what is driver ?

→ is a testing component used in bottom-up approach to unit testing which simulates the higher level module / component that a unit being tested interacts with.

Example :- if module A calls module B, module A is being tested, a driver is created to simulate the behaviour of module B providing inputs and receiving outputs as if module B were functioning normally.

what is stub ?

→ is a testing component used in top down approach to unit testing which simulates the lower level modules allowing higher level of modules to be test independently

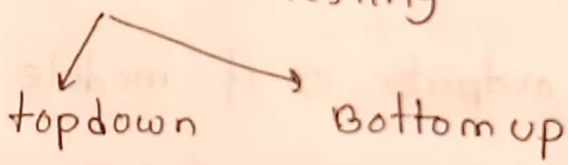
Example :- if module A calls B. and module B is not fully implemented / doesn't exist. a stub is created to simulate the behaviour of B and return predetermine respons

(*) What is integration testing

→ is a software testing phase where individual components/units are tested and combined as group to verify that they can work together on not

(i) Non-incremental integration testing

(ii) incremental integration testing



(*) why incremental integration testing is preferred rather than using non incremental testing ?

(1) Early detection of defects in incremental testing

② more efficient debugging

③ allows continous Validation

④ effective risk management

⑤ Better Resource utilization

⑥ Fasten Feedback loop

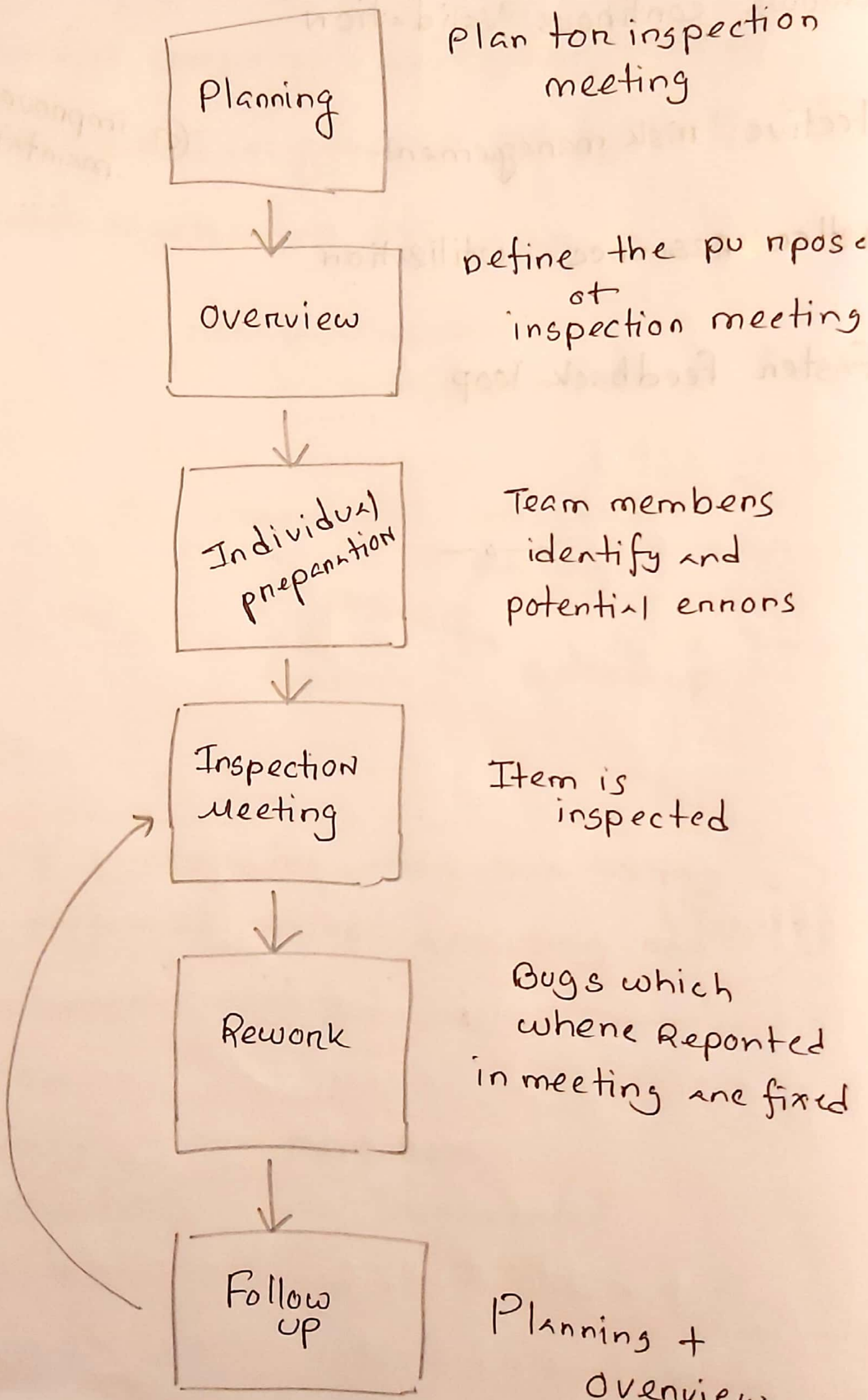
⑦ improves maintainability

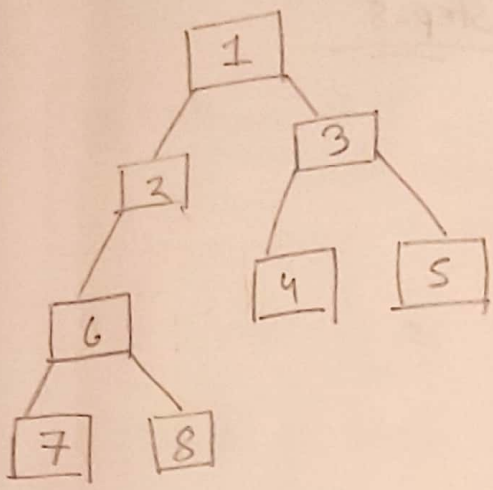
—o—

17 Final

16 Final

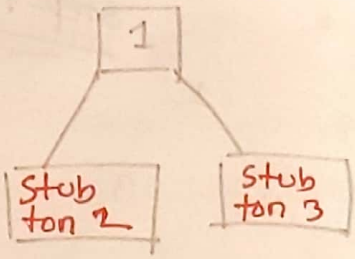
Inspection process

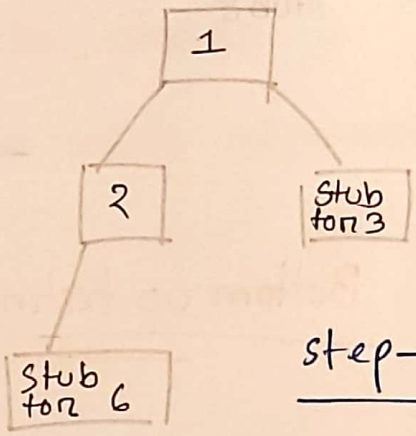




