МАТЕРИАЛЫ ЗАДАНИЙ МЕЖРЕГИОНАЛЬНОЙ ОЛИМПИАДЫ ШКОЛЬНИКОВ ИМЕНИ И.Я. ВЕРЧЕНКО ПО ПРОФИЛЮ «КОМПЬЮТЕРНАЯ БЕЗОПАСНОСТЬ»

2025/2026 УЧЕБНЫЙ ГОД

ЗАКЛЮЧИТЕЛЬНЫЙ ЭТАП
11 КЛАСС

СОДЕРЖАНИЕ

Задача 1. Музыкальный ребус

Вариант 1

Отдел информационной безопасности музыкального журнала «Седьмая нота» расследует утечку конфиденциальных данных. В ходе расследования установлено, что администратор передает пароли пользователей своему сообщнику используя музыкальные ноты (до, ре, ми, фа, соль, ля, си).

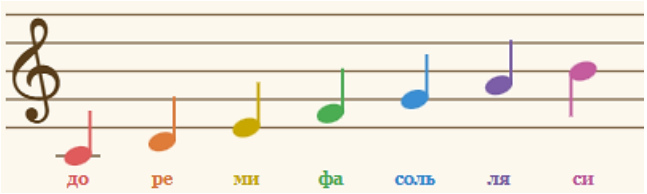


Рисунок 1 – Нотная нотация

В электронной почте администратора обнаружена следующая странная мелодия (рисунок 2). Вам необходимо проанализировать содержимое и определить скомпрометированный пароль.

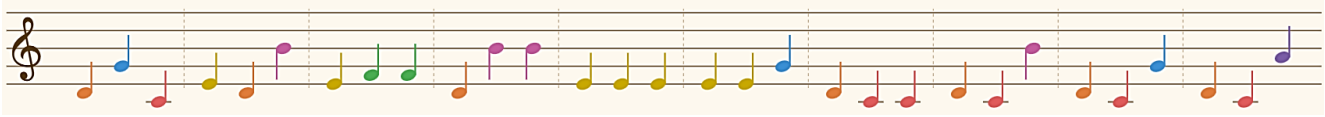


Рисунок 2 – Странная мелодия

Решение

Ноты переводим в цифры:

ДО – 0
РЕ – 1
МИ – 2
ФА – 3
СОЛЬ – 4
ЛЯ – 5
СИ – 6

Получается 7-значная система счисления. Далее группу нот в каждом такте преобразуем в 7-значное число, потом переводим в 10-значное число и смотрим по ASCII-таблице код символа

Ответ: Mozart1756

Вариант 2

Отдел информационной безопасности музыкального журнала «Седьмая нота» расследует утечку конфиденциальных данных. В ходе расследования установлено, что администратор передает пароли пользователей своему сообщнику используя музыкальные ноты (до, ре, ми, фа, соль, ля, си).

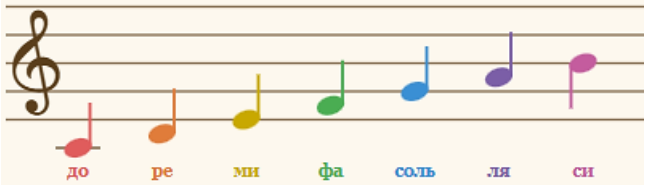


Рисунок 1 – Нотная нотация

В электронной почте администратора обнаружена следующая странная мелодия (рисунок 2). Вам необходимо проанализировать содержимое и определить скомпрометированный пароль.

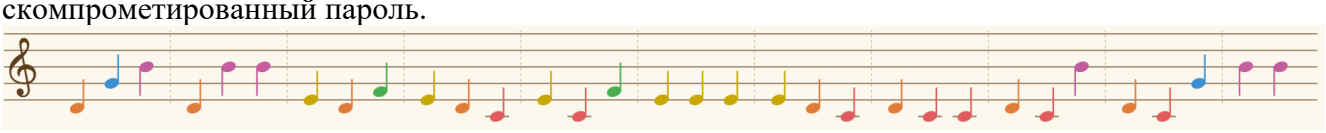


Рисунок 2 – Странная мелодия

Решение

Ноты переводим в цифры:

ДО – 0
РЕ – 1
МИ – 2
ФА – 3
СОЛЬ – 4
ЛЯ – 5
СИ – 6

Получается 7-значная система счисления. Далее группу нот в каждом такте преобразуем в 7-значное число, потом переводим в 10-значное число и смотрим по ASCII-таблице код символа

Ответ: Salieri1750

Задача 2. Хэширование

Вариант 1

Сотрудник компании «DC-Крипто» забыл пароль от доступа к ресурсам компании. В базе данных удалось получить хеш-значение строки его пароля:

`06a9a488671b09fad7b86e4458454f4e`

Из внутренней документации известно, что для получения хеш-значения в компании используется собственный упрощённый алгоритм хеширования, использующий для подмешивания «соль» – константная символьная строка, добавляемая к хешируемой строке по некоторому правилу.

Исходный код функций хеширования представлен в листинге.

Листинг 1. Исходный код функций хеширования

Python

```python
# Нормализует соль до ровно 4 символов
def normalize_salt(salt: str) -> str:
    if len(salt) == 0:
        return ''
    if len(salt) < 4:
        return ''.join(salt[i % len(salt)] for i in range(4))
    else:
        return salt[:4]

# Добавляет соль в пароль
def weave_salt(password: str, salt: str) -> str:
    salt = normalize_salt(salt)
    half = len(password) // 2
    return salt[0:2] + password[:half] + salt[1:3] + password[half:] + salt[2:4]

# Вычисляет хеш строки
def custom_hash(s: str) -> str:
    h = [0xFE, 0xCA, 0xBE, 0xBA, 0xAD, 0xDE, 0xEF, 0xBE]
    for c in s:
        h = [ (h[i] * 65537 * (i+1) + ord(c)) % (2**16) for i in range(len(h))]

    hash_str = ''.join(f"{x:04x}" for x in h)
    hash_4 = s[:2] + s[-2:]
    salt = ''.join(f"{ord(ch):02x}" for ch in hash_4)
    return hash_str[:-8] + salt
```

C++

```cpp
#include <iostream>
#include <string>
#include <sstream>
#include <iomanip>
using namespace std;

// Функция normalize_salt - нормализует соль до 4 символов
string normalize_salt(string salt) {
    if (salt.length() == 0) {
        return "";
    }
    if (salt.length() < 4) {
        string result = "";
        for (int i = 0; i < 4; i++) {
            result += salt[i % salt.length()];
        }
        return result;
    }
```

```
    else {
        return salt.substr(0, 4);
    }
}

// Функция weave_salt – добавляет соль в пароль
string weave_salt(string password, string salt) {
    salt = normalize_salt(salt);
    int half = password.length() / 2;
    string result = "";
    result += salt.substr(0, 2);
    result += password.substr(0, half);
    result += salt.substr(1, 2);
    result += password.substr(half);
    result += salt.substr(2, 2);
    return result;
}

// Функция custom_hash – вычисляет хеш строки
string custom_hash(string s) {
    unsigned int h[8] = { 0xFE, 0xCA, 0xBE, 0xBA, 0xAD, 0xDE, 0xEF, 0xBE };
    for (char c : s) {
        for (int i = 0; i < 8; i++) {
            h[i] = (h[i] * 65537 * (i + 1) + (unsigned int)c) % 65536;
        }
    }
    stringstream hash_stream;
    for (int i = 0; i < 8; i++) {
        hash_stream << hex << setw(4) << setfill('0') << h[i];
    }
    string hash_str = hash_stream.str();
    string hash_4 = s.substr(0, 2) + s.substr(s.length() - 2);
    string salt_hex = "";
    for (char ch : hash_4) {
        stringstream ss;
        ss << hex << setw(2) << setfill('0') << (int)(unsigned char)ch;
        salt_hex += ss.str();
    }
    string final_hash = hash_str.substr(0, hash_str.length() - 8) + salt_hex;
    return final_hash;
}
```

Сформируйте значение хеш-строки для нового пароля пользователя – «newpassword», при условии, что значение «соли» для всех хеш-строк одинаковое.

В ответе укажите значение хеш-строки и значение «соли».

Решение

Последние 8 символов хеш-значения – это и есть «соль». Если взять их и разить на пары и посмотреть по ASCII-таблице, то получим строку XEON.

Далее подставляем значения нового пароля – newpassword и значение «соли» – XEON в функцию weave_salt(), потом вызываем функцию custom_hash() и получаем новое хеш-значение.

Ответ: хеш – 07895f1c41a95f4aaee4d93858454f4e, «соль» – XEON

Вариант 2

Сотрудник компании «DC-Крипто» забыл пароль от доступа к ресурсам компании. В базе данных удалось получить хеш-значение строки его пароля:

`06a0eb6be9e8e595cfb301af5a554d41`

Из внутренней документации известно, что для получения хеш-значения в компании используется собственный упрощённый алгоритм хеширования, использующий для подмешивания «соль» – константная символьная строка, добавляемая к хешируемой строке по некоторому правилу.

Исходный код функций хеширования представлен в листинге.

Листинг 1. Исходный код функций хеширования

Python

```python
# Нормализует соль до ровно 4 символов
def normalize_salt(salt: str) -> str:
    if len(salt) == 0:
        return ''
    if len(salt) < 4:
        return ''.join(salt[i % len(salt)] for i in range(4))
    else:
        return salt[:4]

# Добавляет соль в пароль
def weave_salt(password: str, salt: str) -> str:
    salt = normalize_salt(salt)
    half = len(password) // 2
    return salt[0:2] + password[:half] + salt[1:3] + password[half:] + salt[2:4]

# Вычисляет хеш строки
def custom_hash(s: str) -> str:
    h = [0xFE, 0xCA, 0xBE, 0xBA, 0xAD, 0xDE, 0xEF, 0xBE]
    for c in s:
        h = [ (h[i] * 65537 * (i+1) + ord(c)) % (2**16) for i in range(len(h))]

    hash_str = ''.join(f"{x:04x}" for x in h)
    hash_4 = s[:2] + s[-2:]
    salt = ''.join(f"{ord(ch):02x}" for ch in hash_4)
    return hash_str[:-8] + salt
```

C++

```cpp
#include <iostream>
#include <string>
#include <sstream>
#include <iomanip>
using namespace std;

// Функция normalize_salt - нормализует соль до 4 символов
string normalize_salt(string salt) {
    if (salt.length() == 0) {
        return "";
    }
    if (salt.length() < 4) {
        string result = "";
        for (int i = 0; i < 4; i++) {
            result += salt[i % salt.length()];
        }
        return result;
    }
    else {
        return salt.substr(0, 4);
```

```
    }
}

// Функция weave_salt – добавляет соль в пароль
string weave_salt(string password, string salt) {
    salt = normalize_salt(salt);
    int half = password.length() / 2;
    string result = "";
    result += salt.substr(0, 2);
    result += password.substr(0, half);
    result += salt.substr(1, 2);
    result += password.substr(half);
    result += salt.substr(2, 2);
    return result;
}

// Функция custom_hash – вычисляет хеш строки
string custom_hash(string s) {
    unsigned int h[8] = { 0xFE, 0xCA, 0xBE, 0xBA, 0xAD, 0xDE, 0xEF, 0xBE };
    for (char c : s) {
        for (int i = 0; i < 8; i++) {
            h[i] = (h[i] * 65537 * (i + 1) + (unsigned int)c) % 65536;
        }
    }
    stringstream hash_stream;
    for (int i = 0; i < 8; i++) {
        hash_stream << hex << setw(4) << setfill('0') << h[i];
    }
    string hash_str = hash_stream.str();
    string hash_4 = s.substr(0, 2) + s.substr(s.length() - 2);
    string salt_hex = "";
    for (char ch : hash_4) {
        stringstream ss;
        ss << hex << setw(2) << setfill('0') << (int)(unsigned char)ch;
        salt_hex += ss.str();
    }
    string final_hash = hash_str.substr(0, hash_str.length() - 8) + salt_hex;
    return final_hash;
}
```

Сформируйте значение хеш-строки для нового пароля пользователя – «newpassword», при условии, что значение «соли» для всех хеш-строк одинаковое.

