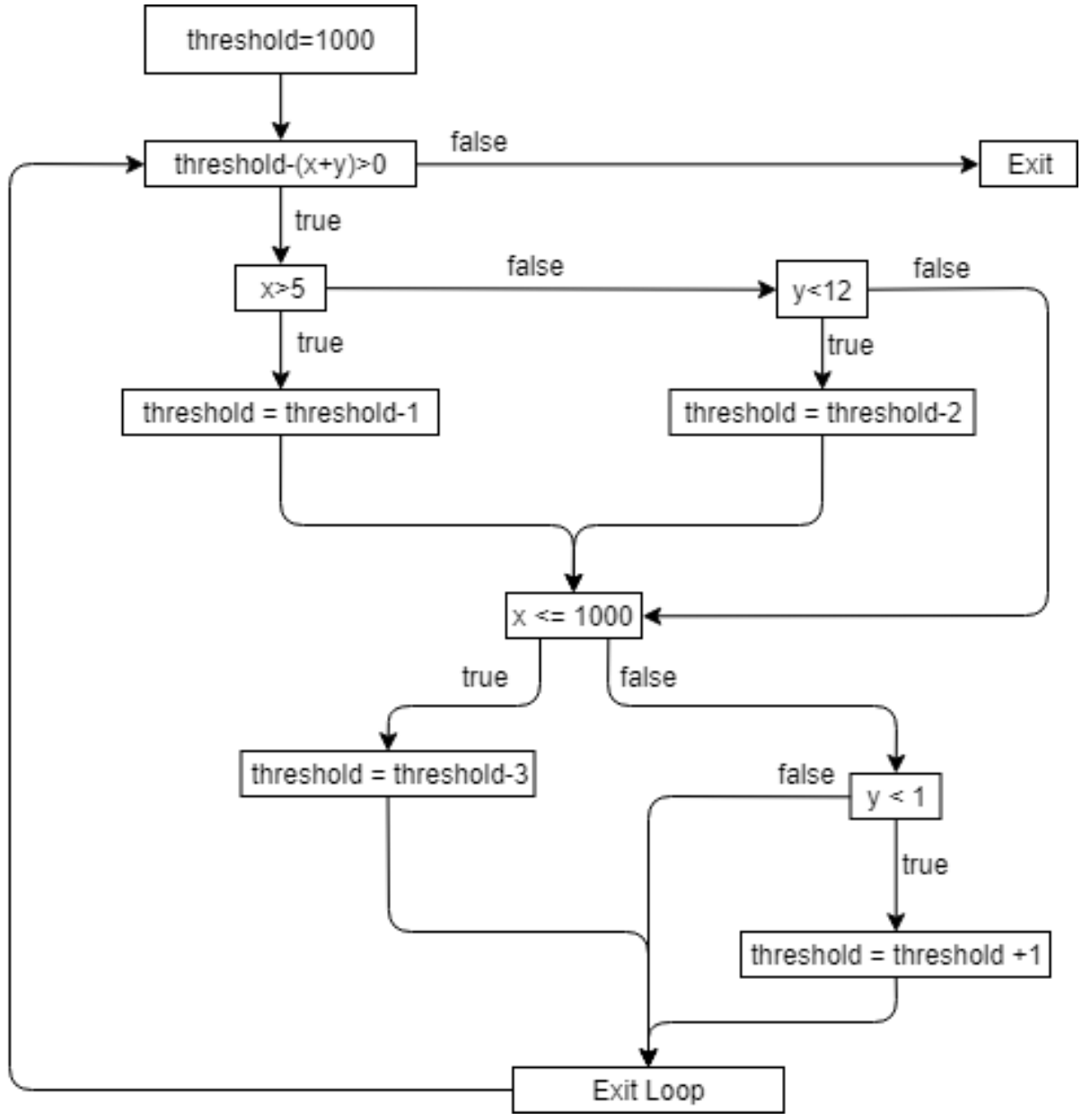
Cohort Exercise 7:



