Expt. No: 1

Problem Statement:

Design and Develop a program in language of your choice to solve the triangle problem defined as follows: Accept 3 integers which are supposed to be 3 sides of a triangle and determine if the 3 values represent as a equilateral, isoceles, scalene triangle or they do not form a triangle at all.

Objectives:

- To solve the triangle problem.
- The test cases should be based on decision table approach.

Theory:

The system should accept 3 positive integers number (a,b,c) which represents 3 sides of a triangle. Based on the input it should specify whether a triangle can be formed or not.

If the above requirement is specified gets satisfied then the system should determine the type of the triangle, which are Equilateral (all 3 sides are equal), Isoceles (two sides are equal), Scalene (all the sides are unequal). Else a suitable message should be displayed. Here we assume that user gives suitable positive integer number as input. From the given requirements we can draw the following conditions:

c1 : $a < b + c$?

c2: $b < a + c$?

c3: $c < a + b$?

c4: $a = b$?

c5 : $a = c$?

c6: $b = c$?

Program:

```
#include<stdio.h>
#include<ctype.h>
#include<conio.h>
#include<process.h>
int main(){
    int a,b,c;
    clrscr();
    printf("Enter the three sides of a triangle");
    scanf("%d%d%d",&a,&b,&c);
    if((a<b+c)&&(b<a+c)&&(c<a+b)){
        if((a==b)&&(b==c)){
            printf("Equilateral Triangle");
        }
        else if((a!=b)&&(a!=c)&&(b!=c)){
            printf("Scalene Triangle");
        }
        else{
            printf("Isoceles Triangle");
        }
        else{
            printf("Triangle cannot be formed");
        }
    getch();
    return 0;
}
```

Decision Table:

It is used to depict complex logical relationships between input data. Its a method used to build a complete set of

The Decision Table is given below:

Conditions	Condition Entries (Rules)										
	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11
C1: $a < b+c$?	F	T	T	T	T	T	T	T	T	T	T
C2: $b < a+c$?	-	F	T	T	T	T	T	T	T	T	T
C3: $c < a+b$?	-	-	F	T	T	T	T	T	T	T	T
C4: $a = b$?	-	-	-	F	T	T	T	F	F	F	T
C5: $a = c$?	-	-	-	T	F	T	F	T	F	F	T
C6: $b = c$?	-	-	-	T	T	F	F	F	T	F	T
Actions	Action Entries										
a1: Not a Triangle	X	X	X								
a2: Scalene										X	
a3: Isosceles							X	X	X		
a4: Equilateral											X
a5: Impossible				X	X	X					

test cases without using the internal structure of the program in question.

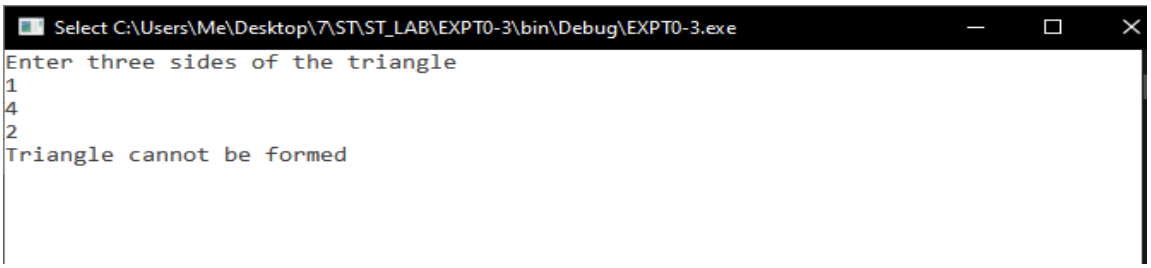
In decision table the symbol '-' indicates doesn't case value. The table shows 6 conditions and sections. All the condition in the decision table are binary. Hence its called 'Limited Entry Decision Table'.