В ответе укажите значение хеш-строки и значение «соли».

Решение

Последние 8 символов хеш-значения – это и есть «соль». Если взять их и разюить на пары и посмотретьпо ASCII-таблице, то получим строку ZUMA.

Далее подставляем значения нового пароля – newpassword и значение «соли» – XEON в функцию weave_salt(), потом вызываем функцию custom_hash() и получаем новое хеш-значение.

Найдена соль: 'ZUMA'

Ответ: хеш – 079a7d0bb1b65f3511adf71f5a554d41, «соль» – ZUMA

Задача 3. Одноразовый код

Вариант 1

Компания «SecureLogix» внедрила двухфактонную аутентификацию (2FA) для доступа к своей системе. При входе система генерирует одноразовый код, состоящий ровно из 6 символов, выбранных из следующего набора:

– 10 цифр (0-9);
– 26 строчных буквы (a-z);
– 3 спецсимвола («!» «@» «#»).

В коде обязательно должна быть хотя бы одна цифра, хотя бы одна буква и ровно два спецсимвола. Спецсимволы не могут стоять рядом.

Атакующий хочет оценить пространство поиска для брутфорса.

Сколько всего различных одноразовых кодов может сгенерировать система? В ответе укажите точное число.

Решение

Шаг 1. Позиции для спецсимволов

Выбираем 2 позиции из 6 так, чтобы они не были соседними.

Всего способов выбрать 2 позиции из 6: $\binom{6}{2} = 15$

Соседние пары позиций: (1,2), (2,3), (3,4), (4,5), (5,6) – это 5 пар.

Допустимых позиций: $15 - 5 = 10$.

Шаг 2. Выбор самих спецсимволов

На каждую из 2 позиций выбираем один из 3 символов (!, @, #). Повторения разрешены.

$$3 \times 3 = 9 \text{ вариантов.}$$

Шаг 3. Заполнение оставшихся 4 позиций

Доступный алфавит: 10 цифр + 26 букв = 36 символов.

Нужно, чтобы среди 4 символов была хотя бы 1 цифра и хотя бы 1 буква. Используем формулу включений-исключений:

$$N = \underbrace{36^4}_{\text{все варианты}} - \underbrace{26^4}_{\text{только буквы}} - \underbrace{10^4}_{\text{только цифры}}$$

Считаем:

– $36^4 = 1\,679\,616$
– $26^4 = 456\,976$
– $10^4 = 10\,000$

$$N = 1\,679\,616 - 456\,976 - 10\,000 = 1\,212\,640$$

Шаг 4. Подсчет итогового числа комбинаций

Перемножаем все три части:

$$\text{Ответ} = \underbrace{10}_{\text{позиции}} \times \underbrace{9}_{\text{спецсимволы}} \times \underbrace{1\,212\,640}_{\text{цифры и буквы}}$$

$$= 90 \times 1\,212\,640 = \boxed{109\,137\,600}$$

Ответ: 109 137 600

Вариант 2

Компания «SecureLogix» внедрила двухфактонную аутентификацию (2FA) для доступа к своей системе. При входе система генерирует одноразовый код, состоящий ровно из 6 символов, выбранных из следующего набора:

– 10 цифр (0-9);

– 20 строчных буквы (a-t);

– 4 спецсимвола («!» «@» «#» «$»).

В коде обязательно должна быть хотя бы одна цифра, хотя бы одна буква и ровно два спецсимвола. Спецсимволы не могут стоять рядом.

Атакующий хочет оценить пространство поиска для брутфорса.

Сколько всего различных одноразовых кодов может сгенерировать система? В ответе укажите точное число.

Решение

Шаг 1. Позиции для спецсимволов

Выбираем 2 позиции из 6 так, чтобы они не были соседними.

Всего способов выбрать 2 позиции из 6: $\binom{6}{2} = 15$

Соседние пары позиций: (1,2), (2,3), (3,4), (4,5), (5,6) – это 5 пар.

Допустимых позиций: $15 - 5 = 10$.

Шаг 2. Выбор самих спецсимволов

На каждую из 2 позиций выбираем один из 4 символов (!, @, #, $). Повторения разрешены.

$$4 \times 4 = 16 \text{ вариантов.}$$

Шаг 3. Заполнение оставшихся 4 позиций

Доступный алфавит: 10 цифр + 20 букв = **30 символов**.

Хотя бы одна цифра и хотя бы одна буква — формула включений-исключений:

$$30^4 - \underbrace{20^4}_{\text{только буквы}} - \underbrace{10^4}_{\text{только цифры}}$$

– $30^4 = 810\,000$

– $20^4 = 160\,000$

– $10^4 = 10\,000$

$$810\,000 - 160\,000 - 10\,000 = \mathbf{640\,000}$$

Шаг 4. Подсчет итогового числа комбинаций

Перемножаем все три части:

$$N = 10 \times 16 \times 640\,000 = \boxed{102\,400\,000}.$$

Ответ: 102 400 000

Задача 4. Ping

Вариант 1

Отдел информационной безопасности компании «ИТ-Сеть» расследует деятельность внутреннего злоумышленника, который, по данным разведки, планирует встретиться со своим сообщником для передачи конфиденциальных данных. Команда аналитиков смогла перехватить часть трафика, где передавалось место встречи злоумышленников, однако понять, как именно оно передавалось им не удалось.

На основе имеющегося трафика, определите передаваемое место встречи.

В качестве ответа укажите место встречи.

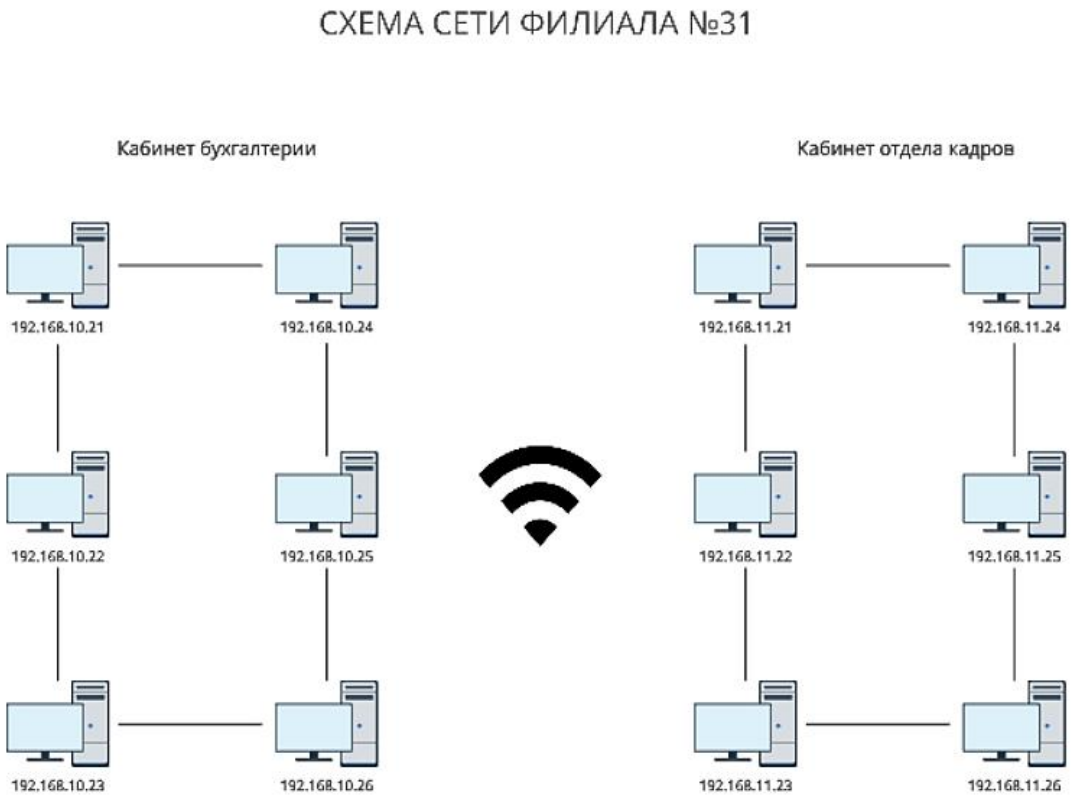


Рисунок 1 – Схема сети

К задаче прилагается:
файл «logs.txt»

Решение

Дана топология сети и файл logs.txt. На схеме мы ничего приметного не находим. Просто два отдела по 6 ПК с указанием IP-адресов.

Рассмотрим файл logs.txt. В нем можно увидеть урезанный дамп трафика с указанием времени, источника, назначения, протокола, длины и информации. Просмотрев весь трафик можно сделать вывод, что это ping-запросы, так как работает протокол ICMP и в информации передается информация, похожая на результаты выполнения команды ping.

При выполнении команды ping идет подсчет пакетов, отправленных от источника к получателю. Как, например, вот здесь:

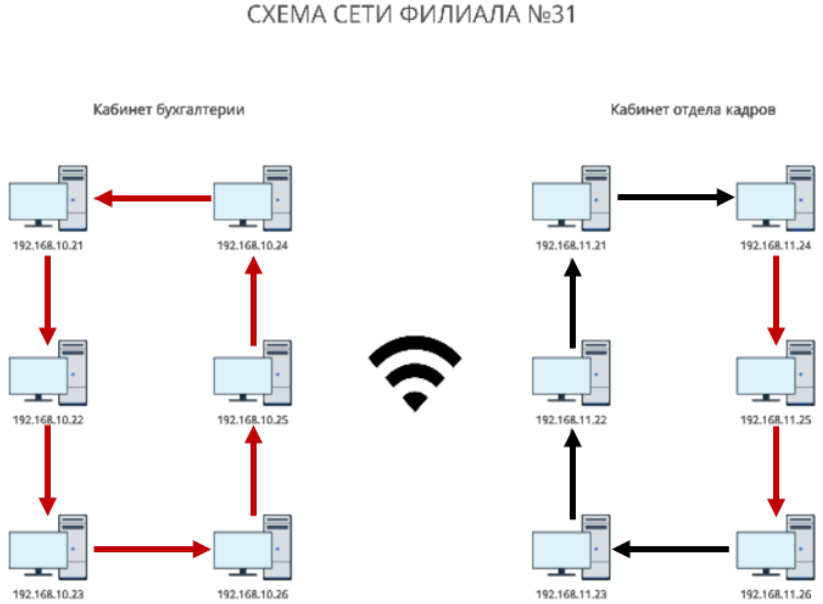
```
└─# ping ya.ru
PING ya.ru (5.255.255.242) 56(84) bytes of data.
64 bytes from ya.ru (5.255.255.242): icmp_seq=1 ttl=240 time=12.5 ms
64 bytes from ya.ru (5.255.255.242): icmp_seq=2 ttl=240 time=6.16 ms
64 bytes from ya.ru (5.255.255.242): icmp_seq=3 ttl=240 time=6.65 ms
64 bytes from ya.ru (5.255.255.242): icmp_seq=4 ttl=240 time=6.64 ms
```

Однако в полученном нами файле, подсчет пакетов идет с разных источников к разным получателям.