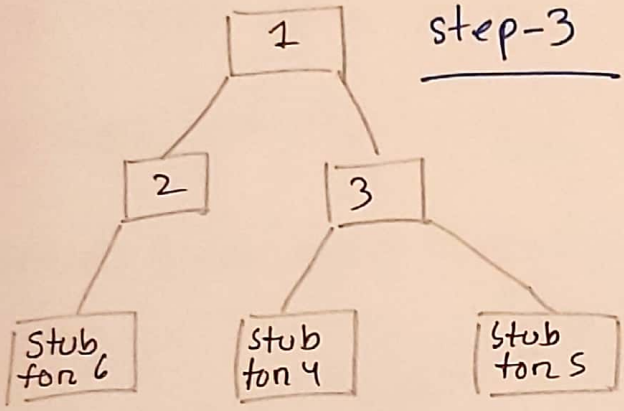
Top down testing →

step-1

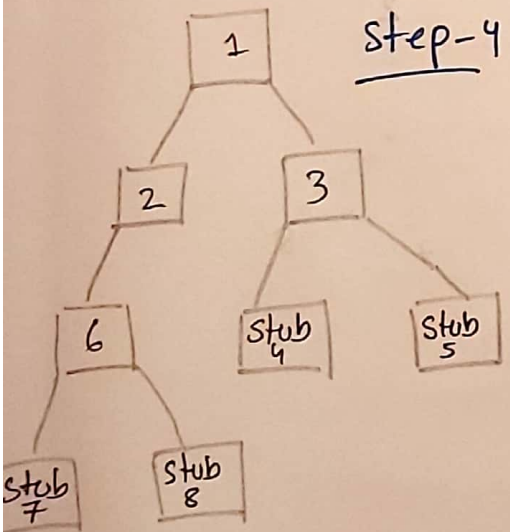




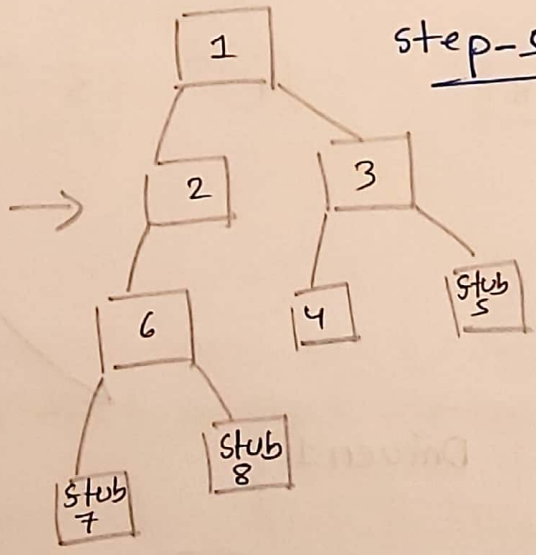
step-2



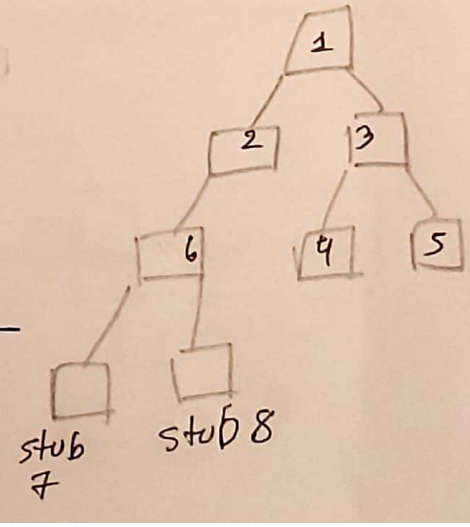
step-3



step-4

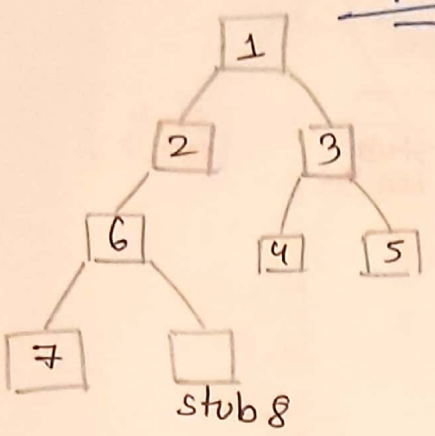


step-5

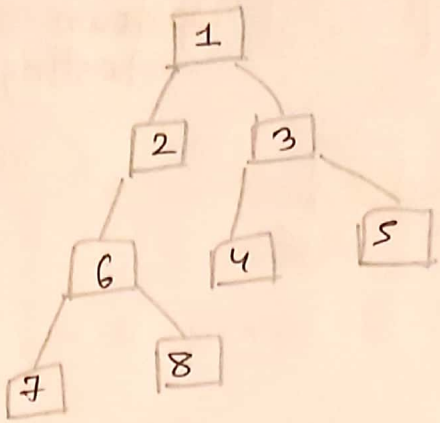


step-6

step-7

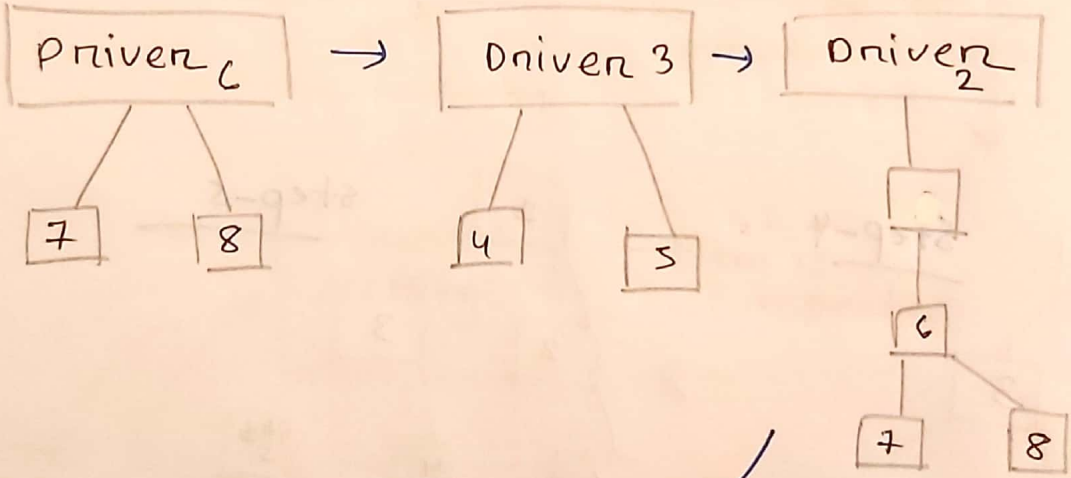


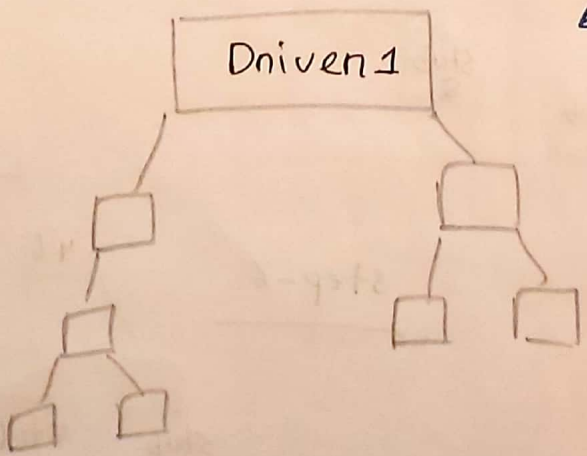
step-8



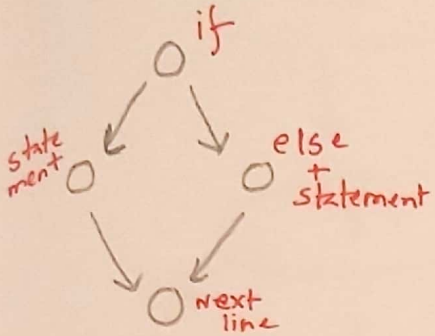
— o —

Bottom up testing

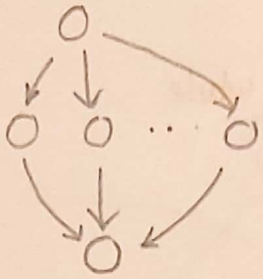




cyclomatic complexity :-



if else



switch case

while do

Method - 1 :-

$$V = E - N + 2P$$

E = edge
N = Node
P = 1
= connected component

Method - 2

$$V = \pi + 1$$

π = predicted Node
(Node has two outgoing edge)

