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 | Вид тестування ПЗ без запуску коду. Виявлення помилок да дефектів в документах(специфікації вимог, самому коді). | Виконується шляхом фактичного використання програми (запуск коду) і визначення того, чи працює функціональність так, як очікується. |
| Перевага №1 | Виявляє баги на початкових етапах, тим самим знижує вартість їх виправлення. | Додаток тестується з точки зору користувача, що підвищує якість ПЗ. |
| Перевага №2 | Відгуки отримані в процесі тестування допомагають покращити функціонування процесу . | Виявлення складних помилок, які могли вислизнути на етапі код ревью (статичного тестування). |
| Перевага №3 (і т.д.) | Підвищує інформованість про різні проблеми якості програмного забезпечення | Може бути автоматизовано за допомогою спеціальних інструментів. |
| Обмеження №1 | Займає багато часу. | Складний процес, який займає багато часу. |
| Обмеження №2 | Здебільшого виконується вручну. | Висока вартість проведення тестування. |
| Обмеження №3 (і т.д.) | Перешкоджає виявленню вразливостей, представлених в середовищі виконання. | Зазвичай виконується після завершення кодування, і знайдені баги виявляються пізніше в життєвому циклі розробки. |
| Висновок | Комплексне тестування коду, яке допомагає знайти більше дефектів у системі. | Завдяки безпосередньому виконанню тестів ПЗ (перевірки функціональної поведінки, продуктивності, надійності та ін.) можемо перевірити і підтвердити якість і ефективність ПЗ. |

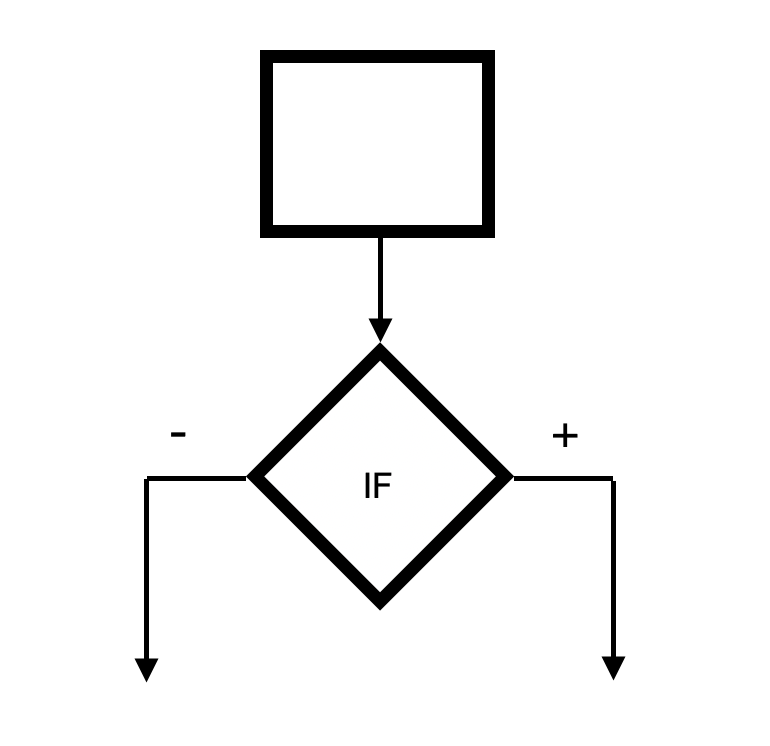
Середній рівень:

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?

* 1. **Коректно. Будь-який тест кейс надає 100% покриття тверджень, таким чином покриває 50% рішень.**
  2. Коректно. Результат будь-якого тесту умови IF буде або правдими, або ні.
  3. Некоректно. Один тест може гарантувати 25% перевірки рішень в цьому випадку.
  4. Некоректно, бо занадто загальне твердження. Ми не можемо знати, чи є воно коректним, бо це залежить від тестованого ПЗ.

