**Sarcina 1**

Afișează pseudocodul de mai jos în formă de schemă logică.

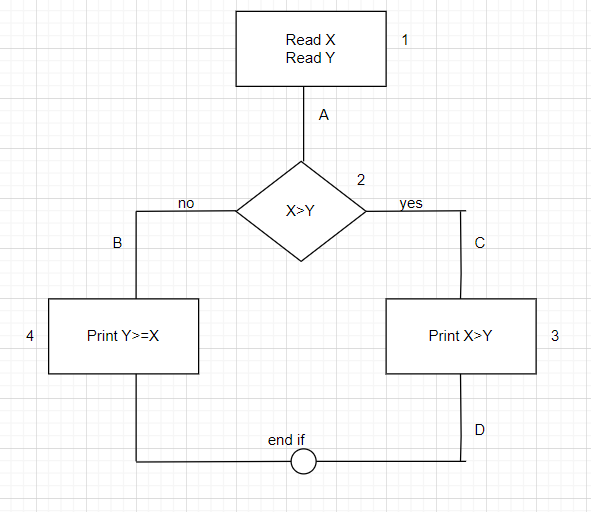
Care este numărul minim de cazuri de testare necesare pentru a garanta o acoperire de 100% statement și 100% decision?

A. Statement coverage = 3, Decision coverage = 3

**B. Statement coverage = 2, Decision coverage = 2 (1-2-3 și 1-2-4 / 2-3 și 2-4)**

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.

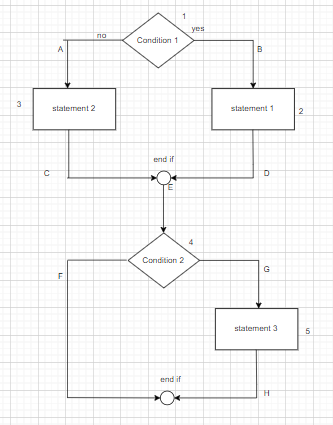
Care este numărul minim de cazuri de testare necesare pentru a garanta o acoperire de 100% path?

A. 1

B. 2

C. 3

**D. Niciun răspuns nu este corect** **(1-2-4-5 și 1-3-4-5 / 1-2-4 și 1-3-4)**

****

### **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.

**Testele statice (white box) ajută la identificarea problemelor cum ar fi erori în cod, defecte de design și contradicții în ceea ce privește cerințele. Întrucât se efectuează la nivel de unit sau integration vor fi efectuate cel mai probabil de programatori.**

**Printre tool-urile folosite ar fi Selenium, Appium, Apache JMeter, Postman, Cucumber, etc.**