**REG No: Sp20-bse-056**

**Name: Zimam Ahmed**

Task 1:

1. int a=5;
2. int b=6;
3. function (int a, int b)
4. {
5. if (a<=0)
6. return -1;
7. if(b<=0)
8. return -1;
9. int t;
10. if (b>a)
11. {
12. t=a;
13. a=b;
14. b=t;
15. }
16. t=a%b;
17. while (t!=0)
18. {
19. a=b;
20. b=t;
21. t=a%b;
22. }
23. return b;
24. }

**Statement coverage:**

Total Executable Statements: 21

Executed Statements:

Statement Coverage = (Executed Statements / Total Executable Statements) \* 100%

Statement Coverage = (12 / 21) \* 100% = 57.143%

So, the statement coverage for this code is 75%.

**Branch coverage:**

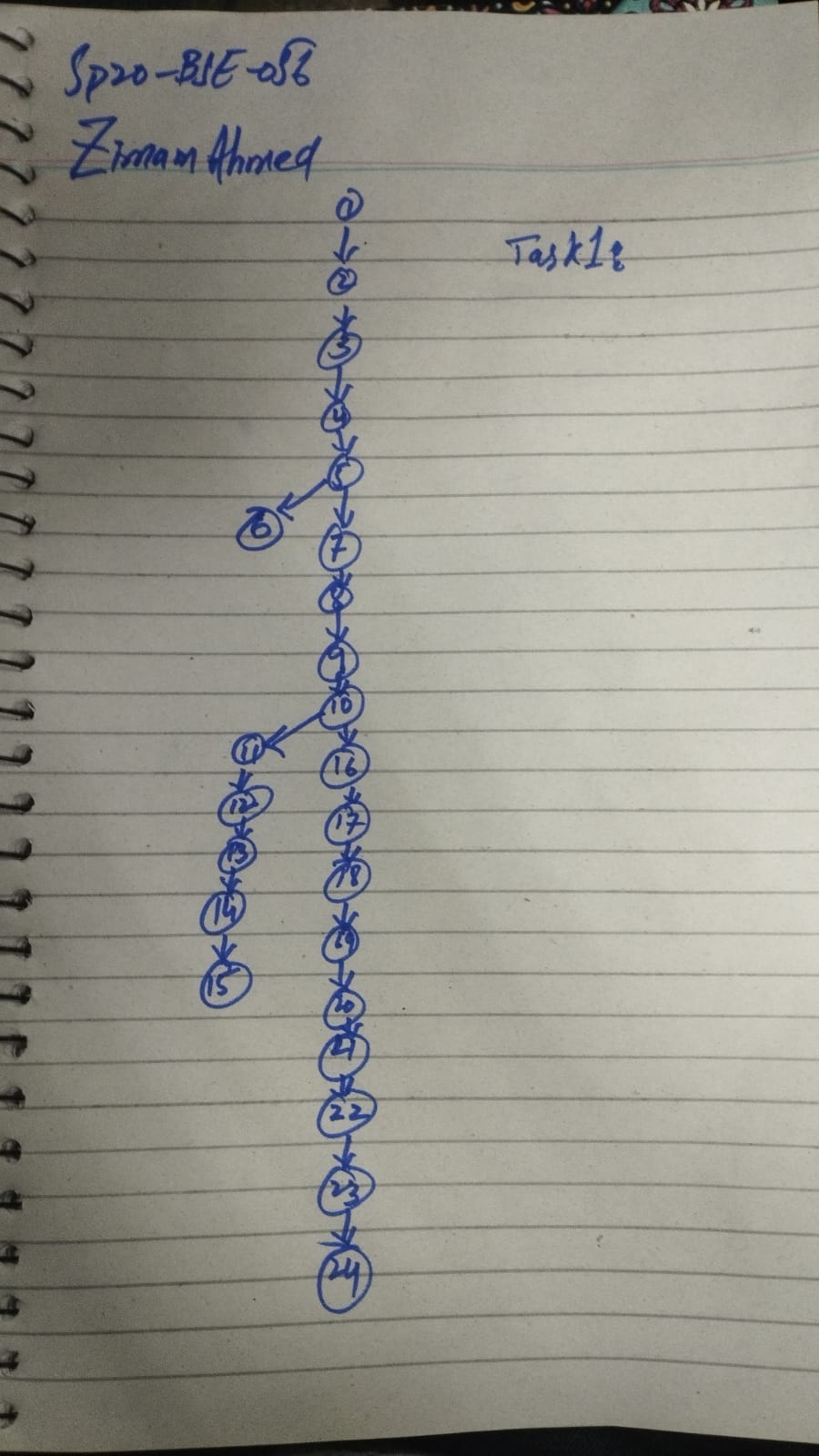
Total Branches: 2

Executed Branches: 2

Branch Coverage = (Executed Branches / Total Branches) \* 100%

Branch Coverage = (2 / 2) \* 100% = 100%

**Path coverage:**



**Cyclomatic complexity:**

Formula : V= E-N+2

V = 23-24+2 = 1

