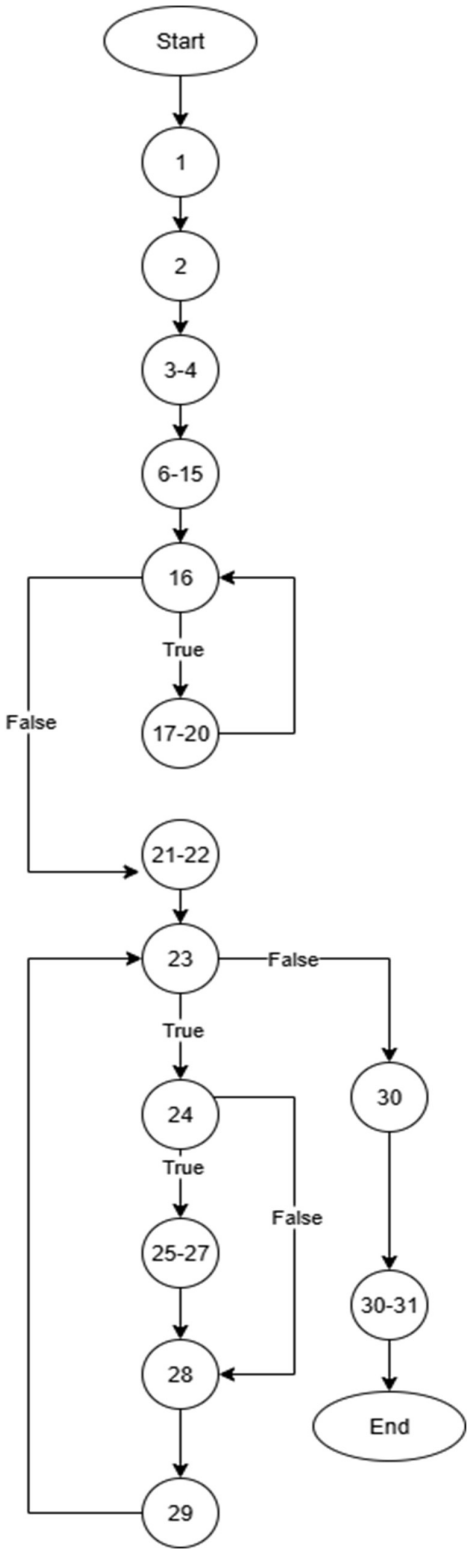
LAB1: Refer the following program to find the smallest integer from an array of numbers entered by the user.

1.1: **Prepare the CFG of the minimum.c Program**



1.2: **Write the test cases to find the errors in the Program:**

1. Test Case1:

```
Enter the size of the array: 4
Enter A[1]=11
Enter A[2]=10
Enter A[3]=12
Enter A[4]=14
Minimum element is: 10
```

2. Test Case2:

```
testing_class1.cpp -o software_testing_clas
Enter the size of the array: 4
Enter A[1]=2
Enter A[2]=3
Enter A[3]=4
Enter A[4]=5
Minimum element is: 3
```

3. Test Case3:

```
Enter the size of the array: 3
Enter A[1]=3
Enter A[2]=2
Enter A[3]=-1
Minimum element is: 2
```

4. Test Case4:

```
Enter the size of the array: 3
Enter A[1]=-2
Enter A[2]=-1
Enter A[3]=-9
Minimum element is: 1
```

5. Test Case5:

```
Enter the size of the array: 1
Enter A[1]=5
Minimum element is: 2147483647
```

6. Test Case6:

```
Enter the size of the array: 2
Enter A[1]=2
Enter A[2]=1
Minimum element is: 2147483647
```

TEST CASES

S.no	Input	Expected Output	Actual Output	Result
1.	Size=4 [11, 10, 12, 14]	10	10	Pass
2.	Size=4 [2, 3, 4, 5]	2	3	Fail
3.	Size=3 [3, 2, -1]	-1	2	Fail

4.	Size=3 [-2, -1, -9]	-9	1	Fail
5.	Size=1 [5]	5	INT_MAX	Fail
6.	Size=2 [2, 1]	1	INT_MAX	Fail

1.3: **Summarize the errors to facilitate the user to debug the program**

The code is not able to handle:

- The last element of the array for checking the minimum integer.
- The first element of the array.
- Not able to handle real numbers; only integers can be handled.
- Not able to handle a negative integer value to determine the minimum integer.

1.4: **Debug the Program for the identified error and execute the same**

```
#include <stdio.h>
#include <limits.h>
1. void MinMax();
2. int main(){
3. MinMax();
4. return 0;
5. }
6. void Minimum(){
7. int array[100];
8. int min = INT_MAX;
9. int Number;
10. int i;
11. int tmpData;
12. printf("Enter the size of the array (1-100): ");
13. scanf("%d", &Number);
14. for (i = 0; i < Number; i++) {
15. printf("Enter A[%d]=", i + 1);
16. while(1){
17. char term;
18. int result = scanf("%d%c", &tmpData, &term);
19. if (result == 2 && term == '\n'){
        break;
20. }
21. else printf("Invalid input! Enter an integer: ");
22. while (getchar() != '\n');
23. }
24. array[i] = tmpData;
25. }
26. i=0;
27. while(i < Number){
28. if(min > array[i]){
29. min = array[i];
30. }
```

```
31. i++;
32. }
33. printf("Minimum element is: %d\n", min);
34. }
```

1.5: **Prepare the set of Test Cases to Re-test the Program**

1. Test Case1:

```
Enter the size of the array (1-100): 4
Enter A[1]=11
Enter A[2]=10
Enter A[3]=12
Enter A[4]=14
Minimum element is: 10
```

2. Test Case2:

```
Enter the size of the array (1-100): 4
Enter A[1]=2
Enter A[2]=3
Enter A[3]=4
Enter A[4]=5
Minimum element is: 2
```

3. Test Case3:

```
Enter the size of the array (1-100): 3
Enter A[1]=3
Enter A[2]=2
Enter A[3]=-1
Minimum element is: -1
```

4. Test Case4:

```
Enter the size of the array (1-100): 3
Enter A[1]=-2
Enter A[2]=-1
Enter A[3]=-9
Minimum element is: -9
```

5. Test Case5:

```
Enter the size of the array (1-100): 1
Enter A[1]=5
Minimum element is: 5
```

6. Test Case6:

```
Enter the size of the array (1-100): 2
Enter A[1]=2
Enter A[2]=1
Minimum element is: 1
```

TEST CASES:

S.no	Input	Expected Output	Actual Output	Result
1.	Size=4 [11, 10, 12, 14]	10	10	Pass

2.	Size=4 [2, 3, 4, 5]	2	2	Pass
3.	Size=3 [3, 2, -1]	-1	-1	Pass
4.	Size=3 [-2, -1, -9]	-9	-9	Pass
5.	Size=1 [5]	5	5	Pass
6.	Size=2 [2, 1]	1	1	Pass