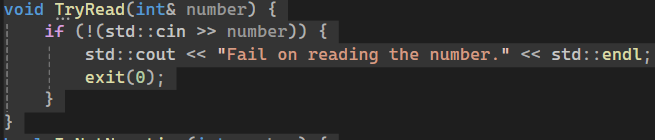
**Лабораторная работа № 2. Мальцевич А. Е. 9 Группа.**

1. **Анализ программы:**

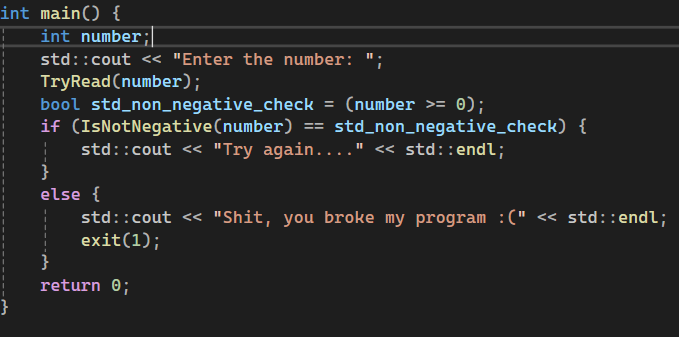
Мы имеем программу, которая содержит следующие функции:

Функция, проверяющая ввод пользователя через “std::cin >> number” – попытка чтения целого числа. (Если ввод корректен – то программа выполняется, иначе – завершается через exit(0)).

 Следующая ф-ия:

1. Вычисляет модуль числа
2. 2)Сравнивает модуль с исходным числом
3. Возвращает True(1), если числа равны при n >= 0, иначе – False(0)

Далее – ф-ия сравнения результатов в теле main:



Сравнение результатов:

Совпадают – “std::cout << "Try again...." << std::endl;” (1),

Иначе – “std::cout << "Shit, you broke my program :(" << std::endl;

exit(1);” (2)

Нам нужен вариант (2), проанализируем, где все может пойти не так.

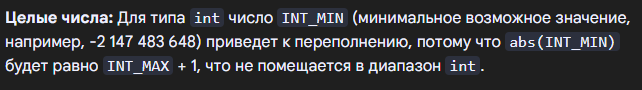
1. **Поиск уязвимости:**

Из-за ф-и проверки ввода, мы не можем ввести ничего кроме целого числа. А значит нужно искать уязвимости в целых числах.

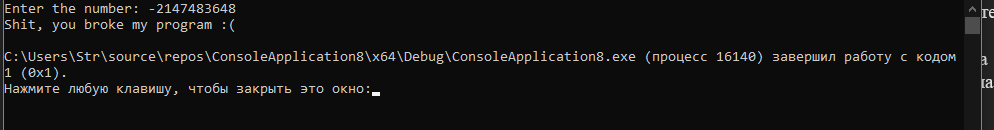
Поймем, что вводить любое сколь угодно большое число бесполезно, значит нужно искать уязвимости при вводе отрицательных чисел.

Базовая проверка при вводе любого отрицательного числа («влезающего в int») вернет нам false. Значит нужно смотреть на другую ф-ию.

Нам нужно, чтобы было (IsNotNegative(number) == std\_non\_negative\_check) (0 == 1, 1 == 0). std\_non\_negative\_check – всегда 1 при number <= 0. Значит смотрим на ф-ю с модулем (abs). Если существует какое-либо число, которое “ломает модуль” , то оно нам и подойдет. Загуглим, посмотрим на ответ от нейронки)



Попробуем ввести это число:



Действительно, данное число – “-2147483648” ломает нашу программу. Так как вызывает переполнение, как и написано выше ответом от Gemini.

Спасибо, что прочитали это, если я написал бред – простите, я сижу на больничном с большой температурой. 😔 😔 😔.