**TASK 2:**

1. int x=1;
2. int y=2;
3. int a=2;
4. int b=1;
5. if ((x<0) && (y<b))
6. {
7. x++;
8. y++;
9. }
10. else
11. {
12. x--;
13. y--;
14. }
15. y=x;
16. while (y>b)
17. {
18. y=y/2;
19. x=y;
20. }
21. x=0;
22. y=0;

**Statement coverage:**

Statement Coverage = (Number of Executed Statements / Total Number of Statements) \* 100%

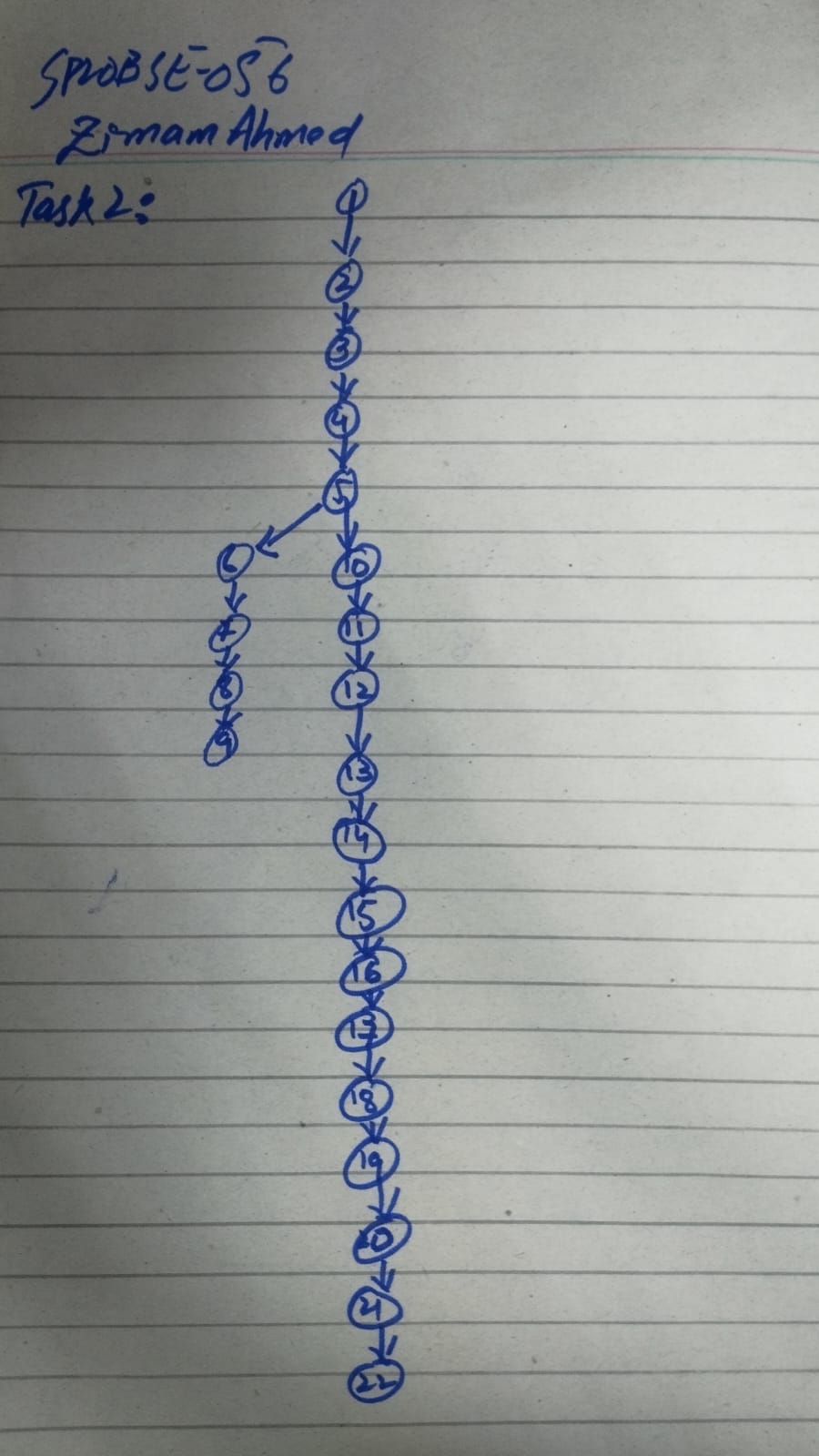
Statement Coverage = (18 / 22) \* 100% ≈ 81%

