**\*Задание 1. \*Представьте, что вы работаете над разработкой простого приложения для записной книжки, которое позволяет пользователям добавлять, редактировать и удалять контакты.  
Ваша задача - придумать как можно больше различных тестов (юнит-тесты, интеграционные тесты, сквозные тесты) для этого приложения. Напишите название каждого теста, его тип и краткое описание того, что этот тест проверяет.**

**Ответ:** Приложение дожно быть разбито на 3 составляющие: пользовательский интерфейс, сервис, осуществляющий CRUD операции с данными, хранилище (репозиторий контактов). Согласно рекомендации Google 70/20/10, в которой говорится, что модульные тесты должны составлять основную часть набора тестов (70%), за которыми следуют интеграционные тесты (20%) и сквозные тесты (10%). Для правильного функционирования этих частей нужно написать соответствующие юнит тесты. Так для интерфеса проверить корректность отображения картинок, шрифтов в засисимости от ориентации и разрешения экрана. Соответствие визуала техническому заданию. Для части сервиса юнит тестами могут стать корректно введенные данные от пользователя(соответсвие номера маске телефона), проверка на дубликаты уже записанных имен или телефонов, проверка операции удаления и корректность обработки ошибок. Эквивалентность возвращаемого контакта запросу(проверка на null). Внутри хранилища   
  
**\*Задание 2. \*Ниже список тестовых сценариев. Ваша задача - определить тип каждого теста (юнит-тест, интеграционный тест, сквозной тест) и объяснить, почему вы так решили.**

2.1 Проверка того, что функция addContact корректно добавляет новый контакт в список контактов"".

**Ответ 2.1:** Т.к рассматривается корректность работы одной функции(модуля) то это юнит-тест. В случае нахождения списка контактов в другом классе(модуле) речь может идти об интеграционном тестировании.

2.2 Проверка того, что при добавлении контакта через пользовательский интерфейс, контакт корректно отображается в списке контактов.

Ответ 2.2: Т.к пользовательский интерфейс представляется обычно отдельным модулем, а список контактов отдельным хранилищем. То в этом случае речь идет об интеграционном тестировании.

2.3 Проверка полного цикла работы с контактом: создание контакта, его редактирование и последующее удаление.

Ответ 2.2: Полный цикл работы с контактами чаще представляется в виде пользовательского интерфейса как отдельного модуля представления данных. Модуля для организации логики операций CRUD над данными(некий сервис). А также самими данными, собранными в хранилище(удаленный репозиторий, БД). В таком случае проверяется работа каждого модуля и взаимная работа всех модулей. Насколько точно происходит отработка методов в каждом модуле, актуальность и правдивость данных полученнных призапросе сервиса из БД и отображения этих данных в пользовательском интерфейсе. Т.о мы получаем процесс полного цикла приложения от начала и до конца. Это end2end или сквозное тестирование