Method - 3

V = No of regions

region = inside cycle + 1

Method - 4

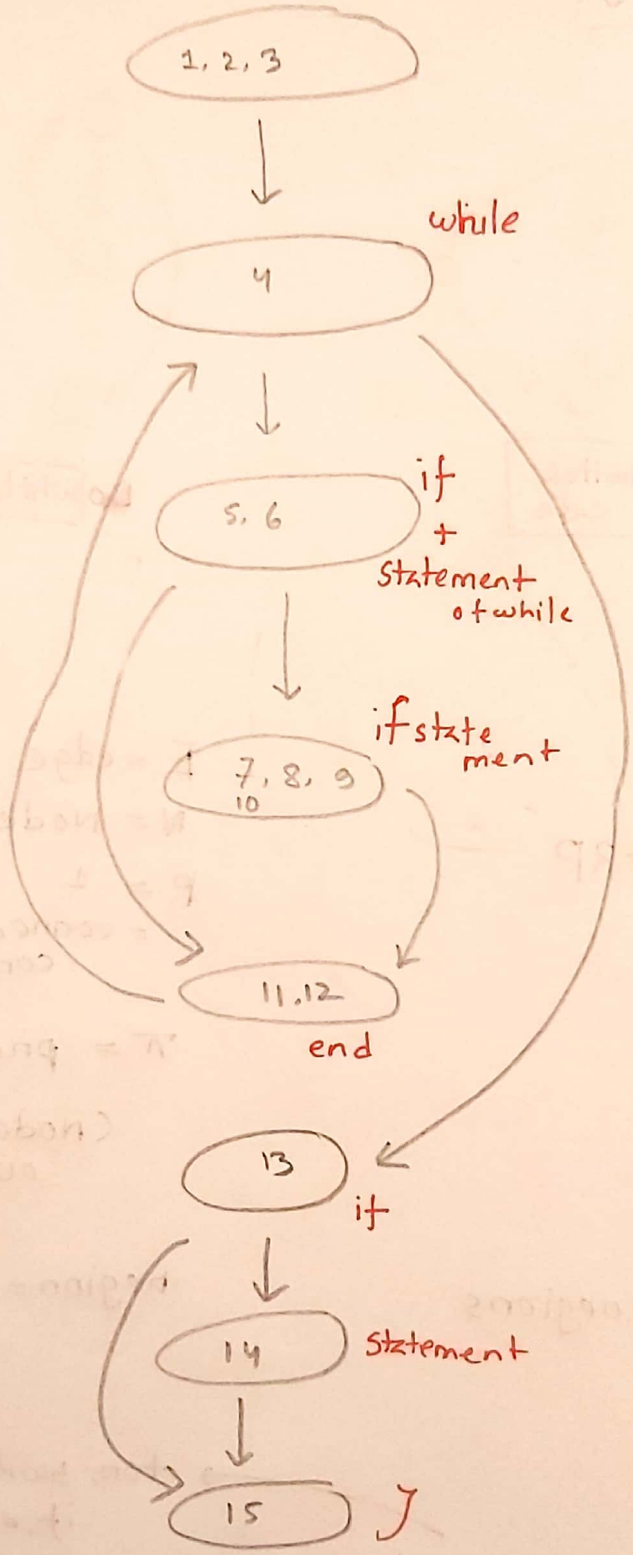
V = binary decision + 1 → ton, switch, if, else

