**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 | Тестування без запуску коду | Тестування із запуском коду |
| Перевага №1 | Раннє тестування | Додаток тестується з точки зору користувача |
| Перевага №2 | Скорочення витрат і часу на розробку та тестування | Виявлення помилок, які було пропущені |
| Перевага №3 (і т.д.) | Попередження дефектів дизайну або кодування | Може включати автоматизацію |
| Обмеження №1 | Займає багато часу, виконується вручну | Висока вартість |
| Обмеження №2 | Не можливо виявити дефекти в середовищі виконання | Баги виявляються пізніше |
| Обмеження №3 (і т.д.) |  | Займає багато часу |
| Висновок | Мета – запобігання дефектів програмного забезпечення | Мета – пошук і усунення багів |

**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 правдими, 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?

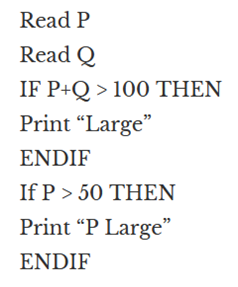
a. 1 – для покриття операторів, 2 – для покриття рішень

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:



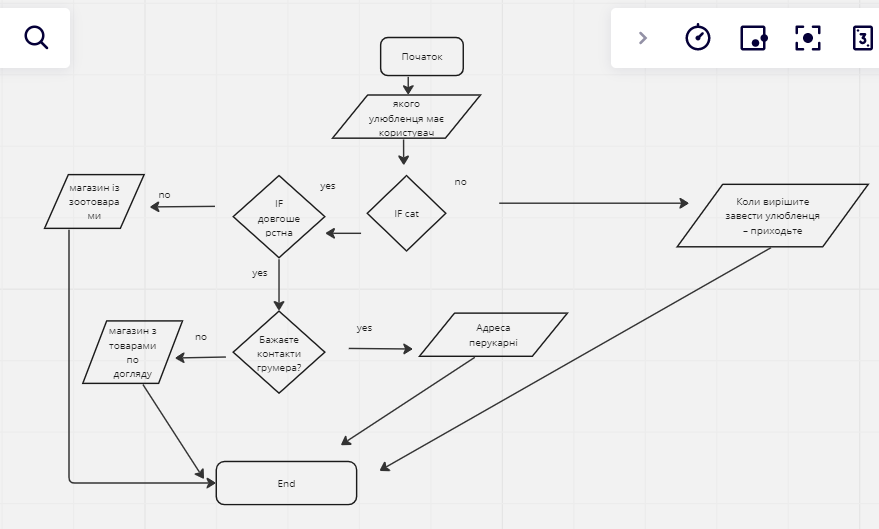
a. 2

b. 1

**c. 3**

d. 4

**Програма максимум:**

****

Нам потрiбно буде 4 тести, щоб переконатися, що всі запитання були поставлені, всі комбінації були пройдені та всі відповіді були отримані.

1. Виконай завдання двох попередніх рівнів.

2. Продовжуємо розвивати стартап для застосунку, який дозволяє обмінюватися фотографіями котиків.

Є алгоритм:

Запитай, якого улюбленця має користувач.

Якщо користувач відповість, що має кота, то запитай, яка порода його улюбленця: «короткошерста чи довгошерста?»

Якщо клієнт відповість «довгошерста», то запитай: «ви бажаєте отримати контакти найближчого грумера?»

Якщо клієнт відповість «так», то скажи: «Надайте адресу найближчої котячої перукарні»

Інакше

Скажи: «Запропонуй магазин з товарами по догляду за шерстю»

Закінчити

Інакше

Скажи «Запропонуй обрати магазин із зоотоварами»

Закінчити

Якщо клієнт не має кота

Скажи “Коли вирішите завести улюбленця – приходьте”

Закінчити

Завдання:

1. Намалюй схему алгоритму (в інструменті на вибір, наприклад, у вбудованому Google Docs редакторі, figjam чи miro)

2. Який потрібен мінімальний набір тест-кейсів, щоб переконатися, що всі запитання були поставлені, всі комбінації були пройдені та всі відповіді були отримані?