Que-

Consider a simple program to classify a triangle. Its inputs are a triple of positive integers (say x, y, z) and the data type for input parameters ensures that these will be integers greater than 0 and less than or equal to 100. The program output may be one of the following words: [Scalene; Isosceles; Equilateral; Not a triangle]

1. Design the boundary value test cases.
2. Draw the cause effect graph. and then Design the test cases using Decision table technique.
3. Consider the program above. Write down the code for it and design the CFG graph.
4. Design statement and branch coverage test cases.
5. Design path coverage test cases using McCabe's cyclomatic Metric

Ans:

Code: package OOPS_lab;

```
public class triangle {

    static void tri(int x, int y, int z) {
        if (x >= y + z || y >= x + z || z >= x + y)
        System.out.println("Not a Triangle!");
        else if (x == y && x == z)
        System.out.println("Equilateral Triangle!");
        else if (x != y && y != z)
        System.out.println("Scalene Triangle!");
        else
        System.out.println("Isosceles Triangle!");
        }
        public static void main(String[] args) {
        tri(50,70,90);
        }
}
```

Design the boundary value test cases.

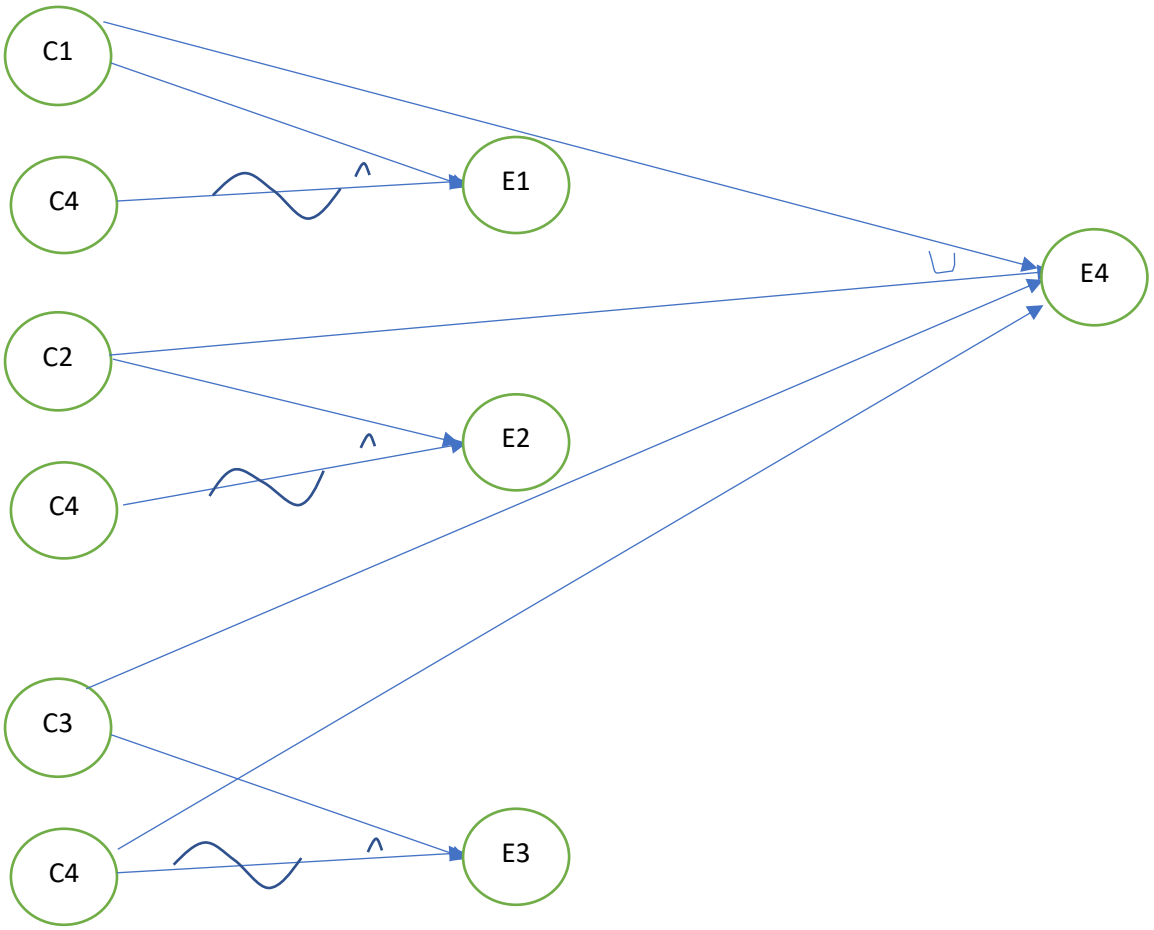
Test Case	X	Y	Z	Expected O/P
1	50	50	1	Isosceles
2	10	30	20	Not a triangle
3	50	50	50	Equilateral
4	50	50	99	Isosceles
5	50	50	100	Not a triangle
6	50	70	90	Scalene
7	50	2	50	Isosceles
8	50	99	50	Isosceles
9	50	100	50	Not a triangle
10	1	50	50	Isosceles