**Branch coverage:**

Branch Coverage = (Number of Executed Branches / Total Number of Branches) \* 100%

Branch Coverage = (0 / 2) \* 100% = 0%

**Path coverage:**



**Cyclomatic complexity:**

Formula : V= E-N+2

V = 21-22+2 = 1

**Task 3:**

1. int x=1;
2. int y=2;
3. int a=2;
4. int b=1;
5. if ((x<a) || (y<b))
6. {
7. x--;
8. y--;
9. }

**Statement coverage:**

Statement Coverage = (Number of Executed Statements / Total Number of Statements) \* 100%

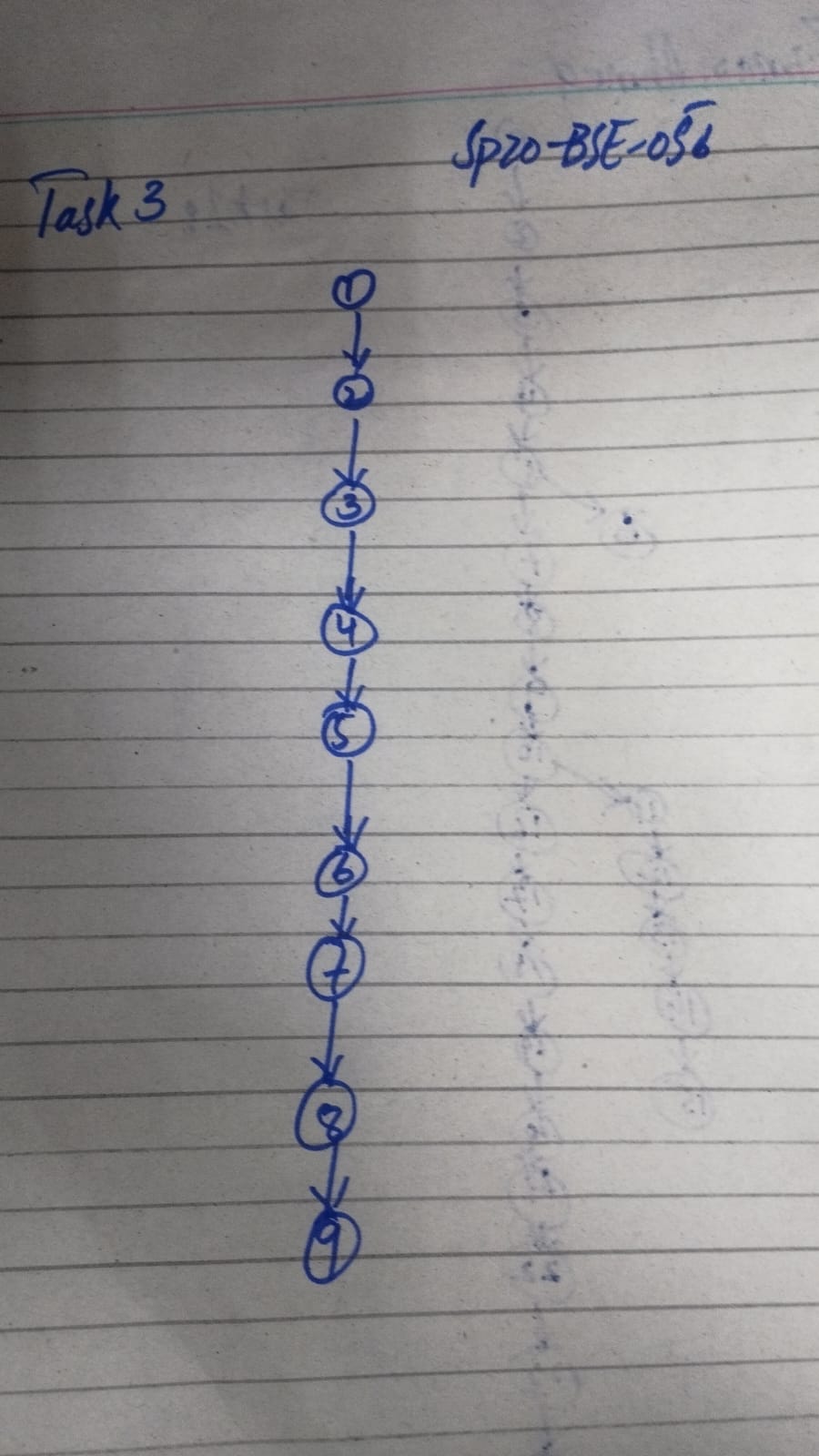
Statement Coverage = (9 / 9) \* 100% ≈ 100%

