**Лабораторная работа 1 «Разработка тестового сценария проекта»**

Цель: получить навыки разработки тестовых сценариев.

**Задание No 1**

Написать программу решения квадратного уравнения ах2 + bх + с = 0.

import math

a = int(input('a = '))

b = int(input('b = '))

c = int(input('c = '))

d = (b\*b) - (4 \* a \* c)

sd = math.sqrt(d)

x1 = (0 - b - sd) / (2\*a)

x2 = (0 - b + sd) / (2\*a)

print(*f*'x1 = {x1}, x2 = {x2}')

**Задание No 2**

Найти минимальный набор тестов для программы нахождения вещественных корней

квадратного уравнения ах2 + bх + с = 0. Решение представлено в таблице.



Таким образом, для этой программы предлагается минимальный набор функциональных

тестов, исходя из 7 классов выходных данных.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Номер теста** | **a** | **b** | **c** | **Ожидаемый результат** | **Что проверятеся** | **Результат** |
| **1** | 2 | -5 | 2 | x1 = 0.5, x2 = 2.0 | Случай вещественных корней | x1 = 0.5, x2 = 2.0 |
| **2** | 3 | 2 | 5 | Сообщение | Случай комплексных  корней | ValueError: math domain error |
| **3** | 3 | -12 | 0 | x1 = 0.0, x2 = 4.0 | Нулевой корень | x1 = 0.0, x2 = 4.0 |
| **4** | 0 | 0 | 10 | Сообщение | Неразрешимое уравнение | ZeroDivisionError: float division by zero |
| **5** | 0 | 0 | 0 | Сообщение | Неразрешимое уравнение | ZeroDivisionError: float division by zero |
| **6** | 0 | 5 | 17 | Сообщение | Неразрешимое уравнение | ZeroDivisionError: float division by zero |
| **7** | 9 | 0 | 0 | x1 = 0.0, x2 = 0.0 | Нулевые корни | x1 = 0.0, x2 = 0.0 |

**Заповеди по отладки программного средства, предложенные Г. Майерсом.**

Заповедь 1. Считайте тестирование ключевой задачей разработки ПС, поручайте его самым

квалифицированным и одаренным программистам, нежелательно тестировать свою собственную

программу.

Заповедь 2. Хорош тот тест, для которого высока вероятность обнаружить ошибку, а не

тот, который демонстрирует правильную работу программы.

Заповедь 3. Готовьте тесты как для правильных, так и для неправильных данных.

Заповедь 4. Документируйте пропуск тестов через компьютер, детально изучайте

результаты каждого теста, избегайте тестов, пропуск которых нельзя повторить.

Заповедь 5. Каждый модуль подключайте к программе только один раз, никогда не изменяйте программу, чтобы облегчить ее тестирование.

Заповедь 6. Пропускайте заново все тесты, связанные с проверкой работы какой-либо

программы ПС или ее взаимодействия с другими программами, если в нее были внесены

изменения (например, в результате устранения ошибки).

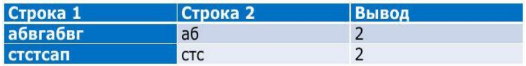
**Задание No 3**

Разработайте набор тестовых сценариев (как позитивных, так и негативных) для

следующей программы:

Имеется консольное приложение (разработайте самостоятельно). Ему на вход подается 2

строки. На выходе приложение выдает число вхождений второй строки в первую. Например:



Набор тестовых сценариев запишите в виде таблицы, приведенной выше.

1. Позитивный тест:

Входные данные:

Первая строка: "hello world"

Вторая строка: "llo"

Ожидаемый вывод: 1

2. Позитивный тест:

Входные данные:

Первая строка: "abababab"

Вторая строка: "ab"

Ожидаемый вывод: 4

3. Позитивный тест:

Входные данные:

Первая строка: "aaaaaa"

Вторая строка: "a"

Ожидаемый вывод: 6

4. Негативный тест:

Входные данные:

Первая строка: ""

Вторая строка: "abc"

Ожидаемый вывод: 0

5. Негативный тест:

Входные данные:

Первая строка: "abc"

Вторая строка: ""

Ожидаемый вывод: 0

6. Негативный тест:

Входные данные:

Первая строка: "abc"

Вторая строка: "def"

Ожидаемый вывод: 0

frist\_stroke = input("Введите первую строку: ")

second\_stroke = input("Введите вторую строку: ")

bank = frist\_stroke.count(second\_stroke)

print(bank)

**Задание No 4**

Оформить отчет.