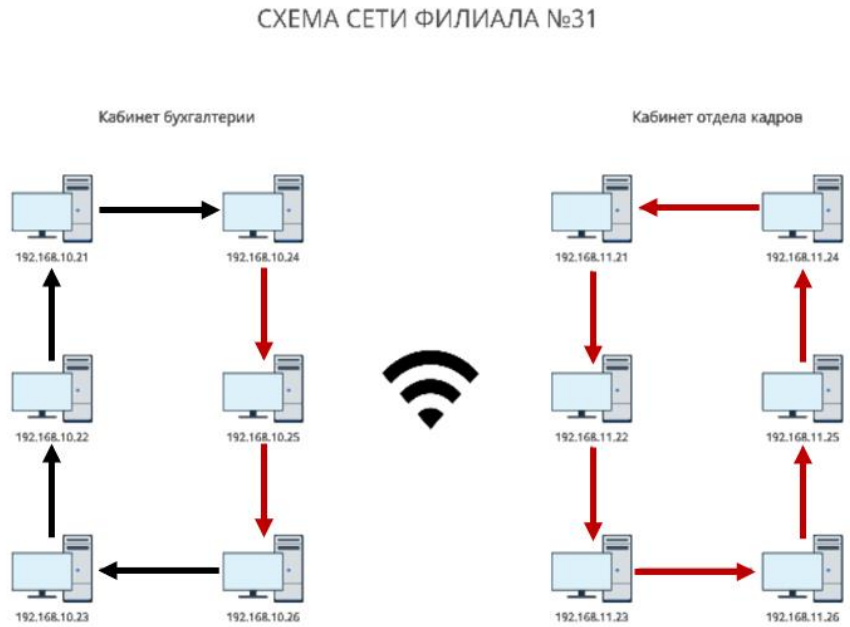
Время	Источник	Назначение	Протокол	Длина	Информация
12:00:00.000	192.168.10.21	192.168.10.22	ICMP	64	Src port: 45000, Dst port: 45000, Data: 64 bytes from 192.168.10.22: icmp_seq=1 ttl=240 time=1.5 ms
12:00:01.502	192.168.10.22	192.168.10.23	ICMP	64	Src port: 45000, Dst port: 45000, Data: 64 bytes from 192.168.10.23: icmp_seq=2 ttl=240 time=6.16 ms
12:00:02.003	192.168.10.23	192.168.10.26	ICMP	64	Src port: 45000, Dst port: 45000, Data: 64 bytes from 192.168.10.26: icmp_seq=3 ttl=240 time=6.65 ms
12:00:03.504	192.168.10.26	192.168.10.25	ICMP	64	Src port: 45000, Dst port: 45000, Data: 64 bytes from 192.168.10.25: icmp_seq=4 ttl=240 time=6.64 ms
12:00:04.005	192.168.10.25	192.168.10.24	ICMP	64	Src port: 45000, Dst port: 45000, Data: 64 bytes from 192.168.10.24: icmp_seq=5 ttl=240 time=6.39 ms
12:00:05.000	192.168.10.24	192.168.10.21	ICMP	64	Src port: 45000, Dst port: 45000, Data: 64 bytes from 192.168.10.21: icmp_seq=6 ttl=240 time=6.98 ms
12:00:15.000	192.168.11.24	192.168.11.25	ICMP	64	Src port: 45000, Dst port: 45000, Data: 64 bytes from 192.168.11.25: icmp_seq=1 ttl=213 time=10.0 ms
12:00:15.433	192.168.11.25	192.168.11.26	ICMP	64	Src port: 45000, Dst port: 45000, Data: 64 bytes from 192.168.11.26: icmp_seq=2 ttl=213 time=3.12 ms

Также, можно заметить, что сначала идут запросы из бухгалтерии от ip-адресов 192.168.10.xx, а потом идут запросы из отдела кадров от ip-адресов 192.168.11.xx и потом ситуация повторяется.

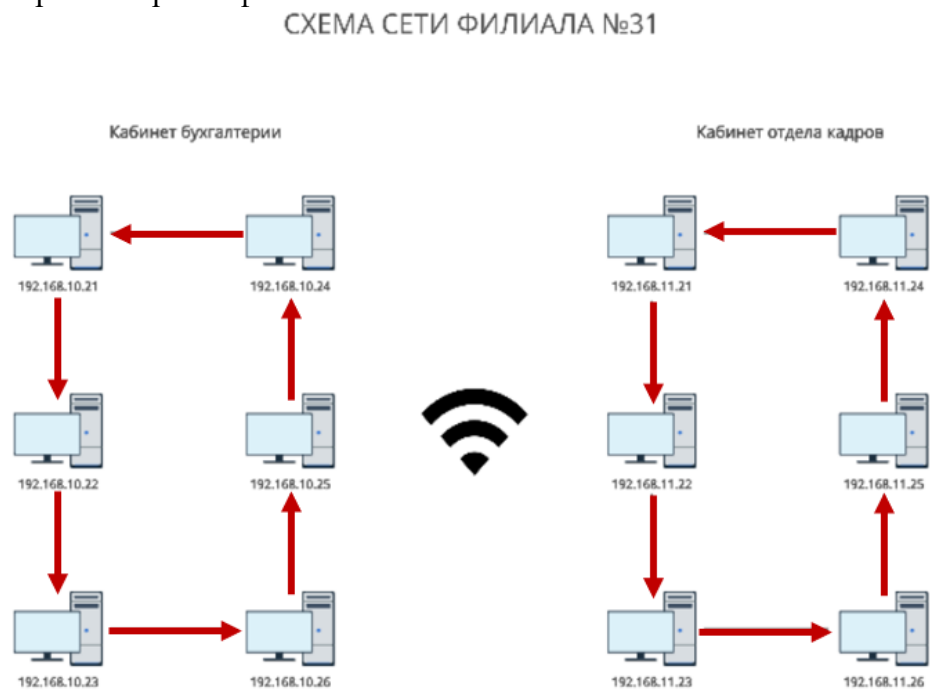
Попробуем провести линию общения компьютеров на схеме сети, основываясь на файле logs.txt.



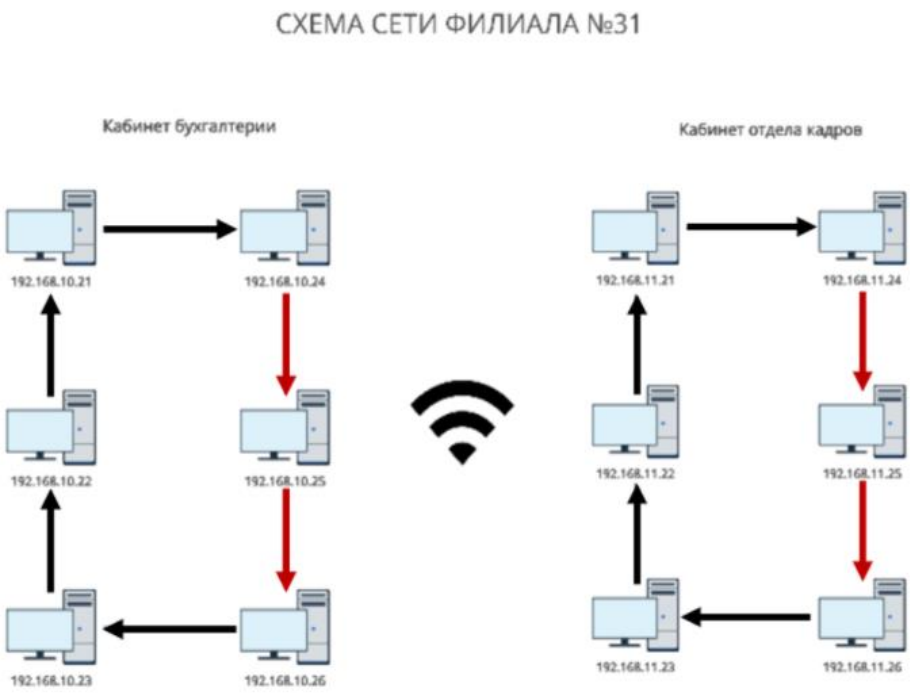
Красными стрелками указано, где команда ping выполнилась и получила ответ, то есть связь между ПК есть, а черными стрелками указано, где команда ping выполнилась и ответа не последовало, то есть связи между ПК нет. Попробуем так дальше идти по файлу logs.txt.



Смотрим третий отрезок файла.



И четвертый отрезок.



Обрисовав на схеме движение команды ping, можно заметить, что вырисовываются нули и единицы, словно именно таким образом попытались зашифровать сообщение. Попробуем расшифровать первые 8 бит, которые мы только что получили.

01100011 в ASCII – это буква «c». Пока это ничего не дает, но мы попробуем продолжить. Рассмотрев таким образом все общение компьютеров в сети, получаем следующую картину:

```
01100011 01100001 01100110 01100101 00100000 001000101 01001110
01001111 01010100
```

Остается только перевести все это в ASCII и получаем – **cafe ENOT**.

Ответ: cafe ENOT

Вариант 2

Отдел информационной безопасности компании «ИТ-Сеть» расследует деятельность внутреннего злоумышленника, который, по данным разведки, планирует встретиться со своим сообщником для передачи конфиденциальных данных. Команда аналитиков смогла перехватить часть трафика, где передавалось место встречи злоумышленников, однако понять, как именно оно передавалось им не удалось.

На основе имеющегося трафика, определите передаваемое место встречи.

В качестве ответа укажите место встречи.

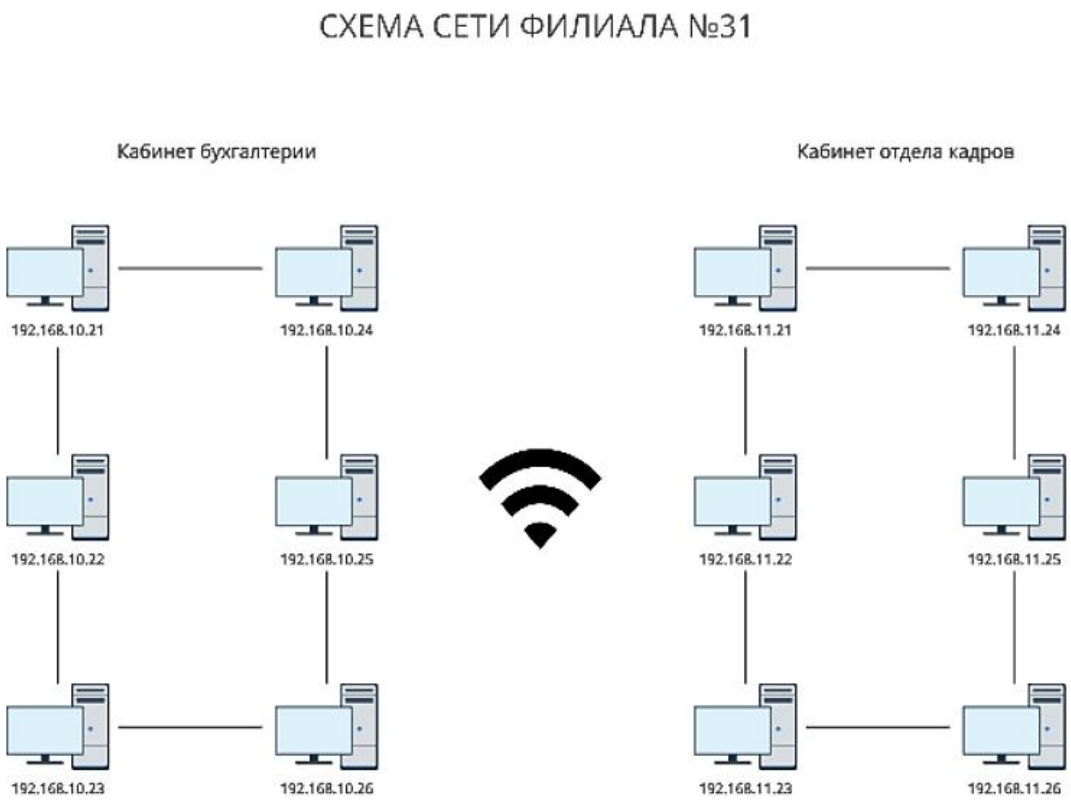


Рисунок 1 – Схема сети

К задаче прилагается:
файл «logs.txt»

Решение

Дана топология сети и файл logs.txt. На схеме мы ничего приметного не находим. Просто два отдела по 6 ПК с указанием IP-адресов.

Рассмотрим файл logs.txt. В нем можем увидеть урезанный дамп трафика с указанием времени, источника, назначения, протокола, длины и информации. Просмотрев весь трафик можно сделать вывод, что это ping-запросы, так как работает протокол ICMP и в информации передается информация, похожая на результаты выполнения команды ping.