Cohort Exercise 8:

There are 5 statements:

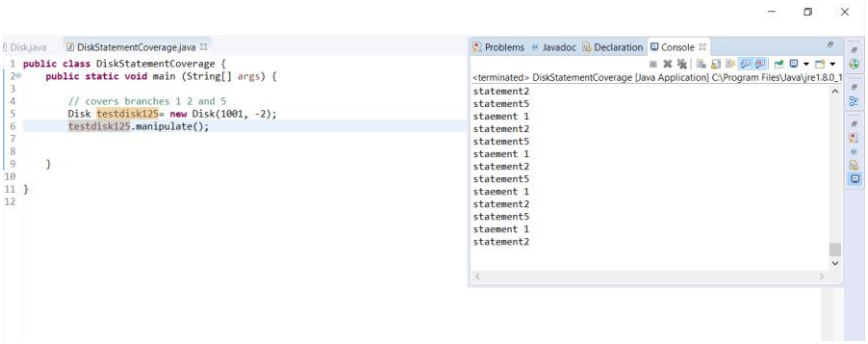
1) (threshold - (x + y)) > 0
2) x > 5
3) y <= 12
4) x <= 1000
5) y < 1

To check for statement coverage, print statements were added after each branch was executed

```
Disk.java  DiskStatementCoverage.java
public class Disk {
    private int x;
    private int y;

    Disk(int x, int y) {
        this.x = x;
        this.y = y;
    }

    public void manipulate () {
        int threshold = 1000;

        while ((threshold - (x + y)) > 0) {
            System.out.println("staement 1");
            if (x > 5) {
                //for debugging
                System.out.println("statement2");
                threshold = threshold - 1;
            }
            else if (y <= 12) {
                System.out.println("statement3");

                threshold = threshold - 2;
            }
            if (x <= 1000) {
                System.out.println("statement4");

                threshold = threshold - 3;
            }
            else if (y < 1) {

                System.out.println("statement5");

                threshold = threshold + 1;
            }
        }
    }
}
```

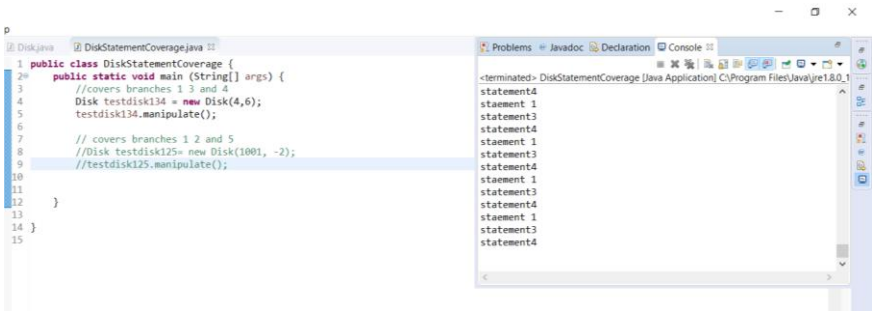
If the statement number is printed, then the statement is covered. To cover all branches, statement 1 must be covered because all the rest of the statements are in the while loop. So for all tests we must ensure x=+y<1000. The 'else if' statements would only be reached if the 'if' statements fail.

We can start by group the statements together, we can group statement 5 and 2 together. To reach statement 2 (y<1) x must be > 1000. Since 1000>5, this effectively covers statement 2, x>5. So the first test must ensure that: x+y<1000, x>1000 and y<1. For the first test, x=1001 and y=-2. This covers statement 1,2 and 5



Then we can group the remaining statements (3&4) together. To cover statement 3, x <5, and y <12. Since 1000 < 5, x<5 would already cover statement 4. So in the second test, y<=12, x<5 and x+y<1000.

For the second test to cover statement 1,3 and 4. X=4 and y=6.