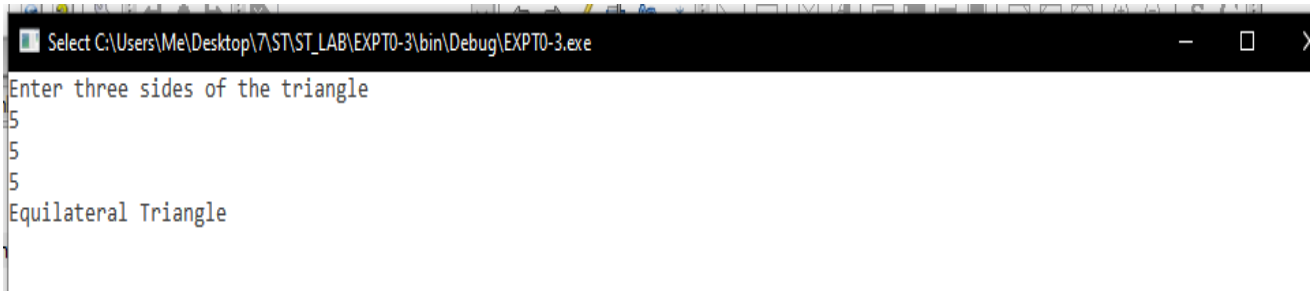
Test Case Table:

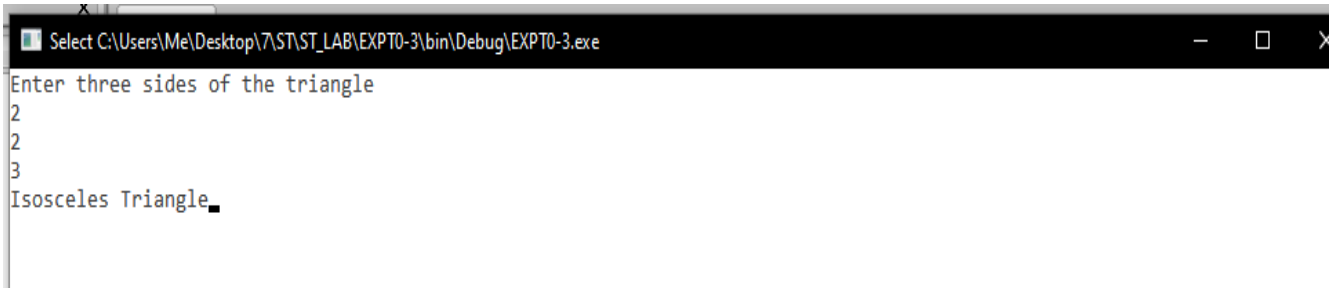
TCID	Test Case	a	b	c	Expected o/p	Actual o/p	Status
1	Testing for requirement 1	4	1	2	Not triangle	Not Triangle	TC Pass
2	Testing for requirement 1	1	4	2	Not Triangle	Not Triangle	TC Pass
3	Testing for requirement 1	1	2	4	Not Triangle	Not Triangle	TC Pass
4	Testing for requirement 2	5	5	5	Equilateral	Equilateral	TC Pass
5	Testing for requirement 2	2	2	3	Isoceles	Isoceles	TC Pass
6	Testing for requirement 2	2	3	2	Isoceles	Isoceles	Tc Pass
7	Testing for requirement 2	3	2	2	Isoceles	Isoceles	TC Pass
8	Testing for requirement 2	3	4	5	Scalene	Scalene	TC Pass

