

**Муниципальный этап всероссийской олимпиады школьников  
по информатике профиль «Информационная безопасность»  
в Красноярском крае в 2025/2026 учебном году**

**Задания и решения**

9 класс

(Время выполнения – 210 мин. Максимальная оценка – 70 баллов)

**ЗАДАНИЕ №1. (0 / 3 баллов)**

Администратором сети в терминале Linux введена следующая команда:  
`tracert -q 2 ya.ru`

Каков результат ее выполнения?

*Варианты ответа:*

- Программа завершит работу после двух прыжков.
- **Для каждого промежуточного узла будет отправлено по два ICMP-запроса.**
- Максимальное время ожидания ответа составит 2 секунды.
- Используется два параллельных потока для ускорения трассировки.

Правильный ответ: Для каждого промежуточного узла будет отправлено по два ICMP-запроса.

Пояснение: Ключ `-q N` в команде `tracert` указывает количество запросов (probes), отправляемых на каждый промежуточный узел (hop). По умолчанию это значение часто равно 3. Если указано `-q 2`, то на каждый хоп будет отправлено по два пакета, что позволяет оценить стабильность задержки (RTT) и наличие потерь.

**ЗАДАНИЕ №2. (0 / 3 баллов)**

В базе данных пароль хранится в виде 5f4dcc3b5aa765d61d8327deb882cf99. Что это за способ защиты?

*Варианты ответа:*

- Шифрование с использованием алгоритма AES.
- **Хэширование с помощью алгоритма MD5.**
- Кодирование в формате Base64.
- Симметричное шифрование с ключом.

Правильный ответ: Хэширование с помощью алгоритма MD5. Хранение паролей в виде хэшей (а не в открытом виде) — базовая мера защиты: даже при утечке базы злоумышленник не получает исходные пароли.

**ЗАДАНИЕ №3. (0 / 1,5 / 3 баллов)**

Несколько тысяч скомпрометированных устройств одновременно отправляют трафик на сервер, и он перестаёт отвечать легитимным пользователям. Как называется такая атака?

*Варианты ответа:*

- Фишинг.
- ARP-спуфинг.
- **DDoS.**
- **Распределённый отказ в обслуживании.**

Правильные ответы: DDoS (распределённый отказ в обслуживании) — перегрузка ресурса большим объёмом трафика из множества источников.

#### ЗАДАНИЕ №4. (0 / 1,5 / 3 баллов)

В строке запроса веб-формы используется следующая конструкция SQL-запроса:

```
... WHERE username = '$user' AND password = '$pass'.
```

Злоумышленник в поле ввода имени пользователя ввёл значение:

```
' OR '1'='1
```

и получил доступ к учётной записи без знания пароля.

Как называется такая уязвимость?

*Варианты ответа:*

- Межсайтовый скриптинг.
- **SQL-инъекция.**
- XSS.
- **SQLi.**

Правильные ответы: SQL-инъекция; SQLi.

Пояснение к ответам:

- **SQL-инъекция** — полное и корректное название уязвимости, при которой вредоносный SQL-код внедряется в запрос через неправильно обработанные пользовательские данные.
- **SQLi** — общепринятое сокращение от SQL injection, широко используемое в сообществе ИБ и допустимое в олимпиадных заданиях.
- **Межсайтовый скриптинг** и **XSS** — это одно и то же, но относится к выполнению JavaScript в браузере, а не к манипуляции SQL-запросами.

#### ЗАДАНИЕ №5. (0 / 1,5 / 3 баллов)

Какой механизм операционной системы ограничивает запуск неизвестных или непроверенных программ для предотвращения заражения?

*Варианты ответа:*

- Антивирусная программа.
- **Контроль учётных записей (UAC).**
- Защитник Windows (Windows Defender).
- **Песочница (Sandbox).**

Правильный ответ: «Контроль учётных записей (UAC)» и «Песочница (Sandbox)».

Пояснение к ответу:

- **Контроль учётных записей (UAC)** – встроенная функция Windows, которая запрашивает разрешение администратора при попытке запуска программ, вносящих изменения в систему. Это препятствует автоматическому запуску вредоносного ПО без ведома пользователя.
- **Песочница (Sandbox)** – механизм изоляции программного обеспечения, позволяющий запускать подозрительные приложения в ограниченной среде без доступа к основной системе. Используется как в ОС (например, Windows Sandbox), так и в браузерах.

### ЗАДАНИЕ №6. (0 / 5 баллов)

Сеть с IP-адресом 192.168.1.1 и маской подсети 255.255.255.0 записывается в формате CIDR как 192.168.1.1/X.

Чему равно значение X?

Формат ответа: запишите целое число X без дополнительных символов, пробелов или пояснений.

Правильный ответ: 24

Возможный ход решения:

Маска подсети 255.255.255.0 в двоичном виде выглядит так:

11111111.11111111.11111111.00000000

Количество единиц в маске – 24.

В CIDR-нотации значение после косой черты (/X) как раз и обозначает число старших единиц в маске подсети. Следовательно,  $X = 24$ .

### ЗАДАНИЕ №7 (0 / 10 баллов)

Найдите наибольшее четырёхзначное число, которое в 135 раз больше суммы своих цифр.

Правильный ответ: 1215

Возможный ход решения:

Требуется найти наибольшее четырёхзначное число  $N$ , для которого выполняется условие:

$$N = 135 \cdot S,$$

где  $S$  – сумма цифр числа  $N$ .