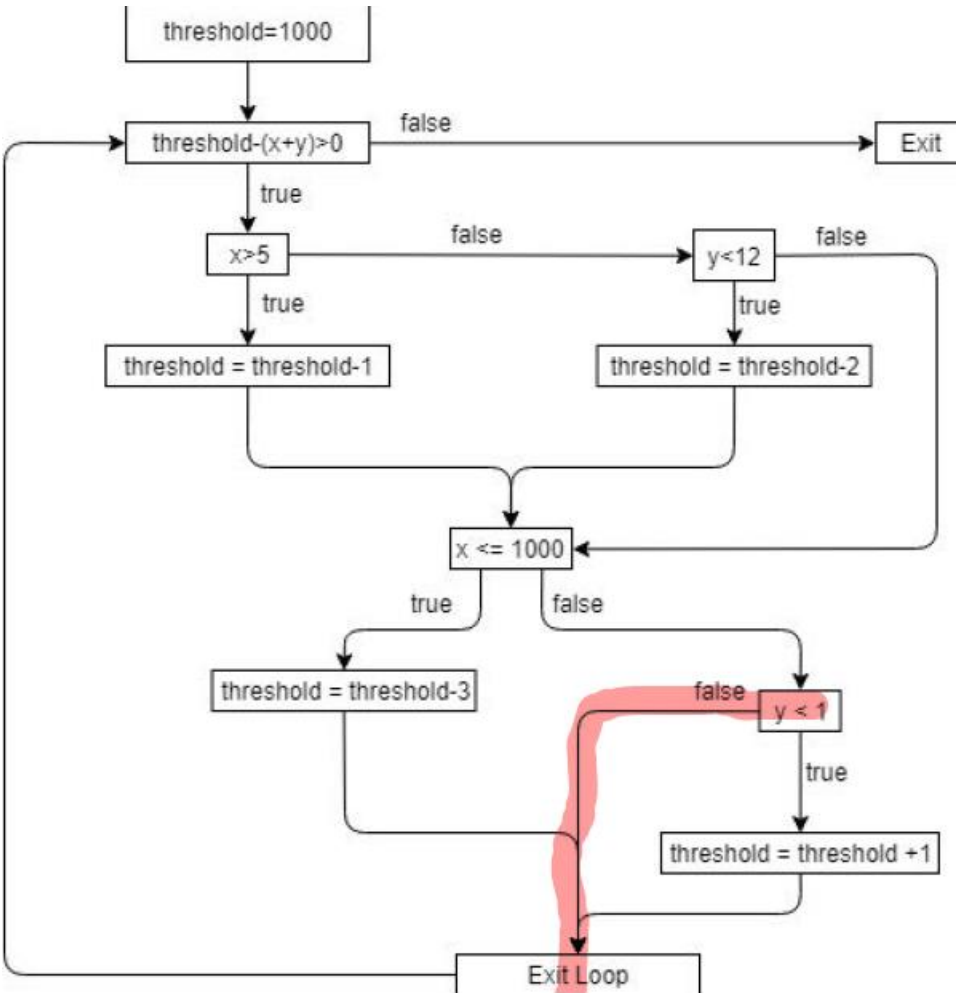
Therefore, I would need a minimum of **2** test cases.

Cohort Exercise 9

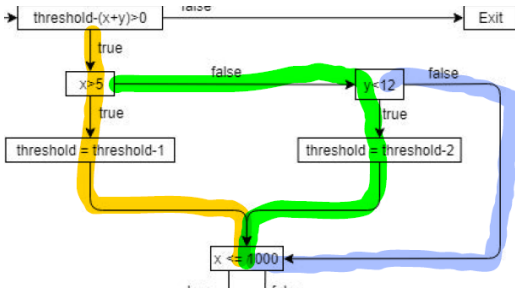
All branches cannot be covered. The red branch cannot be covered. To cover the branch, the following conditions must be satisfied:

