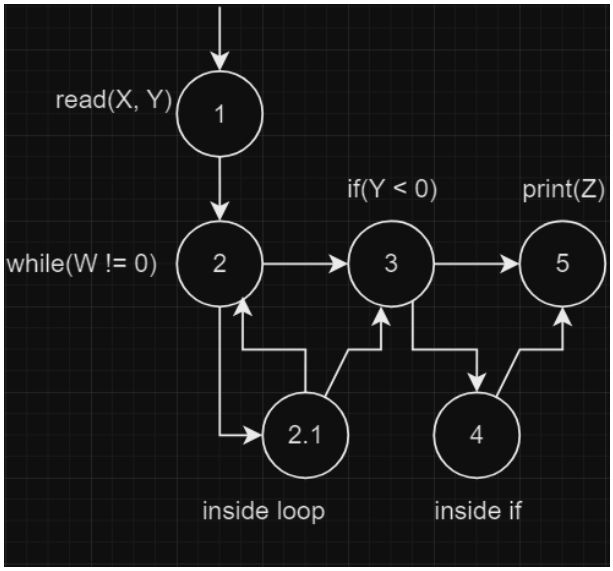
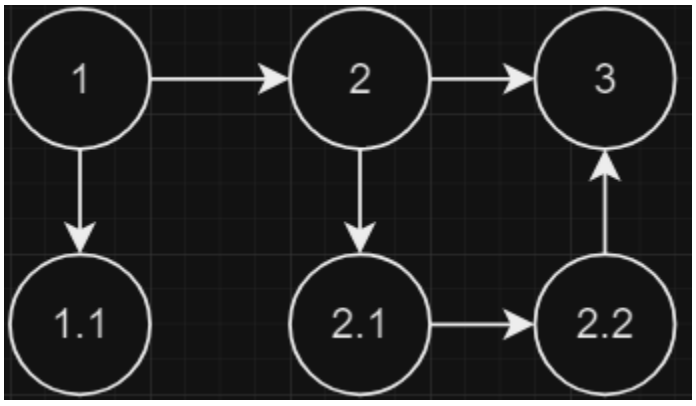
Question 1



There are no infeasible paths. All parts of the code can be reached and no conditions lead to unreachable blocks of code.

Node Coverage: {1, 2, 2.1, 2, 3, 4, 5} Test: (1, -1)
Edge Coverage: {1, 2, 2.1, 2, 3, 4, 5}, {1, 2, 3, 5} Test: (1, -1), (0, 0)

Question 2



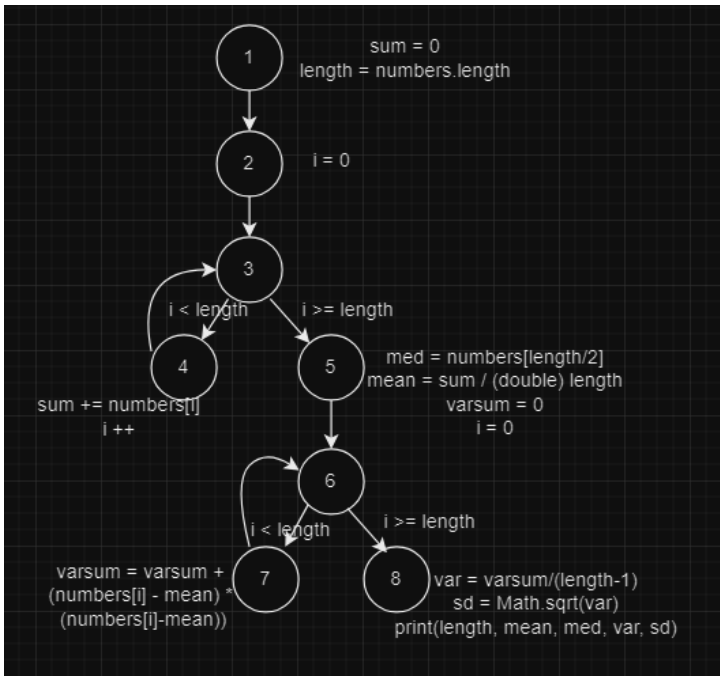
1: if statement
1.1: throw exception
2: while loop
2.1: if statement
2.2: inside if statement

