

题目描述：

已知火星使用的运算符为#、\$，其与地球人的等价公式如下：

$$x\#y = 4*x+3*y+2$$

$$x\$y = 2*x+y+3$$

- 1、其中 x 、 y 是无符号整数
- 2、地球人公式按 C 语言规则计算
- 3、火星公式中，#的优先级高于\$，相同的运算符，按从左到右的顺序计算

现有一段火星人的字符串报文，请你来翻译并计算结果。

输入描述：

火星字符串表达式（结尾不带回车换行）

输入的字符串说明： 字符串为仅由无符号整数和操作符（#、\$）组成的计算表达式。例如：**123#4\$5#67\$78**。

- 1、用例保证字符串中，操作数与操作符之间没有任何分隔符。
- 2、用例保证操作数取值范围为 **32** 位无符号整数。
- 3、保证输入以及计算结果不会出现整型溢出。
- 4、保证输入的字符串为合法的求值报文，例如：**123#4\$5#67\$78**
- 5、保证不会出现非法的求值报文，例如类似这样字符串：

#4\$5 //缺少操作数

4\$5# //缺少操作数

4# \$5 //缺少操作数

4 \$5 //有空格

3+4-5*6/7 //有其它操作符

12345678987654321\$54321 //32 位整数计算溢出

输出描述:

根据输入的火星人字符串输出计算结果（结尾不带回车换行）

示例 1

输入:

7#6\$5#12

输出:

157

说明:

示例:

7#6\$5#12

$= (4 * 7 + 3 * 6 + 2) \$ 5 \# 12$

$= 48 \$ 5 \# 12$

$= 48 \$ (4 * 5 + 3 * 12 + 2)$

$= 48 \$ 58$

$= 2 * 48 + 58 + 3$

$= 157$

```
import copy
```

```
def cal_fir(st):  
    lis = st.split("#")  
    pr = int(lis[0])  
    for i in range(1, len(lis)):  
        pr = 4 * pr + 3 * int(lis[i]) + 2  
    return pr
```

```
exp = input()
new_exps = exp.split("$")
li = copy.deepcopy(new_exps)
for i, tmp in enumerate(new_exps):
    if "#" in tmp:
        li[i] = cal_fir(tmp)
pre = int(li[0])
for i in range(1, len(li)):
    pre = 2 * pre + int(li[i]) + 3
print(pre)
```