# SOFTWARE QUALITY & METRICS
# TEST 3

- ASHISH KUMAR
- 2K18/SE/041

## Tasks:

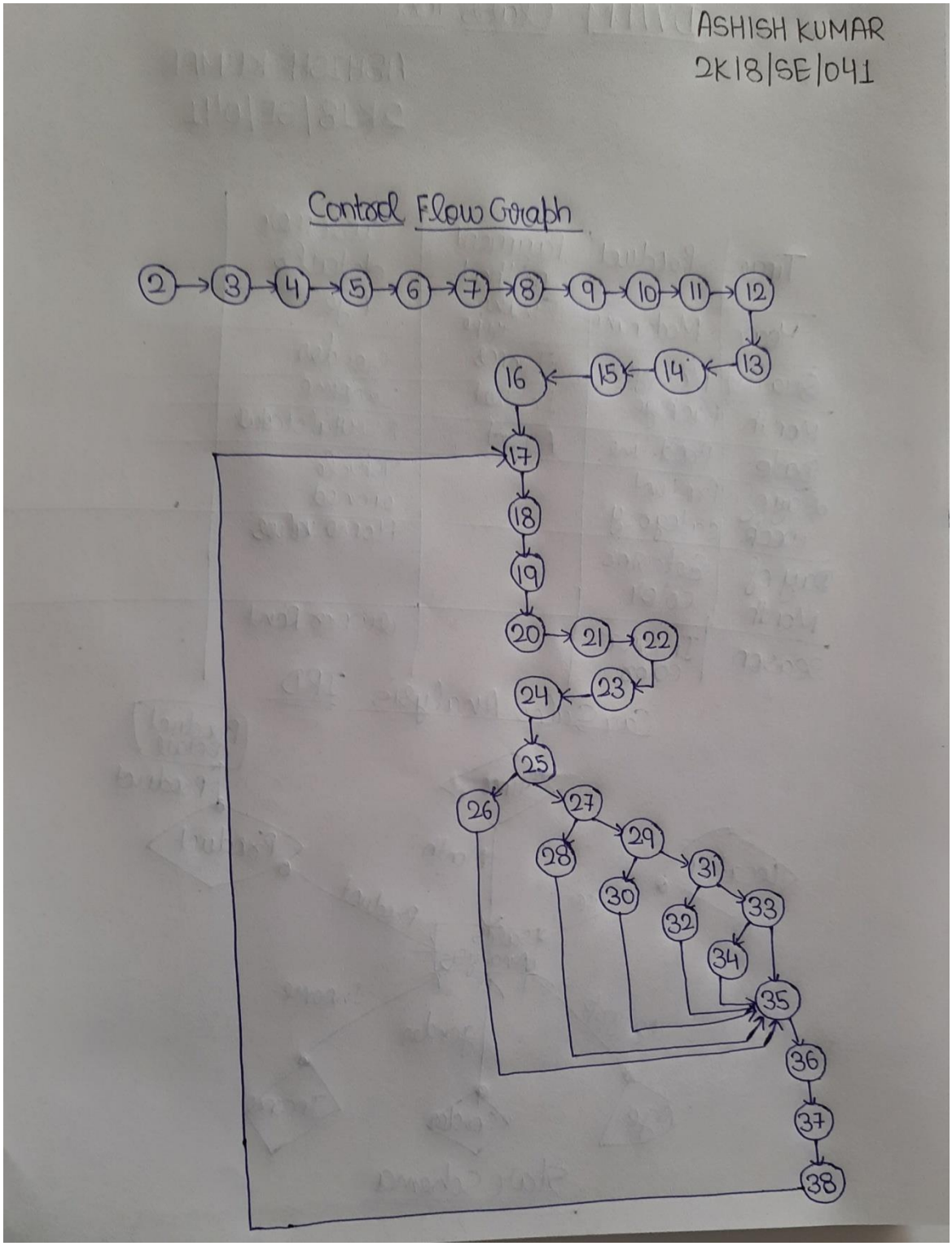
**Ques 1:** Draw the control flow graph then generate a minimal set of test cases to ensure the full coverage of the path, statements and branch criteria of the Calculator program:

```
2. import java.util.Scanner;
3.
4. public class Calculator {
5.
6. public static void main(String[] args) {
7.
8. Scanner input = new Scanner(System.in);
9.
10. Maths Maths = new Maths();
11.
12. double answer = 0;
13. double inputA, inputB;
14. char operator;
15. boolean done = false;
16.
17. while (done = = false) {
18. System.out.print("Please enter your sum: ");
```

```
19.
20. inputA = input.nextDouble();
21. operator = input.next().charAt(0);
22. inputB = input.nextDouble();
23.
24. switch (operator) {
25. case '+': answer = Maths.add(inputA, inputB);
26. break;
27. case '-': answer = Maths.subtract(inputA, inputB);
28. break;
29. case '*': answer = Maths.multiply(inputA, inputB);
30. break;
31. case '/': answer = Maths.divide(inputA, inputB);
32. break;
33. case '^': answer = Maths.power(inputA, inputB);
34. break;
35. }
36.
37. System.out.println(answer);
38. }
```

**Ans:**

a)


ASHISH KUMAR
2K18/SE/041
Control Flow Graph
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38

b) Test cases are written below in the form of table:

For full coverage of the path:

| inputA | inputB | operator |
|---|---|---|
| 20 | 10 | $ |
| 20 | 10 | ^ |
| 20 | 10 | / |
| 20 | 10 | * |
| 20 | 10 | - |
| 20 | 10 | + |

For full statements coverage:

| inputA | inputB | operator |
|---|---|---|
| 20 | 10 | ^ |
| 20 | 10 | / |
| 20 | 10 | * |
| 20 | 10 | - |
| 20 | 10 | + |

For full branch coverage:

| inputA | inputB | operator |
|---|---|---|
| 20 | 10 | $ |
| 20 | 10 | ^ |
| 20 | 10 | / |
| 20 | 10 | * |
| 20 | 10 | - |
| 20 | 10 | + |

**Ques 2:** Consider the following requirements and code:

The program shall take as input an array of three integer numbers.

