1. Поиск ошибок в граничных условиях

Определите входные параметры:

Числовые: мин/макс значения (напр., возраст: 0-120).

Текстовые: пустые строки, спецсимволы (/, @).

Коллекции: пустой список, 1 элемент, макс. размер.

Тестирование:

Тест 1: Ввод минимального значения (0) → ожидаемый результат.

Тест 2: Ввод максимального значения (120) → ожидаемый результат.

Тест 3: Ввод за пределами (121, -1) → ошибка валидации.

Тест 4: Пустой ввод → сообщение "Поле обязательно".

Назначение программы:

Пример: «Расчет пенсии по возрасту: принимает возраст 18-120 лет, выводит размер выплат».

Таблица результатов:

| **№ теста** | **Входные данные** | **Ожидаемый результат** | **Фактический результат** | **Статус** |
| --- | --- | --- | --- | --- |
| 1 | 0 | Ошибка "Минимум 18" | Ошибка | ✅ |
| 2 | 120 | Расчет выполнен | Расчет выполнен | ✅ |
| 3 | -1 | Ошибка | Программа зависла | ❌ |

2. Составьте список рисков:

- Нехватка времени на тестирование.

- Сложная интеграция с API банка.

- Недостаточная квалификация команды в blockchain.

Оцените по параметрам:

Вероятность (1-5): насколько вероятен риск.

Ущерб (1-5): серьезность последствий.

Таблица рисков:

| **Риск** | **Вероятность** | **Ущерб** | **Приоритет (В×У)** | **Действия для снижения** |
| --- | --- | --- | --- | --- |
| Нехватка времени | 4 | 5 | 20 | Увеличить срок на 2 недели |
| Ошибки интеграции с API банка | 3 | 4 | 12 | Тестовый стенд API |
| Сложность в blockchain | 2 | 4 | 8 | Нанять эксперта |

3. Поиск и документирование дефектов

Тест-кейсы:

Шаг 1: Открыть программу.

Шаг 2: Ввести недопустимые данные (напр., буквы в поле "Возраст").

Шаг 3: Нажать "Рассчитать".

Фиксация дефектов:

Что произошло: «Программа закрылась с ошибкой ACCESS\_VIOLATION».

Ожидалось: «Сообщение "Введите число"».

Тачка дефектов:

| **D** | **Название** | **Шаги воспроизведения** | **Ожидаемый результат** | **Фактический результат** | **Серьезность** |
| --- | --- | --- | --- | --- | --- |
| DEF-01 | Крах при вводе текста | 1. Откройте программу. 2. Введите "abc" в поле "Возраст". 3. Нажмите "Рассчитать". | Сообщение об ошибке | Программа закрылась | Critical |

4. #include <stdio.h>  
int main() {  
char s[100], key[20];  
printf("Message: ");  
fgets(s, 100, stdin);  
printf("Key: ");  
fgets(key, 20, stdin);  
for(int i = 0, j = 0; s[i]; i++) {  
if(s[i] >= 'A' && s[i] <= 'Z') {  
s[i] = 'A' + (s[i] + (key[j] & 31) - 'A') % 26;  
j = key[j+1] ? j+1 : 0;  
}  
else if(s[i] >= 'a' && s[i] <= 'z') {  
s[i] = 'a' + (s[i] + (key[j] & 31) - 'a') % 26;  
j = key[j+1] ? j+1 : 0;  
}  
}  
printf("Encrypted: %s", s);  
return 0;  
}

5. #include <stdio.h> // Для ввода/вывода (printf, scanf, fgets, fopen)

#include <string.h> // Для работы со строками (strcmp)

int main() {

// Создаем буферы (массивы символов) для хранения введенных данных

char login[20]; // Для логина (макс 19 символов + '\0')

char password[20]; // Для пароля

char line[100]; // Для чтения строк из файла

// Запрос и ввод логина

printf("Login: ");

// %19s - читает не более 19 символов (предотвращает переполнение буфера)

// login - буфер куда сохранять

scanf("%19s", login);

// Запрос и ввод пароля

printf("Password: ");

scanf("%19s", password);

// Открываем файл с пользователями для чтения

// "r" - режим чтения (read)

FILE \*file = fopen("users.csv", "r");

// Проверка открылся ли файл

if (!file) {

printf("Database error!\n"); // Если файл не открылся

return 1; // Завершаем программу с кодом ошибки

}

// Читаем файл построчно

// fgets читает одну строку (макс 99 символов + '\0')

while (fgets(line, sizeof(line), file)) {

// Буферы для данных из файла

char file\_login[20];

char file\_password[20];

char name[20];

/\* Разбираем строку на компоненты:

%19[^;] - читает до 19 символов до точки с запятой

; - пропускает разделитель

%19[^\n] - читает до перевода строки

sscanf возвращает количество успешно считанных элементов \*/

if (sscanf(line, "%19[^;];%19[^;];%19[^\n]",

file\_login, file\_password, name) == 3) {

/\* Сравнение строк:

strcmp(str1, str2) возвращает 0 если строки равны

Проверяем совпадение логина и пароля \*/

if (strcmp(login, file\_login) == 0 &&

strcmp(password, file\_password) == 0) {

// Если совпали - выводим приветствие

printf("Привет %s\n", name);

// Закрываем файл и завершаем программу

fclose(file);

return 0;

}

}

}

// Если дошли сюда - совпадений не найдено

printf("Invalid login or password!\n");

fclose(file); // Закрываем файл

return 0; // Нормальное завершение

}