EXPERIMENT 1

I. Write a program to find the minimum in three numbers input by the user and write test cases for the program using,
a. Boundary Value Approach
b. Worst Case Approach
c. Robust Approach

CODE

```
#include <stdio.h>

#include <conio.h>

void mini()

{

	int a, b, c;

	printf("Enter the three numbers between Range 1 to 100: ");

	scanf("%d %d %d",&a,&b,&c);

	if (a<1||a>100)

	{

		printf("Invalid");

		return;

	}

	if (b<1||b>100)

	{

		printf("Invalid");

		return;

	}

	if (c<1||c>100)

	{

		printf("Invalid");

		return;

	}

	if(a<b) {

		if(a<c)
```

```
                printf("Minimum : %d",a);
        else
                printf("Minimum : %d",c);
    }
    else {
        if(b<c)
                printf("Minimum : %d",b);
        else
                printf("Minimum : %d",c);
    }
}
int main()
{
    mini();
    getch();
    return(0);
}
```

OUTPUT:

Boundary Case

Test Case ID	A	B	C	Expected O/P	Actual O/P
1.	1	50	50	1	1
2.	2	50	50	2	2
3.	50	50	50	50	50
4.	99	50	50	50	50
5.	100	50	50	50	50
6.	50	1	50	1	1
7.	50	2	50	2	2
8.	50	99	50	50	50
9.	50	100	50	50	50
10.	50	50	1	1	1
11.	50	50	2	2	2
12	50	50	99	50	50
13.	50	50	100	50	50

Robust Case

Test case ID	A	B	C	Expected O/P	Actual O/P
1.	0	50	50	Invalid	Invalid
2.	1	50	50	1	1
3.	2	50	50	22	22

4.	50	50	50	50	50
5.	99	50	50	50	50
6.	100	50	50	50	50
7.	101	50	50	Invalid	Invalid
8.	50	0	50	Invalid	Invalid
9.	50	1	50	1	1
10.	50	2	50	2	2
11.	50	99	50	50	50
12.	50	100	50	50	50
13.	50	101	50	Invalid	Invalid
14.	50	50	0	Invalid	Invalid
15.	50	50	1	1	1
16.	50	50	2	2	2
17.	50	50	99	50	50
18.	50	50	100	50	50
19.	50	50	101	Invalid	Invalid

Worst Case

Test Case ID	A	B	C	Expected O/P	Actual O/P
1.	1	1	1	1	1
2.	1	1	2	1	1
3.	1	1	50	1	1
4.	1	1	99	1	1
5.	1	1	100	1	1
6.	1	2	1	1	1
7.	1	2	2	1	1
8.	1	2	50	1	1
9.	1	2	99	1	1
10.	1	2	100	1	1
11.	1	50	1	1	1
12.	1	50	2	1	1
13.	1	50	50	1	1
14.	1	50	99	1	1
15.	1	50	100	1	1
16.	1	99	1	1	1
17.	1	99	2	1	1
18.	1	99	50	1	1
19.	1	99	99	1	1
20.	1	99	100	1	1
21.	1	100	1	1	1
22.	1	100	2	1	1
23.	1	100	50	1	1
24.	1	100	99	1	1
25.	1	100	100	1	1
26.	2	1	1	1	1
27.	2	1	2	1	1
28.	2	1	50	1	1
29.	2	1	99	1	1
30.	2	1	100	1	1
31.	2	2	1	1	1
32.	2	2	2	2	2
33.	2	2	50	2	2
34.	2	2	99	2	2
35.	2	2	100	2	2
36.	2	50	1	1	1
37.	2	50	2	2	2

38.	2	50	50	2	2
39.	2	50	99	2	2
40.	2	50	100	2	2
41.	2	99	1	1	1
42.	2	99	2	2	2
43.	2	99	50	2	2
44.	2	99	99	2	2
45.	2	99	100	2	2
46.	2	100	1	1	1
47.	2	100	2	2	2
48.	2	100	50	2	2
49.	2	100	99	2	2
50.	2	100	100	2	2
51.	50	1	1	1	1
52.	50	1	2	1	1
53.	50	1	50	1	1
54.	50	1	99	1	1
55.	50	1	100	1	1
56.	50	2	1	1	1
57.	50	2	2	2	2
58.	50	2	50	2	2
59.	50	2	99	2	2
60.	50	2	100	2	2
61.	50	50	1	1	1
62.	50	50	2	2	2
63.	50	50	50	50	50
64.	50	50	99	50	50
65.	50	50	100	50	50
66.	50	99	1	1	1
67.	50	99	2	2	2
68.	50	99	50	50	50
69.	50	99	99	50	50
70.	50	99	100	50	50
71.	50	100	1	1	1
72.	50	100	2	2	2
73.	50	100	50	50	50
74.	50	100	99	50	50
75.	50	100	100	50	50
76.	99	1	1	1	1
77.	99	1	2	1	1
78.	99	1	50	1	1
79.	99	1	99	1	1
80.	99	1	100	1	1
81.	99	2	1	1	1
82.	99	2	2	2	2
83.	99	2	50	2	2
84.	99	2	99	2	2
85.	99	2	100	2	2
86.	99	50	1	1	1
87.	99	50	2	2	2
88.	99	50	50	50	50
89.	99	50	99	50	50
90.	99	50	100	50	50
91.	99	99	1	1	1
92.	99	99	2	2	2
93.	99	99	50	50	50
94.	99	99	99	99	99
95.	99	99	100	99	99

96.	99	100	1	1	1
97.	99	100	2	2	2
98.	99	100	50	50	50
99.	99	100	99	99	99
100.	99	100	100	99	99
101.	100	1	1	1	1
102.	100	1	2	1	1
103.	100	1	50	1	1
104.	100	1	99	1	1
105.	100	1	100	1	1
106.	100	2	1	1	1
107.	100	2	2	2	2
108.	100	2	50	2	2
109.	100	2	99	2	2
110.	100	2	100	2	2
111.	100	50	1	1	1
112.	100	50	2	2	2
113.	100	50	50	50	50
114.	100	50	99	50	50
115.	100	50	100	50	50
116.	100	99	1	1	1
117.	100	99	2	2	2
118.	100	99	50	50	50
119.	100	99	99	99	99
120.	100	99	100	99	99
121.	100	100	1	1	1
122.	100	100	2	2	2
123.	100	100	50	50	50
124.	100	100	99	99	99
125.	100	100	100	100	100