1) X+y < threshold
2) X > 1000
3) Y >= 1

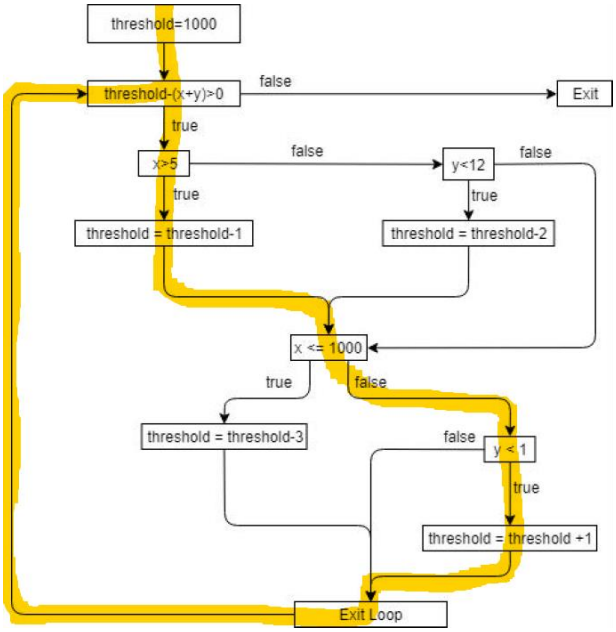
However if conditions 2 and 3 are fulfilled. Then X+Y would be > 1000 which is the initial value of threshold. If conditions 2 and 3 are fulfilled the program would not even enter the while loop and reach the statement y<1. Therefore the red branch cannot be reached.



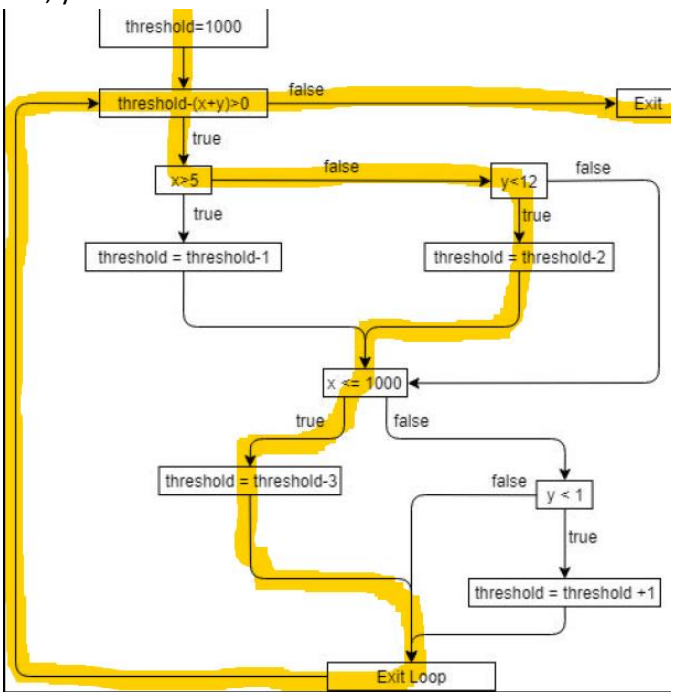
The other branches can be covered in a minimum of 3 tests. From the top half of the diagram (before x<=1000) we already have 3 distinct branches. Therefore, we need at least 3 test cases.



1) first test would infinite loop. AS explained above we can group true branch of x>5 and false branch of x<=1000 together, covering them both in 1 test case. Since threshold never decreases, values of x and y do not change and initial values of x and y are < threshold. This will result in an inifinite loop. X=1001 y=-2.

