

## Experiment No. 3

Aim :- Write a program in C/C++ to read 3 side of a triangle & to determine whether they form scalene, isosceles or equilateral triangle and test the same using basic path testing and find its  $V(G)$  by all the three methods.

Theory :- Basic path helps a tester to compute logical complexity measure,  $V(G)$ , of the code. This value of  $V(G)$ , defines the maximum number of test cases to be designed by identifying basic set of execution paths to ensure that all statements are executed at least once.

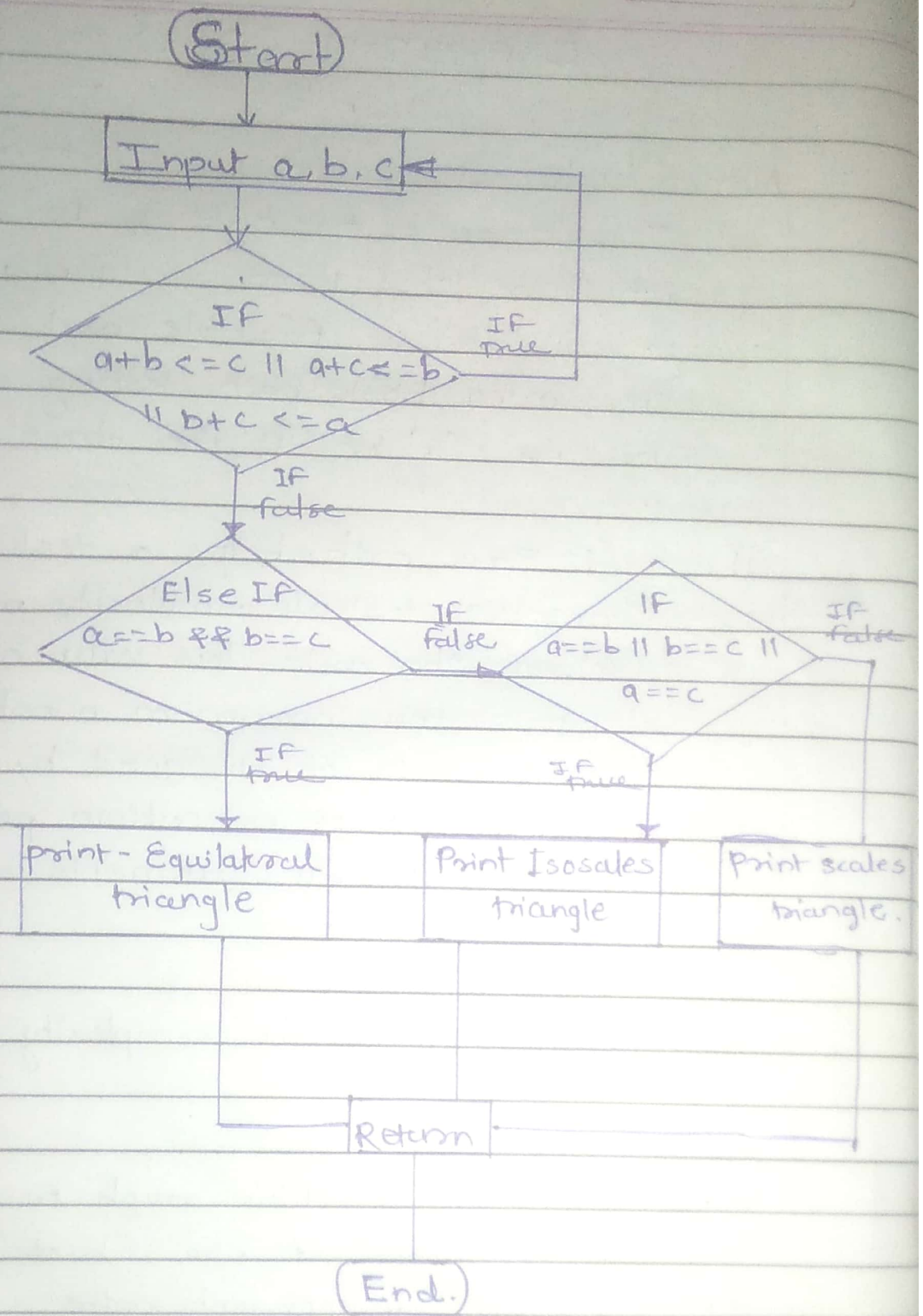
Steps to compute the complexity measure,  $V(G)$  are as under.

Step 1. Construct the Flow graph from the source code or flow charts.

Step 2. Identify independent paths.

Step 3. Calculate Cyclomatic Complexity,  $V(G)$ .