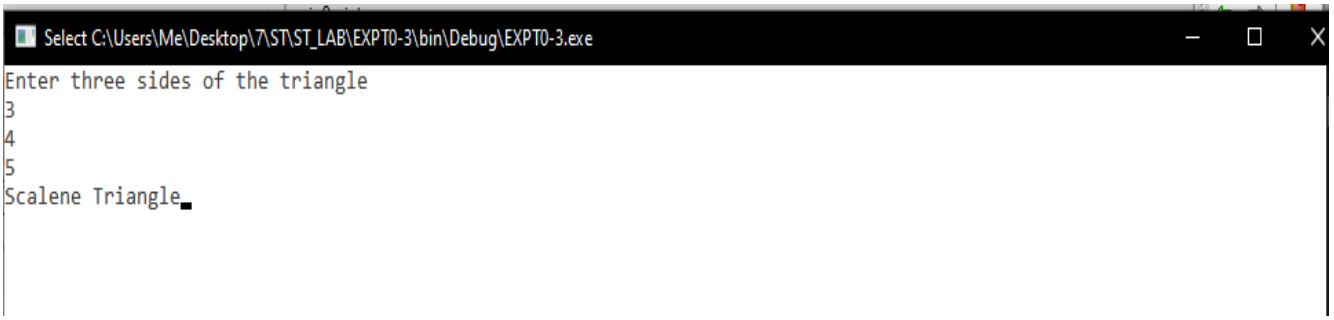
Test Report:

Number of Test Cases Executed : 8
Number of Defects Raised : 0
Number of Test Cases Passed : 8
Number of Test cases Failed : 0


Select C:\Users\Me\Desktop\7\ST\ST_LAB\EXPT0-3\bin\Debug\EXPT0-3.exe
Enter three sides of the triangle
1
4
2
Triangle cannot be formed


Select C:\Users\Me\Desktop\7\ST\ST_LAB\EXPT0-3\bin\Debug\EXPT0-3.exe
Enter three sides of the triangle
5
5
5
Equilateral Triangle


Select C:\Users\Me\Desktop\7\ST\ST_LAB\EXPT0-3\bin\Debug\EXPT0-3.exe
Enter three sides of the triangle
2
2
3
Isosceles Triangle


Select C:\Users\Me\Desktop\7\ST\ST_LAB\EXPT0-3\bin\Debug\EXPT0-3.exe
Enter three sides of the triangle
3
4
5
Scalene Triangle

Conclusion: We have solved the triangle problem by creating necessary conditions using if-else.
- The Decision table approach works well and the decision made were verified against the conditions.

References:

- Paul C Jorgensen 3rd Edition.

Expt: No: 2

Problem Definition:

Design and Develop a program in language of your choice to solve the triangle problem defined as follows: Accept 3 integers which are 3 sides of a triangle and determine if the sides represent an equilateral, Scalene, Isoceles triangle or do not form a triangle at all. Assume the upper limit for the size of any side is 10. Derive test cases of your program based on boundary-value analysis. Execute the test cases and discuss the result.

Objectives:

- Develop a program to check type of triangle given three sides
- Derive test cases based on boundary value analysis.

Theory:

Boundary Value Analysis: It depends on the output and constraints on the output. So its least worried on the input. For this problem it takes 3 sides for input. It yields $(4n+1)$ test cases according to single fault assumption theory. Hence we say total test cases would be $(4 \times 3 + 1) \Rightarrow 12 + 1 \Rightarrow 13$

The max limit of side size = 10. so a, b and c lie between.

$0 \leq a \leq 10$

$0 \leq b \leq 10$

$0 \leq c \leq 10$

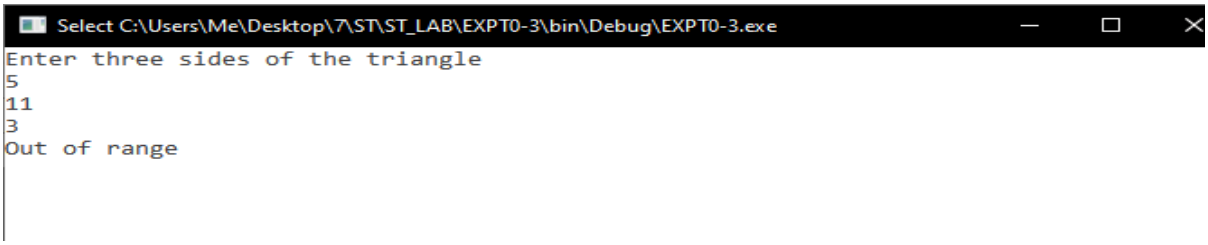
The input of each side must be positive integer to determine type of triangle.