**Branch coverage:**

Branch Coverage = (Number of Executed Branches / Total Number of Branches) \* 100%

Branch Coverage = (1/1) \* 100% = 100%

**Path coverage:**



**Cyclomatic complexity:**

Formula : V= E-N+2

V = 8-9+2 = 1

Task 4:

// input statement for a

1. switch (a)
2. {
3. case 1:
4. print a;
5. break;
6. case 2:
7. print a;
8. break;
9. case 3:
10. print a;
11. default:
12. print a;
13. }

**Statement coverage:**

Statement Coverage = (Number of Executed Statements / Total Number of Statements) \* 100%

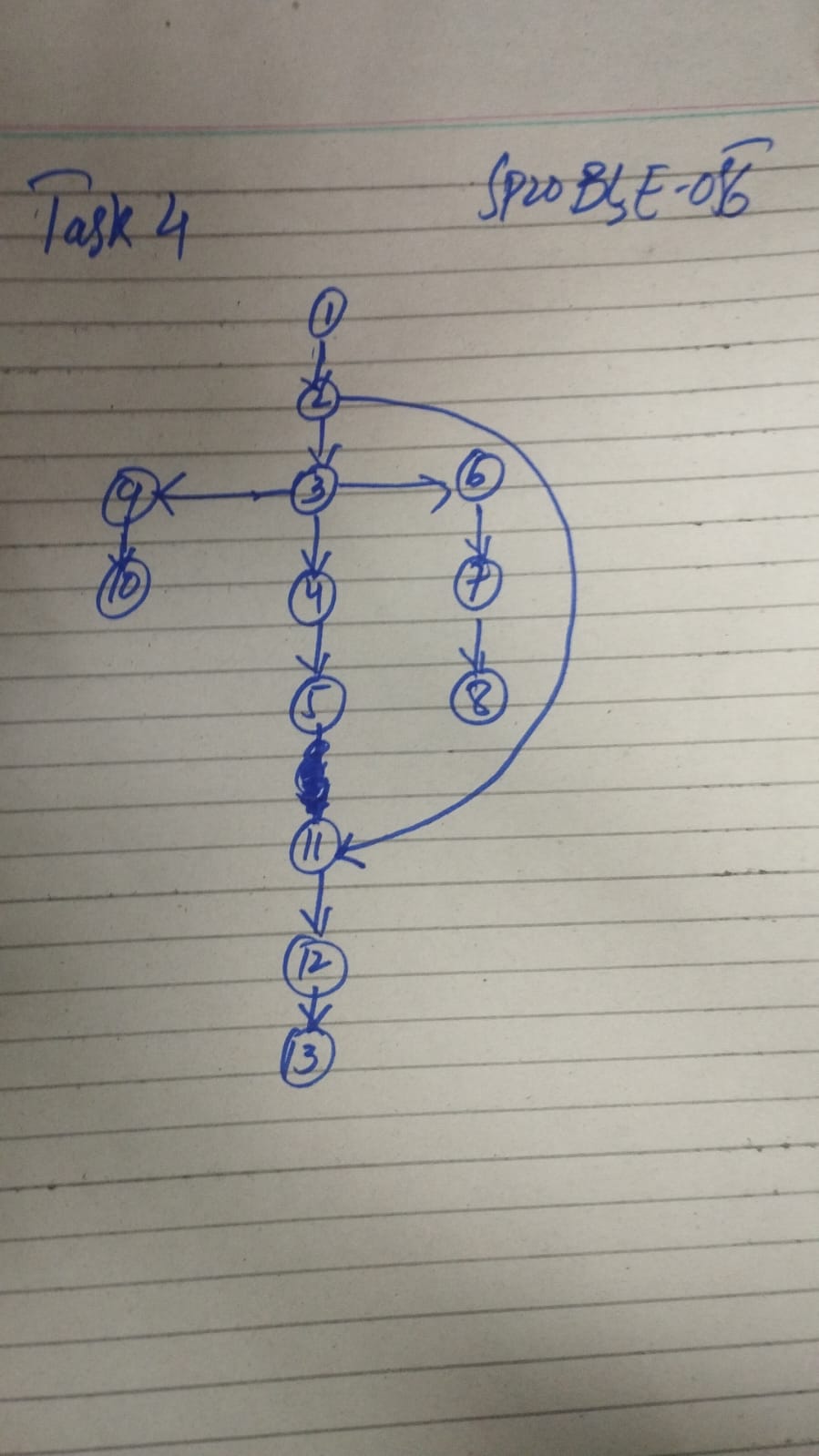
Statement Coverage = (13 / 13) \* 100% ≈ 100%

**Branch coverage:**

Branch Coverage = (Number of Executed Branches / Total Number of Branches) \* 100%

Branch Coverage = (3/3) \* 100% = 100%

**Path coverage:**



**Cyclomatic complexity:**

Formula : V= E-N+2

V = 13-13+2 = 2

**Task 5:**

1. int a=10;
2. for (int i = 0; i <10; i ++)
3. {
4. a++;
5. }
6. print a;

**Statement coverage:**

Statement Coverage = (Number of Executed Statements / Total Number of Statements) \* 100%

