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 | Це методика тестування основною рисою якої є те що вона проводиться без запуску коду. | Це метод тестування який перевіряє наявність помилок коли код виконується |
| Перевага №1 | Баги знаходяться дуже на ранній стадії отже їх виправлення дешевше | Це дуже якісні і ретельні перевірки тому зазвичай виявляють максимум помилок |
| Перевага №2 | Це тестування допомагає покращити процес розробки | Тестування проходить з точки зору користувача |
| Перевага №3 | Тестування проводиться з різних точок зору (як автор, як замовник, як користувач)) | Може бути автоматизованим |
| Обмеження №1 | Частіше за все займає багато часу тому що проводиться вручную | Займає багато часу через сам складний процес тестування |
| Обмеження №2 | Неможливо знайти помилки які з'являться в ході взаємодії коду з зовнішніми ресурсами | Висока вартість проведення тестування |
| Обмеження №3 |  | Тестування проводиться після повного написання коду |
| Висновок | Одне тестування без іншого не дасть однозначно гарного результату так як ці два види тестують різні сторони програми. І вони як незалежні так і доповнюють одне одного. | |

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).*

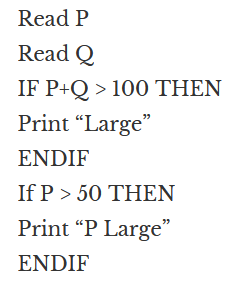
Яке твердження є коректним?

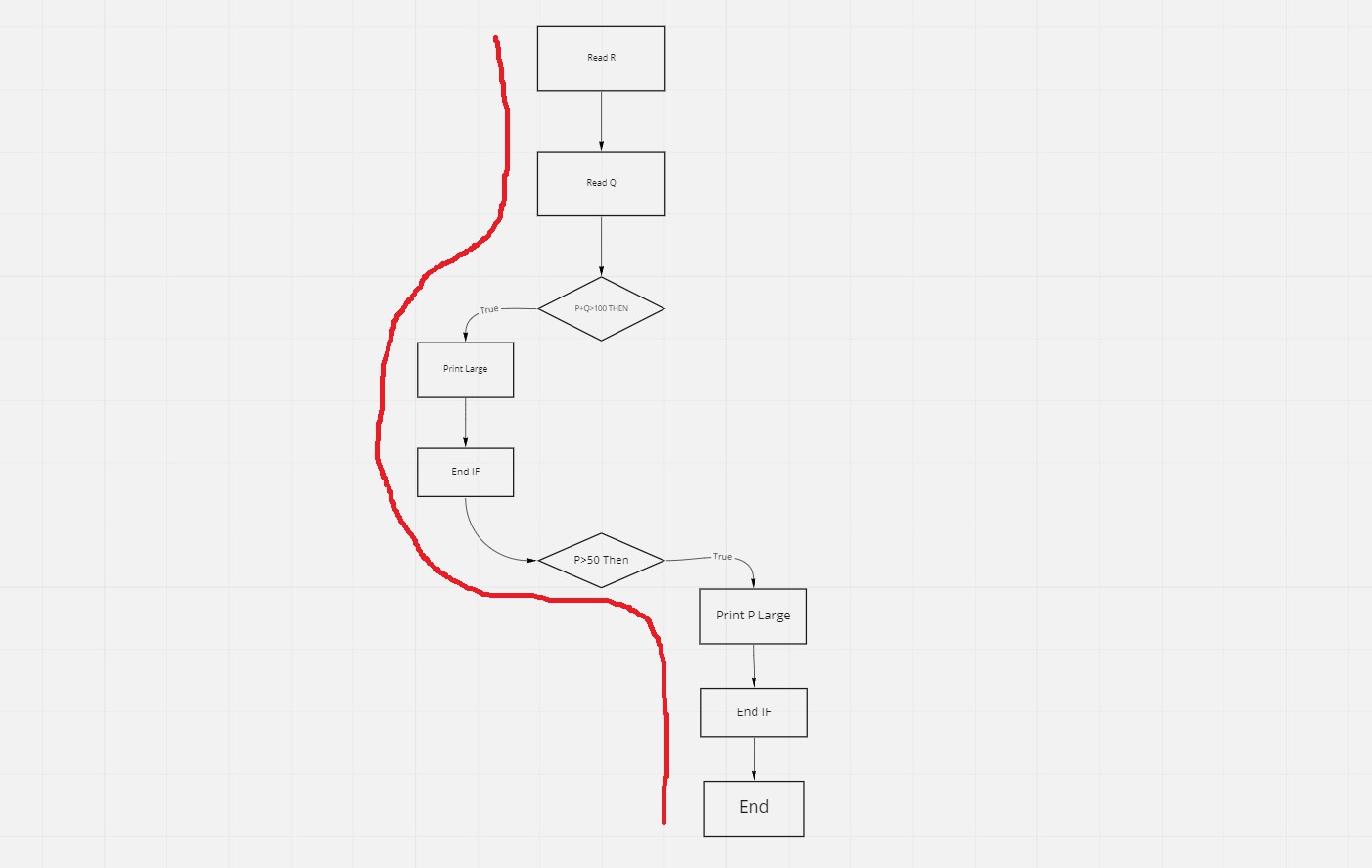
* 1. Коректно. Будь-який тест кейс надає 100% покриття тверджень, таким чином покриває 50% рішень.
  2. Коректно. Результат будь-якого тесту умови IF буде або правдими, або ні.
  3. Некоректно. Один тест може гарантувати 25% перевірки рішень в цьому випадку.
  4. Некоректно, бо занадто загальне твердження.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