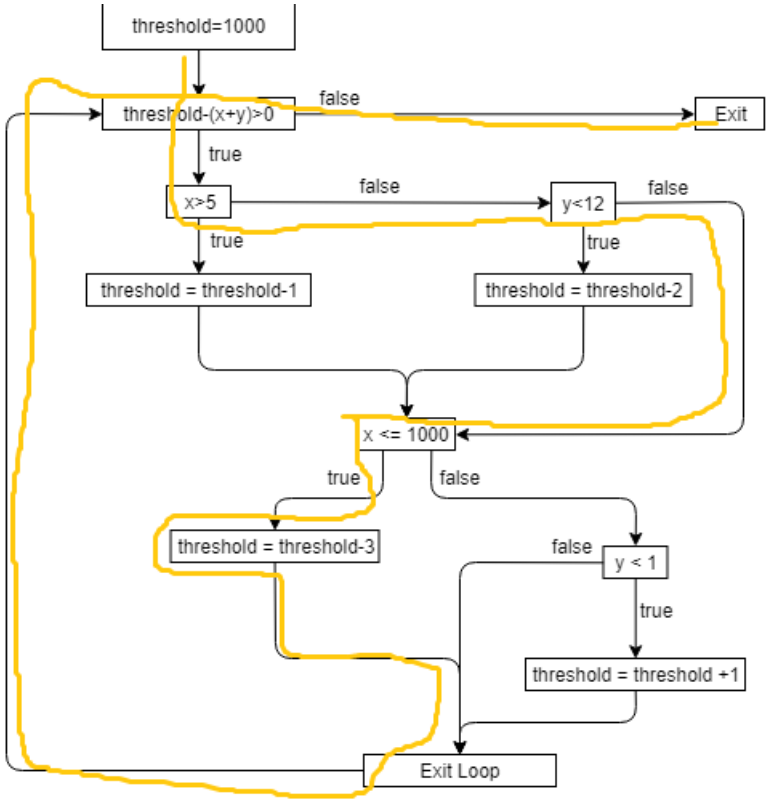


2) x=4, y=6



X=4 is <5 and <1000. Y=6 is <12. In these branches. Threshold would reduce by 5 in each iteration through the while loop. The values of x and y remain constant. Therefore as threshold keeps decreasing x and y would ultimately become greater than threshold. So threshold-(x+y) would be <0 covering the false loop and exiting the loop.

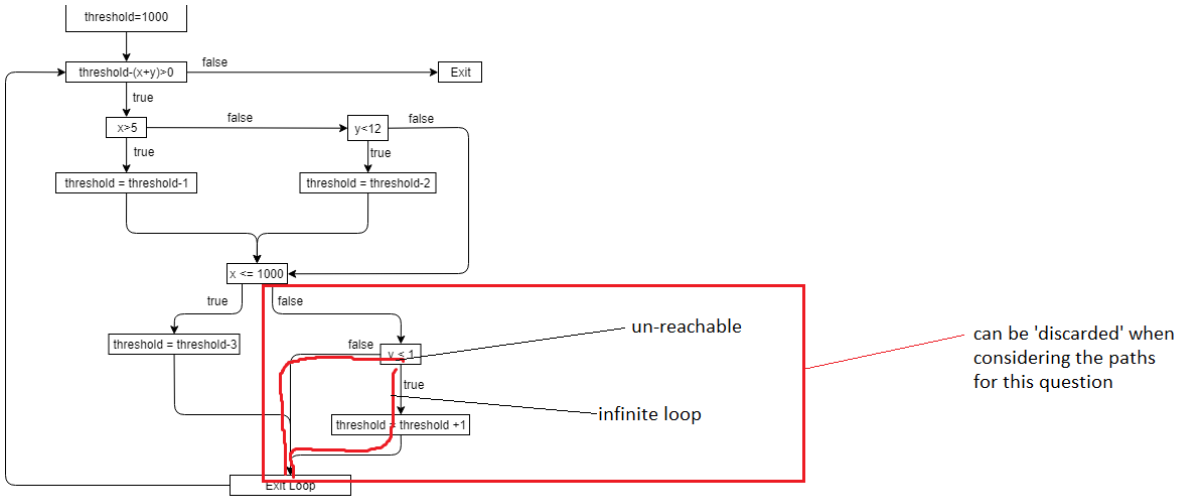
3) X=4 and Y=13.



