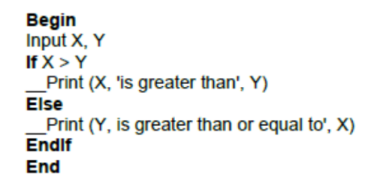
Pentru fiecare sarcină, trebuie să oferi o soluție sub forma unei scheme logice care afișează pseudocodul fiecărei sarcini. (vezi lecția 10 :)) Sarcinile fără soluție nu vor fi luate în considerare.

*Rezolvarea problemelor se poate face pe foi de hârtie și fotografiate, sau în orice aplicație de creare diagrame (de exemplu, Miro, diagrams.net) și realizate capturi de ecran. Apoi trebuie să plasezi fotografiile sau capturile de ecran într-un document text.*

### Sarcina 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

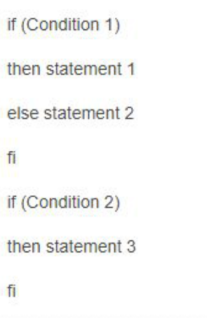
C. Statement coverage = 1, Decision coverage = 2

D. Statement coverage = 2, Decision coverage = 1

### 

### 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

### 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.

**Link-uri utile:**  
<https://www.tutorialspoint.com/software_testing_dictionary/decision_coverage_testing.htm#:~:text=Decision%20coverage%20or%20Branch%20coverage,each%20way%2C%20true%20and%20false>.   
  
<https://www.zyxware.com/articles/4161/what-is-statement-coverage-in-testing>   
  
<https://tutorialshut.com/white-box-techniques/path-coverage/#:~:text=Path%20Coverage%20testing%20is%20a,coverage%20and%20100%25%20branch%20coverage>.