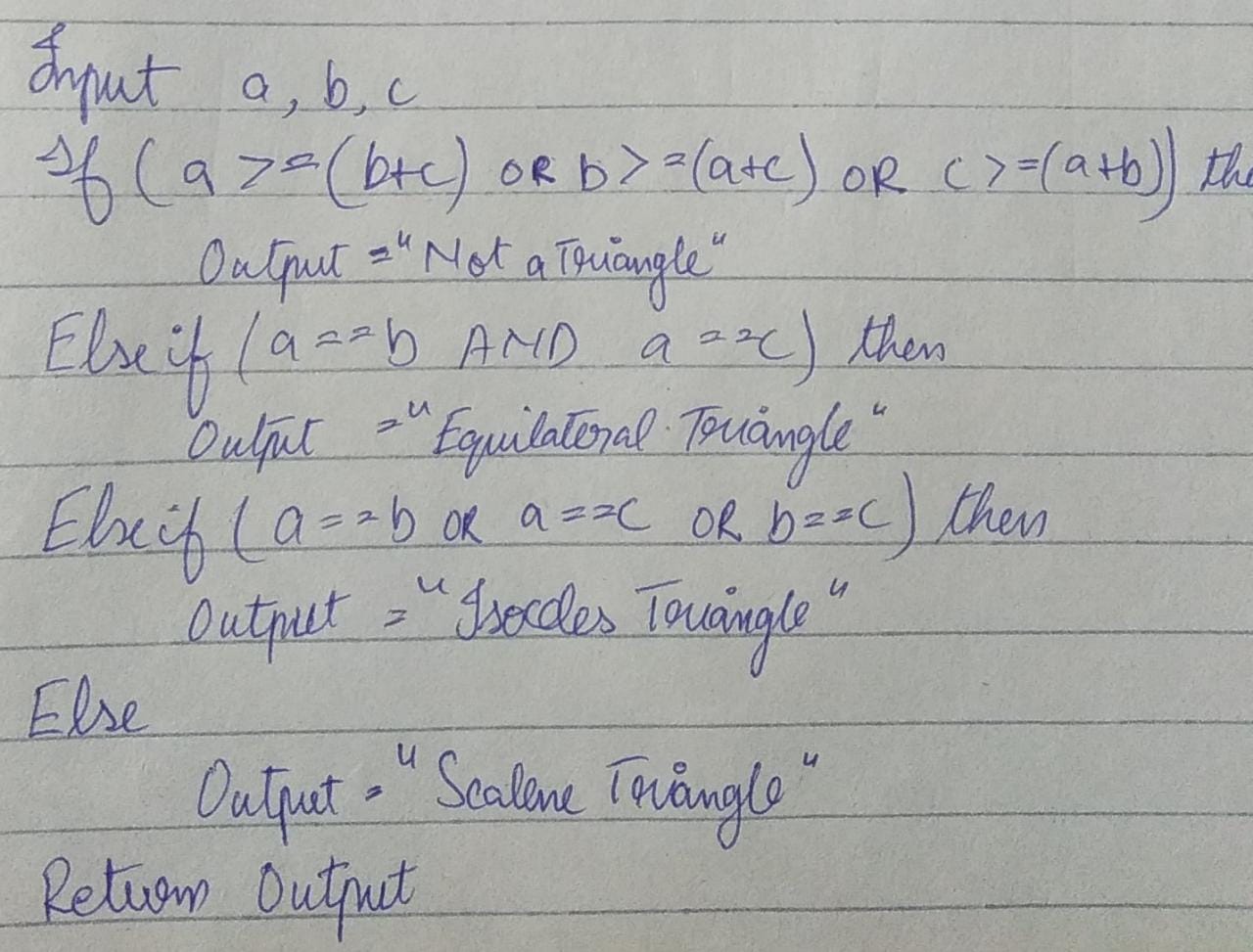
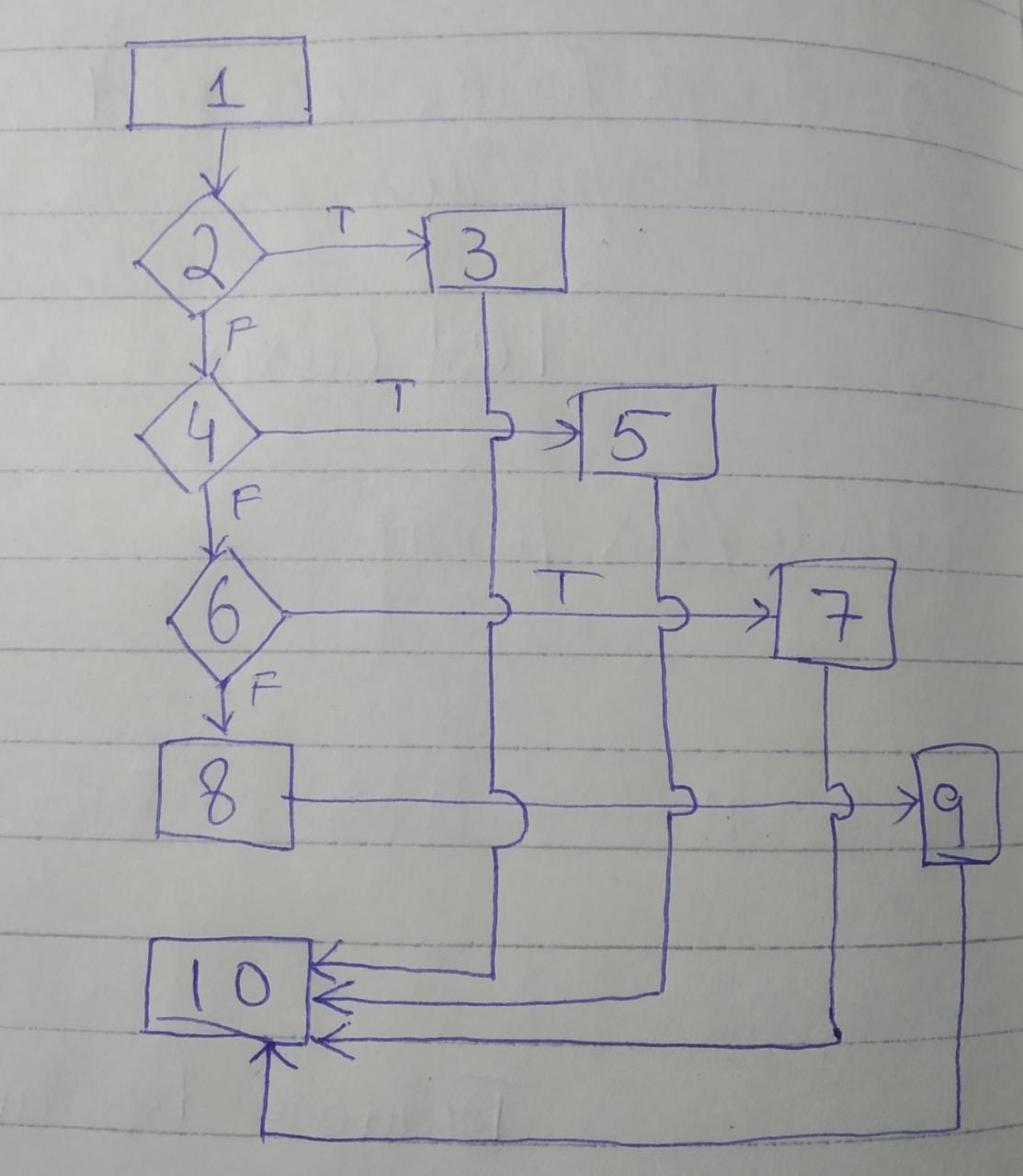
**Sample Code & Control Flow Graph Below**