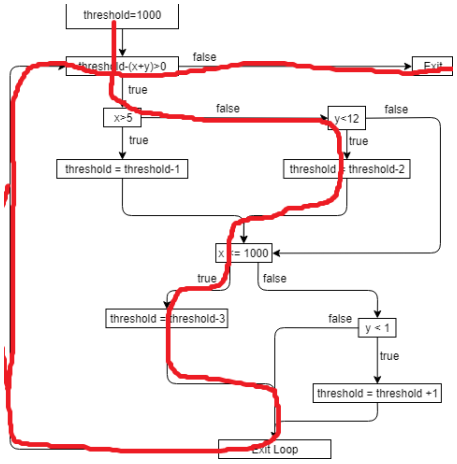
X=4 <5 <1000. Y=13>12. With each iterations of the while loop, threshold would decrease by 5 therefore, since values of x and y remain the same. Threshold would gradually decrease and be less than (x+y). resulting in threshold –(x+y) to be <0 and cauing the program to exit the while loop.

Cohort exercise 10

Since loop can only execute at most 100 times. This rules out the infinite loop in the program i.e. when x > 1000 and y<1. Since the red branch is Cohort Ex 9 is un-reachable. This means the branches leading from x <=1000 false cannot be considered.



This leaves us with four possible paths. The three branches leading from threshold – (x+y) >0 statement and the path that covers the threshold –(x+y) > 0 false statement. However one of these paths, will execute for a minimum of 197 times.

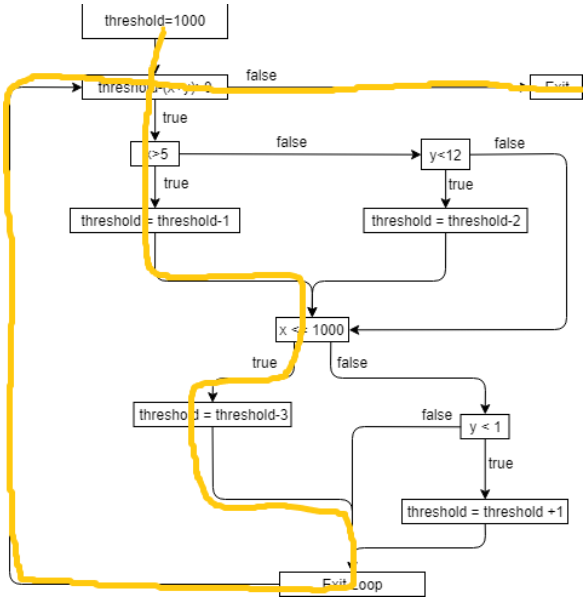


In this path, X <=5 and Y <12. Therefore the maximum values of x and y are 5 and 11 respectively. In this path, threshold decreases by 5 every time the program is executed. This means for threshold to be < x+y. The program must run for a minimum of [1000-(11+5)]/5=196.8. **197** rounds before it terminates. Since 197>100. This path is not valid.

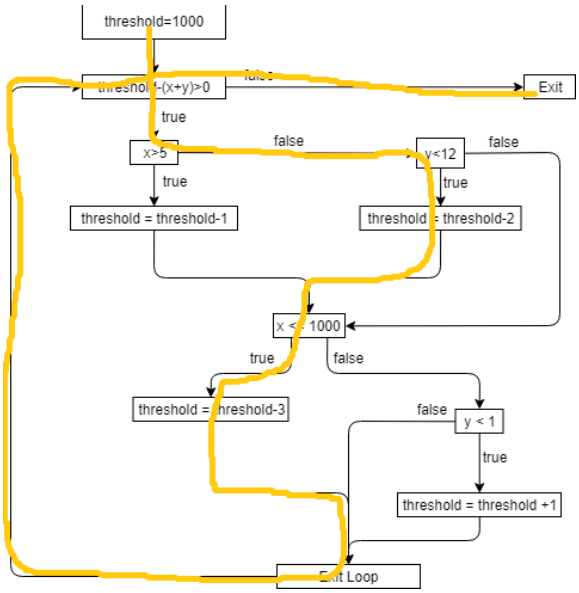
Therefore this leaves us with **3** possible paths that can execute for at most 100 times.