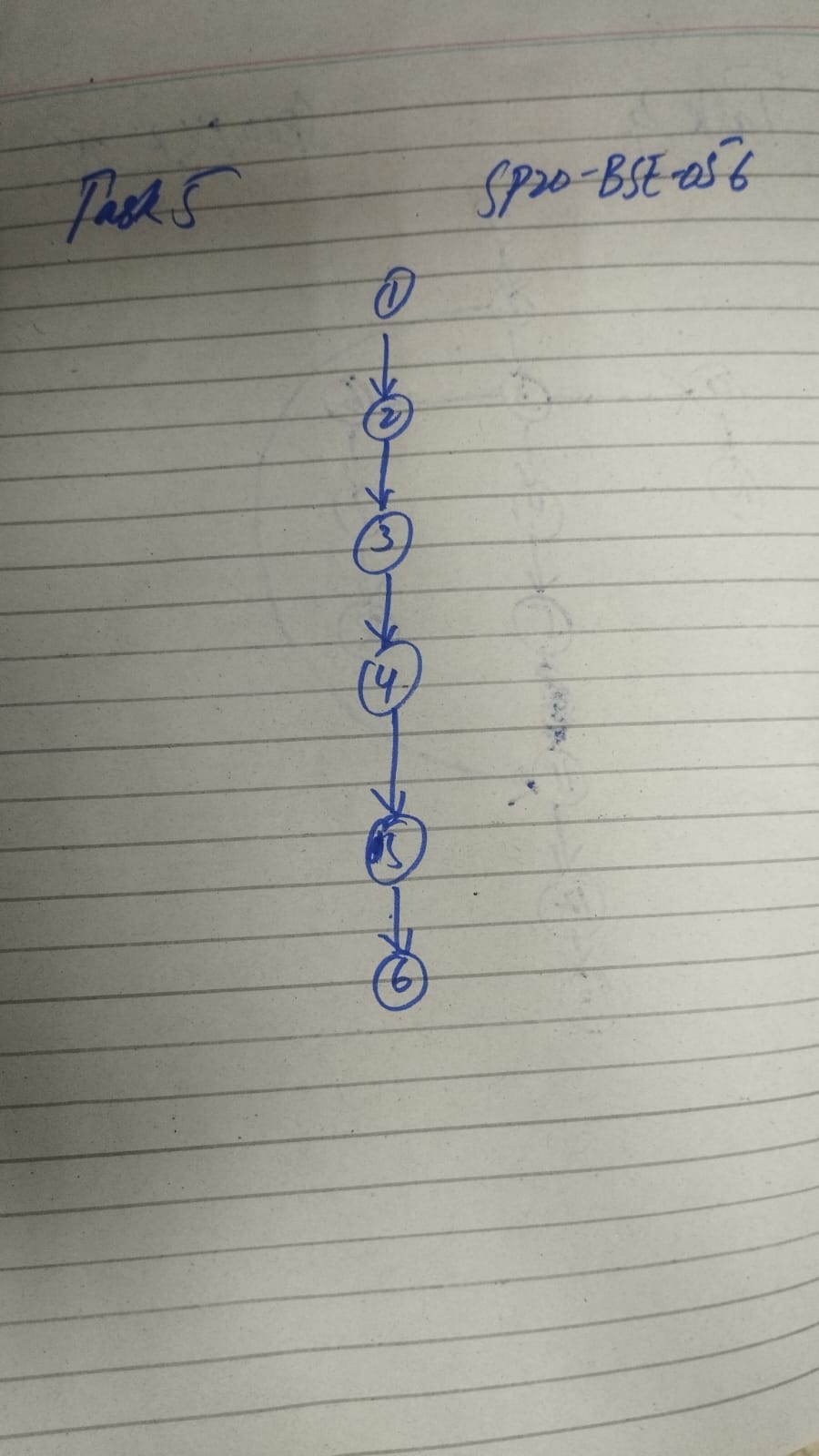
Statement Coverage = (4 / 6) \* 100% ≈ 66%

**Branch coverage:**

Branch Coverage = (Number of Executed Branches / Total Number of Branches) \* 100%

Branch Coverage = (1/1) \* 100% = 100%

**Path coverage:**



**Cyclomatic complexity:**

Formula : V= E-N+2

V = 5-6+2 = 1

**Task 6:**

1. int a=10;
2. for (int i = 0; i &lt; 10; i ++)
3. {
4. if (i == 5)
5. {
6. a--;
7. }
8. }

**Statement coverage:**

Statement Coverage = (Number of Executed Statements / Total Number of Statements) \* 100%

