**Регрессионное тестирование.**

**Регрессио́нное тести́рование** — собирательное название для всех видов [тестирования программного обеспечения](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F), направленных на обнаружение ошибок в уже протестированных участках [исходного кода](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D1%85%D0%BE%D0%B4%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4). Такие ошибки — когда после внесения изменений в программу, перестаёт работать то, что должно было продолжать работать, — называют регрессионными ошибками.

Регрессионное тестирование включает new bug-fix — проверка исправления вновь найденного дефекта, old bug-fix — проверка, что исправленный ранее и верифицированный дефект не воспроизводится в системе снова, а также side-effect — проверка того, что не нарушилась работоспособность работающей ранее функциональности, если её код мог быть затронут при исправлении некоторых дефектов в другой функциональности. Обычно используемые методы регрессионного тестирования включают повторные прогоны предыдущих тестов, а также проверки, не попали ли регрессионные ошибки в очередную версию в результате слияния кода.

Беря за основу полученное задание, нам необходимо проверить, что авторизация в системе происходит успешно. Прогнав тест-кейсы, мы видим, что авторизация в системе возможна при использовании вариаций:

1. Авторизация по связке почта + пароль.
2. Авторизация по связке логин + пароль.
3. Авторизация по связке лицевой счет + пароль.

**Итоги.**

Проанализировав результаты тест-кейсов, можно сказать, что продукт еще не готов для выхода в релиз. Прогнав 17 тест-кейсов выявлено 13 багов. Само техническое задание требует исправлений и доработки, в нем выявлено много недочетов, которые не позволяют прогнать корректно все тест-кейсы.