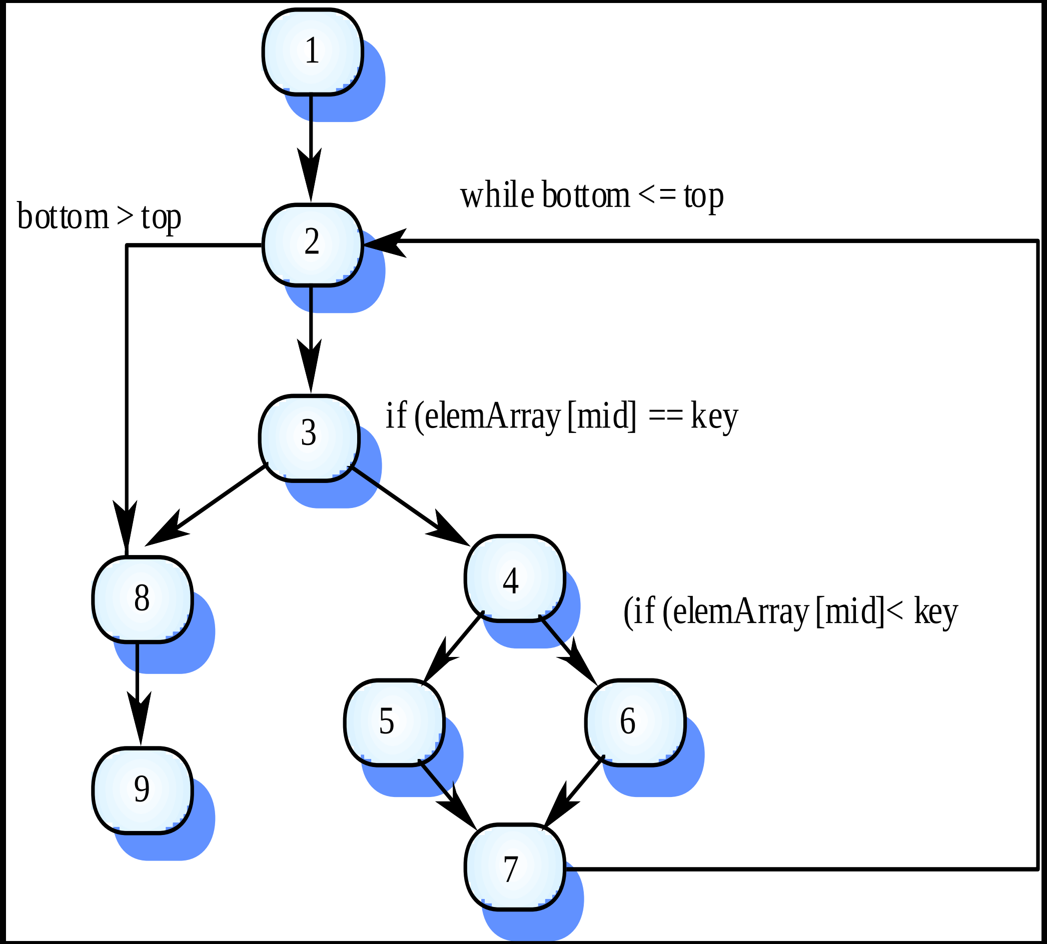
1 2 3 4 5 7 2 8 9

1 2 3 4 5 7 2 3 8 9

1 2 3 4 6 7 2 8 9

1 2 3 4 6 7 2 3 8 9

- In the example, there are less than 8 branching choices because it is not a complete tree



Lab 4.3: 請把 CalculateDiscount.js 的例子變成 branch coverage 100%

請你完成之後執行 npx jest –coverage 然後將結果截圖上傳

Path Coverage

- The highest coverage
- Try to find test cases which can cover all the paths
- equal to exhaustive testing if you want to cover them all, which is impossible
- You need to find infinite test cases which have finite (if the program must stop) or infinite length (if the program may run forever)

1 2 8 9 (finite length)

1 2 3 8 9 (finite length)

1 2 3 4 5 7 2 8 9

1 2 3 4 5 7 2 3 4 5 7 2 8 9

1 2 3 4 5 7 2 3 4 6 7 2 3 8 9

1 2 (3 4 5 7 2) * 8 9 (finite length but very long sequence)

