1. True or False?
   Statement coverage always satisfies decision coverage.

- ○ True
- ◉ False

2. Given the code below, which set of test cases will achieve 100% statement coverage?

```
If a < 5 or b > 7
        X = 50;
        c = a + b;
Else
        X = 25;
        c = a – b;
If X = 50 and c > 6
        Z = 10;
Else
        Z = 12;
```

- Test Case 1: a=2, b=10, c=12, X=25

  Test Case 2: a=3, b=4, c=4, X=25

- Test Case 1: a=2, b=10, c=12, X=50

  Test Case 2: a=5, b=1, c=4, X=25

Don’t know why this is wrong

- Test Case 1: a=3, b=10, c=13, X=50

  Test Case 2: a=5, b=1, c=4, X=25

- Test Case 1: a=3, b=10, c=13, X=50

  Test Case 2: a=1, b=2, c=3, X=50

3. Given the code below, which set of test cases will achieve 100% decision coverage?

```
If a < 5 or b > 7
        X = 50;
        c = a + b;
Else
        X = 25;
        c = a – b;
If X = 50 and c > 6
        Z = 10;
Else
        Z = 12;
```

- Test Case 1: a=3, b=10, c=13, X=50

  Test Case 2: a=1, b=2, c=3, X=50

- Test Case 1: a=3, b=10, c=13, X=50

  Test Case 2: a=3, b=4, c=4, X=25

- Test Case 1: a= 3, b=10, c=13, X=50

  Test Case 2: a=5, b=1, c=4, X=25

- Test Case 1: a=2, b=10, c=12, X=25

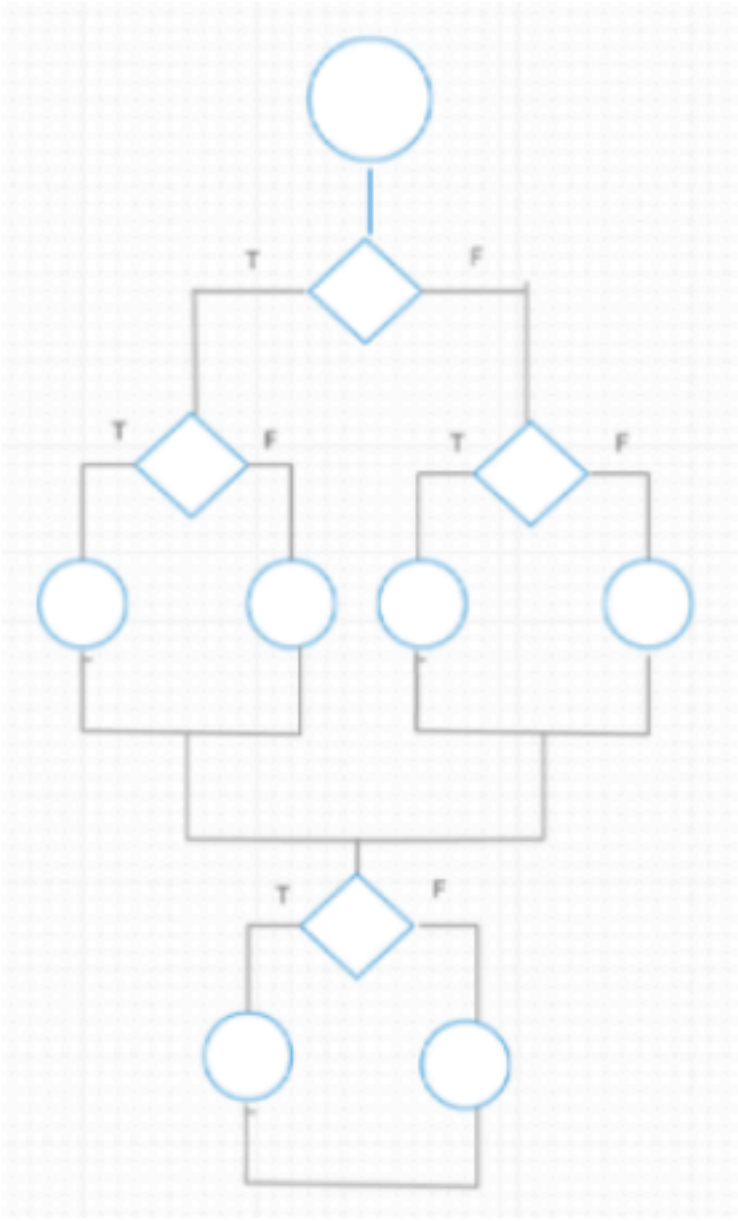
  Test Case 2: a=3, b=4, c=4, X=25

4. Given the code below, how many test cases are needed to achieve 100% multiple condition coverage?

```
If a < 10 or b < 5 or c > 15 or d > 2
        X = 10;
Else
        X = 20;
```

- ○ 4
- ○ 2
- ○ 8
- ◉ 16

5. What is the cyclomatic complexity of the given control flow diagram?



- (●) 5
- ( ) 2
- ( ) 8
- ( ) 4

6. Given the code below, what is the correct set of DU Paths?

```
Get a, b
X = 0                    } Node 1

If a >= 5 (Predicate I)
        Then c = x + 3 (Node 2)
        Else c = 0 (Node 3)

