

### Must have рівень:

1. Зроби порівняння статичних та динамічних технік тестування. Наведи переваги та можливі обмеження при використанні кожної з них.

	Статистична техніка тестування	Динамічна техніка тестування
Основна інформація		
Перевага №1	Reduces the cost of fixing identified bugs by detecting them at early stages of the software development lifecycle	This is a thorough examination that reviews the entire functionality of the application, ensuring that quality meets the highest standards.
Перевага №2	Feedback obtained during this testing helps improve process functioning, which also assists the team in avoiding similar defects and bugs	The dynamic testing process is well-established, with the application tested from the user's perspective, which enhances software quality.
Перевага №3 (і т.д.)	Increases awareness of various software quality issues.	It helps detect complex errors that may have slipped through during code review.
Обмеження №1	The static testing process can be time-consuming, as it is primarily performed manually.	High cost of testing.
Обмеження №2	It hinders the detection of vulnerabilities that manifest in the runtime environment.	Dynamic testing is usually performed after coding is complete, meaning that bugs are detected later in the software development lifecycle.
Обмеження №3 (і т.д.)	Testers must have a strong background in programming and the specific technologies used in the application.	Since dynamic testing is a complex process, it can be time-consuming.

Висновок	Although static testing requires significant time for lively discussions and meetings, it is still worth investing that time to prevent defects from appearing at the later stages of product development. Therefore, static testing is rightfully considered an important step toward developing error-free software.	The importance of dynamic testing is also immense. By directly executing software tests (verifying functional behavior, performance, reliability, and other critical aspects), the team can assess and confirm the quality and effectiveness of the software.
----------	--	---

### Середній рівень:

1. Виконай завдання попереднього рівня.
2. Наступне твердження стосується покриття рішень:  
*Коли код має одну 'IF' умову, не має циклів (LOOP) або перемикачів (CASE), будь-який тест, який ми виконаємо, дасть результат 50% покриття рішень (decision coverage).*

Яке твердження є коректним?

- a. Коректно. Будь-який тест кейс надає 100% покриття тверджень, таким чином покриває 50% рішень.
  - b. Коректно. Результат будь-якого тесту умови IF буде або правдивим, або ні.
  - c. Некоректно. Один тест може гарантувати 25% перевірки рішень в цьому випадку.
  - d. Некоректно, бо занадто загальне твердження. Ми не можемо знати, чи є воно коректним, бо це залежить від тестованого ПЗ.
3. Є псевдокод: Switch PC on -> Start MS Word -> IF MS Word starts THEN -> Write a poem -> Close MS Word.

Скільки тест кейсів знадобиться, щоб перевірити його функціонал?

- a. 1 – для покриття операторів, 2 – для покриття рішень

- b. 1 – для покриття операторів, 1 – для покриття рішень
- c. 2 – для покриття операторів, 2 – для покриття рішень
- d. 2 – для покриття операторів, 1 – для покриття рішень

4. Скільки потрібно тестів для перевірки тверджень коду:

```
Read P
Read Q
IF P+Q > 100 THEN
Print "Large"
ENDIF
If P > 50 THEN
Print "P Large"
ENDIF
```

- a. 2
- b. 1
- c. 3
- d. 4