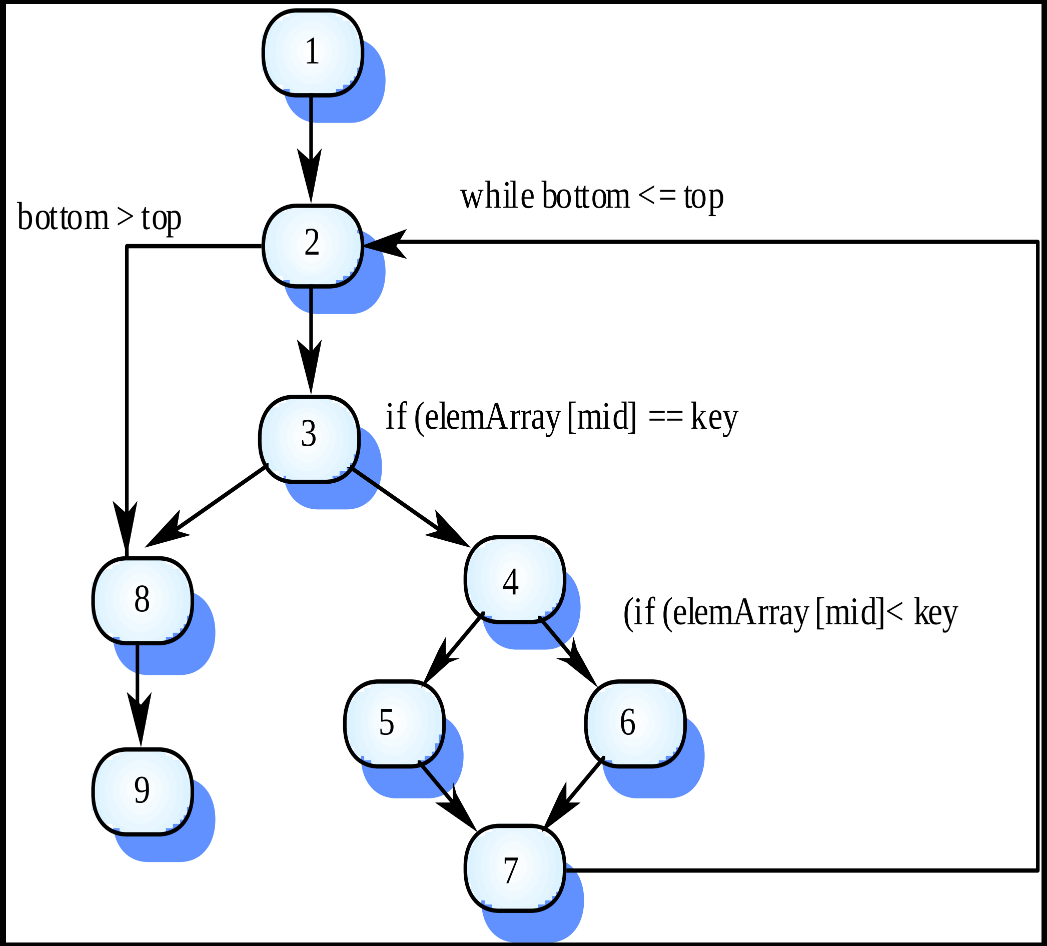
1 2 (3 4 5 7 2 | 3 4 6 7 2) * 8 9 (finite length with permutations)