[if, elseif, else]
(3-1) + 1
= 3

method - 5

using connection matrix

Prime Number from book

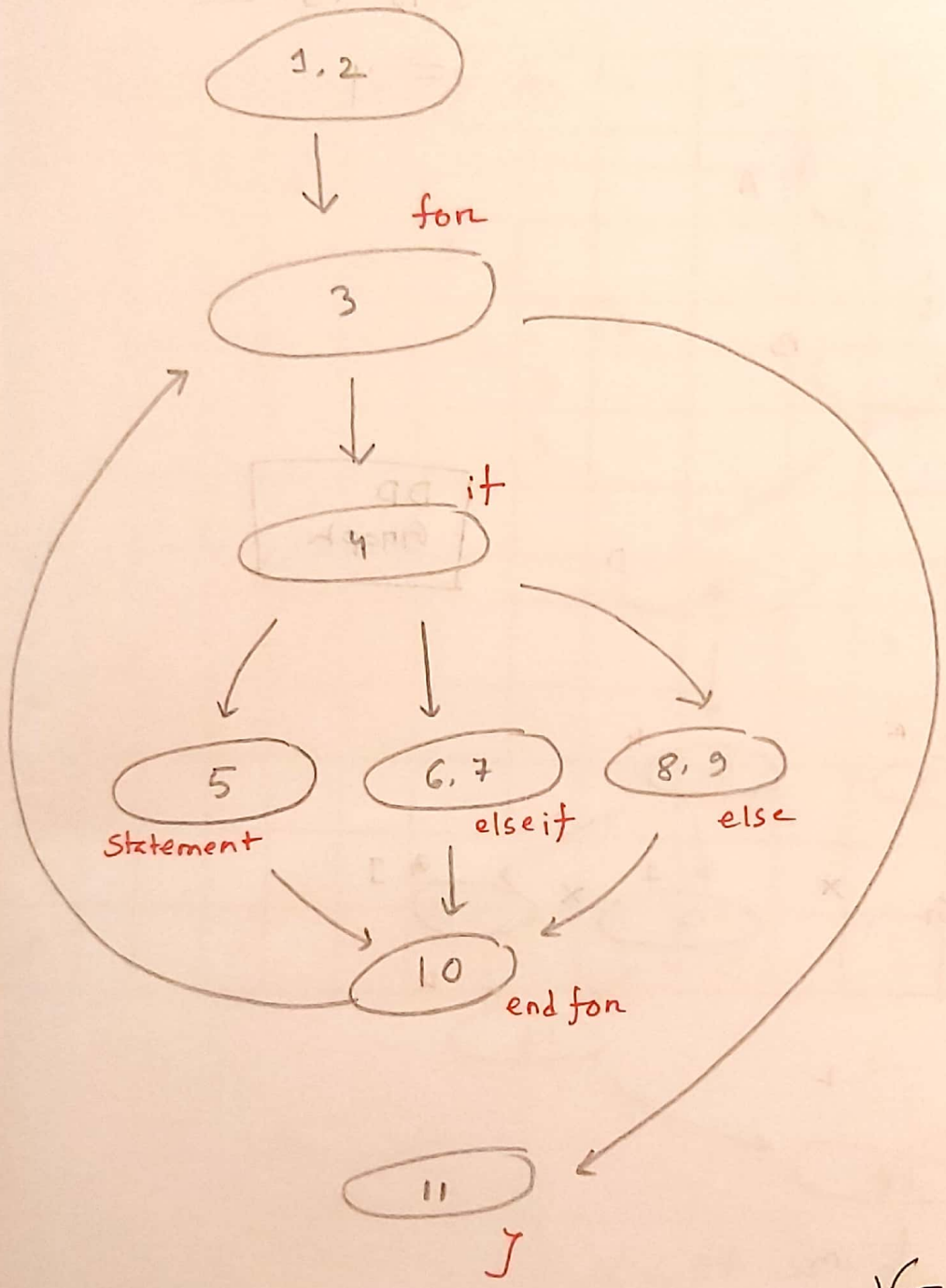


$V(G) = E - N + 2$

$= 10 - 8 + 2$

$= 4$

19
TT-1



$V = E - N + 2$

$= 10 - 8 + 2$

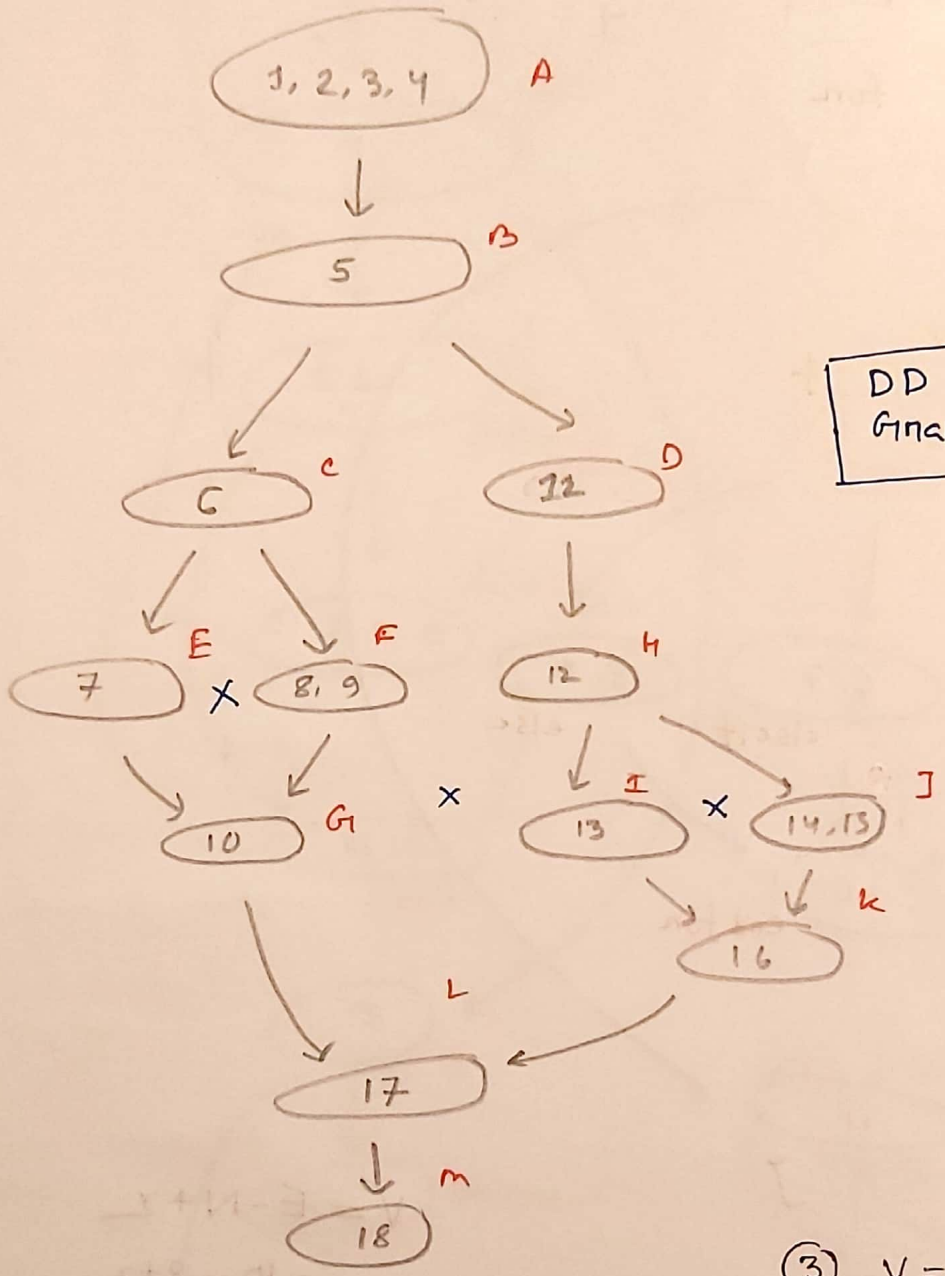
$= 4$

(18) Final (c)

(1) $V = E - N + 2p$

$= 15 - 13 + 2$

$= 4$



(2) $V = \pi + 1$

$= \text{No predicted node (B, C, D)} + 1$

$= 4$

(3) V = No of regions

$= 3 + 1$

$= 4$

④ connection matrix

	A	B	C	D	E	F	G	H	I	J	K	L	M	
A		1												1 - 1 = 0
B			1	1										2 - 1 = 1
C					1	1								2 - 1 = 1
D								1						1 - 0 = 0
E							1							1 - 1 = 0
F							1							1 - 1 = 0
G												1		1 - 1 = 0
H									1	1				2 - 1 = 1
I											1			1 - 1 = 0
J											1			1 - 1 = 0
K												1		1 - 1 = 0
L													1	1 - 1 = 0
M														