11	2	50	50	Isosceles
12	99	50	50	Isosceles
13	100	50	50	Not a triangle

2. Draw the cause effect graph. and then Design the test cases using Decision table technique.

Causes: -
C1: Side ‘X’, ’Y’ and ‘Z’ are equal.
C2: Side ‘X’ and ’Y’ are equal.
C3: Side ‘X’, ‘Y’ and ‘Z’ are different.
C4: sum of two sides is greater than third one.

Effects: -
E1: Equilateral Triangle.
E2: Isosceles Triangle.
E3: Scalene Triangle.
E4: Not a Triangle.

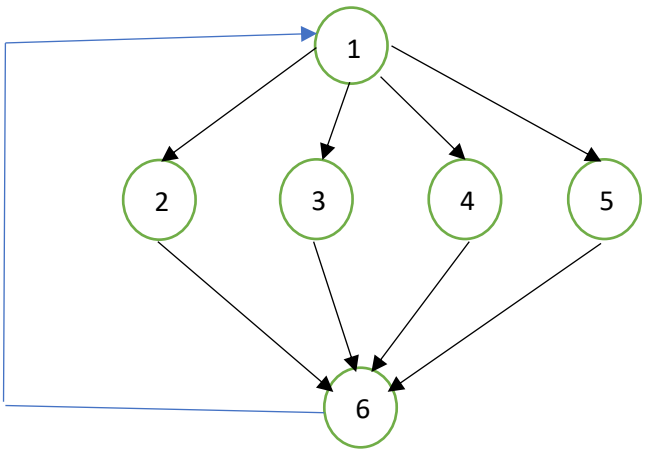


Decision Table:

Action	Tc1	Tc2	Tc3	Tc4
C1	1	0	0	0
C2	0	1	0	0
C3	0	0	1	0
C4	0	0	0	1
E1	1	0	0	0
E2	0	1	0	0
E3	0	0	1	0
E4	0	0	0	1

3. Consider the program above. Write down the code for it and design the CFG graph.

```
public class triangle {
        static void tri(int x, int y, int z) {
                1.if (x >= y + z || y >= x + z || z >= x + y)
                  System.out.println("Not a Triangle!");
                2.else if (x == y && x == z)
                  System.out.println("Equilateral Triangle");
                3.else if (x != y && y != z)
                  System.out.println("Scalene Triangle!");
                4.else
                  System.out.println("Isosceles Triangle!");
                5.}
                  public static void main(String[] args) {
                6.tri(50,70,90);
                  }
        }
```



4. Design statement and branch coverage test cases.
Statement Coverage Test Cases: -
Test Cases 1: - {(x = 100, y = 50, z = 50)}