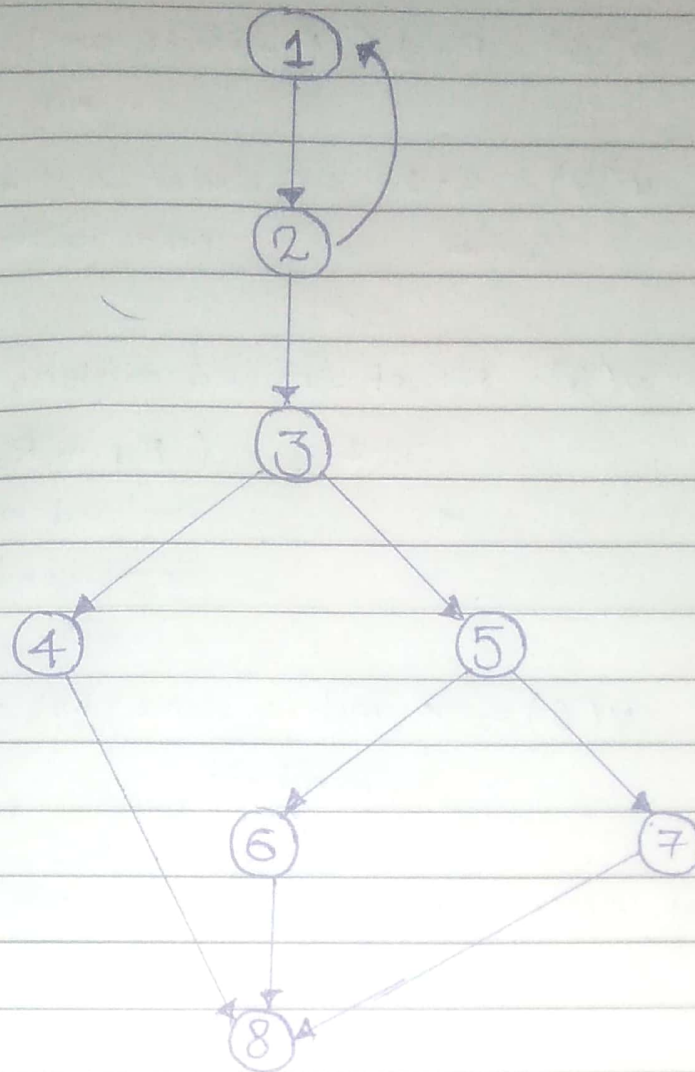
Step 4. Design the test cases.



Flow Chart

Converting Flow Chart into Directed Graph





Directed Graph.

Calculating Cyclomatic Complexity  $V(G)$  by three methods:

$M=1$ .  $V(G) = e - n + 2$  (where  $e$  are edges &  $n$  are nodes)

$$\begin{aligned} V(G) &= 10 - 8 + 2 \\ &= 2 + 2 \\ &= 4. \end{aligned}$$

$M=2$ .  $V(G) = P+1$  ( $P$  - No. of predicate nodes with out degree = 2)

$$V(G) = 3+1 = 4 \quad (\text{Nodes 2, 4 \& 6 are predicate nodes with 2 outgoing edges})$$

$M=3$ .  $V(G) = \text{No. of enclosed regions} + 1$

$$= 3+1 = 4 \quad (R_1 \& R_2 \& R_3 \text{ are the enclosed regions \& 1 corresponds to one outer region})$$

$V(G) = 4$  and is same by all the three methods.

Test cases for each path are:

Test cases	valid Input	Expected results.
Enlist 1 <sup>st</sup> Path	$a, b, c$ : valid input	if $s = b$ or $b = c$ or $a = c$ then 'isosce triangle' display
Enlist 2 <sup>nd</sup> Path	$a, b, c$ : valid input	if $a \neq b \neq c$ then 'scalene triangle' displayed.
Enlist 3 <sup>rd</sup> Path	$a, b, c$ : valid input	if $a = b = c$ then 'Equilateral triangle' displayed.
Enlist 4 <sup>th</sup> Path	$a, b, c$ : invalid input	Go to Enter Value of $a, b, c$ .



Program :-

```
#include <stdio.h>
#include <iostream>
using namespace std;
int main() {
    int a, b, c;
    first:
    cout << "Enter value of sides for triangle:\n";
    cin >> a >> b >> c;
    if (a + b <= c || a + c <= b || b + c <= a) {
        cout << "Invalid Entries\n";
        goto first;
    }
    else {
        if (a == b && b == c)
            cout << "Equilateral Triangle";
        else if (a == b || b == c || c == a)
            cout << "Isosceles Triangle";
        else
            cout << "Scalene Triangle";
    }
}
```

Conclusion :- Thus we have studied & executed program to read 3 sides of triangle & to determine whether they form scalene, isosceles or equilateral triangle and tested the same using basic path testing calculated  $V(G)$  by all the three methods.

 F:\#BLACKHEART\#SOFTWARE\_TESTING\Practical Programs\Practical\_No\_3.exe

Enter value of sides for trianlge :

6 10 6

Isosceles Triangle

 F:\#BLACKHEART\#SOFTWARE\_TESTING\Practical Programs\Practical\_No\_3.exe

Enter value of sides for trianlge :

12 15 17

Scalene Triangle

 F:\#BLACKHEART\#SOFTWARE\_TESTING\Practical Programs\Practical\_No\_3.exe

Enter value of sides for trianlge :

10 10 20

Invalid Entries

Enter value of sides for trianlge :

28 28 28

Equilateral Triangle