* Code(Source YouTube)



* Control Flow Graph



**Basis Path Testing**

Basis Path Testing is the testing based on 4 steps. The steps are given below:

1. Construct a Control Flow Graph
2. Calculation of Cyclomatic Complexity(V(G)=E~(No. of Edges)-N~(No. of Vertices/Nodes)+2(P~No. of connected components))
3. Set of Paths
4. Testing of each path in the Set

This testing technique ensures, 100% statement & branch coverage.

Example:

Triangle Problem

* Cyclomatic Complexity
  + (V(G)=E~(No. of Edges)-N~(No. of Vertices/Nodes)+2(P~No. of connected components))
  + Calculation: V(G)=12-10+2=**4**
* Set of Paths
  + 1->2->3->10 (Path-1)
  + 1->2->4->5->10 (Path-2)
  + 1->2->4->6->7->10 (Path-3)
  + 1->2->4->6->8->9->10 (Path-4)
* Test Cases
  + Path-1
    - Input(a=10,b=5,c=5), Output(Not A Triangle), Expected Output(Not A Triangle)
  + Path-2
    - Input(a=10,b=10,c=10), Output(Equilateral Triangle), Expected Output(Equilateral Triangle)
  + Path-3
    - Input(a=5,b=5,c=8), Output(Isosceles Triangle), Expected Output(Isosceles Triangle)
  + Path-4
    - Input(a=1,b=2,c=3), Output(Scalene Triangle), Expected Output(Scalene Triangle)

**Branch Coverage Testing**

Now, applying branch coverage testing on above will give the coverage of all the edges as we are covering all the branches at all conditions, in branch coverage testing.

So, the Set of Paths to be tested will be:

* 1->2->3->10
* 1->2->4->6->8->9->10
* 1->2->4->5->10
* 1->2->4->6->7->10

So, actually the above Basis Path testing and Branch Coverage Testing is working the same for our case, also the test cases be same. But, for longer branching, Branch Coverage makes sure to cover all the edges.