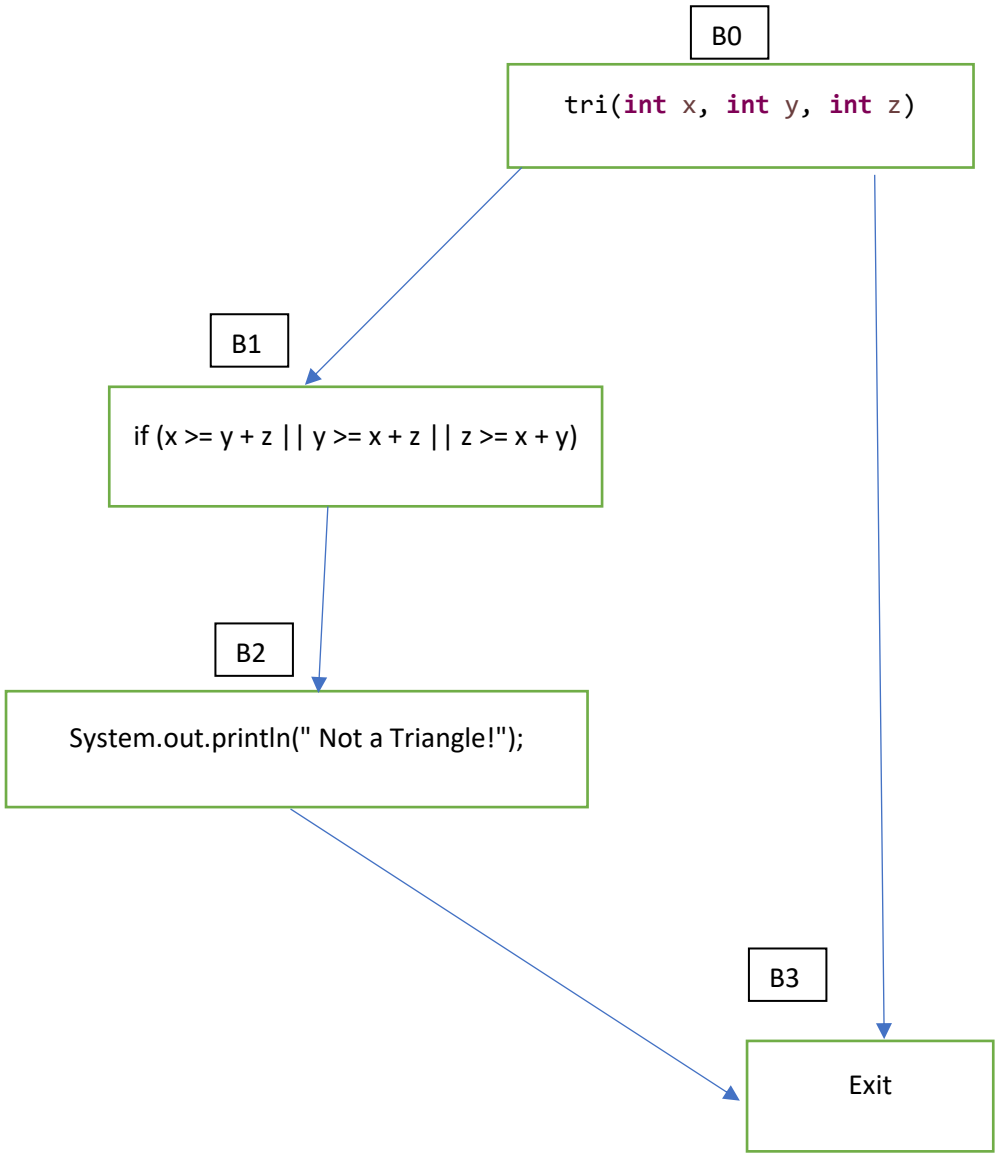
Test Case 2: - {(x = 50, y = 70, z = 90)}

