（ps：所有的问题都应该能够用自己的语言来进行描述，就是自己的重新组织能力，对问题的准确描述，是解决问题的第一步）

Monty Hall问题

这个是关于概率的问题~

3个门中有1个是有礼物的，另外两个有羊，当你选择1个门之后，主人会把另外一个没有礼物的门打开，这时候你可以选择换门，是否该换呢？理论上来说，换门之后选中礼物的概率由原来的1/2变成2/3,虽然到现在还没有真正明白为什么？

问题的难点其实还在于最后的概率计算，要求的是整数，但是返回的一直一般来说应该是小数，是向上取整还是向下取整呢？

答案为什么是向上取整而不是向下取整呢

LongestConsec 问题

这个问题是寻找字符串数组中连续K个字符串的中最长的那个

Testing 1-2-3

原来最大的问题是我观察不仔细，没有注意到空格在哪里

Integers: Recreation Two

我的理解（a\*\*2+b\*\*2）\*(c\*\*2+d\*\*2)=e\*\*2+f\*\*2

输入a、b、c、d，返回e、f

但我不太明白它们的解法~是数学功底不好么

Primitive

目的，给定一个字符串，判定其是属于哪种数字类型，其中只有整数

这个问题中，是思维有点跟不上。刚开始脑中第一想法是将其当成字符串处理，一个一个字符串比对，但是这样的方法肯定是不对的，因为其不够简单和优美。

我们更重要的是学会黎勇已有的工具，而不是重复造轮子。

利用Long.parseLong方法，可以将字符串转换成long型。

（关于正则表达式匹配）

1、句点符号匹配所有通配符，但只是匹配一个

2、方括号也只能匹配单个字符，匹配方括号中有的字符，而不是所有字符

3、圆括号中使用“|”运算符进行或运算，可以匹配多个字符

4、表示匹配次数的符号

\* 0次或者多次

+ 1次或者多次

？ 0次或者1次

{n} 恰好n次

{n,m}从n次到m次

（ps：特定数量的需要大括号，而其它的只需要把符号用上）

5、“^”符号称为否符号，用在方括号内，表示不想匹配的字符

6、在Java中，你必须对每一个向前的斜杠（“\”）进行转义处理

7、常用符号

\d [0-9]

\D [^0-9]

\w [A-Z0-9]

\W [^A-Z0-9]

\s [\t\n\r\f]

\S [^\t\n\r\f]

（ps：自己正则表达式的功夫不到家，得要自己看答案了）

1. 匹配输入字符串结尾的位置

\* Multiply two ints, but take care of overflow. If the result cannot be

\* accurately stored in an int, throw an ArithmeticException.

理解题目有误，这里是抛出异常，而不是让我来捕获异常和处理异常