1 2 (3 4 5 7 2) ω 8 9

………

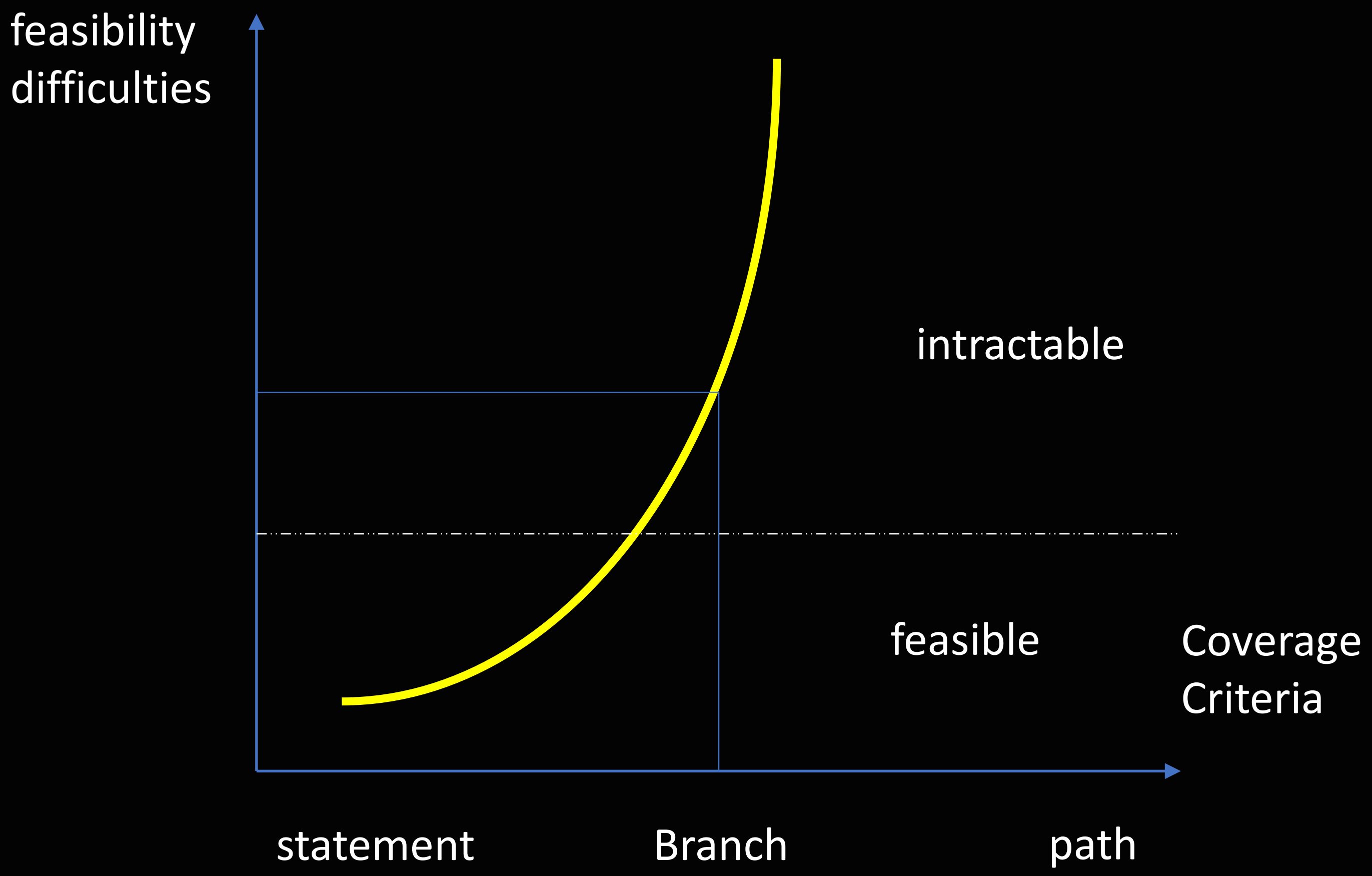
………

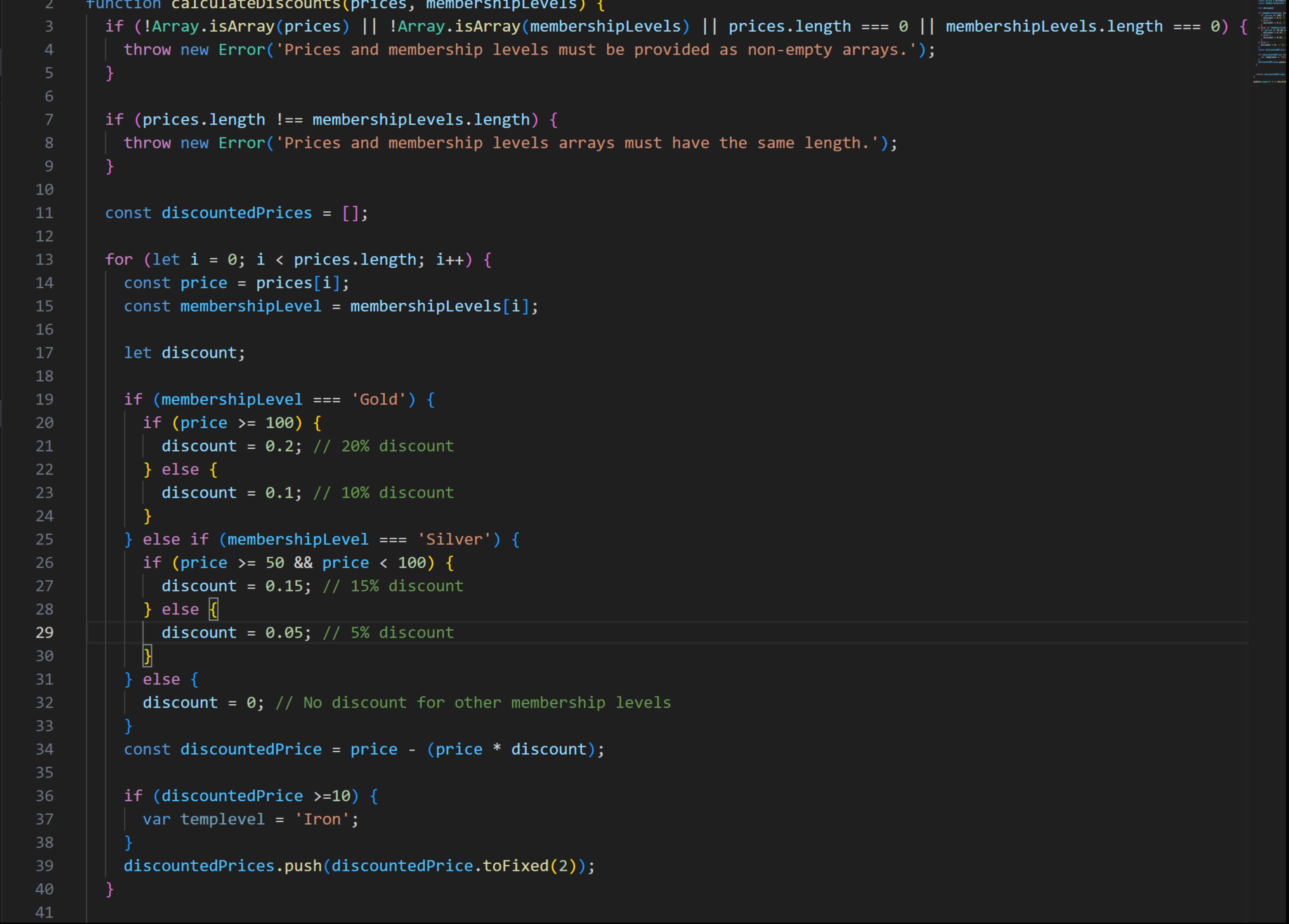
- where * is finite iteration in automata theory
- where ω is infinite iteration in automata theory



Summary about branch Coverage

- 理論上非常有趣，你如果想達到的標準是 branch coverage，實務上他已經很困難了
- 聽說過 branch coverage 是金管會證交系統的驗收標準





```
function calculateDiscounts(prices, membershipLevels) {
  if (!Array.isArray(prices) || !Array.isArray(membershipLevels) || prices.length === 0 || membershipLevels.length === 0) {
    throw new Error('Prices and membership levels must be provided as non-empty arrays.');
  }

  if (prices.length !== membershipLevels.length) {
    throw new Error('Prices and membership levels arrays must have the same length.');
  }

  const discountedPrices = [];

  for (let i = 0; i < prices.length; i++) {
    const price = prices[i];
    const membershipLevel = membershipLevels[i];

    let discount;

    if (membershipLevel === 'Gold') {
      if (price >= 100) {
        discount = 0.2; // 20% discount
      } else {
        discount = 0.1; // 10% discount
      }
    } else if (membershipLevel === 'Silver') {
      if (price >= 50 && price < 100) {
        discount = 0.15; // 15% discount
      } else {
        discount = 0.05; // 5% discount
      }
    } else {
      discount = 0; // No discount for other membership levels
    }
    const discountedPrice = price - (price * discount);

    if (discountedPrice >=10) {
      var templevel = 'Iron';
    }
    discountedPrices.push(discountedPrice.toFixed(2));
  }

```

Homework: 請撰寫 homework/calculateGrade.* 的測試案例，使其變成 branch coverage 100%

請你完成之後執行 coverage 工具，展示 coverage 的效果，然後將結果截圖上傳