1) X+Y > 1000 (eg. x=y=1000)
   Program will execute only once.

2) This path where X > 5 and Y < 1000-X and threshold decreases by 4 each time the program executes, the program can be executed for at most 100 times. Consider the values where X=500 and Y=100 The program would run for [1000-(500+100)]/4=*100* times.



3) Path where X <=5 and Y>=12. And threshold decreases by 5 each time program executes. Consider X=5 and Y=900. The program would execute for [1000-(5+900)]/5=19 times. 19 <=100.

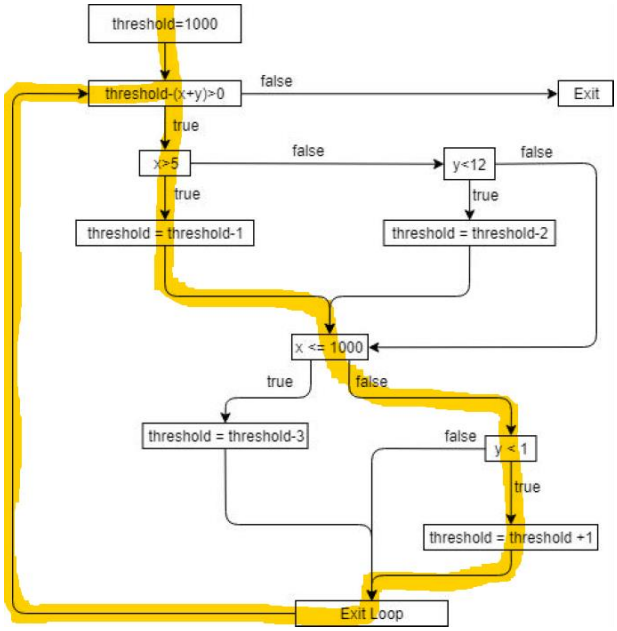


Cohort Exercise 11

Excluding the un-reachable branch discussed in Cohort Exercise 9. Since each is, else if statement only has **one** condition. There are not complex conditions in the statements of the control flow diagram. Since Cohort Exercise 9 covers all reachable branches, the test suite would also cover conditions. (except the unreachable one)

Cohort Exercise 12

If statement



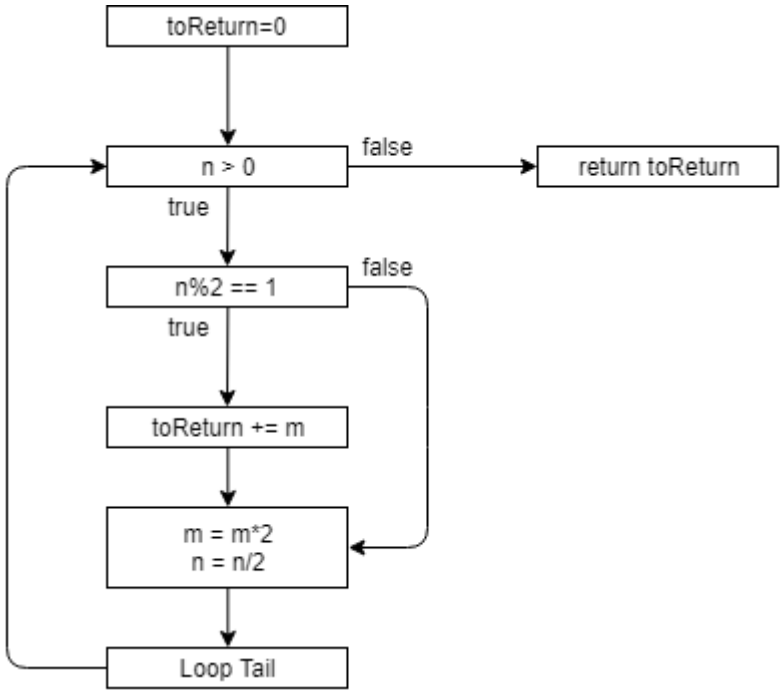
This path would result in an infinite loop because at every execution threshold = threshold +1-1 therefore threshold will never change. Additionally since initial values of x+y are less then 1000 x and y do not change throughout the iteration of the program, x and y will always be < 1000. Therefore, when then following conditions are fulfilled an infinite loop occurs:

