

1. 停机问题的证明定义在没有输入的函数上，能否改成在带输入的函数上？注意这时 `halt(p, i)` 函数接受两个参数，其中 `i` 是输入  
定义一个邪恶程序：

```
1 void Evil() {
2     if (!Halt(Evil, Evil)) return;
3     else while 1;
4 }
```

分析Evil的行为即可知，Halt函数在输入Evil情况下返回真，那么程序进入无限循环，但这与Evil停机相矛盾；反之如果Halt函数在输入Evil下返回假，则Evil本应无限循环，但却返回，导致悖论。

2. 假设我们把符号分析的抽象域改成{自然数、负、𐀀}三个值，其中自然数表示所有正数和零，请写出加法和除法的计算规则，并给出一个式子，在该抽象域上得到的结果不如原始分析精确。

运算 +/÷	自然数	负	𐀀
自然数	自然数/自然数	𐀀/负	𐀀/𐀀
负	𐀀/负	负/自然数	𐀀/𐀀
𐀀	𐀀/𐀀	𐀀/𐀀	𐀀/𐀀

给出式子：

- 加法： `1 + (-2) = -1` → 自然数 + 负 = 𐀀
- 除法是准确的（根据要求的确保除数不为0、且整除的情况下）