При выполнении команды ping идет подсчет пакетов, отправленных от источника к получателю. Как, например, вот здесь:

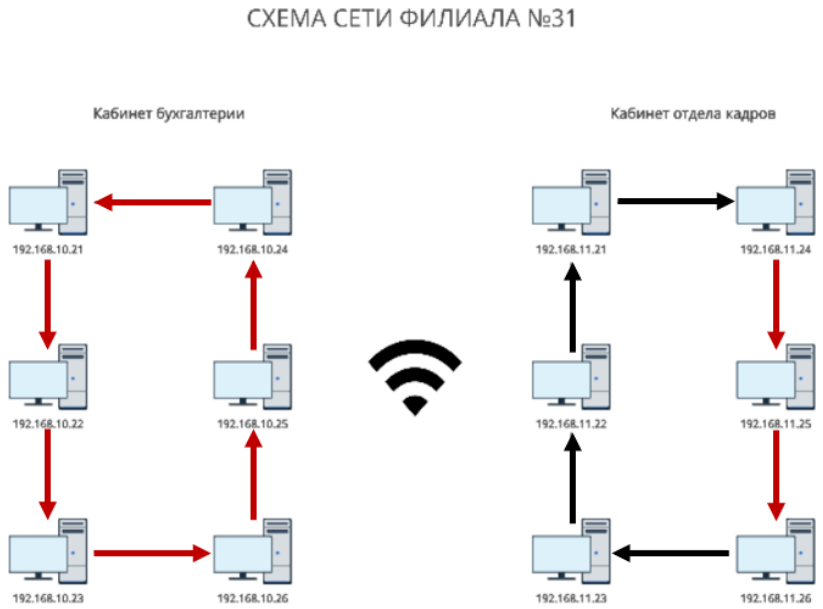
```
└─# ping ya.ru
PING ya.ru (5.255.255.242) 56(84) bytes of data.
64 bytes from ya.ru (5.255.255.242): icmp_seq=1 ttl=240 time=12.5 ms
64 bytes from ya.ru (5.255.255.242): icmp_seq=2 ttl=240 time=6.16 ms
64 bytes from ya.ru (5.255.255.242): icmp_seq=3 ttl=240 time=6.65 ms
64 bytes from ya.ru (5.255.255.242): icmp_seq=4 ttl=240 time=6.64 ms
```

Однако в полученном нами файле, подсчет пакетов идет с разных источников к разным получателям.

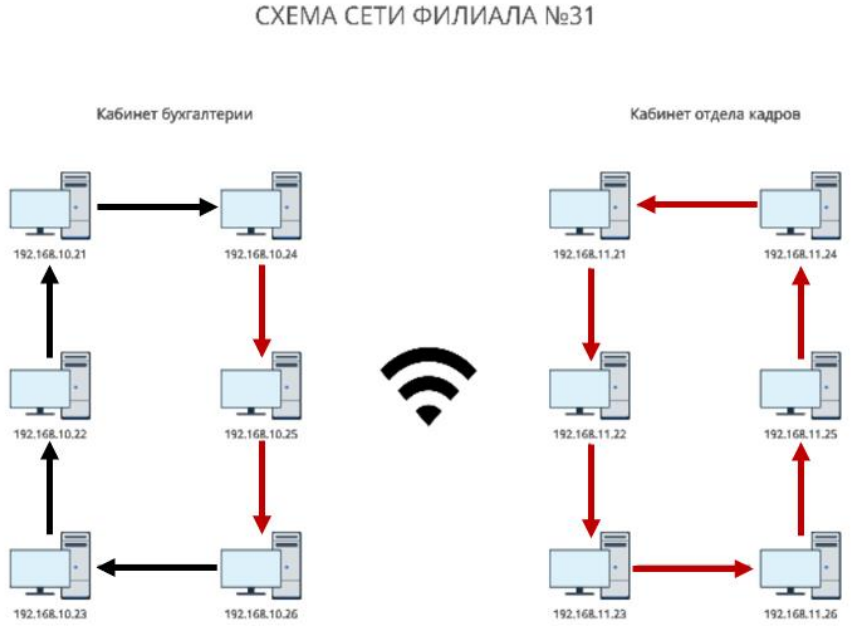
Время	Источник	Назначение	Протокол	Длина	Информация
12:00:00.000	192.168.10.21	192.168.10.22	ICMP	64	Src port: 45000, Dst port: 45000, Data: 64 bytes from 192.168.10.22: icmp_seq=1 ttl=240 time=1.5 ms
12:00:01.502	192.168.10.22	192.168.10.23	ICMP	64	Src port: 45000, Dst port: 45000, Data: 64 bytes from 192.168.10.23: icmp_seq=2 ttl=240 time=6.16 ms
12:00:02.003	192.168.10.23	192.168.10.26	ICMP	64	Src port: 45000, Dst port: 45000, Data: 64 bytes from 192.168.10.26: icmp_seq=3 ttl=240 time=6.65 ms
12:00:03.504	192.168.10.26	192.168.10.25	ICMP	64	Src port: 45000, Dst port: 45000, Data: 64 bytes from 192.168.10.25: icmp_seq=4 ttl=240 time=6.64 ms
12:00:04.005	192.168.10.25	192.168.10.24	ICMP	64	Src port: 45000, Dst port: 45000, Data: 64 bytes from 192.168.10.24: icmp_seq=5 ttl=240 time=6.39 ms
12:00:05.000	192.168.10.24	192.168.10.21	ICMP	64	Src port: 45000, Dst port: 45000, Data: 64 bytes from 192.168.10.21: icmp_seq=6 ttl=240 time=6.98 ms
12:00:15.000	192.168.11.24	192.168.11.25	ICMP	64	Src port: 45000, Dst port: 45000, Data: 64 bytes from 192.168.11.25: icmp_seq=1 ttl=213 time=10.0 ms
12:00:15.433	192.168.11.25	192.168.11.26	ICMP	64	Src port: 45000, Dst port: 45000, Data: 64 bytes from 192.168.11.26: icmp_seq=2 ttl=213 time=3.12 ms

Также, можно заметить, что сначала идут запросы из бухгалтерии от ip-адресов 192.168.10.xx, а потом идут запросы из отдела кадров от ip-адресов 192.168.11.xx и потом ситуация повторяется.

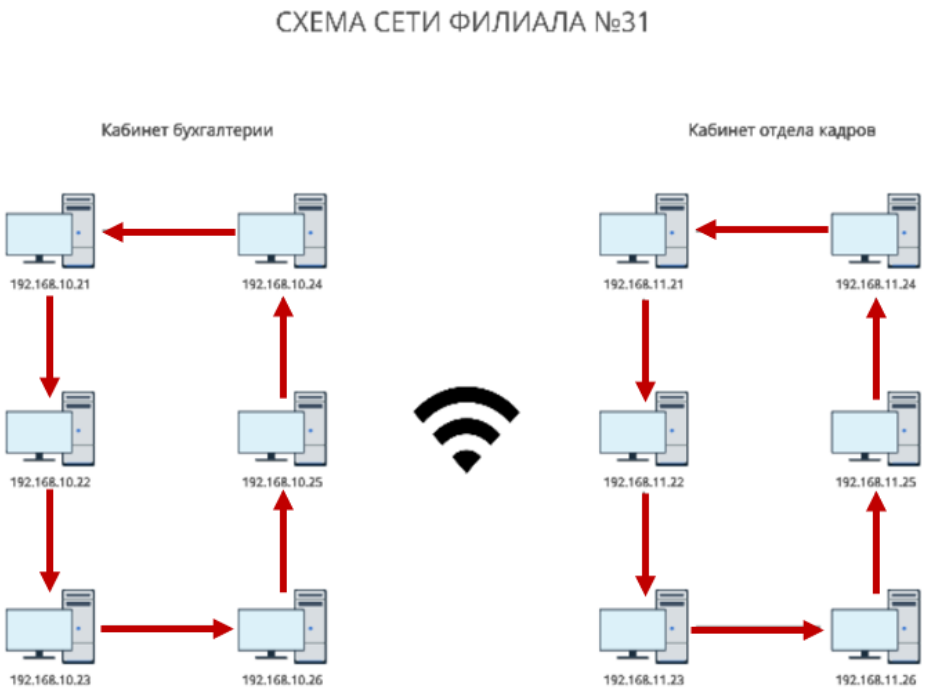
Попробуем провести линию общения компьютеров на схеме сети, основываясь на файле logs.txt.



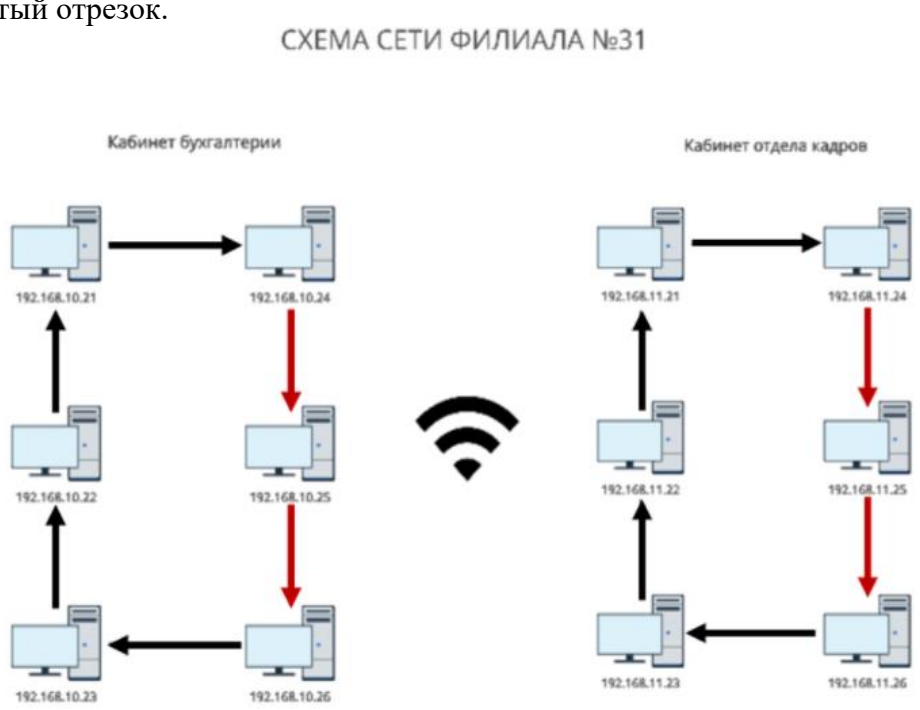
Красными стрелками указано, где команда ping выполнилась и получила ответ, то есть связь между ПК есть, а черными стрелками указано, где команда ping выполнилась и ответа не последовало, то есть связи между ПК нет. Попробуем так дальше идти по файлу logs.txt.



Смотрим третий отрезок файла.



И четвертый отрезок.



Обрисовав на схеме движение команды ping, можно заметить, что вырисовываются нули и единицы, словно именно таким образом попытались зашифровать сообщение. Попробуем расшифровать первые 8 бит, которые мы только что получили.

01110000 в ASCII – это буква «p». Пока это ничего не дает, но мы попробуем продолжить. Рассмотрев таким образом все общение компьютеров в сети, получаем следующую картину:

```
01110000 01100001 01110010 01101011 00100000 01001101 01010101
01011000 01000001
```

Остается только перевести все это в ASCII и получаем – **park MUXA**.

Ответ: **park MUXA**

Задача 5. Виженер

Вариант 1

Шифр Виженера – это метод многоалфавитной подстановки, в котором используется ключевое слово, задающее последовательность сдвигов.

Принцип работы:

1. Каждой букве алфавита сопоставляется число: А=0, Б=1, В=2, … Я=32.
2. Ключ повторяется циклически, чтобы совпадать по длине с текстом.
3. Каждая буква текста сдвигается на значение соответствующей буквы ключа по модулю 33:

$$C_i = (P_i + K_i) \bmod 33$$

Где, P_i – номер буквы открытого текста,
K_i – номер буквы ключа,
C_i – номер буквы шифротекста.