The program shall output the greatest number among the elements of the array.

The program shall order the elements of the array in decreasing order.

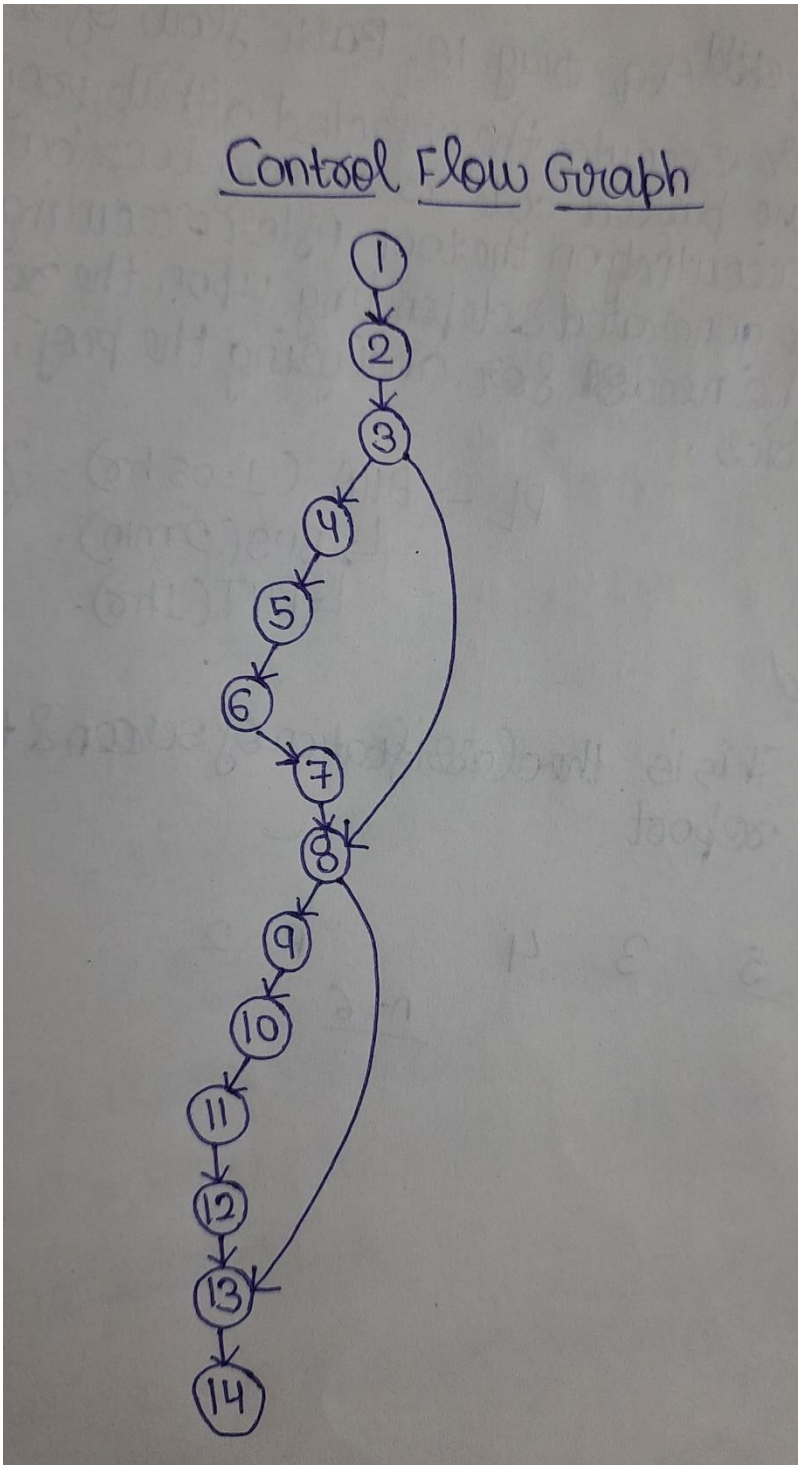
```
1.  int[] order(int v[]) {
2.      int tmp;
3.      if (v[0]<v[1]) {
4.      tmp = v[0];
5.      v[1] = v[1];
6.      v[1] = tmp;
7.      }
8.      if (v[1]<v[2]) {
9.       tmp = v[0];
10.     v[1] = v[2];
11.     v[2] = tmp;
12.     }
13.     return v;
14.  }
```

a) Create the control flow graph.

b) Write the minimum number of test cases for each of the coverage criteria: Node (statement) coverage, Edge (branch) coverage, Path coverage.

**Ans:**

a)



b) i) For node coverage, following array of 3 elements can be taken as test case :

v[] = {10, 11, 12}

ii) For edge coverage, following arrays of 3 elements can be taken as test case :

v[] = {10, 11, 12}

v[] = {12, 11, 10}

iii) For Path coverage, following arrays of 3 elements can be taken as test case :

v[] = {10, 11, 12)}

v[] = {11, 12, 12}

v[] = {5, 5, 6}

v[] = {7, 7, 7}