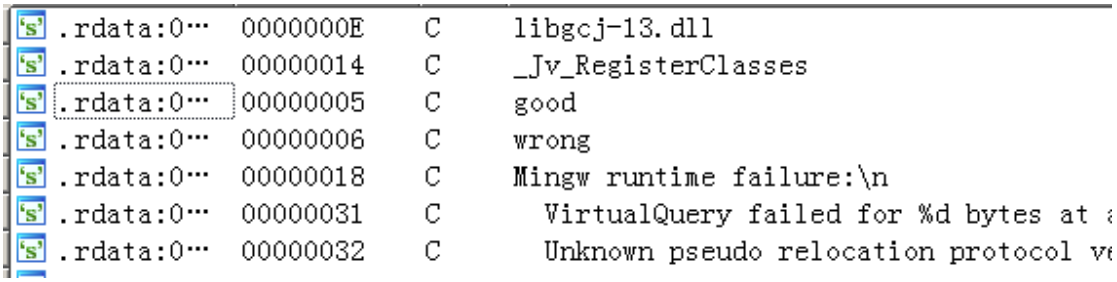
10000000 Write UP

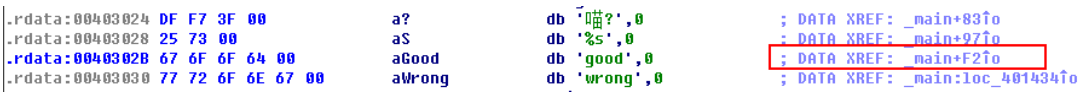
沉迷迷路不能自拔，抬头一看发现时间没了…赶紧做个水题压压惊

这个题目界面是这样的：

算你有良心留了个字符串…不过也无所谓

打开 IDA，直接 Shift+F12 进入字符串界面：





在红框的位置右键寻找到它被引用的位置，F5 开始反编译至伪 C，发现代码不完整，于是再稍微往上找一找，找到了这段代码的完整版：

```
int __cdecl main()
{
  char *v0; // edi@1
  signed int i; // ecx@1
  char *v2; // edi@4
  signed int j; // ecx@4
  char v5; // [sp+14h] [bp-34h]@4
  char v6; // [sp+15h] [bp-33h]@7
  char v7; // [sp+16h] [bp-32h]@7
  char v8; // [sp+17h] [bp-31h]@7
  char v9; // [sp+18h] [bp-30h]@7
  char v10; // [sp+19h] [bp-2Fh]@7
  char v11; // [sp+1Ah] [bp-2Eh]@7
  char v12; // [sp+1Bh] [bp-2Dh]@7
  char v13; // [sp+1Ch] [bp-2Ch]@7
  char v14; // [sp+1Dh] [bp-2Bh]@7
  char v15; // [sp+1Eh] [bp-2Ah]@7
  char v16; // [sp+1Fh] [bp-29h]@7
  char v17; // [sp+20h] [bp-28h]@7
  int v18; // [sp+28h] [bp-20h]@1
  char v19; // [sp+2Ch] [bp-1Ch]@1
  int v20; // [sp+3Ch] [bp-Ch]@7

  __main();
  v18 = 0;
  v0 = &v19;
  for ( i = 16; i; --i )
```

```
  *v0++ = 0;
v2 = &v5;
for ( j = 20; j; --j )
  *v2++ = 0;
v5 = -26;
v6 = -20;
v7 = -31;
v8 = -25;
v9 = -70;
v10 = -12;
v11 = -27;
v12 = -13;
v13 = -12;
v14 = -12;
v15 = -27;
v16 = -13;
v17 = -12;
v20 = 0;
puts("喵?");
scanf("%s", &v18);
LOBYTE(v20) = 0;
while ( *((_BYTE *)&v18 + v20) )
  *((_BYTE *)&v18 + v20++) |= 0x80u;
if ( strcmp((const char *)&v18, &v5) )
  printf("wrong");
else
  printf("good");
```

（最后还有个 return 0，不截了）

代码很明显了：输入的字符串（v18）逐位与 0x80u（u 应该是 unsigned）做或运算，结果再与 v5 做比较，相同则 good，否则 wrong

那么问题来了：v5 是什么？

在 java 里先试了一下这里的负数组成的字符串去运算（在网上查找了一下，决定试试与运算和异或运算），发现是乱码，不科学，于是回头去找汇编码：

```
.
mov     [esp+48h+var_34], 0E6h
mov     [esp+48h+var_33], 0ECh
mov     [esp+48h+var_32], 0E1h
mov     [esp+48h+var_31], 0E7h
mov     [esp+48h+var_30], 0BAh
mov     [esp+48h+var_2F], 0F4h
mov     [esp+48h+var_2E], 0E5h
mov     [esp+48h+var_2D], 0F3h
mov     [esp+48h+var_2C], 0F4h
mov     [esp+48h+var_2B], 0F4h
mov     [esp+48h+var_2A], 0E5h
mov     [esp+48h+var_29], 0F3h
mov     [esp+48h+var_28], 0F4h
```

应该就是这个了。h 表示是十六进制。

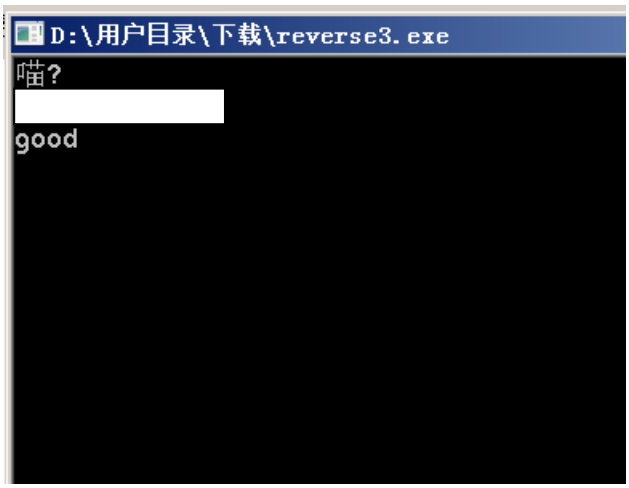
```java
public static void main(String[] args) {
    char[] v = new char[13];
    v[0] = (char) 0x0e6;
    v[1] = (char) 0x0ec;
    v[2] = (char) 0x0e1;
    v[3] = (char) 0x0e7;
    v[4] = (char) 0x0ba;
    v[5] = (char) 0x0f4;
    v[6] = (char) 0x0e5;
    v[7] = (char) 0x0f3;
    v[8] = (char) 0x0f4;
    v[9] = (char) 0x0f4;
    v[10]= (char) 0x0e5;
    v[11]= (char) 0x0f3;
    v[12]= (char) 0x0f4;
    char[] s = new char[13];
    for(int i=0; i<13; i++)
        s[i] = (char) (v[i] & 0x80);
    System.out.println(s);
    for(int i=0; i<13; i++)
        s[i] = (char) (v[i] ^ 0x80);
    System.out.println(s);
}
```

运行结果：

```
????????????
```

? ? ? ? ? ? ? ? ? ? ? ? ?

把看起来比较正常的那个结果输入到程序，看到程序闪过一个 good，那应该就是了（闪退的控制台截图不易啊啊啊）



输入结果：

？？？？？？？？？？？？？？？？？？？？？？

提交的时候，注意删掉flag:

卦星

1年前 回复

首页 上一页 3 4 5 6 跳转： 页 确定

这 就很尴尬了