White Box Testing

CYCLOMATIC COMPLEXITY

- ***Cyclomatic complexity*** is a software metric that provides a quantitative measure of the logical complexity of a program.

- Cyclomatic complexity **defines number of independent paths** which can be further used in development of test cases.

Complexity is computed in one of three ways:

1. The **number of regions** of the flow graph corresponds to the cyclomatic complexity.

2. Cyclomatic complexity *V*(*G*) for a flow graph *G* is defined as

$$V(G) = E - N + 2$$ (**McCabe Complexity Measure**)

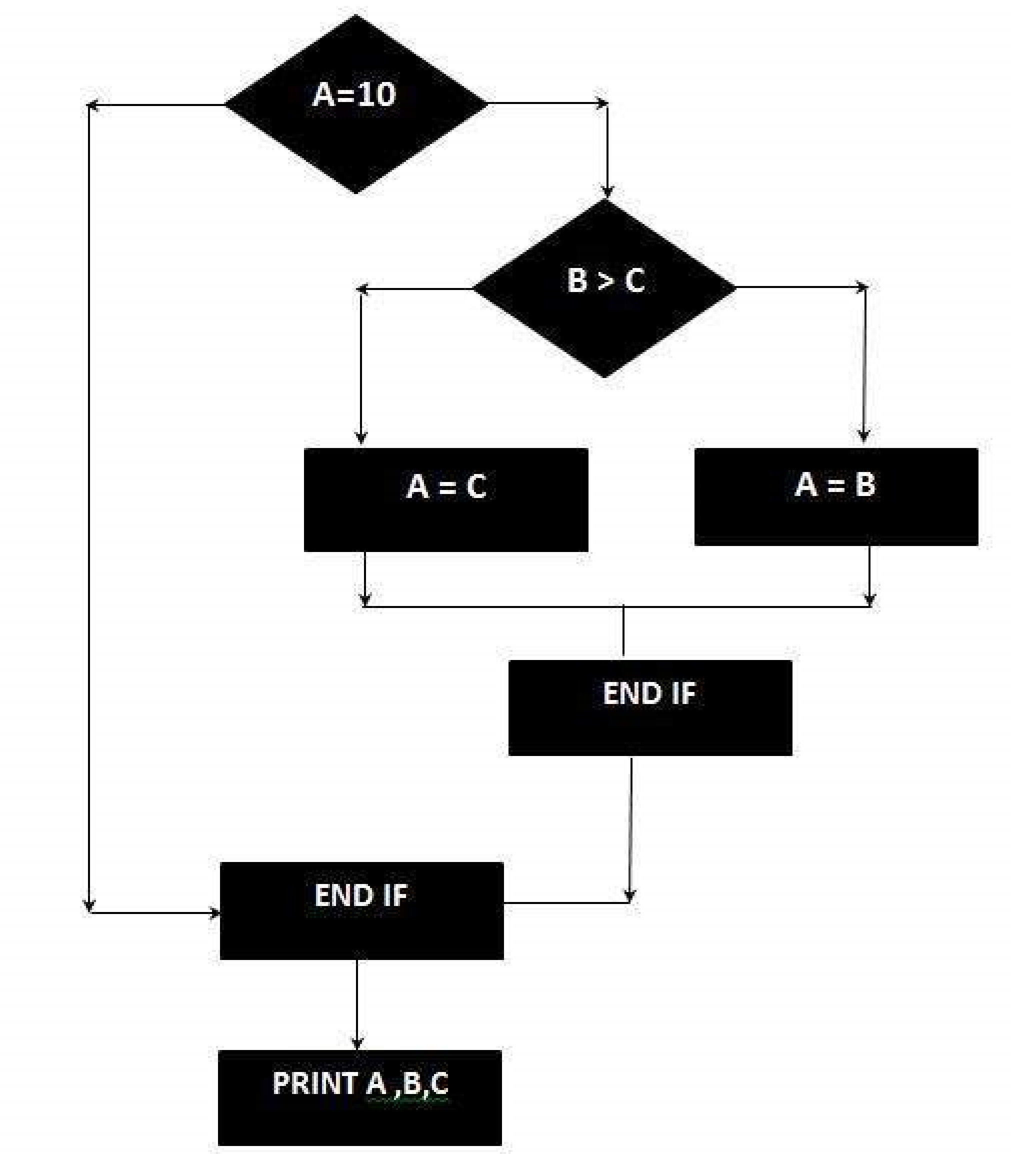
where *E* is the number of flow graph edges and *N* is the number of flow graph nodes.

3. Cyclomatic complexity *V*(*G*) for a flow graph *G* is also defined as

$$V(G) = P + 1$$

where *P* is the number of predicate nodes contained in the flow graph *G*.

```
IF A = 10 THEN
IF B > C  THEN
A = B
ELSE
A = C
ENDIF
ENDIF
Print A
Print B
Print C
```



A=10

B > C

A = C

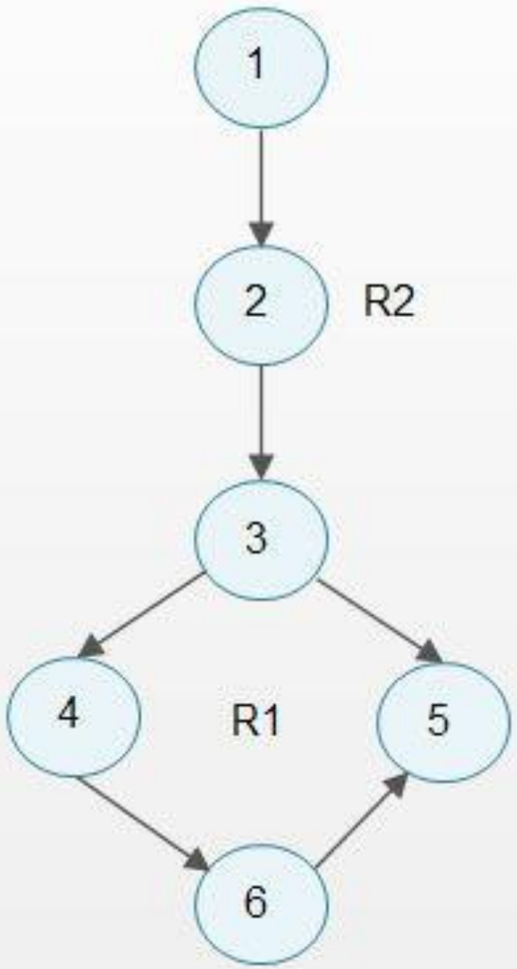
A = B

END IF

END IF

PRINT A ,B,C

```
procedure greater;
integer: x, y, $z = 0$;
enter the value of x;
enter the value of y;
if $x > y$ then
$z = x$;
else
$z = y$;
end greater
```



Flow graph to Find the Greater beetween Two Numbers

Determine all independent paths through the program

For the flow graph depicted the independent paths are listed below.

P1: 1-2-3-4-6

P2: 1-2-3-5-6

Compute the cyclomatic complexity

CC = 2 as there two regions R1 and R2

or

CC 6 edges - 6 nodes + 2 =2

or

CC 1 predicate node + 1 = 2.