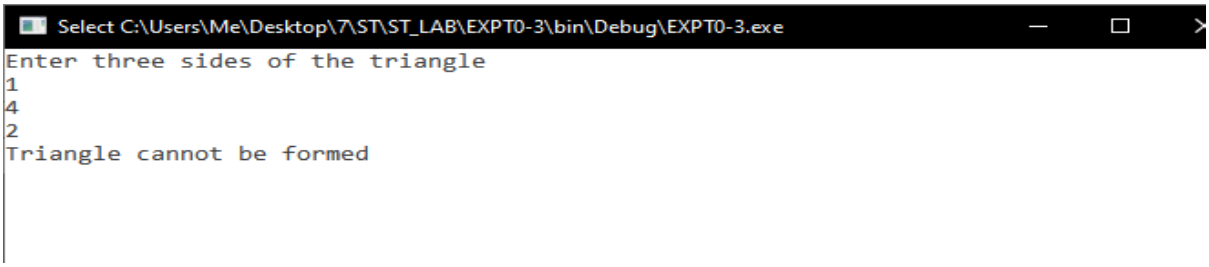
PROGRAM:

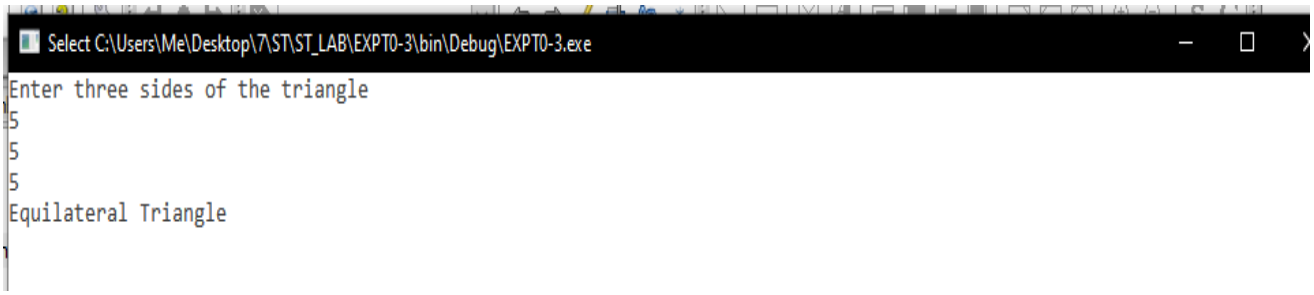
```
#include<stdio.h>
#include<ctype.h>
#include<conio.h>
#include<process.h>
int main(){
    int a,b,c;
    clrscr();
    printf("Enter three sides of a triangle");
    scanf("%d%d%d",&a,&b,&c);
    if((a>10)||(b>10)||(c>10)){
        printf("Out of range");
        getch();
        exit(0);
    }
    if((a<b+c)&&(b<a+c)&&(c<b+a)){
        if((a==b)&&(b==c)){
            printf("Equilateral Triangle");
        }
        else if((a!=b)&&(a!=c)&&(b!=c)){
            printf("Scalene Triangle");
        }
        else
            printf("Isosceles Triangle"); }
    else{
        printf("Triangle cannot be formed");
    }
    getch();
    return 0;
}
```