Расшифровка выполняется вычитанием:

$$P_i = (C_i - K_i) \bmod 33$$

Пример:
Открытый текст – *ПРИВЕТ*
Ключ – *КЛЮЧ*
Преобразуем в числа:

П = 16	Р = 17	И = 9	В = 2	Е = 5	Т = 19
К = 11	Л = 12	Ю = 31	Ч = 24	К = 11	Л = 12

Шифруем по формуле:
(16 + 11) mod 33 = 27 → *Ъ*
(17 + 12) mod 33 = 29 → *Ь*
(9 + 31) mod 33 = 7 → *Ж*
(2 + 24) mod 33 = 26 → *Щ*
(5 + 11) mod 33 = 16 → *П*
(19 + 12) mod 33 = 31 → *Ю*

Шифротекст: *ЪЬЖЩПЮ*

Системный администратор издательства «Книжный мир» обнаружил странное вложение в письме с утверждением к печати рассказа одного из известных писателей. Ранее в Интернет выложили копию базы данных издательства с конфиденциальными данными, однако пароль от базы данных не был опубликован.

Вам необходимо подтвердить факт компрометации пароля от базы данных. Системный администратор предложил несколько ключей для расшифровки сообщения:

1.	МАРШАК
2.	ТЮТЧЕВ
3.	ЕСЕНИН
4.	ПУШКИН
5.	ОГАРЁВ

В качестве ответа предоставьте пароль от базы данных в зашифрованном сообщении.

Текст сообщения:

ГЯЛХЧЪЮДФЙМБРЫЭЦНЫКЭАЦИАПТОПШЙЭУЬЮЙТВВЕУМЫФАБШЧЕЛСГЩЫБЖШЁЁТПБЩНЩМЦВГЖБНЭШЮИЮЛЬЬЬЬП
ЫЫПГИККЬЯШЙШЕХПХЖОСАЭУДПКЯЪУАХЬРЮХЖЫСАВУЕВЬСФЕШЭИЪЫШРУТОАВЬУЫЮГЕШЦУЫПХЭЭКМЕЕБОСАВУ
ЕШИЫФХЭОЧЪКИЬЩЩЬЦЮШАЪЩФЧУШНПШЧШШЦИЕЫЪПЩТЩЬАЛЬЁЪУККХЫПЮЛЫЕЦЕБЖСУНЕЕКЩСАРШАЩУЬЭФЭТМП
ФДЭФЫНЬЯЭЬССЮЯЪУМТЭЬВЪЧЩЭОККХЬЧУИПШЮШИДЁЦБВХЖЦЦИЭУЩЫНРЯШЙВИЫКЭБЪЬЯВВВУЫЮШЧОКЫЙСЬКЙ
РТЩГИПУЮПЕЁЁЮЕАШЬЩТЦЧХЖОКИЕВЬЙЫМББУАСЯЭЬЕУММУПГКСГЪВИЬУЬЩЁШЧУЬАВДПКЦЖАБЧЧГЮЧЖЧШЩФБ
ЧПЫРАВАШЧРЮЙШЫЗАПАЪЩЙЩПЮШАШТАШЬШИЮЮЧЖЧАТАШАЦНЯПКЭЫНХЬВИЙУЬЫЧЛШЦТБШКЛЧРПЁУЫЗПВШЕШСД
ФЁШЧЯНАШЪШИАГЪБЭИОГДУФКЬЫЮЭФКТАБЖЬФБЦЬКЭИЪБЁЛЪИЯРУЩЩЖМТВВУМТВФИПМТВЕШЧИЯЮФЖФЫНЬЙШЫ
ЕШПМЭФЦНУЫДКЫЬЬКШАЦТВЁШЧЩББЕГУТСГИККХЮГЕФИШНЕБЭЭСАПАЧЛДЩШАЭОЗЭШЯЛЧЧЮОХБОНЩУЖЩТНЩФБ
УФШЬУББЧЪЦУШДУУЬВЖППЦИЩЕЪЩСЪЭШЙХИХЪЬЫЩКЬАЬДЩМЫГТЗЩХЫНЕГКРШГШКЮШЬСШЬКЖАФГЭЫЕМБХЭЭЬС
ФЯШОИПЭВЕУЦБСЛБАМЫФЭЗЫНСПБФЙЪАПДБШДРЫЖЩЩУЬЩХКЩФЭФАЖНЬЕШИЙЁЦЬСШВЬМЮГЫФЙХЦСЦИУМЫШЙЭМ
ДЯЮЮЖФКЩПЧБЧСЮБВДШЯТЯЬИЩКНЫАЭШЕЁГСЬЩАЙЮБЪЁМНСУДТИШЭТАЙЮЮПФИЩЛЬАЖЙЦИЫПЬЕПМЦЧЁЧСУЬТВ
ЙЭИШПБШТИЦЕЫЬЩЩЬСПЭМДЭШХШЦЦТБЮЖЫЧТЫЬЗЫНСЪЬЁКБЦЭШЙХЧЮЮЧЪУЛНЫЬЙЖУЮГЦЖЧУЬСЛБЬНЮФФИЙЦИ
ФКШГСЯЪЬЗЙВЦЬГБМЧЪШХБШЧЪЮББМНЯФЯФПКЯФДЬБНЩШЯБГСЭФЯШЪНЫПЮИКЗЪШИЪКПЫЖУРШСШШБЖЬСЩШЬЁУ
РШЮЮДКЦМЫЬЙЖЛЬБЁЧЧШЮЮШГЭСХЬШЁУЫЙЪЯЦВСШМЁЖНЧЁШЗИКЪЩШЯБЬЕЮФКБМББЬБЭМЦМВБУФПБЦЪБЭЛЬБЁ
ЭФКТБШДЁТШАЖЫШЧПУДЛНЩНЧЧШЦЪМТЯШЬШЮШТКШДЧШЫЪЩЦШШИЫЮЪЩФЭЩПЛЩКЭАМЬШСЕЭЬХЬЫЮДУЪЩГЛШИЫО
ПТЁКСЯЫУЪУЫЯЫУЬЩЪЯВБУФШТСШОЦЖСЬЬДЮШЮФЯЭЬЫЦАЖЙЦИЫПЬДПФТЬЕЪУЫИЩЬЕМНЫФЙЁЩЪААУЙЭЕЛЯОДХ
ЧЧГЁЖЧФТЭБУФЦТФЕКШНЭЛШКЬЯЫУЁМФЛРЯЭШЦИЩГШЫЧЩЛВКЛИХКЧШШЦИЕЕКУЮЬВХЖЫНЫШШЁКМЮГЦШЧСЩЮЦ
ЖНФМУЬКМРСКИШПЫЯФДЬУЫЯОЦЖЫСАИЬЗЦЗББВКШНАФДЗПЦЙОЕПУЫНФЁГКПСКШЕНЦЬСШЁЖЗПГБУШЕЦБГШЬХБ
АБУЧАТЫВЕТИЁГАЁЁХЯСУЬПЙЫКАЙЭЧЩЮАЙУММВЁИУКЦВТАЙХЩПЧУПЙТЧАЖЦКЫКЫШХЧПЗВЕЪЬЯВОЕЪИХКФБЦ
СШГФГУУЮГЦЖМДТШФИКЦЁПБЭЪЩЦОЁЁЁХЦЭБЭЬФИЗУКМНЖФЦЖЛИМЭУЗЩЫБЯЬДУЪЪГМЭШЦИЩХАНФМУЁЖЭЩЦБВ
ЗПЩЫШЮШЫЬЯЫУЁККСГЛЭШНЯЖУЙЭЦИФЁШЙЫЩНФЪУСЫФБШМСЯВЬЧОЧСШБИЩЛСПЭЪЩСАФЯФЬХТЫОВЧНЕЮАИКРС
СЬЁЮКЁШЭЗЫНСФЯУЛЧРПЁУАУЦФХЙХСГЯВДПТСАЖЫЩТВПДДКЭШАЬГЮЦЫПЧЕПЦЫКЭЪЪСЮПИЁУУТЬБЭЪЧОФЬЬП
ЦЫКЭЁЩКЬШБЙХЩЬЬБУФЬЮФЧФЧНЕФЭЗЩЪЩФЧЁУТЭЮЯЁЁТЯВДШЬЫЫЮЭЬЮХИЬЯШОЧЧЕУАКЩЯЪЬВАИЫАУКЧСЮКХ
ЙПЫЮЮШЩЦНСЭОБЮЛЮНАУУШЦАХЭЬНЩКЭБЧЦТСГБЫ

К задаче прилагается:

файл «message.txt»

Решение

Имеется зашифрованное сообщение, 5 вариантов ключей. Судя по названию задания это шифр Виженера. Согласно ему – каждая буква текста заменяется на другую букву, сдвинутую на количество позиций в алфавите, определяемое соответствующей буквой ключевого слова.

