Design and develop a program in a language of your choice to solve the triangle problem defined as follows: Accept three integers which are supposed to be the three sides of a triangle and determine if the three values represent an equilateral triangle, isosceles triangle, scalene triangle, or they do not form a triangle at all. Assume the upper limit for the size of any side is 10. Derive test cases for your program based on equivalence class partitioning, execute the test cases and discuss the results.

Test Data : Enter the 3 Integer Value( a , b And c )

Pre-condition : 1 ≤ a ≤ 10 , 1 ≤ b ≤ 10 and 1 ≤ c ≤ 10 and a < b + c , b < a + c and c < a + b

Brief Description : Check whether given value for a equilateral, isosceles , Scalene triangle or Not a triangle

Code :

#include<stdio.h>

int main()

{

int a,b,c,c1,c2,c3;

char istriangle;

do {

printf("\nEnter 3 integers which are sides of triangle\n");

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

printf("\na=%d\tb=%d\tc=%d",a,b,c);

c1 = a>=1 && a<=10;

c2= b>=1 && b<=10;

c3= c>=1 && c<=10;

if (!c1)

printf("\nThe value of a=%d is not the range of permitted value", a);

if (!c2)

printf("\nThe value of b=%d is not the range of permitted value", b);

if (!c3)

printf("\nThe value of c=%d is not the range of permitted value", c);

} while (! (c1 && c2 && c3));

// to check is it a triangle or not

if( a<b+c && b<a+c && c<a+b )

istriangle='y';

else

istriangle ='n';

if (istriangle=='y')

if ((a==b) && (b==c))

printf("Equilateral triangle\n");

else if ((a!=b) && (a!=c) && (b!=c))

printf("Scalene triangle\n");

else

printf("Isosceles triangle\n");

else

printf("Can't form a triangle\n");

return 0;}

**Triangle Problem -Equivalence Class Test cases for input data**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Weak Normal Equivalence Class Testing** | | | | | | | |
| **Case id** | **Description** | **Input data** | | | **Expected output** | **Actual output** | **Comments** |
| a | b | c |
| 1 | Enter value for a , b and c | 5 | 5 | 5 | Equilateral triangle | Equilateral triangle | Pass |
| 2 | Enter value for a , b and c | 2 | 2 | 3 | Isosceles triangle | Isosceles triangle | Pass |
| 3 | Enter value for a , b and c | 3 | 4 | 5 | Scalene triangle | Scalene triangle | Pass |
| 4 | Enter value for a , b and c | 4 | 1 | 2 | Not a triangle | Not a triangle | Pass |
| **Weak Robust Equivalence Class Testing** | | | | | | | |
| 5 | Enter one invalid input and two valid value for a, b and c | -1 | 5 | 5 | Value of a is not in the range of permitted values | Value of a is not in the range of permitted values | Pass |
| 6 | Enter one invalid input and two valid value for a, b and c | 5 | -1 | 5 | Value of b is not in the range of permitted values | Value of b is not in the range of permitted values | Pass |
| 7 | Enter one invalid input and two valid value for a, b and c | 5 | 5 | -1 | Value of c is not in the range of permitted values | Value of c is not in the range of permitted values | Pass |
| 8 | Enter one invalid input and two valid value for a, b and c | 11 | 5 | 5 | Value of a is not in the range of permitted values | Value of a is not in the range of permitted values | Pass |
| 9 | Enter one invalid input and two valid value for a, b and c | 5 | 11 | 5 | Value of b is not in the range of permitted values | Value of b is not in the range of permitted values | Pass |
| 10 | Enter one invalid input and two valid value for a, b and c | 5 | 5 | 11 | Value of c is not in the range of permitted values | Value of c is not in the range of permitted values | Pass |
| **Strong Robust Equivalence Class Testing** | | | | | | | |
| 11 | Enter one invalid input and two valid value for a , b and c | -1 | 5 | 5 | Value of a is not in the range of permitted values | Value of a is not in the range of permitted values | Pass |
| 12 | Enter one invalid input and two valid value for a , b and c | 5 | -1 | 5 | Value of b is not in the range of permitted values | Value of b is not in the range of permitted values | Pass |
| 13 | Enter one invalid input and two valid value for a , b and c | 5 | 5 | -1 | Value of c is not in the range of permitted values | Value of c is not in the range of permitted values | Pass |
| 14 | Enter two invalid input and one valid value for a , b and c | -1 | -1 | 5 | Value of a,b is not in the range of permitted values | Value of a,b is not in the range of permitted values | Pass |
| 15 | Enter two invalid input and one valid value for a , b and c | 5 | -1 | -1 | Value of b,c is not in the range of permitted values | Value of b,c is not in the range of permitted values | Pass |
| 16 | Enter two invalid input and one valid value for a , b and c | -1 | 5 | -1 | Value of a,c is not in the range of permitted values | Value of a,c is not in the range of permitted values | Pass |
| 17 | Enter all invalid inputs | -1 | -1 | -1 | Value of a,b,c is not in the range of permitted values | Value of a,b,c is not in the range of permitted values | Pass |

**Equivalence classes:**

D1 = {<a, b, c>: a = b = c}

D2 = {<a, b, c>: a = b, a ≠ c}

D3 = {<a, b, c>: a = c, a ≠ b}

D4 = {<a, b, c>: b = c, a ≠ b}

D5 = {<a, b, c>: a ≠ b, a ≠ c, b ≠ c}

D6 = {<a, b, c>: a ≥ b + c}

D7 = {<a, b, c>: b ≥ a + c}

D8 = {<a, b, c>: c ≥ a + b}