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

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



Програма максимум:

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*

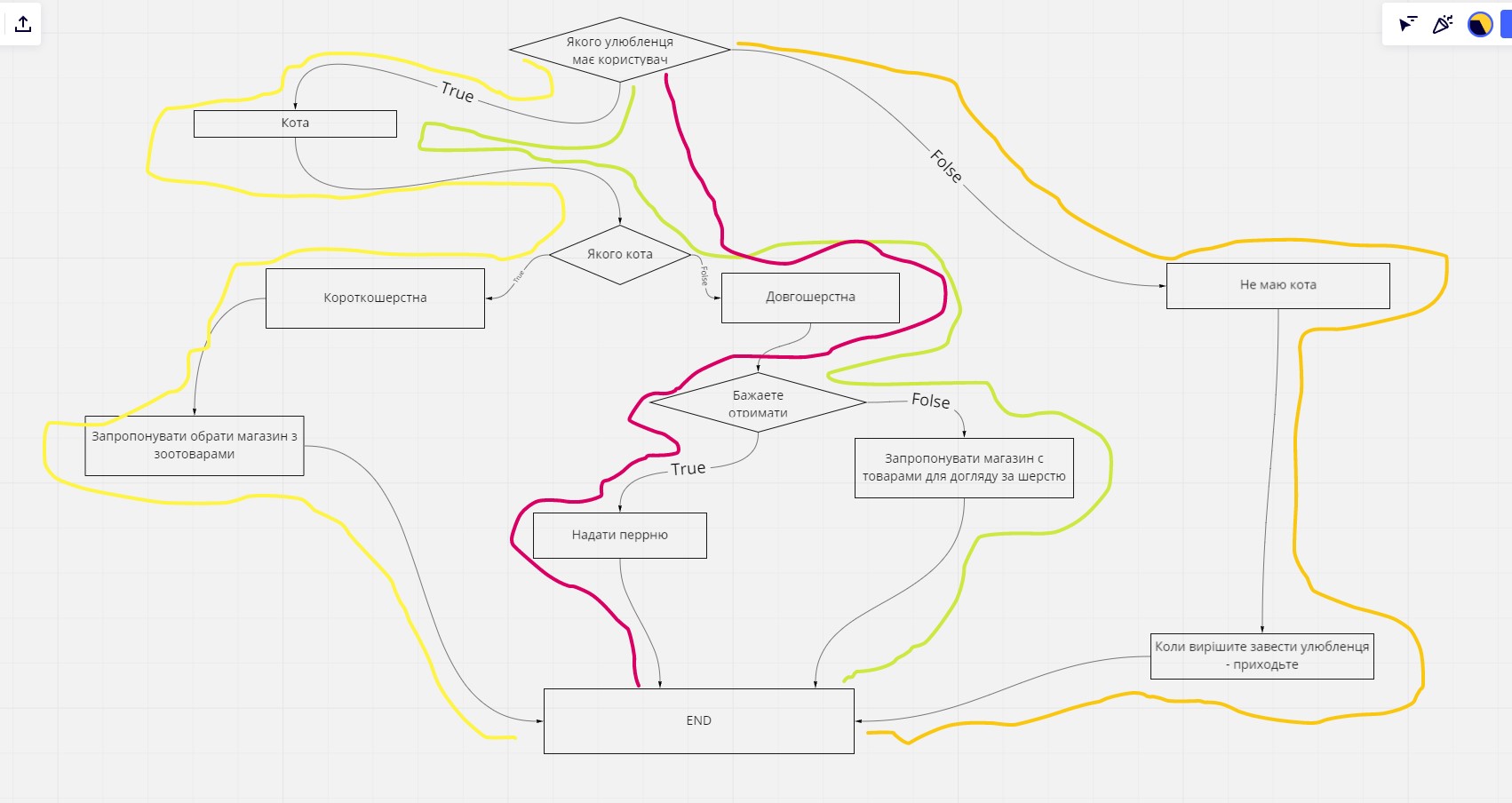
*Якщо клієнт не має кота*

*Скажи “Коли вирішите завести улюбленця – приходьте”*

*Закінчити*

Завдання:

1. Намалюй схему алгоритму (в інструменті на вибір, наприклад, у вбудованому Google Docs редакторі, [figjam](https://www.figma.com/figjam/) чи [miro](https://miro.com/))\



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

* 5 тест-кейсві.