Соответственно, можно написать программную реализацию этого шифра.

```
def vigenere_decrypt(ciphertext, key,
alphabet="абвгдеёжзийклмнопрстуфхцчшщъыьэюя"):
    """Дешифрование шифра Виженера"""
    decrypted = []
    for i in range(len(ciphertext)):
        letter = ciphertext[i]
        if letter in alphabet:
            letter_num = alphabet.index(letter)
            key_letter = key[i % len(key)]
            key_num = alphabet.index(key_letter)
            new_num = (letter_num - key_num) % len(alphabet)
            decrypted.append(alphabet[new_num])
        else:
            decrypted.append(letter)
    return ''.join(decrypted)

# Зашифрованный текст (вставь свой)
ciphertext =
"ГЯЛХЧЪЮДФЙМБРЫЭЦНЫКЭАЦИАПТОПШЙЭУЬЮЙТВВЕУМЫФАБШЧЕЛСГЩЫБЖШЁЁТПБЩНЩМЦВГЖБНЭШЮИЮЛЬЬЬЬ
ПЫЫПГИККЬЯШЙШЕХПХЖОСАЭУДПКЯЪУАХЬРЮХЖЫСАВУЕВЬСФЕШЭИЪЫШРУТОАВЬУЫЮГЕШЦУЫПХЭЭКМЕЕБОСАВ
УЕШИЫФХЭОЧЪКИЬЩЩЬЦЮШАЪЩФЧУШНПШЧШШЦИЕЫЪПЩТЩЬАЛЬЁЪУККХЫПЮЛЫЕЦЕБЖСУНЕЕКЩСАРШАЩУЬЭФЭТМ
ПФДЭФЫНЬЯЭЬССЮЯЪУМТЭЬВЪЧЩЭОККХЬЧУИПШЮШИДЁЦБВХЖЦЦИЭУЩЫНРЯШЙВИЫКЭБЪЬЯВВВУЫЮШЧОКЫЙСЬК
ЙРТЩГИПУЮПЕЁЁЮЕАШЬЩТЦЧХЖОКИЕВЬЙЫМББУАСЯЭЬЕУММУПГКСГЪВИЬУЬЩЁШЧУЬАВДПКЦЖАБЧЧГЮЧЖЧШЩФ
БЧПЫРАВАШЧРЮЙШЫЗАПАЪЩЙЩПЮШАШТАШЬШИЮЮЧЖЧАТАШАЦНЯПКЭЫНХЬВИЙУЬЫЧЛШЦТБШКЛЧРПЁУЫЗПВШЕШС
ДФЁШЧЯНАШЪШИАГЪБЭИОГДУФКЬЫЮЭФКТАБЖЬФБЦЬКЭИЪБЁЛЪИЯРУЩЩЖМТВВУМТВФИПМТВЕШЧИЯЮФЖФЫНЬЙШ
ЫЕШПМЭФЦНУЫДКЫЬЬКШАЦТВЁШЧЩББЕГУТСГИККХЮГЕФИШНЕБЭЭСАПАЧЛДЩШАЭОЗЭШЯЛЧЧЮОХБОНЩУЖЩТНЩФ
```

```
БУФШЬУББЧЪЦУШДУУЬВЖППЦИЩЕЪЩСЪЭШЙХИХЪЬЫЩКЬАЬДЩМЫГТЗЩХЫНЕГКРШГШКЮШЬСШЬКЖАФГЭЫЕМБХЭЭЬ
СФЯШОИПЭВЕУЦБСЛБАМЫФЭЗЫНСПБФЙЪАПДБШДРЫЖЩЩУЬЩХКЩФЭФАЖНЬЕШИЙЁЦЬСШВЬМЮГЫФЙХЦСЦИУМЫШЙЭ
МДЯЮЮЖФКЩПЧБЧСЮБВДШЯТЯЬИЩКНЫАЭШЕЁГСЬЩАЙЮБЪЁМНСУДТИШЭТАЙЮЮПФИЩЛЬАЖЙЦИЫПЬЕПМЦЧЁЧСУЬТ
ВЙЭИШПБШТИЦЕЫЬЩЩЬСПЭМДЭШХШЩЦТБЮЖЫЧТЫЬЗЫНСЪЬЁКБЦЭШЙХЧЮЮЧЪУЛНЫЬЙЖУЮГЦЖЧУЬСЛБЬНЮФФИЙЦ
ИФКШГСЯЪЬЗЙВЦЬГБМЧЪШХБШЧЪЮБМНЯФЯФПКЯФДЬБНЩШЯБГСЭФЯШЪНЫПЮИКЗЪШИЪКПЫЖУРШСШШБЖЬСЩШЬЁ
УРШЮЮДКЦМЫЬЙЖЛЬБЁЧЧШЮЮШГЭСХЬШЁУЫЙЪЯЦВСШМЁЖНЧЁШЗИКЪЩШЯБЬЕЮФКБМББЬБЭМЦМВБУФПБЦЪБЭЛЬБ
ЁЭФКТБШДЁТШАЖЫШЧПУДЛНЩНЧЧШЦЪМТЯШЬШЮШТКШДЧШЫЪЩЦШШИЫЮЪЩФЭЩПЛЩКЭАМЬШСЕЭЬХЬЫЮДУЪЩГЛШИЫ
ОПТЁКСЯЫУЪУЫЯЫУЬЩЪЯВБУФШТСШОЦЖСЬЬДЮШЮФЯЭЬЫЦАЖЙЦИЫПЬДПФТЬЕЪУЫИЩЬЕМНЫФЙЁЩЪААУЙЭЕЛЯОД
ХЧЧГЁЖЧФТЭБУФЦТФЕКШНЭЛШКЫЬЯЫУЁМФЛРЯЭШЦИЩГШЫЧЩЛВКЛИХКЧШШЦИЕЕКУЮЬВХЖЫНЫШШЁКМЮГЦШЧСЩЮ
ЦЖНФМУЬКМРСКИШПЫЯФДЬУЫЯОЦЖЫСАИЬЗЦЗБББВКШНАФДЗПЦЙОЕПУЫНФЁГКПСКШЕНЦЬСШЁЖЗПГБУШЕЦБГШЬХ
БАБУЧАТЫВЕТИЁГАЁЁХЯСУЬПЙЫКАЙЭЧЩЮАЙУММВЁИУКЦВТАЙХЩПЧУПЙТЧАЖЦКЫКЫШХЧПЗВЕЪЬЯВОЕЪИХКФБ
ЦСШГФГУУЮГЦЖМДТШФИКЦЁПБЭЪЩЦОЁЁЁХЦЭБЭЬФИЗУКМНЖФЦЖЛИМЭУЗЩЫБЯЬДУЪЪГМЭШЦИЩХАНФМУЁЖЭЩЦБ
ВЗПЩЫШЮШЫЬЯЫУЁККСГЛЭШНЯЖУЙЭЦИФЁШЙЫЩНФЪУСЫФБШМСЯВЬЧОЧСШБИЩЛСПЭЪЩСАФЯФЬХТЫОВЧНЕЮАИКР
ССЬЁЮКЁШЭЗЫНСФЯУЛЧРПЁУАУЦФХЙХСГЯВДПТСАЖЫЩТВПДДКЭШАЬГЮЦЫПЧЕПЦЫКЭЪЪСЮПИЁУУТЬБЭЪЧОФЪЬ
ПЦЫКЭЁЩКЬШБЙХЩЬЬБУФЪЮФЧФЧНЕФЭЗЪЩФЧЁУТЭЮЯЁЁТЯВДШЬЫЫЮЭЬЮХИЬЯШОЧЧЕУАКЩЯЪЬВАИЫАУКЧСЮК
ХЙПЫЮЮШЩЦНСЭОБЮЛЮНАУУШЦАХЭЬНЩКЭБЧЦТСГБЫ"

# 5 ключей
keys = [
    "ПУШКИН",
    "ТЮТЧЕВ",
    "ЕСЕНИН",
    "МАРШАК",
    "ОГАРЁВ"
]

alphabet = "АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"

print("Расшифровка для 5 ключей:\n")
print("=" * 60)

for i, key in enumerate(keys, 1):
    decrypted = vigenere_decrypt(ciphertext, key, alphabet)
    print(f"\nКлюч {i}: '{key}'")
    print("-" * 40)
    print("Полный результат:")
    print("-" * 40)
    print(decrypted)
    print("=" * 60)

print("\nВсего вариантов: 5")
```

Осмысленный текст у нас получился только при ключе – **ПУШКИН**.

```
УЛУКОМОРЬЯДУБЗЕЛЕНЫЙЗЛАТАЯЦЕПЬНАДУБЕТОМИДНЕМИНОЧЬЮКОТУЧЕНЫЙВСЁХОДИТПОЦЕПИКРУГОМИДЕ
ТНАПРАВОПЕСНЬЗАВОДИТНАЛЕВСКАЗКУГОВОРИТТАМЧУДЕСАТАМЛЕШИЙБРОДИТРУСАЛКНАВЕТВЯХСИДИТТА
МНАНЕВЕДОМЫХДОРОЖКАХСЛЕДЫНЕВИДАННЫХЗВЕРЕЙИЗБУШКАТАМНАКУРЬИХНОЖКАХСТОИТБЕЗОКОНБЕЗДВ
ЕРЕЙТАМЛЕСИДОЛВИДЕНИЙПОЛНЫТАМОЗАРЕПРИХЛЫНУТВОЛНЫНАБРЕГПЕСЧАНЫЙИПУСТОЙИТРИДЦАТЬВИТЯ
ЗЕЙПРЕКРАСНЫХЧРЕДОЙИЗВОДВЫХОДЯТЯСНЫХИСНИМИДЯДЬКАИХМОРСКОЙТАМКОРОЛЕВИЧМИМОХОДОМПЛЕН
ЯЕТГРОЗНОГОЦАРЯТАМВОБЛАКАХПЕРЕДНАРОДОМЧЕРЕЗЛЕСАЧЕРЕЗМОРЯКОЛДУННЕСЕТБОГАТЫРЯВТЕМНИЦ
ЕТАМЦАРЕВНАТУЖИТАБУРЫЙВОЛКЕЙВЕРНОСЛУЖИТТАМСТУПАСБАБОЮЯГОЙИДЕТБРЕДЕТСАМАСОБОЙТАМЦАР
ЬКАЩЕЙНАДЗЛАТОМЧАХНЕТТАМРУССКИЙДУХТАМРУСЬЮПАХНЕТИТАМЯБЫЛИМЕДЯПИЛУМОРЯВИДЕЛДУБЗЕЛЕН
ЫЙПОДНИМСИДЕЛИКОТУЧЕНЫЙСВОИМНЕСКАЗКИГОВОРИЛОДНУЯПОМНЮСКАЗКУЕТУПОВЕДАЮТЕПЕРЬЯСВЕТУД
ЕЛАДАВНОМИНУВШИХДНЕЙПРЕДАНЬЯСТАРИНЫГЛУБОКОЙВТОЛПЕМОГУЧИХСЫНОВЕЙСДРУЗЬЯМИВГРИДНИЦЕВ
ЫСОКОЙВЛАДИМИРСОЛНЦЕПИРОВАЛМЕНЬШУЮДОЧЬОНВЫДАВАЛЗАКНЯЗЯХРАБРОГОРУСЛАНАИМЕДИЗТЯЖКОГО
СТАКАНАЗАИХЗДОРОВЬЕВЫПИВАЛНЕСКОРОЕЛИПРЕДКИНАШИНЕСКОРОДВИГАЛИСЬКРУГОМКОВШИСЕРЕБРЯНЫ
ЕЧАШИСКИПЯЩИМПИВОМИВИНОМОНИВЕСЕЛЬЕВСЕРДЦЕЛИЛИШИПЕЛАПЕНАКРАЯМИХВАЖНЧАШНИКИНОСИЛИИНИ
ЗКОКЛАНЯЛИСЬГОСТЯМПРОЕКТИЗМЕНИТЬКЛЮЧИКЭТОГОШИФРАСЛИЛИСЬРЕЧИВШУМНЕВНЯТНЫЙЖУЖЖИТГОСТ
ЕЙВЕСЕЛЫЙКРУГНОВДРУГРАЗДАЛСЯГЛАСПРИЯТНЫЙИЗВОНКИХГУСЛЕЙБЕГЛЫЙЗВУКВСЕСМОЛКЛИСЛУШАЮТБ
АЯНАИСЛАВИТСЛАДОССТНЫЙПЕВЕЦЛЮДМИЛУПРЕЛЕСТИРУСЛАНАИЛЕЛЕМСВИТЫЙИМВЕНЕЦНОСТРАСТЬЮПЫЛК
ОЙУТОМЛЕННЫЙНЕЕСТНЕПЬЕТРУСЛАНВЛЮБЛЕННЫЙПАРОЛЬОТБАЗЫДАННЫХСТИХОТВОРЕНИЕНАДРУГАМИЛОГ
ОГЛЯДИТВЗДЫХАЕТСЕРДИТСЯГОРИТЩИПЛЯУСОТНЕТЕРПЕНЬЯСЧИТАЕТКАЖДЫЕМГНОВЕНЬЯВУНЬИСПАСМУ
```

РНЫМЧЕЛОМЗАШУМНЫМСВАДЕБНЫМСТОЛОМСИДЯТТРИВИТЯЗЯМЛАДЫЕБЕЗМОЛВНЫЗАКОВШОМПУСТЫМПАЗЫБИЛ
ИКУБКИКРУГОВЫЕИБРАНШАНЕПРИЯТНЫМИННЕСЛЫШАТВЕЩЕГОБАЯНАПОТУПИЛИСМУЩЕННЫЙВЗГЛЯДТОТРИСО
ПЕРНИКАРУСЛАНАВДУШЕНЕСЧАСТНЫЕТАЯТЛЮБВИИНЕНАВИСТИЯДОДИНРОГДАЙВОИТЕЛЬСМЕЛЫЙМЕЧОМРАЗД
ВИНУВШИЙПРЕДЕЛЫБОГАТЫХКИЕВСКИХПОЛЕЙДРУГОЙФАРЛАФКРИКУННАДМЕННЫЙВПИРАХНИКЕМНЕПОБЕЖДЕ
ННЫЙНОВОИНСКРОМНЫЙСРЕДЬМЕЧЕЙПОСЛЕДНИЙПОЛНЫЙСТРАСТНОЙДУМЫМЛАДОЙХАЗАРСКИЙХАНРАТМИРЫВ
СЕТРОЕБЛЕДНЫИУГРЮМЫИПИРВЕСЕЛЫЙИМНЕВПИР

Ответ: СТИХОТВОРЕНИЕ

Вариант 2

Шифр Виженера – это метод многоалфавитной подстановки, в котором используется ключевое слово, задающее последовательность сдвигов.