Test Case 3: - {(x = 50, y = 50, z = 50)}

Test Case 4: - {(x = 50, y = 50, z = 1)}

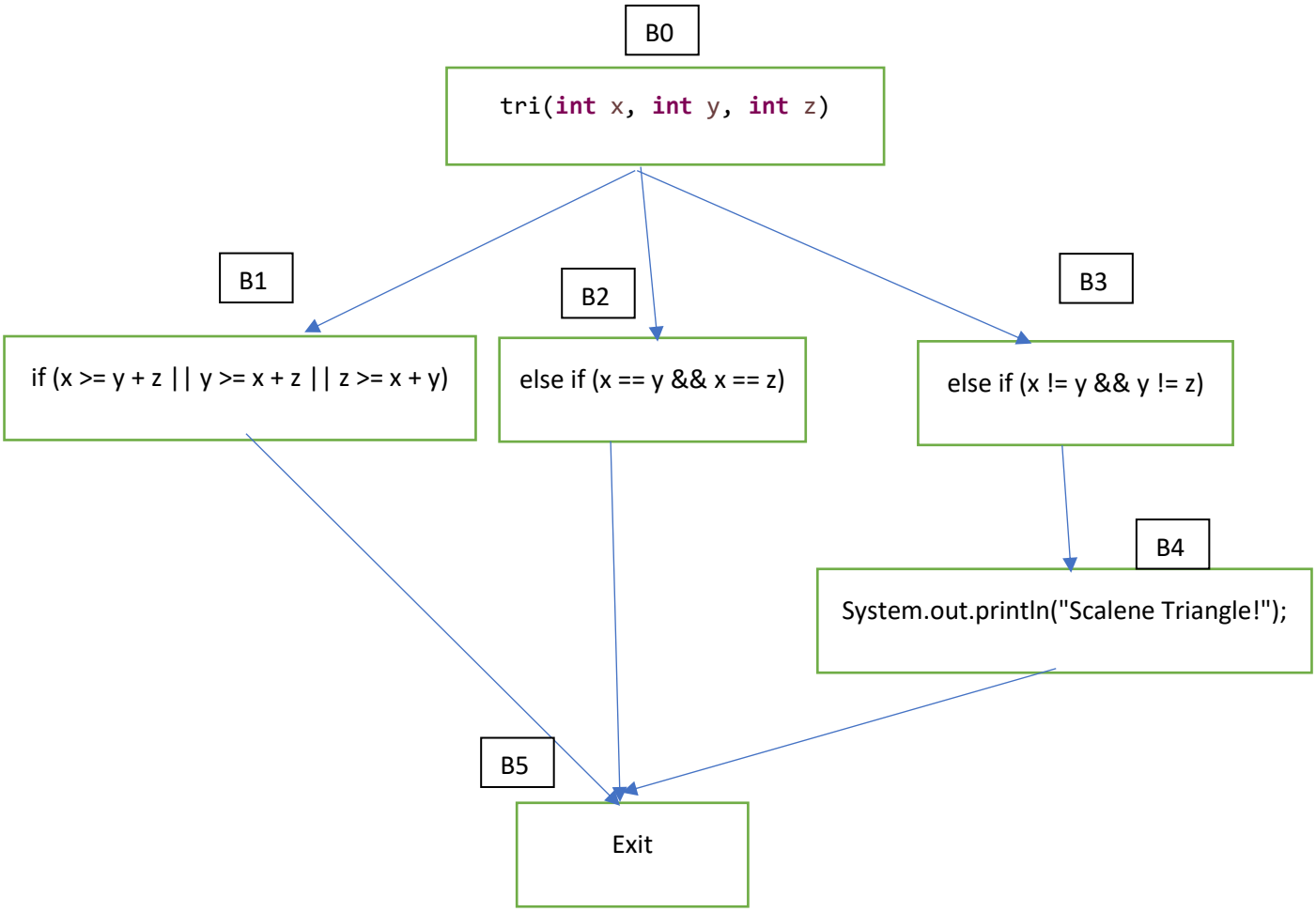
Branch Coverage Test Case:

