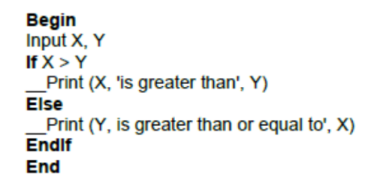
Tema 7-Tehnici de proiectare White Box

1. Afișează pseudocodul de mai jos în formă de schemă logică.



Care este numărul minim de cazuri de testare necesare pentru a garanta o acoperire de 100% statement și 100% decision?

A. Statement coverage = 3, Decision coverage = 3

B. Statement coverage = 2, Decision coverage = 2

C. Statement coverage = 1, Decision coverage = 2

D. Statement coverage = 2, Decision coverage = 1

Raspuns: **B. Statement coverage = 2, Decision coverage = 2**

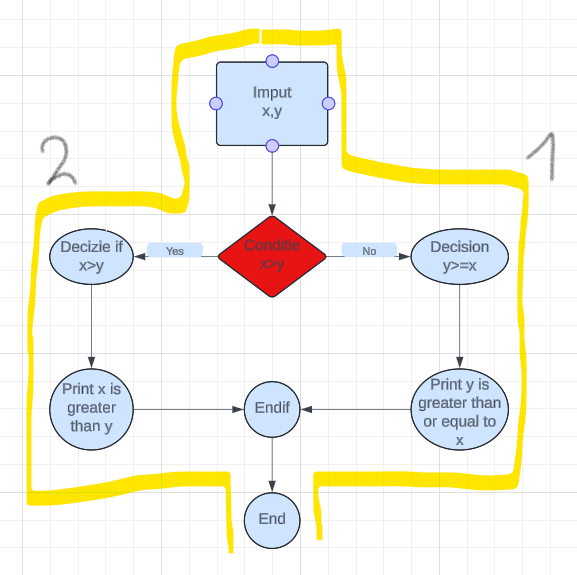
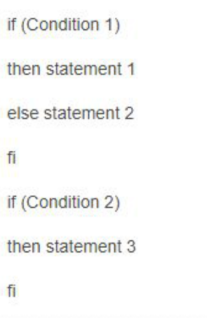


Diagrama 1

### **Sarcina 2.**

Afișează pseudocodul de mai jos în formă de schemă logică.



Care este numărul minim de cazuri de testare necesare pentru a garanta o acoperire de 100% path?

A. 1

B. 2

C. 3

D. Niciun răspuns nu este corect

Raspuns: **D. Niciun răspuns nu este corect**

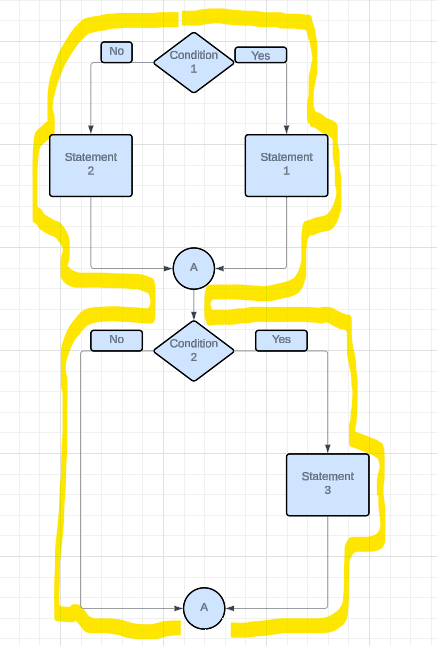
****

Diagrama 2

### **Sarcina 3.**

În cadrul unei echipe care dezvoltă o aplicație Java, clarificați cine din echipă este responsabil pentru teste white-box/static și ce tool-uri sunt utilizate în acest scop?

Pentru această sarcină, este necesar să efectuați un research de sine stătător și să căutați exemple de tool-uri și framework-uri folosite pentru testarea aplicației scrisă în Java.

Persoana din echipa responsabilă cu testarea white box/static este testerul sau programatorul, în funcție de nivelul testării. La nivelul unit testing si integration testing, va fii realizat de programator dar poate fii realizat si de un tester, dar cu experienta. La nivel de end to end testing, testarea este realizata de către tester. Testarea White Box este realizata doar la nivelul de unit testing; la celelalte doua nivele este realizata testarea Black Box.

Programele folosite de developer în limbajul de programare Java pot fii:

* **Site 24x7** (este un tool folosit pentru a monitoriza și evalua performanța bazelor de date; monitorizează call-urile ce vin de la bazele de date; urmărește performanțele specifice unui nou features sau a unui modul Java).
* **Manage Engine Applications Manager** (este un tool folosit de dezvoltator pentru a monitoriza aplicațiile Java și a performanței infrastructurii. Acesta este un program multifunctional ce se integrează perfect cu diverse platforme ca: slack, opmanager. Suporta mai multe platforme ca Windows, Mac Os si Linux.)
* **Junit** (este un instrument open source de testare unitară pentru limbajul de programare Java. Este un instrument de dezvoltare Java bun deoarece, se pot creea teste pe parcursul dezvoltării, oferind suport pentru pluginuri ca Gradle; astfel testarea este mai eficienta și minutioasa. Junit permite scrierea rapidă a codului, sporind calitatea. Este compatibil cu mai multe limbaje de programare ca Java, Python etc. Este o aplicație ușor de utilizat dar nu poate fi folosită la nivelele superioare de testare).
* **NetBeans** ( este un instrument open source gratuit de dezvoltare Java care permite dezvoltarea aplicațiilor desktop, mobile si web. NetBeans IDE este excelent pentru dezvoltatorii C/C++, PHP, Java, JavaScript și HTML. Acest instrument oferă suport lingvistic și se integrează perfect cu bazele de date și instrumente de gestionare a codului sursă. Avantajele acestuia includ suport pentru pluginuri cum ar fii NetBeans Editor CSV si JVI pentru centrul de actualizare NB. Deasemenea acesta va permite sa construiti un cod, sa validati probleme si sa le raportati).