If b < 4 (Predicate II)
        Then b = c + 4 (Node 4)
        Else b = c + 2 (Node 5)
```

- Def1(a) = USEI(a)
  Def1(b) = USEII(b)
  Def1(x) = USE2(x)
  Def2(c) = USE4(c)
  Def3(c) = USE4(c)
- **Def1(a) = USEI(a)**
  Def1(b) = USEII(b)
  Def1(x) = USE2(x)
  Def2(c) = USE4(c) || USE5(c)
  Def3(c) = USE4(c) || USE5(c)
- Def1(a) = USEI(a)
  Def1(b) = USEII(b)
  Def2(c) = USE4(c) || USE5(c)
  Def3(c) = USE4(c) || USE5(c)
- Def1(a) = USEI(a)
  Def1(b) = USEII(b)
  Def1(x) = USE2(x)
  Def2(c) = USE4(c) || USE5(c)

7. Based on the code and the DU paths below, what coverage does the test case provide?
A = 2; B=2

```
Get a, b
X = 0          } Node 1

If a >= 5 (Predicate I)
        Then c = x + 3 (Node 2)
        Else c = 0 (Node 3)

If b < 4 (Predicate II)
        Then b = c + 4 (Node 4)
        Else b = c + 2 (Node 5)
```

- ○ 4/7
- ◉ 3/7
- ○ 5/7
- ○ 2/7

8. True or False? Huang's Theorem helps reduce the number of iterations over a path during anomaly testing.

- ◉ True
- ○ False

9. Static analysis techniques are applied during program execution.

- ○ True
- ◉ False

10. What is the correct path condition representation for a False False path given the code below?

```
If (x <= 0) or (y <= 0)          (0)
then
        x = x2                   (1)
        y = y2
    else
        x = x + 1                (2)
        y = y + 1
    endif
if (x < 1) or (y < 1)
        then
        x = x + 1                (3)
        y = y + 1
        else
        x = x – 1                (4)
        y = y - 1
        endif
```

- [ ] [(x_0 > 0) and (y_0 > 0)] and [(x_0 >= 0) or (x_0 >= 0)]
- [ ] [(x_0 > 0) or (y_0 > 0)] and [((x_0 >= 0) or (y_0>= 0)]
- [x] [(x_0 > 0) and (y_0 > 0)] and [((x_0 >= 0) and (y_0>= 0)]
- [ ] [(x_0 > 0) or (y_0 > 0)] and [(x_0 >= 0) and (y_0 >= 0)]