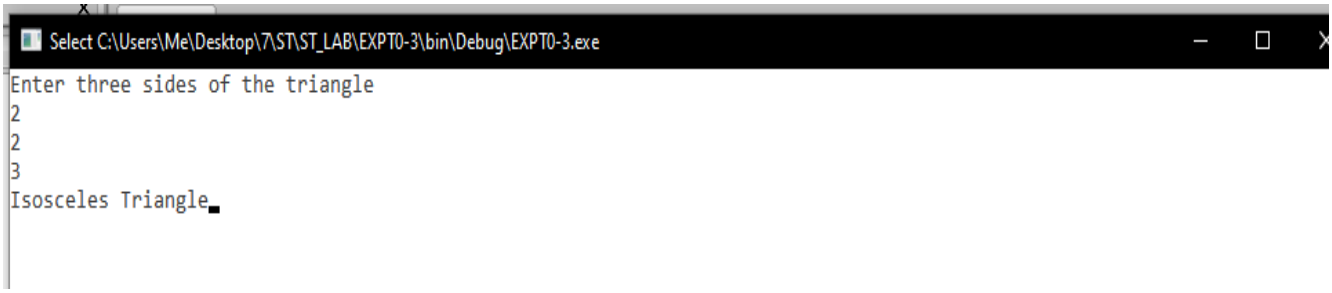
TC ID	Test Case Description	Input Data			Expected Output	Actual Output	Status
		A	B	C			
1.	For A input is not given	X	3	6	Not Triangle	Not Triangle	TC Pass
2	For B input is not given	3	X	6	Not Triangle	Not Triangle	TC Pass
3	For C input is not given	4	7	X	Not Triangle	Not Triangle	TC Pass
4	I/p of C is negative	5	5	-1	Not Triangle	Not Triangle	TC Pass
5	Two sides are same one side is different	5	5	1	Isosceles	Isosceles	TC Pass
6	All sides of input are equal	5	5	5	Equilateral	Equilateral	TC Pass
7	Two sides are same one side is different	5	5	9	Isogeles	Isosceles	TC Pass
8	If input is out of range	5	5	10	Not Triangle	Not Triangle	TC Pass
9	Two sides are same one side is different	5	1	5	Isosceles	Isosceles	TC Pass
10	Two sides are same one side is different	5	2	5	Isosceles	Isosceles	TC Pass
11	Twosides are same one side is different	5	9	5	Isosceles	Isosceles	TC Pass
12	Two sides are same one side is different (A=C=5, B is out of range ie B=10)	5	10	5	Not Triangle	Not Triangle	TC Pass

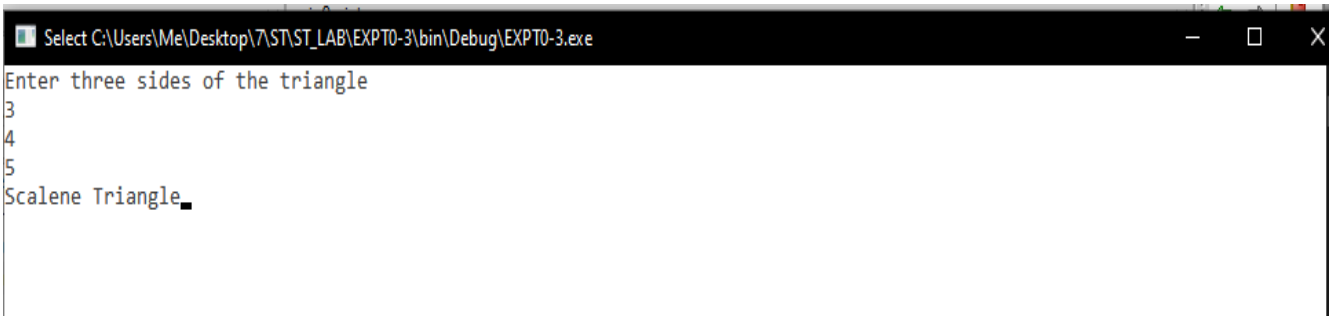
TC IO	Test Case Description	Input Data			Expected Output	Actual Output	Status
		A	B	C			
13	Two sides are same one side is different	1	5	5	Isosceles	Isosceles	TC Pass
14	Two sides are same one side is different	2	5	5	Isosceles	Isosceles	TC Pass
15	Two sides are same one side is different	9	5	5	Isosceles	Isosceles	TC Pass
16	Two sides are same one side is out of range	10	5	5	Not Triangle	Not Triangle	TC Pass