Test Cases 1: -not a triangle
{(x = 100, y = 50, z = 50)}

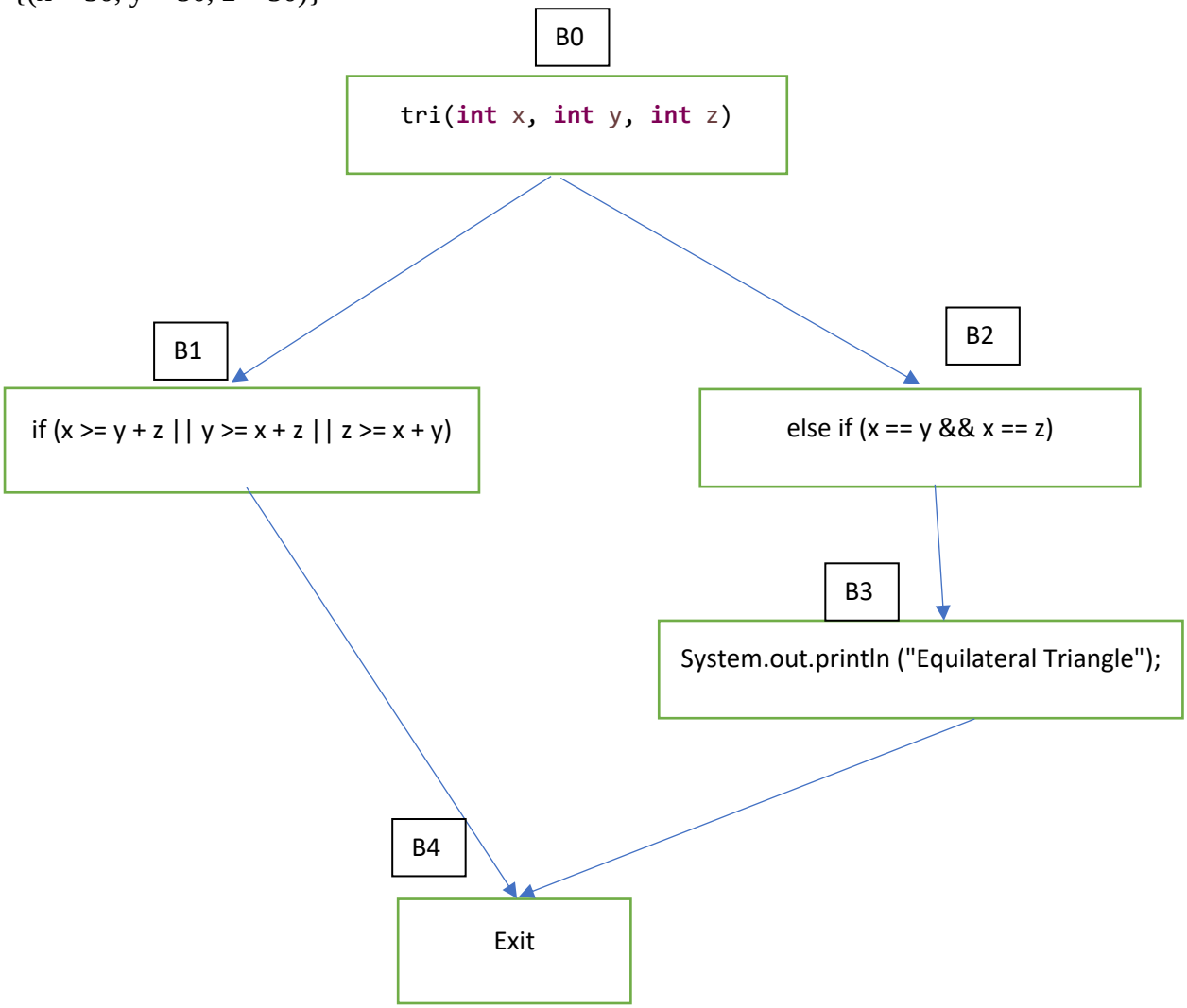


Test Case 2: Scalene triangle

{(x = 50, y = 70, z = 90)