1) X+y < 1000
2) X > 1000
3) Y <1

Test case: X = 1001, Y = -2

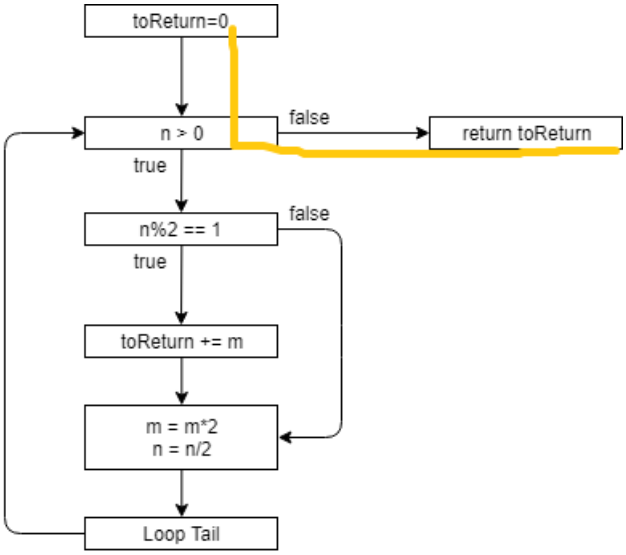
Cohort Exercise 13

Control flow:



1) testcase 1

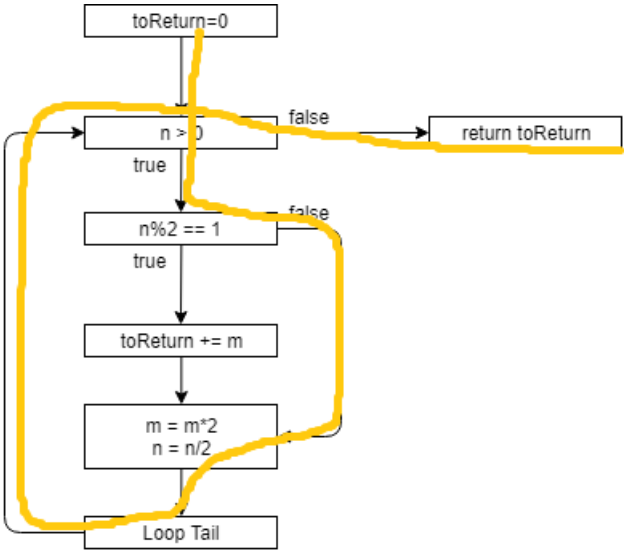
   Branch 1: n<-2, m=5test



2) testCase 2

   n>0 and n is even (e.g. n=4, m=3)

   n = n/2. At every execution of program, n will be halved eventually becoming 0. Program terminates



3) testcase 3

n >0 and n is odd. Eg. (n=7 and m=2)

At every execution of program n will be halved and it will still be odd 7/2 =3. 3/2=1. Eventually n will reach 0, ½=0. And the program terminates