3

$\therefore \quad V = 3 + 1$

$= 4$

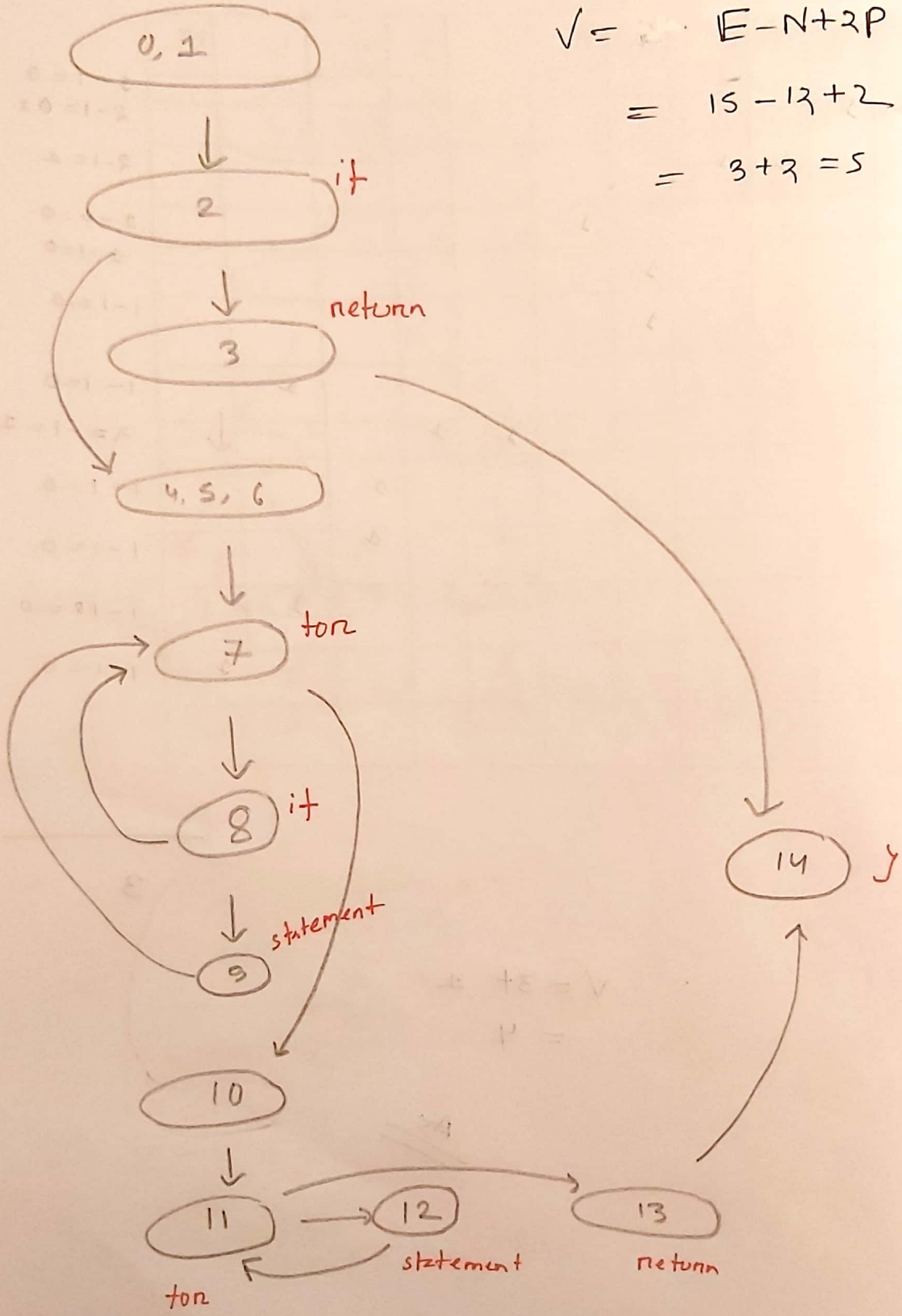
Ans

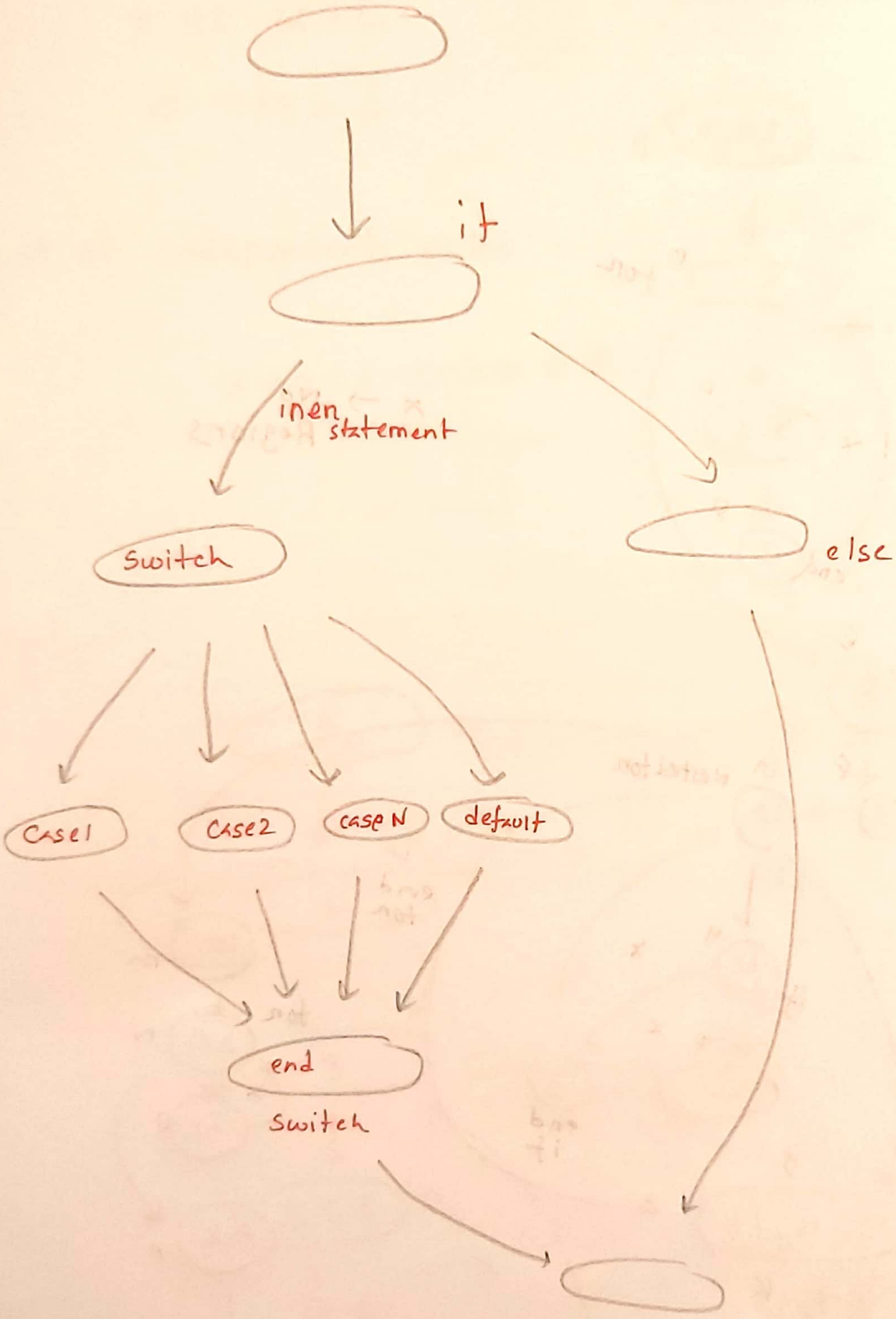
17 (1)

