**Задача 1.**

Для приложения записной книжки можно создать множество различных тестов разных типов. Вот несколько примеров различных тестов для такого приложения:

1. **Юнит-тесты:**
   * *Тест добавления контакта:* Этот тест проверяет, что функция добавления контакта корректно добавляет контакт в записную книжку.
   * *Тест редактирования контакта:* Проверяет, что функция редактирования контакта обновляет информацию о контакте верно.
   * *Тест удаления контакта:* Убеждается, что функция удаления контакта удаляет контакт из записной книжки и не оставляет лишних записей.
2. **Интеграционные тесты:**
   * *Тест взаимодействия с базой данных:* Проверяет, что приложение правильно взаимодействует с базой данных для сохранения и получения контактов.
   * *Тест взаимодействия с пользовательским интерфейсом:* Проверяет, что пользовательский интерфейс отображает и обрабатывает данные корректно.
3. **Сквозные тесты:**
   * *Тест создания контакта и его поиска:* Создает контакт в записной книжке, а затем выполняет поиск по имени или другим атрибутам, чтобы убедиться, что контакт найден.
   * *Тест удаления контакта и его отсутствия после удаления:* Удаляет контакт и затем проверяет, что контакт больше не существует в записной книжке.
4. **Тесты на обработку ошибок:**
   * *Тест на обработку пустых полей при добавлении контакта:* Проверяет, как приложение обрабатывает попытку добавления контакта с недостающей информацией.
   * *Тест на обработку некорректных данных:* Проверяет, как приложение обрабатывает попытку добавления некорректных данных в записную книжку.
5. **Тесты на производительность:**
   * *Тест на скорость добавления большого количества контактов:* Измеряет, как быстро приложение может обработать добавление большого числа контактов.
   * *Тест на скорость поиска контакта:* Измеряет, как быстро приложение может найти контакт в записной книжке.
6. **Тесты на безопасность:**
   * *Тест на безопасность хранения данных*: Проверяет, что данные контактов хранятся в зашифрованной форме и защищены от несанкционированного доступа.

**Задача 2.**

Давайте определим тип каждого из предложенных тестовых сценариев:

1. **"Проверка того, что функция addContact корректно добавляет новый контакт в список контактов":**
   * *Тип теста:* Юнит-тест.
   * *Обоснование:* Этот сценарий проверяет корректность работы конкретной функции addContact, и поэтому является юнит-тестом, который охватывает отдельный компонент (функцию) приложения.
2. **"Проверка того, что при добавлении контакта через пользовательский интерфейс, контакт корректно отображается в списке контактов":**
   * *Тип теста:* Интеграционный тест.
   * *Обоснование:* Этот сценарий проверяет взаимодействие между пользовательским интерфейсом и бэкэндом приложения, а также правильность отображения контактов в пользовательском интерфейсе. Таким образом, он охватывает несколько компонентов приложения и является интеграционным тестом.
3. **"Проверка полного цикла работы с контактом: создание контакта, его редактирование и последующее удаление":**
   * *Тип теста:* Сквозной тест (или энд-ту-энд тест).
   * *Обоснование:* Этот сценарий охватывает весь жизненный цикл контакта, начиная с его создания через пользовательский интерфейс, затем редактирования и, наконец, удаления. Такие тесты проверяют, как взаимодействуют различные компоненты приложения в полном цикле, включая пользовательский интерфейс, бэкэнд и базу данных, и поэтому являются сквозными (или энд-ту-энд) тестами.

Таким образом, каждый из этих тестовых сценариев соответствует своему типу теста в зависимости от того, какие компоненты приложения и взаимодействие между ними они проверяют.