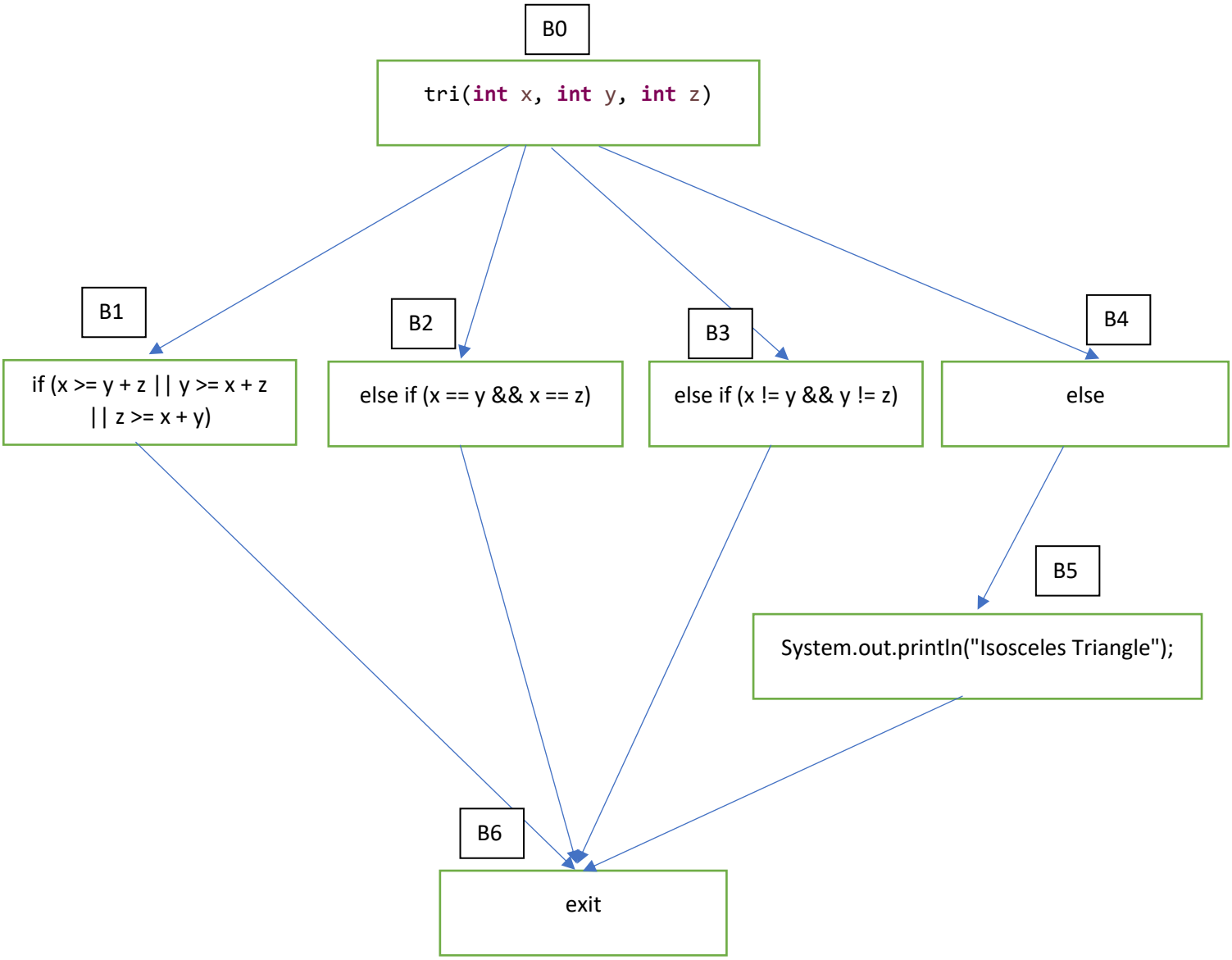


Test Case 3: - Equilateral Triangle
{(x = 50, y = 50, z = 50)}



Test Case 4: Isosceles Triangle

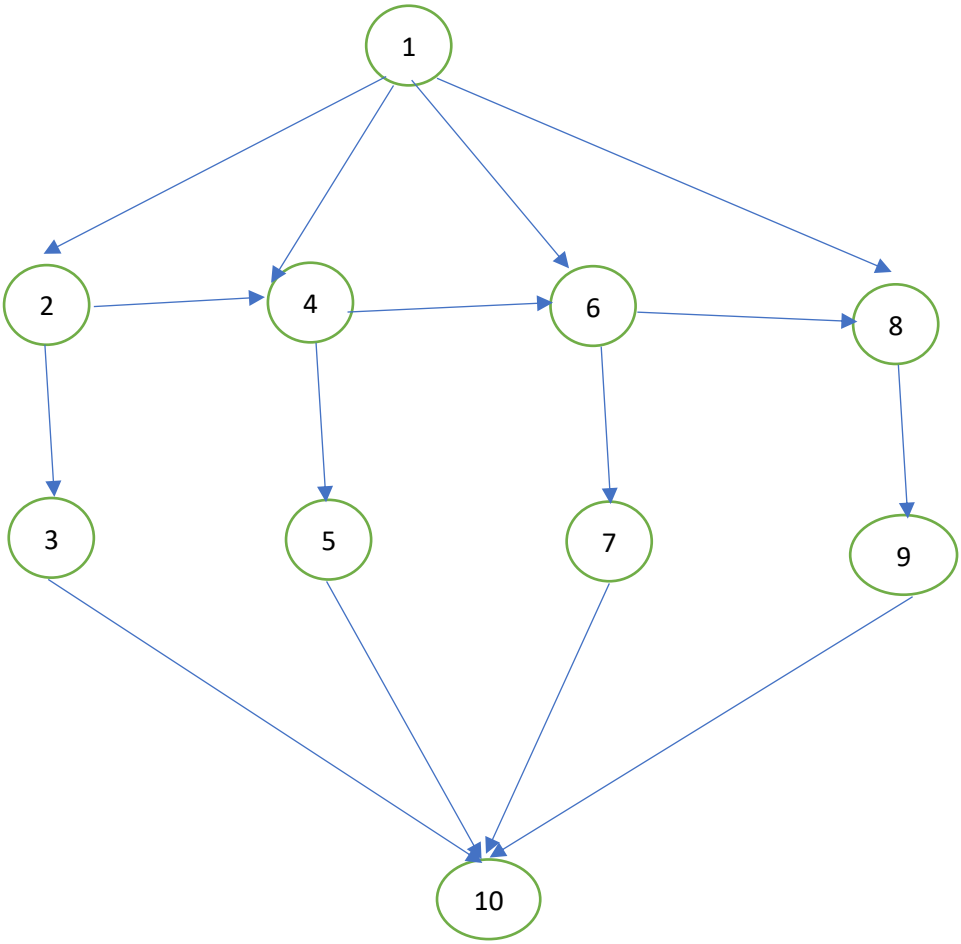
{(x = 50, y = 50, z = 1)}



5. Design path coverage test cases using McCabe's cyclomatic metric.

1. tri(int x, int y, int z)
2. if (x>= y + z || y > x + z || z > = x + y)
3. System.out.println("Not a Triangle!");
4. else if (x == y && x == z)
5. System.out.println("Equilateral Triangle!");
6. else if (x != y && y != z)
7. System.out.println("Scalene Triangle!");
8. else
9. System.out.println("Isosceles Triangle");

10. exit



N = 10, E = 12

Cyclomatic Metrix = E – N +2 = 12-10+2 = 4.

Set of all execution paths: {(B0, B1, B5, B9), (B0, B2, B6, B9), (B0, B3, B7, B9), (B0, B4, B8, B9)}