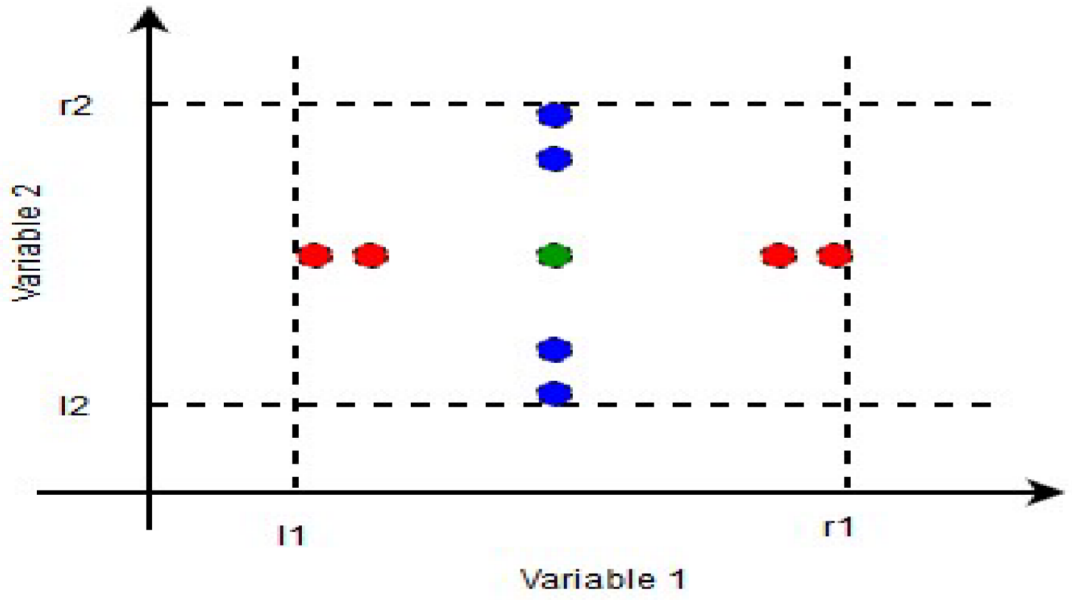
Exp No: 07	TITLE: BOUNDARY VALUE ANALYSIS	Date: 03/06/2022

AIM: Design the Test cases for triangle problem with Software Testing Technique: Boundary Value Analysis using C.

Boundary Value Analysis (BVA) is a black box software testing technique where test cases are designed using boundary values. BVA is based on the *single fault assumption*, also known as critical fault assumption which states that failures are rarely the product of two or more simultaneous faults. Hence while designing the test cases for BVA we keep all but one variable to the nominal value and allowing the remaining variable to take the extreme value.

Test Case Design for BVA:

While designing the test cases for BVA first we determine the number of input variables in the problem. For each input variable, we then determine the range of values it can take. Then we determine the extreme values and nominal value for each input variable.
Consider an input variable t taking values in the range [l, r].Extreme values for t are –



t = l

t = l + 1

t = r-1

t = r

The nominal value for the variable can be any value in the range (l, r).
In most of the BVA implementations, it is taken as the middle value of the range (r+l)/2.
The figure on the right shows the nominal and extreme values for boundary value of analysis of a two variable problem.

Exp No: 07	TITLE: BOUNDARY VALUE ANALYSIS	Date: 03/06/2022

Under the single fault assumption, the total number of test cases in BVA for a problem with n inputs is 4n+1.
The 4n cases correspond to the test cases with the four extreme values of each variable keeping the other n-1 variable at nominal value. The one additional case is where all variables are held at a nominal value.

One of the common problem for Test Case Design using BVA is the *Triangle Problem* that is discussed below –
Triangle Problem accepts three integers – a, b, c as three sides of the triangle .We also define a range for the sides of the triangle as [l, r] where l>0. It returns the type of triangle (Scalene, Isosceles, Equilateral, Not a Triangle) formed by a, b, c.

For a, b, c to form a triangle the following conditions should be satisfied –

a < b+c

b < a+c

c < a+b

If any of these conditions is violated output is Not a Triangle.

- For Equilateral Triangle all the sides are equal.
- For Isosceles Triangle exactly one pair of sides is equal.
- For Scalene Triangle all the sides are different.

The table shows the Test Cases Design for the Triangle Problem.The range value [l, r] is taken as [1, 100] and nominal value is taken as 50.

The total test cases is,

4n+1 = 4*3+1 = 13

SOURCE CODE:

```
#include<stdio.h>

void main() {

        int x, y, z;

        printf("Enter three sides of a triangle : ");

        scanf("%d %d %d", &x,&y,&z);

        if(x + y > z && y + z > x && z + x > y) {

                if(x == y && x == z) {
```

```
                printf("The triangle is: Equilateral triangle\n");
            } else if(x == y || y == z || z == x) {
                printf("The triangle is: Isosceles triangle\n");
            } else {
                printf("The triangle is: Scalene triangle\n");
            }
        } else {
            printf("Not a triangle.\n");
        }
}
```

OUTPUT:

1. Enter three sides of a triangle : 5 1 5

   Isosceles

2. Enter three sides of a triangle : 10 5 5

   Not a triangle

3. Enter three sides of a triangle : 5 5 1

   Isosceles

4. Enter three sides of a triangle : 5 10 5

   Not a triangle

5. Enter three sides of a triangle : 5 5 5

   Equilateral

Exp No: 07	TITLE: BOUNDARY VALUE ANALYSIS	Date: 03/06/2022

Test Cases:

BOUNDARY VALUE ANALYSIS FOR TRIANGLE PROBLEM						
DATE : 03 - 06 – 2022				DONE BY : HN LAKSHMI NARASIMHA		
STATUS : All Testcases Passed				ORIGIN : GPREC		
TEST ID	A	B	C	EXPECTED OUTPUT	PROGRAM OUTPUT	TESTED OUTCOME
1	1	5	5	Isosceles Triangle	Isosceles Triangle	Pass
2	2	5	5	Isosceles Triangle	Isosceles Triangle	Pass
3	9	5	5	Isosceles Triangle	Isosceles Triangle	Pass
4	10	5	5	Not a Triangle	Not a Triangle	Pass
5	5	1	5	Isosceles Triangle	Isosceles Triangle	Pass
6	5	2	5	Isosceles Triangle	Isosceles Triangle	Pass
7	5	9	5	Isosceles Triangle	Isosceles Triangle	Pass
8	5	10	5	Not a Triangle	Not a Triangle	Pass
9	5	5	1	Isosceles Triangle	Isosceles Triangle	Pass
10	5	5	2	Isosceles Triangle	Isosceles Triangle	Pass
11	5	5	9	Isosceles Triangle	Isosceles Triangle	Pass
12	5	5	10	Not a Triangle	Not a Triangle	Pass
13	5	5	5	Equilateral Triangle	Equilateral Triangle	Pass