Принцип работы:

1. Каждой букве алфавита сопоставляется число: А=0, Б=1, В=2, … Я=32.
2. Ключ повторяется циклически, чтобы совпадать по длине с текстом.
3. Каждая буква текста сдвигается на значение соответствующей буквы ключа по модулю 33:

$$C_i = (P_i + K_i) \bmod 33$$

Где, P_i – номер буквы открытого текста,
K_i – номер буквы ключа,
C_i – номер буквы шифротекста.

Расшифровка выполняется вычитанием:

$$P_i = (C_i - K_i) \bmod 33$$

Пример:
Открытый текст – *ПРИВЕТ*
Ключ – *КЛЮЧ*
Преобразуем в числа:

П = 16	Р = 17	И = 9	В = 2	Е = 5	Т = 19
К = 11	Л = 12	Ю = 31	Ч = 24	К = 11	Л = 12

Шифруем по формуле:
(16 + 11) mod 33 = 27 → *Ъ*
(17 + 12) mod 33 = 29 → *Ь*
(9 + 31) mod 33 = 7 → *Ж*
(2 + 24) mod 33 = 26 → *Щ*
(5 + 11) mod 33 = 16 → *П*
(19 + 12) mod 33 = 31 → *Ю*

Шифротекст: *ЪЬЖЩПЮ*

Системный администратор издательства «Книжный мир» обнаружил странное вложение в письме с утверждением к печати рассказа одного из известных писателей. Ранее в Интернет выложили копию базы данных издательства с конфиденциальными данными, однако пароль от базы данных не был опубликован.

Вам необходимо подтвердить факт компрометации пароля от базы данных. Системный администратор предложил несколько ключей для расшифровки сообщения:

1.	МАРШАК

2.	ТЮТЧЕВ
3.	ЕСЕНИН
4.	ПУШКИН
5.	ОГАРЁВ

В качестве ответа предоставьте пароль от базы данных в зашифрованном сообщении.

```
ШЭШШЧЪУВБММБЁЩЙЩНЫАЫМЩИАЕРЫТШЙТСИБЙТЧАСЦМЫЙЮНЫЧЕБППЫЬБЬЦТИТПЦЧЪЬМЦЧБУДНЭНЬХБЛЬСЪИТ
ЫЫЕБХНКЬФЦЦЫЕХЕУУССАТСРТКЯПСМШЬРУУУЮСАЧССЕЬСЙГЕАИЪРЦЭЦТОХАИЦЫЮШГЕЩУЫЕУЙАКМЪГНССАЧС
СЫИЫЙУЙСЧЪАЖИЬЩЬЛЬЕГЪЩЙХАЫНПНХЕЫЦИЪЩЖТЩТОЪМОЬЁПСЧНХЫЕЬШЮЕЦЪЯУФУНЪГЧЬСАЁЦМЬУЬТТЙХМП
ЙВЙЧЫНСЭЙЯССУЭЖЦМТТЪОЭЧЩТМЧНХЬМСХТШЮНЖРИЦБЧУУЩЦИТСЁЮНРФЦЦЕИЫАЫНЭЬЯЧАОЦЫЮНХЫНЫЙЖЪЧМ
РТОБХТУЮЕГТИЮЕХЦИЬТЦМУУСКИЪАИМЫМЦЯАГСЯТЪСЦММИНПНСГСАХЯУЬОДЕЪУЬХАРТКЦЬЮНЪЧГУХУЪШЩЙЯ
ДТЫРХАМЫЧРУЗЕЮЗАЕЮЖЬЙЩЕЬЕГШТХЦИЫИЮУХУЪАТХЦМЩНЯЕИЙЮНХСАХМУЬРХШЫЦТЦЦЧОЧРЕДАЮЗПЧЦСЫСД
ЙДЕЪЯНХЦЖЫИАШШНАИОШВАЧКЬРЬЙЧКТХЯУЯФБЛЪЧАИЪЦДШЭИЯЁСЁЬЖМЗАОЦМТЧТХТМТЧГЕЪИЯУТУЧЫНСЗЕЮ
ЕШЕКЙЧЦНИЩРНЫЬСИЕГЦТЧДЕЪЩБЦГПЦТСШЖЧНХЮШГБЛШНЪЯЙАСАЕЮДОДЩНЮЙСЗЭНЭШЪЧЮДУНСНЩИЕЁХНЩЙЯ
АЧШЬИЯНЪЪЦИЦРЦУЬЧЕЬТЦИОГЖЬСЪТЦЦШИХПЪЗЬКЬХЪРЬМЫШРФЬХЫГГПНРШШЦЧБШЬЖЦИНЖАЙБЙЮЕМЦУЙАЬС
ЙЭЕСИПТАСЦЦБЖЙНГМЫЙЫФЮНСЕЯБМЪАЕВНЫДРРЕЁЬУЬОУЧЬФЭЙЮУРЬЕНЖЦИЦЬЖЦОЯМЮШЩБМХЦЖФХЦМЫНЗЙП
ДЯУЬУЧКЩЕХНЪСЮЦАРЫЯТФЪХЬКНРЮЙЫЕЁШПИЬАЙУЯЖИМНЖСРХИШТРММЮЮЕТХЬЛЬХЕЦЩИЫЕЪСТМЦМДДФУЬЗА
ЦАИШЕЯЕХИЦЪЩИЬЩЬЖНЙПДЭНУЕЩЦТЦЬУЮЧТРЪФЮНСПЪТНБЦТЦЦШЧЮУХЖЦЛНРЪЦЙУЮШФУЪУЬЖЙНЯНЮЙТХМЦИ
ЙИЕЁСЯПЪФМВЦСБНПЧЪНУНЫЧЪУЯНПНЯЙЭБТКЯЙВИДНЩНЭНЁСЭЙЭЕЭНЫЕЬХНЗЪНЖЖНПЫЬСЭЫСШНЯУЯСЩНЪТЦ
РШУЬРНЦМРЪЦЙЛЬЦДДЪШЮУЦПАСХСЦТЦЫЙПЭГЕСШВДУРЧЁНЁХНЪЩНЭНЯЕЮЙИНПБББСЯЙПЦМЧЯАЧПБЛШНАЛЬЦД
ЙЧКТЦЦРИТШХЕЗЫЧПИВШРЩНМХЕЩЪМЗЭЕЯШЮНРЧЫДЧНЩЖЬЦШНЖЗБЪЩЙЫЁТЛЩАЫМПЬШЖГЙЯХЬРЬРЦЪЩШЙЕЛЫО
ЕРТНСЯРСЖЦЫЯРСИЬЪЯЧЯАЧШТЖЦЫЩЖССЪРБШЮЙЭЙЯЫЦХЕЦЩИЫЕЪРТФТСГЖЦЫИОЪСПНЫЙЗТЬЪАХСЦАЕЛФМРШ
ЧЧШДУЪФТТЯАЧЦТЙГЧЫНЭБЦЧЮЬЯРСТПФЛЁЭЙЫЦИОЯЕСЩБЗССЦФЬЗАЗЩЗСНДЖХМИЪСЙАЪТХХНАЪМЗАХЦЫЖНБ
РМЬЯУДТТЫТХБЙЫЕМЦИНАИТЧЬЕФМИЙЮЗЫЧПЙЯБМКБТМТЙСЯФСЦЪЬЮТМСЕНЩУЮМНББСЯАЪЪПЕХЙОЦИСГЧЬФЬ
СГНСЗАЧВНПСАДШДЪФНИМЙОНХСАРПЦИМСПЬКЁУЮФБЪААЮФНРИЁЪРЦУБЁЬНШЩБЗАЖИНЦЁВЕЫБНТЦФЮСМЧЯАЪ
ШНХАРЙЧАЁСМИМНТЯАГЪЩУУУЯЧЕЙДЕЫСТНЯТТЪЩАЙЕАКТЮЦЗЬЙНДЯЕЭЧАШБНЩСЯСЕЮТЦЫАЫЖХЛШДХЧЬЫЮНГ
УЭНЮТЪПНЩБЦЭЕЫИПИЕЭТЦТЦИЕЯЫЫАЦЧНЗАРПЁПСЦТЦТНКЦЦДНММЬИЪТЮЧРИСОПЧЦЧЦРЙЪЪЙЭАЧХТЬАСЮИХ
ИУНЫЬПЭЪОЭЩТИЦРИЙЬЗСЧИЮШНЦЖЯУЦЪБУЩНЧИВШРЧЧЩСХЩИВПВНШЬЫТСИЪНЫТМОПШЦХСЪЫСШЙЮТТШЬЁЦЛС
НЫТМОЫЧПУЪТЯУЮУЮТИТЯХЦИЙХТЬЦОЭЧЯРЦИЫСЧФАРЫДЧЦДХНЪАТАОСЬЪАЮРНМЬОЖЕХИЮЦЬНЧЮНТВЕАХЦХМ
ЖЯНАХАЙОФТИЯАЦЬРХПСИСЭНВЖТЪТРМОЦХЫЙУФЦЩ
```

К задаче прилагается:
файл «message.txt»

Решение

Имеется зашифрованное сообщение, 5 вариантов ключей. Судя по названию задания это шифр Виженера. Согласно ему – каждая буква текста заменяется на другую букву, сдвинутую на количество позиций в алфавите, определяемое соответствующей буквой ключевого слова.