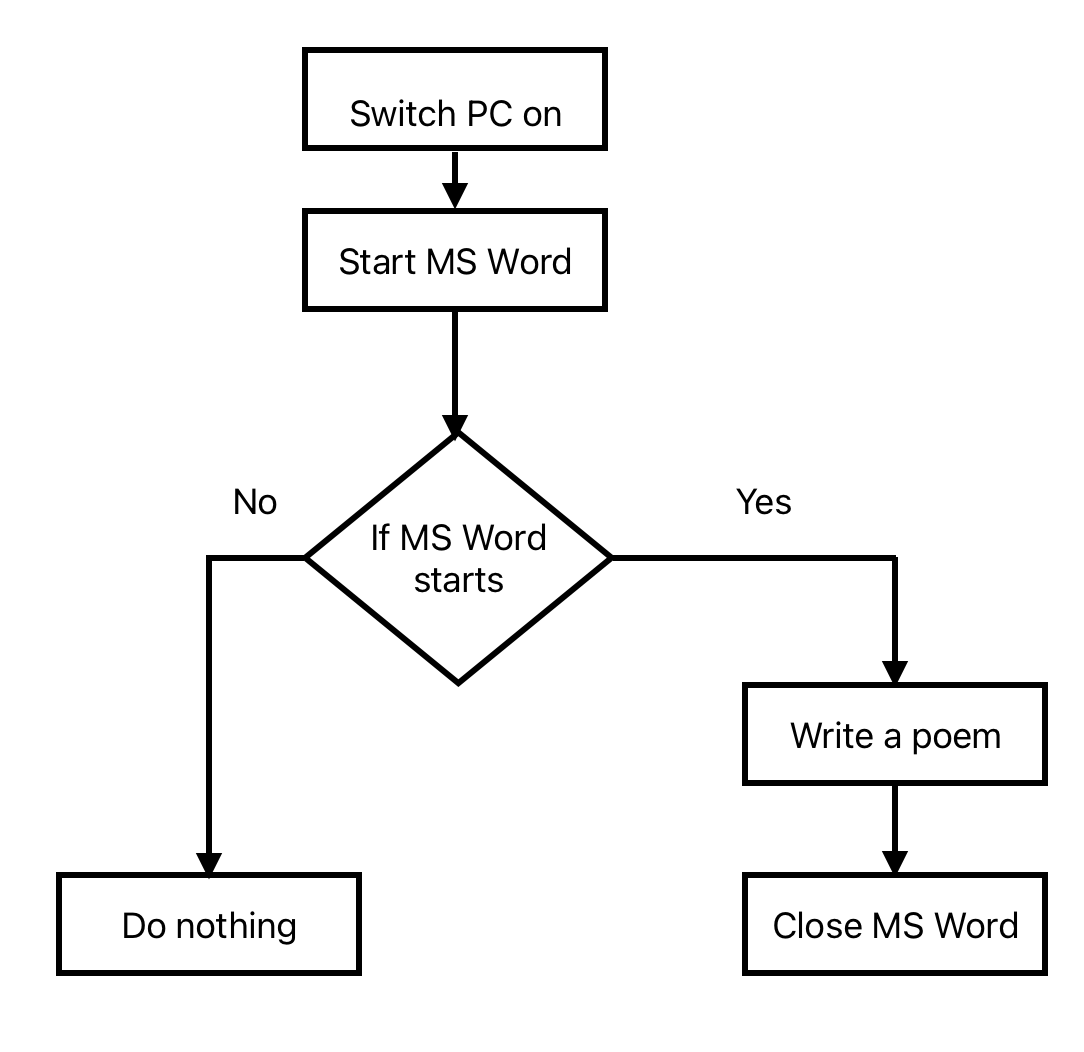


Твердження А коректне, тому що ми ми маємо тільки одну умову з варіантами Так чи Ні, тобто маємо кінцевий результат 100%, 2 кейси, кожний з яких покриває 50%.