3: return result

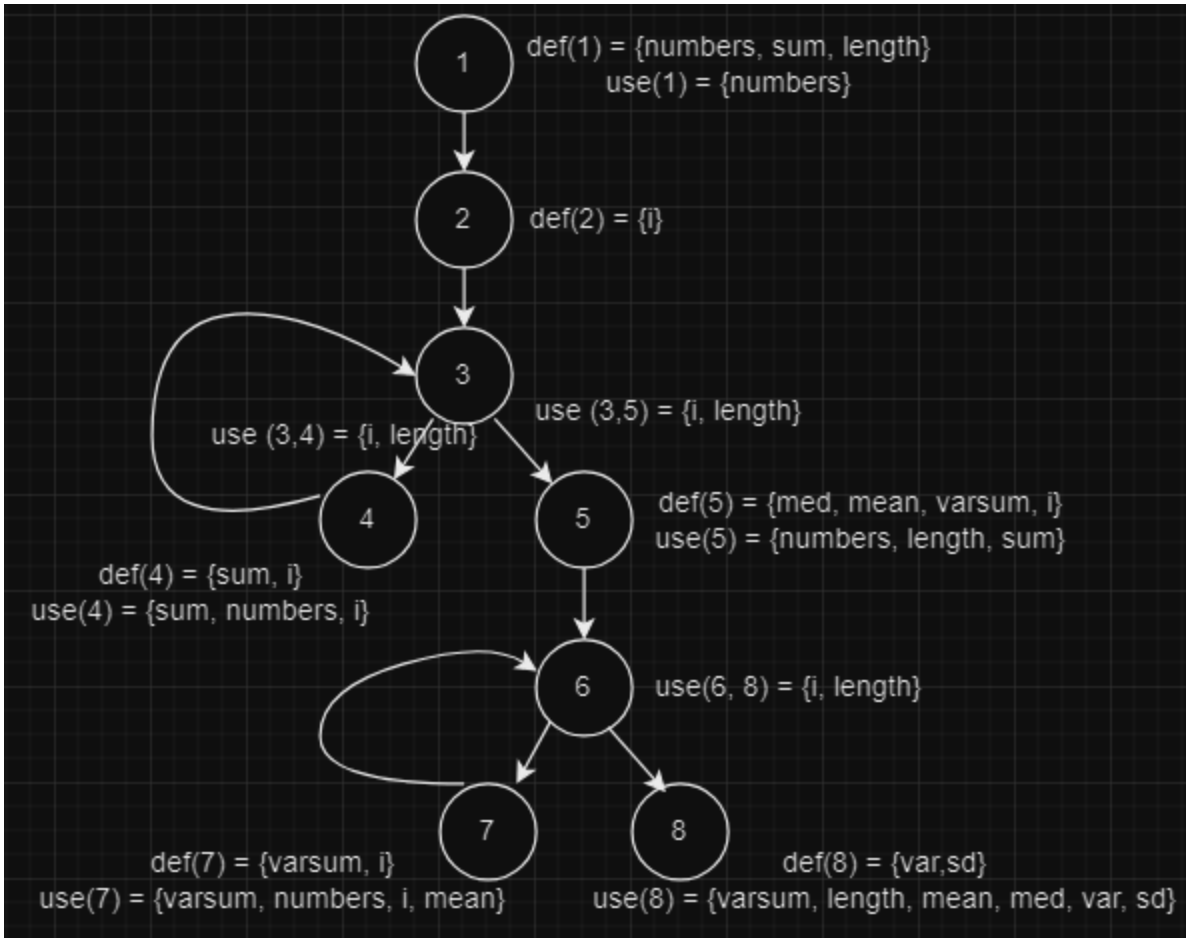
Test requirements for Node Coverage:{1, 1.1, 2, 2.1, 2.2, 3}
Test requirements for Edge Coverage:{(1,1.1), (1,2), (2, 2.1), (2, 3), (2.1, 2.2), (2.2, 3)}
Test requirements for Edge Pair Coverage: {(1, 2, 3), (1, 2, 2.1), (2, 2.1, 2.2), (2, 2.1, 2), (2.1, 2, 2.1), (2.1, 2.2, 2), (2.2, 2, 3), (2.1, 2, 3)}

Node coverage but not edge coverage {null, “sa”}
Edge coverage but not EPC: not possible
EPC {“saaa”}
TR(PPC): 1 → 1.1, 1→2→2.1→2.2→2→3, 1→2→2.1→2→3
PPC: {null, “sa”, “saaa”}

Question 3



CFG



DFG

Du pairs

numbers	14, 15, 17
sum	14, 15, 45
length	13, 15, 16, 18
i	23, 24, 25, 56, 57
var	
varsum	78
mean	58, 57
med	58