Напишем программу, которая перебирает все четырёхзначные числа в порядке убывания (от 9999 до 1000). Для каждого числа вычисляется сумма его цифр, после чего проверяется выполнение условия  $N = 135 \cdot S$ . Как только такое число найдено, оно выводится, и программа завершается — таким образом гарантируется, что найдено наибольшее подходящее число.

Решение на языке Python

```
# Перебираем все четырёхзначные числа от 9999 вниз до 1000
for n in range(9999, 999, -1):
    # Вычисляем сумму цифр числа
    s = sum(int(digit) for digit in str(n))
    # Проверяем условие
    if n == 135 * s:
        print(n)
        break
```

### ЗАДАНИЕ №8 (0 / 10 баллов)

Злоумышленник хочет получить доступ к сейфу, который защищён кодовым замком. Кодом является комбинация из трёх цифр, которые нужно нажать одновременно (Порядок не важен). На ввод одной комбинации уходит 1 секунда. Посчитайте, сколько времени потребуется для перебора всех возможных комбинаций. Ответ запишите в секундах.

Правильный ответ: 120.

Возможный ход решения:

У нас 10 цифр (от 0 до 9). Если бы порядок имел значение, было бы  $10 \cdot 9 \cdot 8 = 720$  вариантов. Но так как кнопки нажимаются одновременно, то, например, комбинации 1-2-3, 2-3-1 и 3-1-2 — это одно и то же.

Из трёх разных цифр можно составить ровно 6 разных порядков (потому что: первая — 3 варианта, вторая — 2, третья — 1  $\rightarrow 3 \cdot 2 \cdot 1 = 6$ ).

Значит, чтобы получить число настоящих комбинаций, нужно 720 разделить на 6:

$$720 : 6 = 120$$

На каждую комбинацию уходит 1 секунда  $\rightarrow$  всего нужно **120 секунд**.

### ЗАДАНИЕ №9 (0 / 15 баллов)

Предоставлен **Python-скрипт** (encode.py), реализующий алгоритм кодирования флага, а также **строка** (data.txt) с закодированным значением.

Требуется инвертировать алгоритм — восстановить флаг.

**Формат флага**

vsosh{...}

**Критерии приёма**

Ответом является флаг. Флаг принимается, если отправленная строка полностью совпадает с эталоном (включая регистр, фигурные скобки и длину). Пробелы и переносы недопустимы.

Правильный ответ: vsosh{c4n\_y0u\_s0lv3\_m3}

Возможный вариант программы декодирования на языке Python: Приложение № 1.

## ЗАДАНИЕ №10 (0 / 15 баллов)

В архиве (logs.7z) находится множество небольших текстовых файлов с записями. Почти все строки состоят из случайных шестнадцатеричных наборов символов, но в одном из файлов спрятана необычная запись, отличающаяся по структуре. Найдите аномалию и определите значение скрытого флага.

### Формат флага

forensic{<64-символьная\_hex-строка>}

### Критерии приёма

Ответом является флаг. Флаг принимается, если отправленная строка полностью совпадает с эталоном (включая регистр, фигурные скобки и длину). Пробелы и переносы недопустимы.

Правильный ответ: forensic{281c8af49dede2798a4885bd90e532a4365c14afec62159e762baed7292f7c41}

### Возможный ход решения:

В коллекции файлов необходимо идентифицировать строку, содержащую маркер forensic{...} либо метку meta:, и извлечь токен флага по регулярному выражению.

```
#!/usr/bin/env bash
set -euo pipefail
ROOT="forensic-a"
# 1) быстрый поиск строки с флагом
grep -R --line-number --text "forensic{" "$ROOT" || true
# 2) альтернатива через meta:
grep -R --line-number --text "^meta:" "$ROOT" || true
# 3) чистая экстракция токена флага (если встречается вхождение)
find "$ROOT" -type f -print0 \
| xargs -0 grep -aNo "forensic{[0-9a-fA-F]\{64\}}" 2>/dev/null || true
```

```
1  -*- coding: utf-8 -*-
2  #!/usr/bin/env python3
3
4  import sys
5  import base64
6
7  def rol(b, r):
8      r %= 8
9      return ((b << r) | (b >> (8 - r))) & 0xFF
10 def ror(b, r):
11     r %= 8
12     return ((b >> r) | ((b << (8 - r)) & 0xFF)) & 0xFF
13 def lcg32(x):
14     return (1103515245 * x + 12345) & 0xFFFFFFFF
15
16 def keystream_byte(i, seed=0xC0FFEE):
17     v = lcg32(i ^ seed)
18     return (v >> 16) & 0xFF
19
20 def decode(b64: str) -> bytes:
21     b64 = b64.strip()
22     data = base64.b64decode(b64)
23     out = bytearray()
24     for i, v in enumerate(data):
25         v = (v - (i % 5)) & 0xFF
26         v = ror(v, 3)
27         k = keystream_byte(i)
28         b = v ^ k
29         out.append(b)
30     return bytes(out)
31
32 def main():
33     if len(sys.argv) >= 2:
34         encoded = sys.argv[1]
35     else:
36         encoded = sys.stdin.read().strip()
37     if not encoded:
38         return
39     try:
40         decoded_bytes = decode(encoded)
41     except Exception as e:
42         print("Ошибка при декодировании:", e, file=sys.stderr)
43         sys.exit(1)
44
45     try:
46         decoded_text = decoded_bytes.decode('utf-8')
47     except UnicodeDecodeError:
48         decoded_text = decoded_bytes.decode('utf-8', errors='replace')
49
50     print(decoded_text)
51
52 if __name__ == "__main__":
53     main()
```