Соответственно, можно написать программную реализацию этого шифра.

```
def vigenere_decrypt(ciphertext, key,
alphabet="абвгдеёжзийклмнопрстуфхцчшщъыьэюя"):
    """Дешифрование шифра Виженера"""
    decrypted = []
    for i in range(len(ciphertext)):
        letter = ciphertext[i]
        if letter in alphabet:
            letter_num = alphabet.index(letter)
            key_letter = key[i % len(key)]
            key_num = alphabet.index(key_letter)
            new_num = (letter_num - key_num) % len(alphabet)
            decrypted.append(alphabet[new_num])
        else:
            decrypted.append(letter)
    return ''.join(decrypted)

# Зашифрованный текст (вставь свой)
ciphertext = "
ШЭШШЧЪУВБММБЁЩЙЩНЫАЫМЩИАЕРЫТШЙТСИБЙТЧАСЦМЫЙЮНЫЧЕБППЫЬБЬЦТИТПЦЧЪЬМЦЧБУДНЭНЬХБЛЬСЪИТ
ЫЫЕБХНКЬФЦЦЫЕХЕУУССАТСРТКЯПСМШЬРУУУЮСАЧССЕЬСЙГЕАИЪРЦЭЦТОХАИЦЫЮШГЕЩУЫЕУЙАКМЪГНССАЧС
СЫИЫЙУЙСЧЪАЖИЬЩЬЛЬЕГЪЩЙХАЫНПНХЕЫЦИЪЩЖТЩТОЪМОЬЁПСЧНХЫЕЬШЮЕЦЪЯУФУНЪГЧЬСАЁЦМЬУЬТТЙХМП
```

```
ЙВЙЧЫНСЭЙЯССУЭЖЦМТТЪОЭЧЩТМЧНХЬМСХТШЮНЖРИЦБЧУУЩЦИТСЁЮНРФЦЦЕИЫАЫНЭЬЯЧАОЦЫЮНХЫНЫЙЖЪЧМ
РТОБХТУЮЕГТИЮЕХЦИЬТЦМУУСКИЪАИМЫМЦЯАГСЯТЪСЦММИНПНСГСАХЯУЬОДЕЪУЬХАРТКЦЬЮНЪЧГУХУЪШЩЙЯ
ДТЫРХАМЫЧРУЗЕЮЗАЕЮЖЬЙЩЕЬЕГШТХЦИЫИЮУХУЪАТХЦМЩНЯЕИЙЮНХСАХМУЬРХШЫЦТЦЦЧОЧРЕДАЮЗПЧЦСЫСД
ЙДЕЪЯНХЦЖЫИАШШНАИОШВАЧКЬРЬЙЧКТХЯУЯФБЛЪЧАИЪЦДШЭИЯЁСЁЬЖМЗАОЦМТЧТХТМТЧГЕЪИЯУТУЧЫНСЗЕЮ
ЕШЕКЙЧЦНИЩРНЫЬСИЕГЦТЧДЕЪЩБЦГПЦТСШЖЧНХЮШГБЛШНЪЯЙАСАЕЮДОДЩНЮЙСЗЭНЭШЪЧЮДУНСНЩИЕЁХНЩЙЯ
АЧШЬИЯНЪЪЦИЦРЦУЬЧЕЬТЦИОГЖЬСЪТЦЦШИХПЪЗЬКЬХЪРЬМЫШРФЬХЫГГПНРШШЦЧБШЬЖЦИНЖАЙБЙЮЕМЦУЙАЬС
ЙЭЕСИПТАСЦЦБЖЙНГМЫЙЫФЮНСЕЯБМЪАЕВНЫДРРЕЁЬУЬОУЧЬФЭЙЮУРЬЕНЖЦИЦЬЖЦОЯМЮШЩБМХЦЖФХЦМЫНЗЙП
ДЯУЬУЧКЩЕХНЪСЮЦАРЫЯТФЪХЬКНРЮЙЫЕЁШПИЬАЙУЯЖИМНЖСРХИШТРММЮЮЕТХЬЛЬХЕЦЩИЫЕЪСТМЦМДДФУЬЗА
ЦАИШЕЯЕХИЦЪЩИЬЩЬЖНЙПДЭНУЕЩЦТЦЬУЮЧТРЪФЮНСПЪТНБЦТЦЦШЧЮУХЖЦЛНРЪЦЙУЮШФУЪУЬЖЙНЯНЮЙТХМЦИ
ЙИЕЁСЯПЪФМВЦСБНПЧЪНУНЫЧЪУЯНПНЯЙЭБТКЯЙВИДНЩНЭНЁСЭЙЭЕЭНЫЕЬХНЗЪНЖЖНПЫЬСЭЫСШНЯУЯСЩНЪТЦ
РШУЬРНЦМРЪЦЙЛЬЦДДЪШЮУЦПАСХСЦТЦЫЙПЭГЕСШВДУРЧЁНЁХНЪЩНЭНЯЕЮЙИНПББСЯЙПЦМЧЯАЧПБЛШНАЛЬЦД
ЙЧКТЦЦРИТШХЕЗЫЧПИВШРЩНМХЕЩЪМЗЭЕЯШЮНРЧЫДЧНЩЖЬЦШНЖЗБЪЩЙЫЁТЛЩАЫМПЬШЖГЙЯХЬРЬРЦЪЩШЙЕЛЫО
ЕРТНСЯРСЖЦЫЯРСИЬЪЯЧЯАЧШТЖЦЫЩЖССЪРБШЮЙЭЙЯЫЦХЕЦЩИЫЕЪРТФТСГЖЦЫИОЪСПНЫЙЗТЬЪАХСЦАЕЛФМРШ
ЧЧШДУЪФТТЯАЧЦТЙГЧЫНЭБЦЧЮЬЯРСТПФЛЁЭЙЫЦИОЯЕСЩБЗССЦФЬЗАЗЩЗСНДЖХМИЪСЙАЪТХХНАЪМЗАХЦЫЖНБ
РМЬЯУДТТЫТХБЙЫЕМЦИНАИТЧЬЕФМИЙЮЗЫЧПЙЯБМКБТМТЙСЯФСЦЪЬЮТМСЕНЩУЮМНББСЯАЪЪПЕХЙОЦИСГЧЬФЬ
СГНСЗАЧВНПСАДШДЪФНИМЙОНХСАРПЦИМСПЬКЁУЮФБЪААЮФНРИЁЪРЦУБЁЬНШЩБЗАЖИНЦЁВЕЫБНТЦФЮСМЧЯАЪ
ШНХАРЙЧАЁСМИМНТЯАГЪЩУУУЯЧЕЙДЕЫСТНЯТТЪЩАЙЕАКТЮЦЗЬЙНДЯЕЭЧАШБНЩСЯСЕЮТЦЫАЫЖХЛШДХЧЬЫЮНГ
УЭНЮТЪПНЩБЦЭЕЫИПИЕЭТЦТЦИЕЯЫЫАЦЧНЗАРПЁПСЦТЦТНКЦЦДНММЬИЪТЮЧРИСОПЧЦЧЦРЙЪЪЙЭАЧХТЬАСЮИХ
ИУНЫЬПЭЪОЭЩТИЦРИЙЬЗСЧИЮШНЦЖЯУЦЪБУЩНЧИВШРЧЧЩСХЩИВПВНШЬЫТСИЪНЫТМОПШЦХСЪЫСШЙЮТТШЬЁЦЛС
НЫТМОЫЧПУЪТЯУЮУЮТИТЯХЦИЙХТЬЦОЭЧЯРЦИЫСЧФАРЫДЧЦДХНЪАТАОСЬЪАЮРНМЬОЖЕХИЮЦЬНЧЮНТВЕАХЦХМ
ЖЯНАХАЙОФТИЯАЦЬРХПСИСЭНВЖТЪТРМОЦХЫЙУФЦЩ"

# 5 ключей
keys = [
    "ПУШКИН",
    "ТЮТЧЕВ",
    "ЕСЕНИН",
    "МАРШАК",
    "ОГАРЁВ"
]
alphabet = "АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"
print("Расшифровка для 5 ключей:\n")
print("=" * 60)
for i, key in enumerate(keys, 1):
    decrypted = vigenere_decrypt(ciphertext, key, alphabet)
    print(f"\nКлюч {i}: '{key}'")
    print("-" * 40)
    print("Полный результат:")
    print("-" * 40)
    print(decrypted)
    print("=" * 60)

print("\nВсего вариантов: 5")
```

Осмысленный текст у нас получился только при ключе – *ЕСЕНИН*.

```
УЛУКОМОРЬЯДУБЗЕЛЕНЫЙЗЛАТАЯЦЕПЬНАДУБЕТОМИДНЕМИНОЧЬЮКОТУЧЕНЫЙВСЁХОДИТПОЦЕПИКРУГОМИДЕ
ТНАПРАВОПЕСНЬЗАВОДИТНАЛЕВСКАЗКУГОВОРИТТАМЧУДЕСАТАМЛЕШИЙБРОДИТРУСАЛКНАВЕТВЯХСИДИТТА
МНАНЕВЕДОМЫХДОРОЖКАХСЛЕДЫНЕВИДАННЫХЗВЕРЕЙИЗБУШКАТАМНАКУРЬИХНОЖКАХСТОИТБЕЗОКОНБЕЗДВ
ЕРЕЙТАМЛЕСИДОЛВИДЕНИЙПОЛНЫТАМОЗАРЕПРИХЛЫНУТВОЛНЫНАБРЕГПЕСЧАНЫЙИПУСТОЙИТРИДЦАТЬВИТЯ
ЗЕЙПРЕКРАСНЫХЧРЕДОЙИЗВОДВЫХОДЯТЯСНЫХИСНИМИДЯДЬКАИХМОРСКОЙТАМКОРОЛЕВИЧМИМОХОДОМПЛЕН
ЯЕТГРОЗНОГОЦАРЯТАМВОБЛАКАХПЕРЕДНАРОДОМЧЕРЕЗЛЕСАЧЕРЕЗМОРЯКОЛДУННЕСЕТБОГАТЫРЯВТЕМНИЦ
ЕТАМЦАРЕВНАТУЖИТАБУРЫЙВОЛКЕЙВЕРНОСЛУЖИТТАМСТУПАСБАБОЮЯГОЙИДЕТБРЕДЕТСАМАСОБОЙТАМЦАР
ЬКАЩЕЙНАДЗЛАТОМЧАХНЕТТАМРУССКИЙДУХТАМРУСЬЮПАХНЕТИТАМЯБЫЛИМЕДЯПИЛУМОРЯВИДЕЛДУБЗЕЛЕН
ЫЙПОДНИМСИДЕЛИКОТУЧЕНЫЙСВОИМНЕСКАЗКИГОВОРИЛОДНУЯПОМНЮСКАЗКУЕТУПОВЕДАЮТЕПЕРЬЯСВЕТУД
ЕЛАДАВНОМИНУВШИХДНЕЙПРЕДАНЬЯСТАРИНЫГЛУБОКОЙВТОЛПЕМОГУЧИХСЫНОВЕЙСДРУЗЬЯМИВГРИДНИЦЕВ
ЫСОКОЙВЛАДИМИРСОЛНЦЕПИРОВАЛМЕНЬШУЮДОЧЬОНВЫДАВАЛЗАКНЯЗЯХРАБРОГОРУСЛАНАИМЕДИЗТЯЖКОГО
СТАКАНАЗАИХЗДОРОВЬЕВЫПИВАЛНЕСКОРОЕЛИПРЕДКИНАШИНЕСКОРОДВИГАЛИСЬКРУГОМКОВШИСЕРЕБРЯНЫ
ЕЧАШИСКИПЯЩИМПИВОМИВИНОМОНИВЕСЕЛЬЕВСЕРДЦЕЛИЛИШИПЕЛАПЕНАКРАЯМИХВАЖНЧАШНИКИНОСИЛИИНИ
ЗКОКЛАНЯЛИСЬГОСТЯМПРОЕКТИЗМЕНИТЬКЛЮЧИКЭТОГОШИФРАСЛИЛИСЬРЕЧИВШУМНЕВНЯТНЫЙЖУЖЖИТГОСТ
ЕЙВЕСЕЛЫЙКРУГНОВДРУГРАЗДАЛСЯГЛАСПРИЯТНЫЙИЗВОНКИХГУСЛЕЙБЕГЛЫЙЗВУКВСЕСМОЛКЛИСЛУШАЮТБ
АЯНАИСЛАВИТСЛАДОССТНЫЙПЕВЕЦЛЮДМИЛУПРЕЛЕСТИРУСЛАНАИЛЕЛЕМСВИТЫЙИМВЕНЕЦНОСТРАСТЬЮПЫЛК
ОЙУТОМЛЕННЫЙНЕЕСТНЕПЬЕТРУСЛАНВЛЮБЛЕННЫЙНАДРУГАМИЛОГОГЛЯДИТВЗДЫХАЕТСЕРДИТСЯГОРИТЩИП
```

ЛЯУСОТНЕТЕРПЕНЬЯСЧИТАЕТКАЖДЫЕМГНОВЕНЬЯВУНЫНЬИСПАСМУРНЫМЧЕЛОМЗАШУМНЫМСВАДЕБНЫМСТОЛО
МСИДЯТТРИВИТЯЗЯМЛАДЫЕБЕЗМОЛВНЫЗАКОВШОМПУСТЫМПАЗЫБИЛИКУБКИКРУГОВЫЕИБРАНШАНЕПРИЯТНЫМ
ПАРОЛЬОТБАЗЫДАННЫХСЛОВОСОЧЕТАНИЕИННЕСЛЫШАТВЕЩЕГОБАЯНАПОТУПИЛИСМУЩЕННЫЙВЗГЛЯДТОТРИС
ОПЕРНИКАРУСЛАНАВДУШЕНЕСЧАСТНЫЕТАЯТЛЮБВИИНЕНАВИСТИЯДОДИНРОГДАЙВОИТЕЛЬСМЕЛЫЙМЕЧОМРАЗ
ДВИНУВШИЙПРЕДЕЛЫБОГАТЫХКИЕВСКИХПОЛЕЙДРУГОЙФАРЛАФКРИКУННАДМЕННЫЙВПИРАХНИКЕМНЕПОБЕЖД
ЕННЫЙНОВОИНСКРОМНЫЙСРЕДЬМЕЧЕЙПОСЛЕДНИЙПОЛНЫЙСТРАСТНОЙДУМЫМЛАДОЙХАЗАРСКИЙХАНРАТМИРЫ
ВСЕТРОЕБЛЕДНЫИУГРЮМЫИПИРВЕСЕЛЫЙИМНЕВПИР

Тут мы можем увидеть пароль от базы данных – **СЛОВОСОЧЕТАНИЕ**.

Ответ: СЛОВОСОЧЕТАНИЕ