

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	Не потребує запуску програми чи додатку. Дає змогу знайти помилки на ранній стадії. Включає рев'ю та статистичний аналіз	Потребує запуску програми/додатку. Включає перевірку продуктивності (завантаження процесора, використання пам'яті, час відповіді, etc)
Перевага №1	Менше часу та витрат	Можливість автоматизації
Перевага №2	Більш раннє виявлення дефектів	Добре комбінується з регресією
Перевага №3 (і т.д.)	Більш шансів попередити дефект/проблему	Дозволяє провалідувати інтеграції
Обмеження №1	Рідкісні події можуть бути упущені	Потребує більше ресурсів
Обмеження №2	Причина проблеми не буде так очевидна як у випадку з кодом	Не покриває всі можливі стани
Обмеження №3 (і т.д.)	Для комплексних алгоритмів може бути не дуже ефективною (наприклад якщо є заморочений security механізм)	Можуть бути ускладнення з інтеграціями, бо саме до хз коду ми доступу (зазвичай не маємо)
Висновок	Добре підходить для виявлення широкого спектру дефектів,	Використовується для перевірки вже імплементованого коду.

	вельючаючи за межами тест кейсів.	Покриває саме випадки описані у тест кейсах
--	-----------------------------------	---

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 правдимим, 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

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 kota, 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 kota

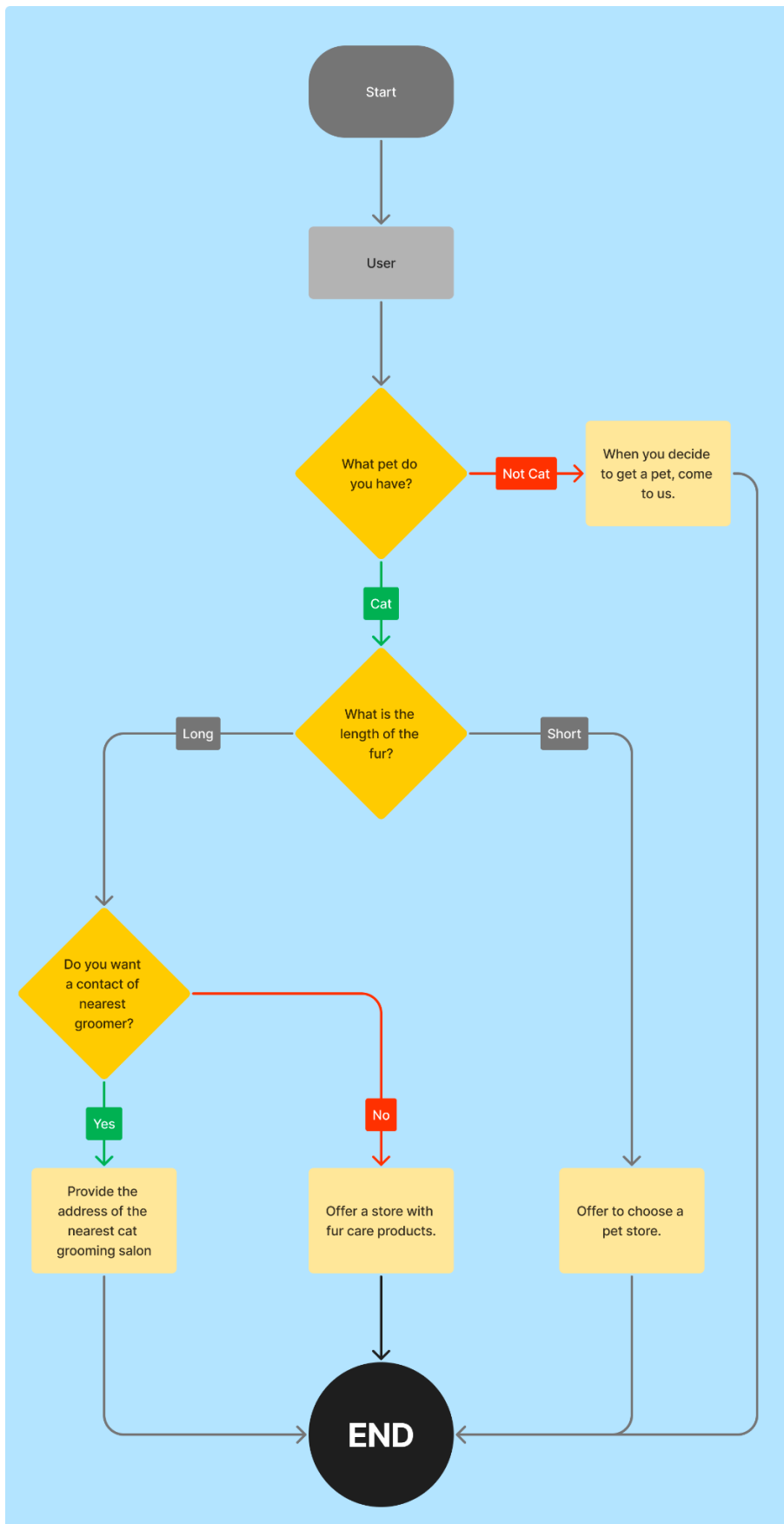
Скажи «Коли вирішите завести улюбленця – приходьте»

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?



- Мінімальний набір тест-кейсів, щоб переконатися, що всі запитання були

поставлені - 1

- Мінімальний набір тест-кейсів, щоб переконатися, що всі комбінації були пройдені та всі відповіді були отримані? - 4