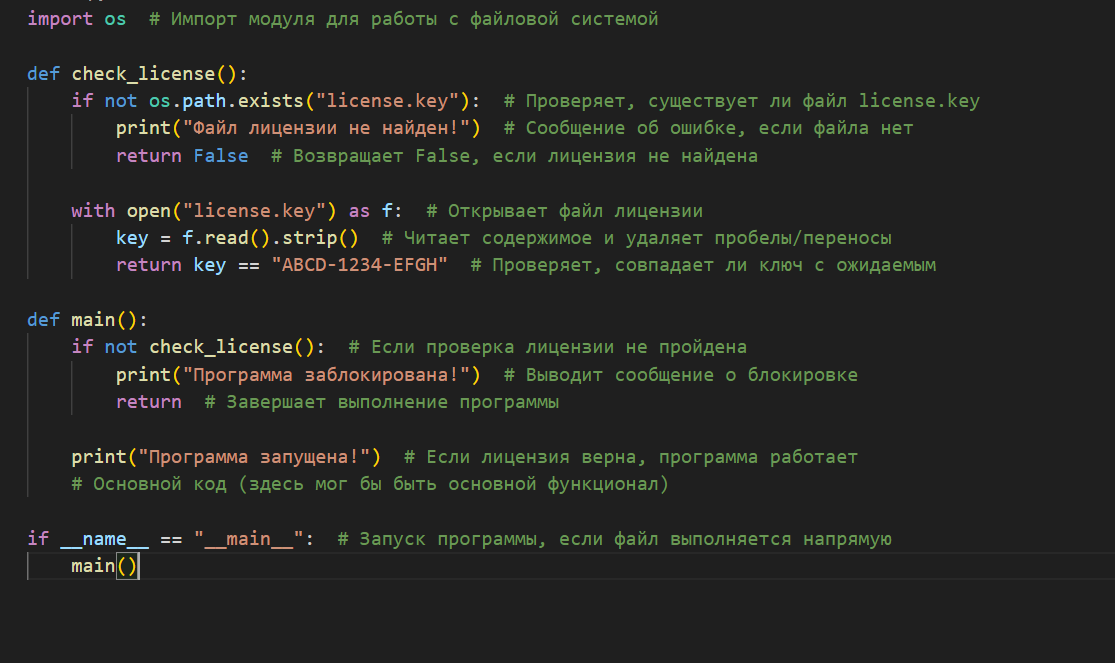
Пара 14

Прокомментировать код:



Проверка трех случаев:

1. License.key отсутствует:

Output:

Файл лицензии не найден!

Программа заблокирована!

Программа завершается, файл не найден

1. **Есть файл лицензии (license.key содержит ABCD-1234-EFGH)**

Output:

Программа запущена!

Программа работает

1. **Лицензия не соответствует (license.key содержит другой** ключ, например, XXXX-0000-YYYY)

Output:

Программа заблокирована!

Программа будет завершена

**Ответы на вопросы**

**Задача 4: Ответы на вопросы**

**1. Основные цели защиты ПО от исследования:**

Защита интеллектуальной собственности.

Предотвращение взлома и несанкционированного использования.

Блокировка реверс-инжиниринга и модификации кода.

**2. Важность защиты ПО для бизнеса:**

Сохранение конкурентных преимуществ.

Предотвращение утечек данных и финансовых потерь.

Защита от пиратства и нелегального копирования.

**3. Риски при отсутствии защиты:**

Утечка алгоритмов и данных.

Внедрение вредоносного кода.

Финансовые потери из-за пиратства.

**4. Реверс-инжиниринг и его угрозы:**

**Реверс-инжиниринг** – анализ кода для понимания его работы.

**Угрозы:**

Кража алгоритмов.

Поиск уязвимостей для взлома.

Создание копий и пиратских версий.

**5. Основные методы реверс-инжиниринга:**

Декомпиляция (преобразование бинарного кода в исходный).

Дизассемблирование (анализ машинного кода).

Динамический анализ (отладка и трассировка).

**6. Уязвимости в недостаточно защищённых программах:**

Жёстко зашитые ключи (как в примере).

Отсутствие шифрования данных.

Уязвимости к инъекциям и перехвату управления.

**7. Распространённые методы защиты:**

Обфускация кода.

Шифрование данных и лицензий.

Антиотладочные техники.

**8. Шифрование данных и его виды:**

**Симметричное** (один ключ, например, AES).

**Асимметричное** (публичный + приватный ключ, RSA).

**9. Обфускация кода и её польза:**

Запутывание кода, затрудняющее анализ.

Усложняет реверс-инжиниринг.

**10. Инструменты обфускации:**

**Python:** PyArmor, Oxyry, Ghidra.

**Java/C#:** ProGuard, Dotfuscator.

**11. Лицензирование для защиты ПО:**

Проверка легальности использования.

Ограничение функционала без лицензии.

**12. Эффективные типы лицензий:**

Аппаратные (USB-ключи).

Онлайн-активация.

Подписки с проверкой сервера.

**13. Тестирование защитных механизмов:**

Пентестинг (попытки взлома).

Фаззинг (автоматический ввод неверных данных).

**14. Важность аудитов безопасности:**

Выявление уязвимостей до релиза.

Соответствие стандартам (ISO 27001).

**15. Автоматизированное тестирование защиты:**

Скрипты для проверки устойчивости к взлому.

Инструменты типа **Burp Suite**, **IDA Pro**.

**16. Примеры успешной защиты:**

Denuvo (защита игр от пиратства).

Серверная валидация лицензий (как в Adobe).

**17. Адаптация защиты под тип ПО:**

**Десктоп:** аппаратные ключи.

**Веб:** JWT-токены, HTTPS.

**18. Известные случаи взлома:**

Взлом игр с Denuvo (со временем уязвимости находили).

Утечка исходников Windows 2000.

**19. Машинное обучение в защите ПО:**

Анализ поведения для обнаружения аномалий.

Генерация сложнообфусцированного кода.

**20. Юридические аспекты:**

Патентование алгоритмов.

DMCA (защита от обхода защиты).

**21. Новые тренды в защите:**

**Контейнеризация** (Docker, изоляция кода).

**Zero Trust** (постоянная проверка доступа).

**22. Влияние технологий на защиту:**

Усложнение реверс-инжиниринга из-за виртуализации (VMProtect).

**23. Уязвимости в IoT и AI:**

**IoT:** слабая защита прошивок.

**AI:** кража моделей и данных обучения.