$V = E - N + 2P$

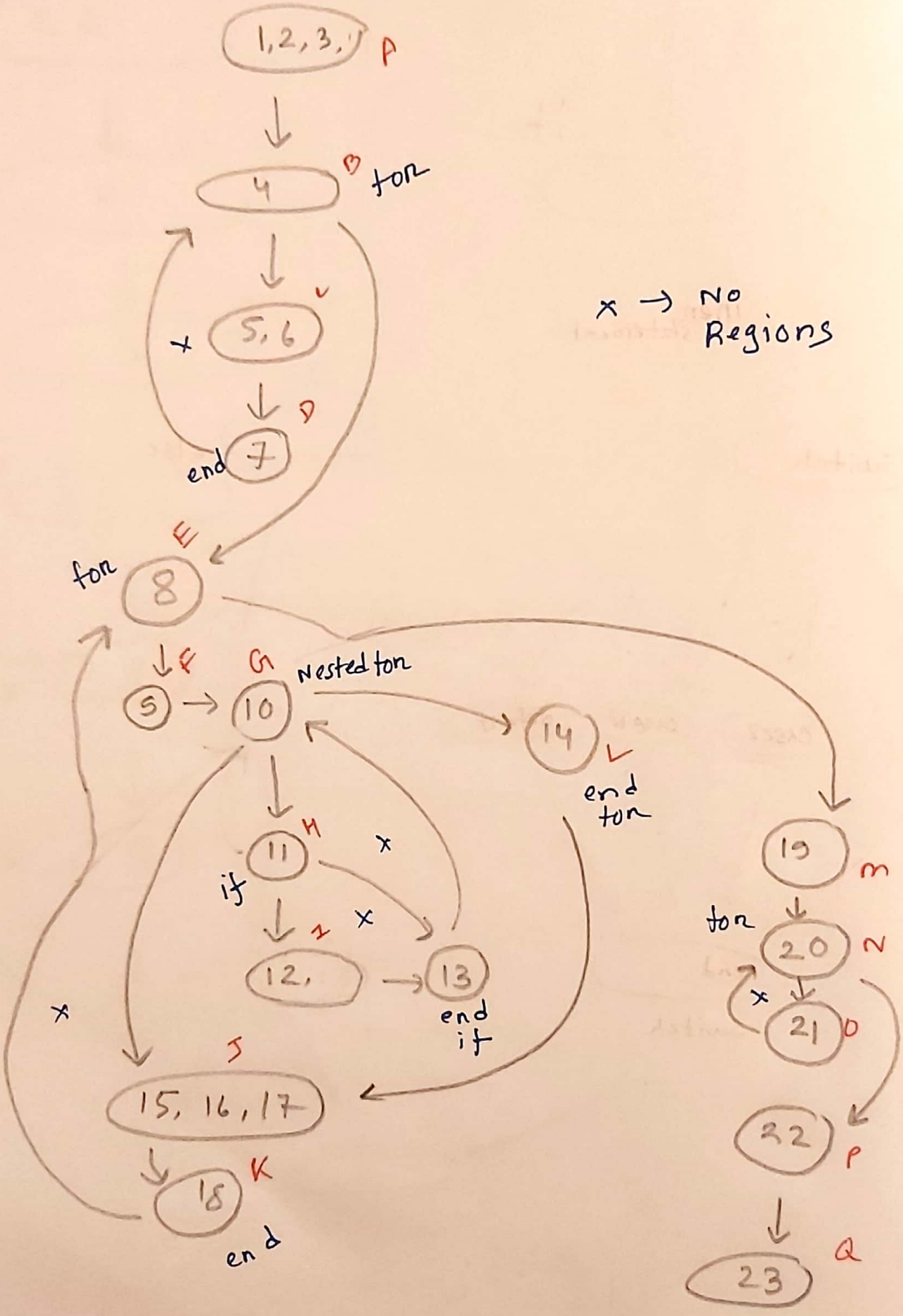
$= 15 - 13 + 2$

$= 3 + 2 = 5$




if
inen statement
switch
else
case1
case2
case N
default
end
switch

18 Term
Test 2



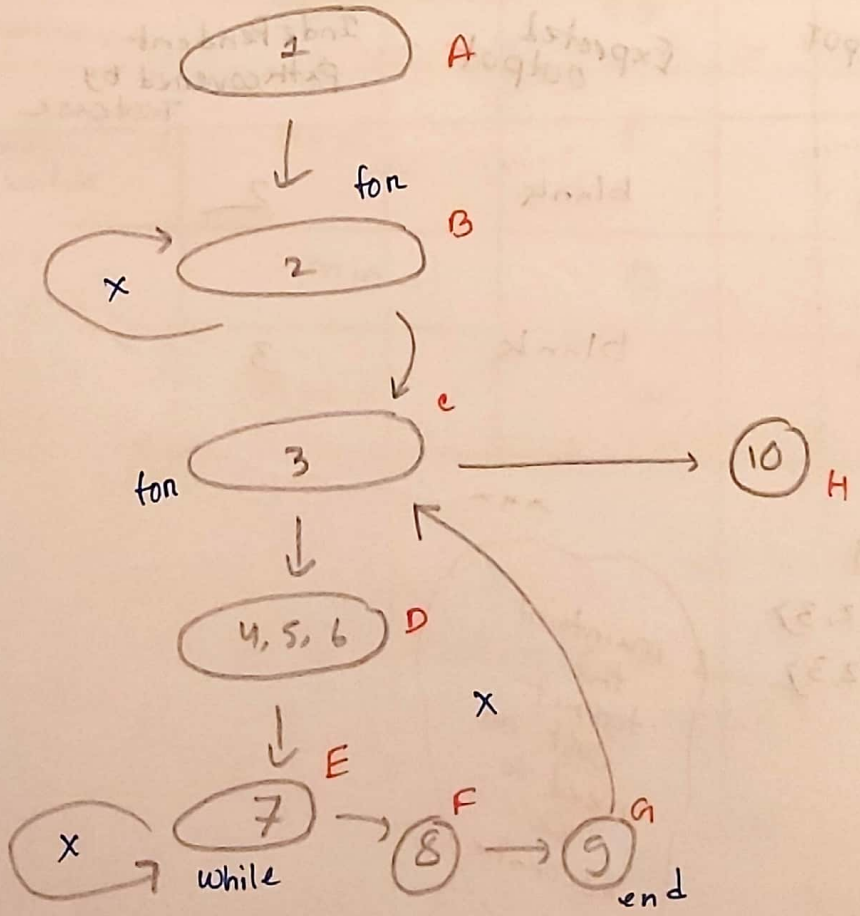
$$V = E - N + 2P$$
$$= 22 - 18 + 2$$
$$= 4 + 2 = 6$$

List of Independent paths :-

$A \rightarrow B \rightarrow C \rightarrow D \rightarrow B \rightarrow E$ ... (do in halls)

---

16 Final



$V = 10 - 8 + 2$
$= 4$

Independent paths :-

1. A B B C D E F G C H
2. A B C H
3. A B B C H
4. A B B C D E E F G C H

Test case design for independent path

Test case ID	INPUT	Expected output	Inde Pendent path covened by Test case
1	N=0	blank	2
2	N= 1	blank	3
3	N=3 a={1,2,3} P={1,2,3}	maintaint the format just no need to be accurate	
4			

(0, 100]

⊗ Calculates GCD :- A, B range [1, 100]

A = [1, 100]

B = [1, 100]

Boundary Value Test case

		A	B
minimum value	min	1	1
Value just above minimum	min+	2	2
maximum	max	100	100
Value just less maximum	Max-	99	99
Nominal value	Nom	5	10
just below less than minimum	min-	0	0
value above maximum	max+	101	101

∴ No of Variable = 2

(8)

BVC
Boundary Value test cases
$= 4n+1$

Boundary Value checking $= 4 \times 2 + 1 = 8 + 1 = 9$