Select C:\Users\Me\Desktop\7\ST\ST_LAB\EXPT0-3\bin\Debug\EXPT0-3.exe
Enter three sides of the triangle
5
11
3
Out of range


Select C:\Users\Me\Desktop\7\ST\ST_LAB\EXPT0-3\bin\Debug\EXPT0-3.exe
Enter three sides of the triangle
1
4
2
Triangle cannot be formed


Select C:\Users\Me\Desktop\7\ST\ST_LAB\EXPT0-3\bin\Debug\EXPT0-3.exe
Enter three sides of the triangle
5
5
5
Equilateral Triangle


Select C:\Users\Me\Desktop\7\ST\ST_LAB\EXPT0-3\bin\Debug\EXPT0-3.exe
Enter three sides of the triangle
2
2
3
Isosceles Triangle


Select C:\Users\Me\Desktop\7\ST\ST_LAB\EXPT0-3\bin\Debug\EXPT0-3.exe
Enter three sides of the triangle
3
4
5
Scalene Triangle

Test Report:

Number of Test Cases executed : 16

Number of defects raised: 0

Number of Test cases passed: 16

Number of Test cases failed: 0

Conclusion:

- The triangle problem is implemented by using boundary value analysis.
- Implementation of test cases is done and verified.

References:

Paul C Jorgensen 3rd Edition.

Expt. No: 3

Problem Statement:

Design and Develop a program to solve the triangle problem defined as follows: Accept 3 integers which are 3 sides of a triangle and determine if the values represent an equilateral, Scalene, Isosceles Triangle or do not form a triangle at all. Assume size of any side to be upper limit of 10. Derive test cases of your program based on equilance class partitioning, execute the test cases and discuss the result.

Objectives:

- To solve the triangle problem and identify type of triangle.
- Use equivalence class partitioning to execute test cases.

Theory:

Equivalence Class Partitioning: It focuses on the input domain, we can obtain richer set of test cases. What are the possibilities for the three integers a, b, c? They can be equal, exactly one pair can be equal. The max limit of each side is 10 units accordingly. So a, b and c lie between.

