

## 1: ทดสอบด้วยเทคนิค Decision และ Condition ต่างๆ บนโปรแกรมการล็อกอิน

## 1.1: Decision Coverage

TC	!validUser	!validPassword	Count > 3	In-if	count	User	password	Expected output
1	T	T	F	N	0	Admin	Admin	Login successful
2	T	F	F	Y	1	Admin	1234	Login Failed count 1/3
3	F	T	F	Y	2	User	Admin	Login Fail count 2/3
4	F	F	T	Y	3	User	1234	Account Blocked

## 1.2: Condition Coverage

TC	!validUser	!validPassword	Count > 3	In-if	count	User	password	Expected output
1	T	T	F	N	0	Admin	Admin	Login successful
2	F	T	T	Y	1	Admin	1234	Account Blocked

## 1.3: Condition/decision converge

TC	!validUser	!validPassword	Count > 3	In-if	count	User	password	Expected output
1	T	T	F	N	0	Admin	Admin	Login successful
2	F	F	T	Y	3	Admin	1234	Account Blocked

## 1.4: Modified Condition/Decision Coverage (MC/DC)

TC	!validUser	!validPassword	Count > 3	In-if	count	User	password	Expected output
1	T	T	F	N	0	Admin	Admin	Login successful
2	T	F	F	N	1	Admin	1234	Login successful
3	F	T	F	Y	2	User	Admin	Account Blocked
4	F	F	T	Y	3	User	1234	Account Blocked

## 2: ทดสอบด้วยเทคนิค Decision และ Condition ต่างๆ บนโปรแกรมการลงทะเบียน

## 2.1: Decision Coverage

TC	Email.index('@') >= 1	Password >= 8	age > 18	In-if	Email	password	Age	Expected output
1	T	T	T	Y	admin@gmail.com	012345678	25	True
2	T	F	F	N	admin@gmail.com	012345678	17	False
3	T	F	T	N	admin@gmail.com	0123456	25	False
4	T	F	F	N	admin@gmail.com	0123456	17	False
5	F	F	F	N	admingmail.com	0123456	17	False

## 2.2: Condition Coverage

TC	Email.index('@') >= 1	Password >= 8	age > 18	In-if	Email	password	Age	Expected output
1	T	T	T	Y	admin@gmail.com	012345678	25	True
2	T	F	F	N	admin@gmail.com	012345678	17	False

## 2.3: Condition/decision converge

TC	Email.index('@') >= 1	Password >= 8	age > 18	In-if	Email	password	Age	Expected output
1	T	T	T	Y	admin@gmail.com	012345678	25	True
2	T	F	F	N	admin@gmail.com	0123456	17	False
3	F	F	F	N	admingmail.com	0123456	17	False

## 2.4: Modified Condition/Decision Coverage (MC/DC)

TC	Email.index('@') >= 1	Password >= 8	age > 18	In-if	Email	password	Age	Expected output
1	T	T	T	Y	admin@gmail.com	012345678	25	True
2	T	F	F	N	admin@gmail.com	012345678	17	False
3	T	F	T	N	admin@gmail.com	0123456	25	False
4	F	F	F	N	admin@gmail.com	0123456	17	False
5	F	F	F	N	admingmail.com	0123456	17	False

## 2.5: Test Case

```
class TestRegister(unittest.TestCase):  
  
    def test_resgister_successful(self):  
        self.assertEqual(True, register('admin@gmail.com', '012345678', 25))  
  
    def test_resgister_failed_1(self):  
        self.assertEqual(False, register('admin@gmail.com', '012345678', 17))  
  
    def test_resgister_failed_2(self):  
        self.assertEqual(False, register('admin@gmail.com', '0123456', 25))  
  
    def test_resgister_failed_3(self):  
        self.assertEqual(False, register('admin@gmail.com', '0123456', 17))  
  
    def test_resgister_failed_4(self):  
        self.assertEqual(False, register('admingmail.com', '0123456', 17))
```

Name	Stmts	Miss	Cover
Register.py	4	0	100%
test_resigter.py	13	0	100%
TOTAL	17	0	100%

## 3: ทดสอบเมธอดเช็คประเภทของสามเหลี่ยม ด้วยเทคนิค Condition Coverage

## 3.1: Test Case

```

class Test_triangle(unittest.TestCase):
    def test_Equilateral(self):
        self.assertEqual('Equilateral', get_triangle_type(3, 3, 3))

    def test_Isosceles_1(self):
        self.assertEqual('Isosceles', get_triangle_type(7, 7, 3))

    def test_Isosceles_2(self):
        self.assertEqual('Isosceles', get_triangle_type(3, 7, 7))

    def test_Isosceles_3(self):
        self.assertEqual('Isosceles', get_triangle_type(7, 3, 7))

    def test_Scalene(self):
        self.assertEqual('Scalene', get_triangle_type(7, 5, 6))

    def test_Invalid_1(self):
        self.assertEqual('Invalid', get_triangle_type(-1, 2, 4))

    def test_Invalid_2(self):
        self.assertEqual('Invalid', get_triangle_type(2, -7, 5))

    def test_Invalid_3(self):
        self.assertEqual('Invalid', get_triangle_type(9, 6, -5))