Test ID	Formula	Test case	Expected Output
1	Anom, Bnom	(5, 10)	5
2	Anom, Bmin	(5, 1)	1
3	Anom, Bmint	(5, 2)	1
4	Anom, Bmax	(5, 100)	5
5	Anom, Bmax-	(5, 99)	1
6	Amin, Bnom	(1, 10)	1
7	Amint, Bnom	(2, 10)	2
8	Amax, Bnom	(100, 10)	10
9	Amax-, Bnom	(99, 10)	1

⑦ Robustness Testing

$n = 2$

$\therefore$ Test cases $= 6n + 1$

$= 12 + 1 = 13$

Testcase ID		Testcase	Expected output
1 – 9	From BVC Testcase		
10	A_{Nom}, B_{max^+}	5, 101	Invalid
11	A_{Nom}, B_{min^-}	5, 0	Invalid
12	A_{max^+}, B_{Nom}	101, 10	Invalid
13	A_{min^-}, B_{Nom}	0, 10	Invalid

⑧ worst case testing method

$\therefore n = 2$

$\therefore$ no of Testcases $= 5^n$

$= 5^2$

$= 25$

(1 – 9) ⟶ BVC Testcase

(10) Amin, Bmin ⟶ (1, 1) — 1

(11) Amin, Bmax ⟶ (1, 100) — 1

(12) Amin, BMint ⟶ (1, 2) — 1

(13) Amin, Bmax- ⟶ (1, 99) — 1

(14) Amint, Bmint ⟶ (2, 2) ⟶ 2

(15) Amint, Bmin ⟶ (2, 1) ⟶ 1

(16) Amint, Bmax ⟶ (2, 100) ⟶ 2

(17) Amint, Bmax- ⟶ (2, 99) ⟶ 1

(18) Amax, Bmax ⟶ (100, 100) ⟶ 100

(19) Amax, Bmin ⟶ (100, 1) ⟶ 1

(20) Amax, Bmint ⟶ (100, 2) ⟶ 2

(21) Amax, Bmax- ⟶ (100, 99) ⟶ 1

(22) Amax-, Bmax- ⟶ (99, 99) ⟶ 99

(23) Amax-, Bmin ⟶ (99, 1) ⟶ 1

(24) Amax-, Bmint ⟶ (99, 2) ⟶ 1

(25) Amax-, B, max ⟶ (99, 100) ⇒ 1