$0 \leq a \leq 10$

$0 \leq b \leq 10$

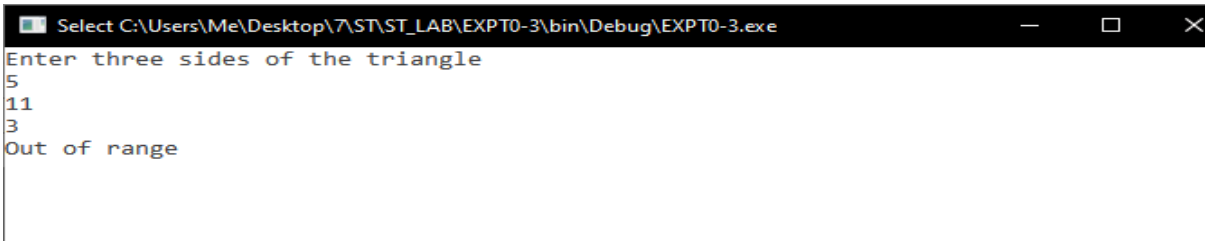
$0 \leq c \leq 10$

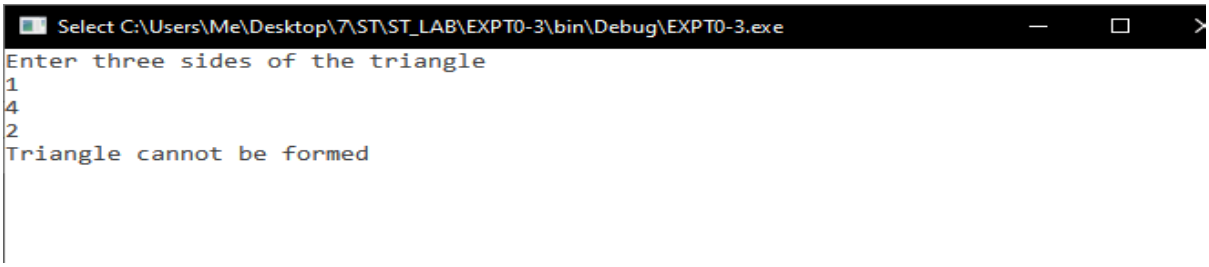
This technique tries to uncover / to define test cases that uncover classes of errors, therefore reducing the total number of test cases that can be developed.

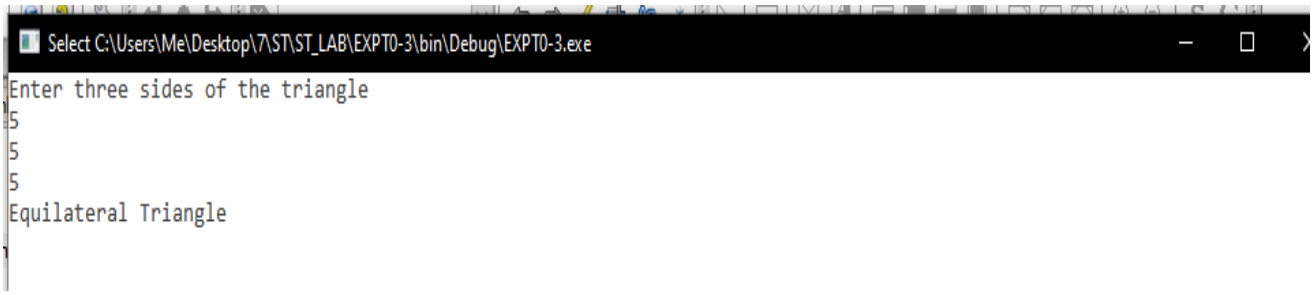
PROGRAM:

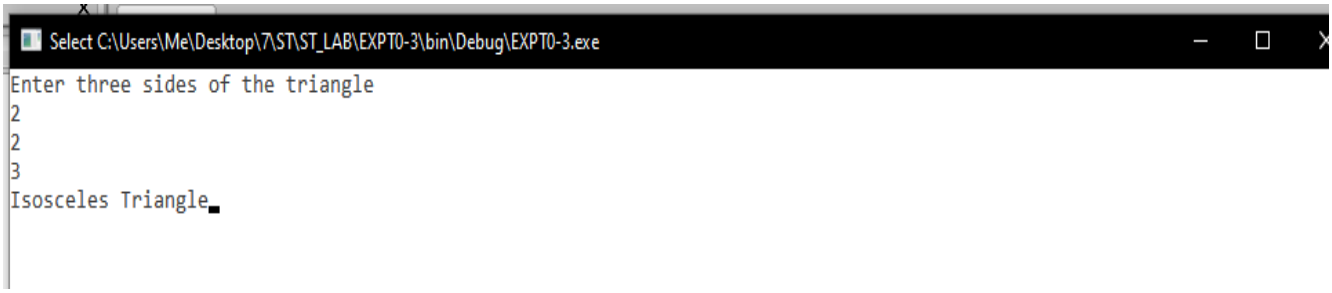
```
#include<stdio.h>
#include<ctype.h>
#include<conio.h>
#include<process.h>
int main(){
    int a,b,c;
    clrscr();
    printf("Enter three sides of a triangle");
    scanf("%d%d%d",&a,&b,&c);
    if((a>10)||(b>10)||(c>10)){
        printf("Out of range");
        getch();
        exit(0);
    }
    if((a<b+c)&&(b<a+c)&&(c<b+a)){
        if((a==b)&&(b==c)){
            printf("Equilateral Triangle");
        }
        else if((a!=b)&&(a!=c)&&(b!=c)){
            printf("Scalene Triangle");
        }
        else
            printf("Isosceles Triangle"); }
    else{
        printf("Triangle cannot be formed");
    }
    getch();
    return 0;
}
```

