

## 一、编程题

### ACM： 火星文计算

#### 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运算符（#、\$）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：根据输入的火星字符串输出计算结果（结尾不带回车换行）

#### 补充说明：

##### 示例1

输入：7#6\$5#12

输出：157

说明：示例：

$$7\#6\$5\#12$$

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

$$=48\$5\#12$$

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

$$=48\$58$$

$$=2*48+58+3$$

$$=157$$

代码：

```
import java.io.*;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;
import java.util.*;
import java.util.Arrays;
```

```

public class Main {
    public static void main(String[] args) throws IOException {
        BufferedReader bufR = new BufferedReader(new InputStreamReader(System.in));

        String strIn;
        while ( (strIn = bufR.readLine()) != null ) {
            System.out.println( getValueOfExpressionHX(strIn) );
        }

        bufR.close();
    }
    // 返回火星文表达式的值
    private static int getValueOfExpressionHX(String str_expresssion) {
        int theResult = Integer.MIN_VALUE;

        int lastIndex_dollar = str_expresssion.lastIndexOf("$");
        if (lastIndex_dollar >= 0) {
            String str_befor = str_expresssion.substring(0, lastIndex_dollar);
            String str_after = str_expresssion.substring(lastIndex_dollar+1);
            theResult      =      (2      *      getValueOfExpressionHX(str_befor))      +
            getValueOfExpressionHX(str_after) + 3;
        } else {
            int lastIndex_jing = str_expresssion.lastIndexOf("#");
            if (lastIndex_jing >= 0) {
                String str_befor = str_expresssion.substring(0, lastIndex_jing);
                String str_after = str_expresssion.substring(lastIndex_jing+1);
                theResult      =      (4      *      getValueOfExpressionHX(str_befor))      +      (3      *
            getValueOfExpressionHX(str_after)) + 2;
            } else {
                theResult = Integer.parseInt(str_expresssion);
            }
        }

        return (theResult);
    }
}

```