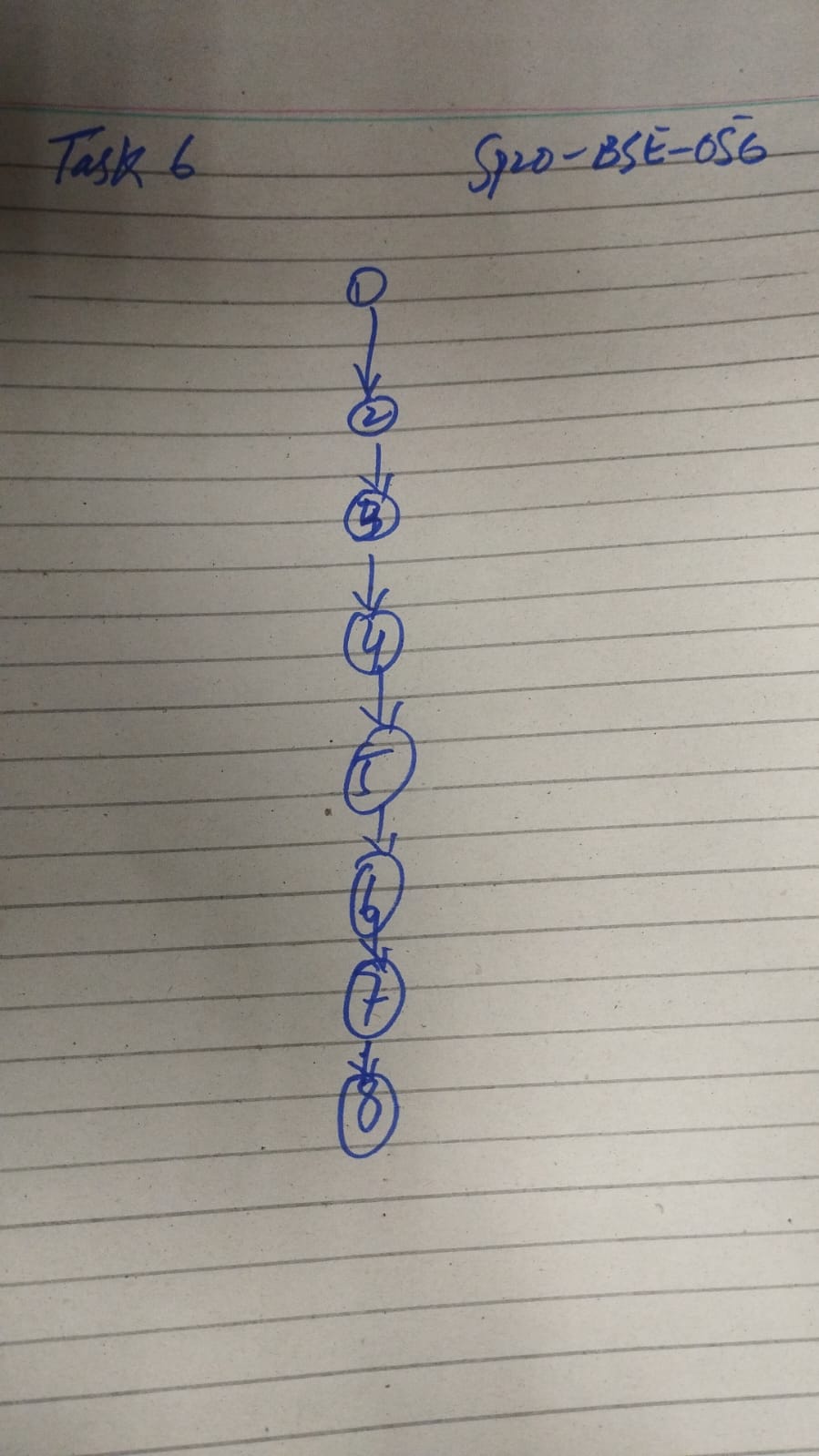
Statement Coverage = (4 / 8) \* 100% ≈ 50%

**Branch coverage:**

Branch Coverage = (Number of Executed Branches / Total Number of Branches) \* 100%

Branch Coverage = (2/2) \* 100% = 100%

**Path coverage:**



**Cyclomatic complexity:**

Formula : V= E-N+2

V = 7-8+2 = 1

**Task 7:**

1. int main ()
2. {
3. int i, j, k;
4. for (int i =1; i &lt; 4; i ++)
5. {
6. for (int j =1; j &lt; 7; j ++)
7. {
8. print “ ”;
9. }
10. for (int k =1; k &lt; (i \* 2); k ++)
11. {
12. print “ \*”;
13. }
14. print “\n”;
15. }
16. for (int i =5; i &gt;= 1; i --)
17. {
18. for (int j =7; j &gt; i; j --)
19. {
20. print “ ”;
21. }
22. for (int k =1; k &lt; (i \* 2); k ++)
23. {
24. print “ \*”;
25. }
26. print “\n”;
27. }
28. }