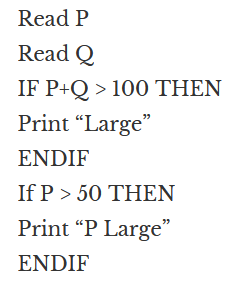
1. Є псевдокод: Switch PC on -> Start MS Word -> IF MS Word starts THEN -> Write a poem -> Close MS Word.

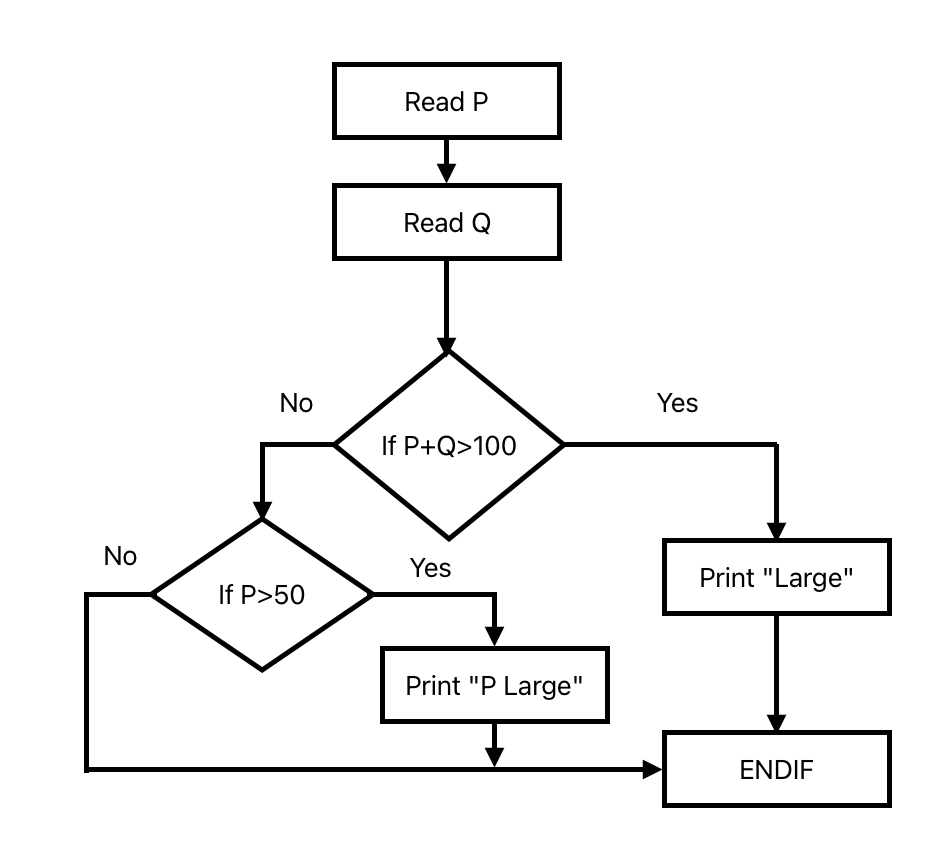
Скільки тест кейсів знадобиться, щоб перевірити його функціонал?

* 1. 1 – для покриття операторів, 2 – для покриття рішень
  2. 1 – для покриття операторів, 1 – для покриття рішень
  3. 2 – для покриття операторів, 2 – для покриття рішень
  4. 2 – для покриття операторів, 1 – для покриття рішень



Я думаю що варіант А, тому що ми маємо тільки один оператор IF, та маємо 2 варіанти рішень - Так чи Ні, хоча про Ні немає чітких умов.

1. Скільки потрібно тестів для перевірки тверджень коду: 
   1. 2
   2. 1
   3. **3**
   4. 4



Я думаю що варіант с.3, тому що ми маємо 2 умови, що приводить до варіантів:

1. IF P+Q>100 YES
2. IF P+Q>100 NO AND P>50 YES
3. IF P+Q>100 NO AND P>50 NO

Тобто маємо 3 тест-кейси.