sd	

Du paths

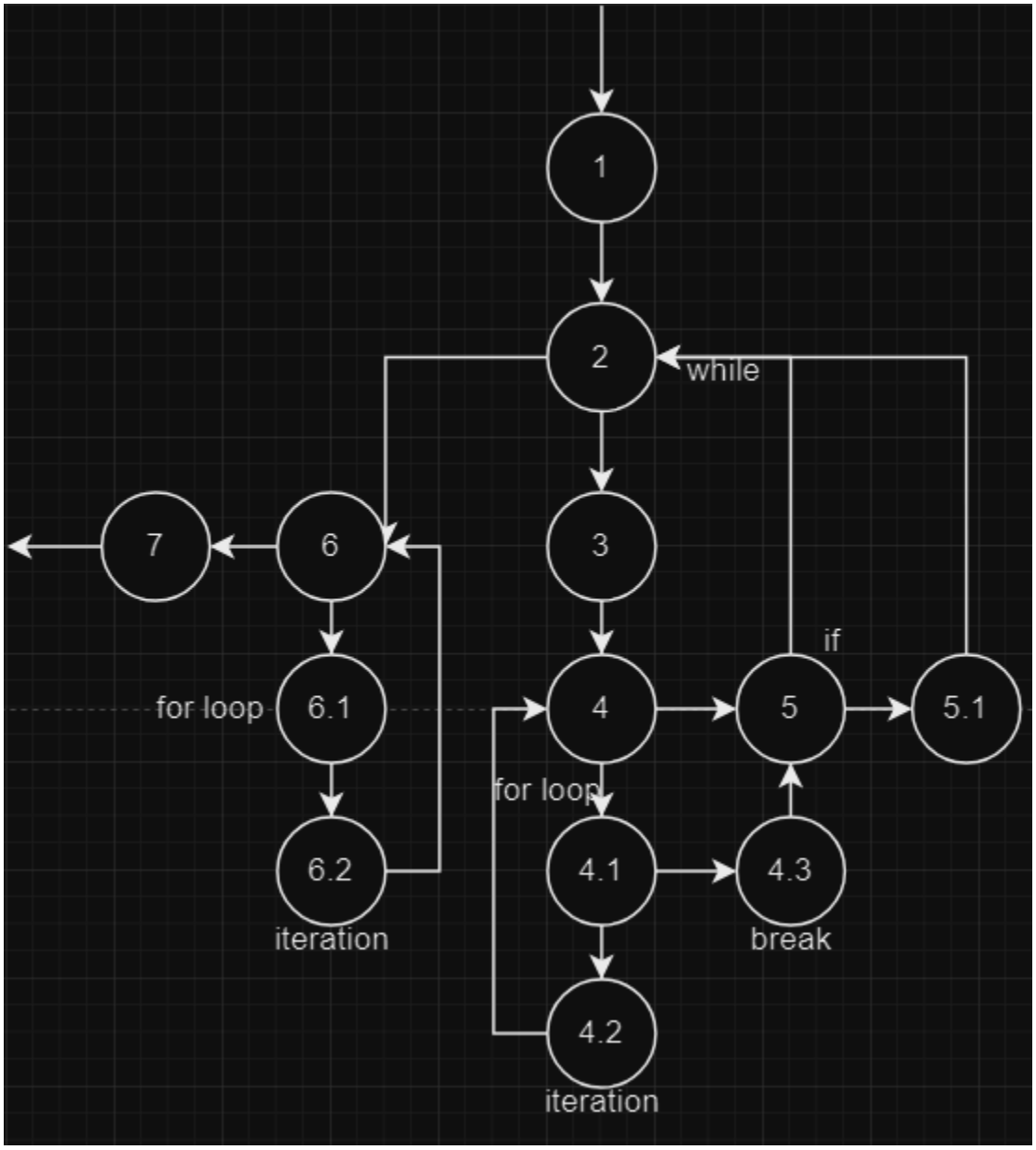
numbers	[1, 2, 3, 4] [1, 2, 3, 5] [1, 2, 3, 5, 6, 7]
sum	[1, 2, 3, 4] [1, 2, 3, 5] [4, 3, 5]
length	[1, 2, 3, 5] [1, 2, 3, 5, 6, 8] [1, 2, 3, 4] [1, 2, 3, 5, 6, 7] [1, 2, 3, 5, 6, 8]
i	[2, 3, 4] [2, 3, 5] [4, 3, 4] [4, 3, 5] [5, 6, 7] [5, 6, 8] [7, 6, 7] [7, 6, 8]
var	
varsum	[7, 6, 8]
mean	[5, 6, 7] [5, 6, 8]
med	[5, 6, 8]
sd	

Test cases: numbers (20), numbers (5, 10, 15)

5: if a length of 0 is provided to the program, the program will run into errors as a line within the code evaluates some number divided by length. If length is 0, dividing by 0 would result in errors.

Question 4

CFG



2: if n = 0, numPrimes < n returns false

3:T:{1, 2, 3, 4, 4.1, 4.2, 4, 5, 5.1, 2, 3, 4, 4.1, 4.3, 5, 2, 6, 6.1, 6.2, 6, 6.1, 6.2, 6, 6.1, 6.2, 6, 7} input is 3