```

Name	Stmts	Miss	Cover
Test_Ractangle.py	16	0	100%
ractangle.py	8	0	100%
TOTAL	24	0	100%

## 3.2: Equivalence partitioning

getTriangle Method		Partition							
		Equilateral	Isosceles			Scalene	Invalid		
EP input	A	3	7	3	7	7	-1	2	9
	B	3	7	7	3	5	2	-7	6
	C	3	3	7	7	6	4	5	-5
Expected output		Equilateral	Isosceles			Scalene	Invalid		

## 3.3: Statement coverage

TC	A	B	C	Path	Expected Output
1	7	5	6	1-2-3-4-9-10	Scalene
2	-1	2	4	1-2-3	Invalid

## 3.4: Decision coverage

TC	A	B	C	Path	Expected Output
1	3	3	3	1-2-4-5-6	Equilateral
2	7	7	3	1-2-4-5-7-8	Isosceles
3	7	5	6	1-2-3-4-9-10	Scalene
4	-1	2	4	1-2-3	Invalid

## 3.5: Condition coverage

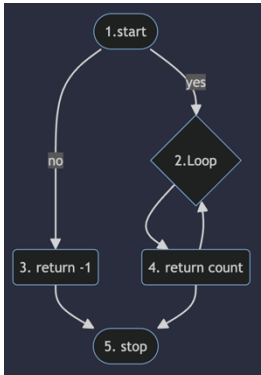
TC	A	B	C	In-if (1)	In-if (2)	Expected output
1	3	3	3	N	Y	Equilateral
2	7	7	3	N	Y	Isosceles
3	7	5	6	N	Y	Scalene
4	-1	2	4	Y	N	Invalid

## 3.6: ความแตกต่าง

Coverage Criteria	Equivalence partitioning	Statement coverage	Decision coverage	Condition coverage
ทุกจุดเข้าและออกในโปรแกรมถูกเรียกใช้อย่างน้อยหนึ่งครั้ง	X	X	X	X
ทุกการตัดสินใจในโปรแกรมต้องคำนึงถึงผลลัพธ์ที่เป็นไปได้ทั้งหมดอย่างน้อยหนึ่งครั้ง		X	X	X
ทุกเงื่อนไขในการตัดสินใจในโปรแกรมได้นำผลลัพธ์ที่เป็นไปได้ทั้งหมดอย่างน้อยหนึ่งครั้ง			X	X
ทุกเงื่อนไขในการตัดสินใจได้แสดงให้เห็นแล้วว่าส่งผลต่อผลลัพธ์ของการตัดสินใจอย่างเป็นอิสระ	X		X	X
ทุกการรวมกันของผลลัพธ์ของเงื่อนไขภายในการตัดสินใจจะถูกเรียกใช้อย่างน้อยหนึ่งรายการ	X	X	X	X

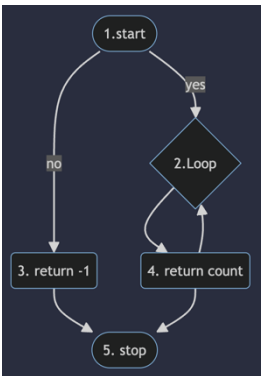
4: ทดสอบด้วยเทคนิค Loop, Path Coverage ต่างๆ

4.1: Loop Coverage



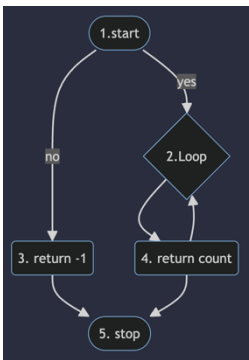
TC	Path	Expected Output
1	1-2-4-5	0 loop
2	1-2-4-2-4-5	1 loop
3	1-2-4-2-4-2-4-5	2 loops

4.2: Edge-pair Coverage



TC	Path	Expected Output
1	1-3-5	
2	1-2-4-5	0 loop
3	1-2-4-2-4-5	1 loops

4.3: McCabe Cyclomatic Coverage



TC	Array	X	Path	Expected Output
1		1	1-3-5	-1
2	1	1	1-2-4-5	0
3	1,2	2	1-2-4-2-4-5	1

## 4.4: Test Case

```
class Test_find_first(unittest.TestCase):  
  
    def test_expexted_case_1(self):  
        self.assertEqual(0, find_first([1], 1))  
  
    def test_expexted_case_2(self):  
        self.assertEqual(1, find_first([1, 2], 2))  
  
    def test_expexted_case_3(self):  
        self.assertEqual(-1, find_first([], 1))
```

Name	Stmts	Miss	Cover
find_first.py	6	0	100%
test_find_first.py	9	0	100%
TOTAL	15	0	100%