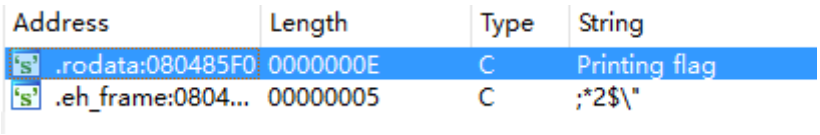
本题是一道 32 位 elf 文件，这次我们用 ida 打开，使用快捷键 shift+F12，可以查看所有字符串，我们看到结果如下：



Address	Length	Type	String
.rodata:080485F0	0000000E	C	Printing flag
.eh_frame:0804...	00000005	C	;*2$\"

转到“Printing flag”所在位置，鼠标右键，选择“Create function”，使用 Tab 键，得到结果如下：

```
int sub_804849B()
{
  signed int i; // [sp+Ch] [bp-2Ch]@1
  unsigned int j; // [sp+10h] [bp-28h]@2
  int v3; // [sp+18h] [bp-20h]@1
  int v4; // [sp+1Ch] [bp-1Ch]@1
  int v5; // [sp+20h] [bp-18h]@1
  int v6; // [sp+24h] [bp-14h]@1
  unsigned int v7; // [sp+28h] [bp-10h]@1
  int v8; // [sp+2Ch] [bp-Ch]@1

  v8 = *MK_FP(__GS__, 20);
  puts("Printing flag");
  v3 = 0x1686F596;
  v4 = 0x5646F537;
  v5 = 0x76765726;
  v6 = 0x37F52756;
  v7 = 0xC6C696B6;
  for ( i = 0; i <= 4; ++i )
  {
    for ( j = *(&v3 + i); j; j >>= 8 )
      putchar((char)(((unsigned __int8)j >> 4) | 16 * j));
  }
  putchar(10);
  return *MK_FP(__GS__, 20) ^ v8;
}
```

这下就很明显了，从 v3 地址开始，每次取一个字节，并将字节的高低位互换，然后输出，使用 python 脚本爆破，即可，得到结果为：“i_has_debugger_skill”。