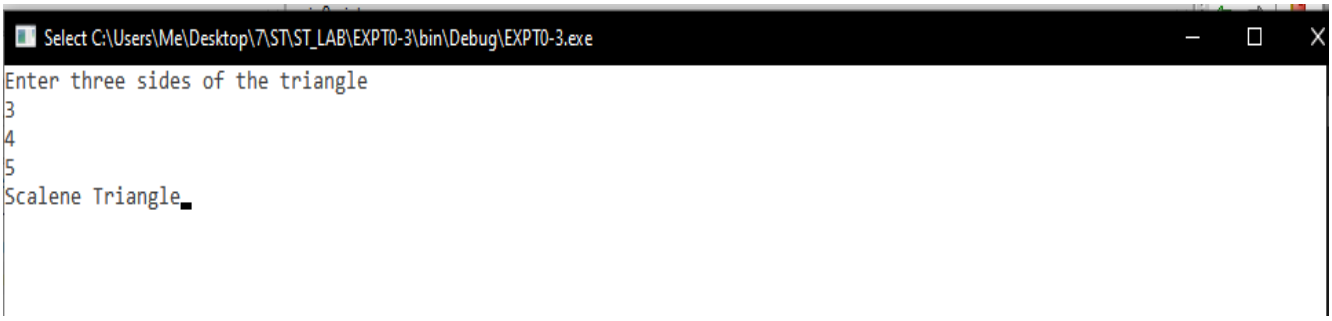
Tc ID	Tc Description	Input Data			Expected Output	Actual Output	Status
		A	B	C			
1	WN1	5	5	5	Equilateral	Equilateral	Pass
2	WN2	2	2	3	Isosceles	Isosceles	Pass
3	WN3	3	4	5	Scalene	Scalene	Pass
4	WN4	4	1	2	Not Triangle	Not Triangle	Pass
5	WR1	-1	5	5	Out of Range	Out of Range	Pass
6	WR2	5	-1	5	Out of Range	Out of Range	Pass
7	WR3	5	5	-1	Out of Range	Out of Range	Pass
8	WR4	11	5	5	Out of Range	Out of Range	Pass
9	WR5	5	11	5	Out of Range	Out of Range	Pass
10	WR6	5	5	11	Out of Range	Out of Range	Pass


Select C:\Users\Me\Desktop\7\ST\ST_LAB\EXPT0-3\bin\Debug\EXPT0-3.exe
Enter three sides of the triangle
5
11
3
Out of range


Select C:\Users\Me\Desktop\7\ST\ST_LAB\EXPT0-3\bin\Debug\EXPT0-3.exe
Enter three sides of the triangle
1
4
2
Triangle cannot be formed


Select C:\Users\Me\Desktop\7\ST\ST_LAB\EXPT0-3\bin\Debug\EXPT0-3.exe
Enter three sides of the triangle
5
5
5
Equilateral Triangle


Select C:\Users\Me\Desktop\7\ST\ST_LAB\EXPT0-3\bin\Debug\EXPT0-3.exe
Enter three sides of the triangle
2
2
3
Isosceles Triangle


Select C:\Users\Me\Desktop\7\ST\ST_LAB\EXPT0-3\bin\Debug\EXPT0-3.exe
Enter three sides of the triangle
3
4
5
Scalene Triangle

Test Report:

Number of Test Cases executed: 10

Number of Defects reported: 0

Number of Test cases passed: 10

Number of Test cases failed: 0

Conclusion:

- We have implemented a program to solve the triangle problem.
- Used the equivalence class partitioning technique to create and verify against test cases.

References:

- Paul C Jorgensen 3rd Edition