

Problem 6: Big Ben's Big Brain Bamboozles a Bovine

4+2+3=9 Points

Problem ID: `rotate`

Rank: 2+2+3

Sponsored Bounty: \$100 [X-Camp](#) Discount Code for **every member** of the first team to solve any test set of this problem and fill out this [Google Form](#)!

Introduction

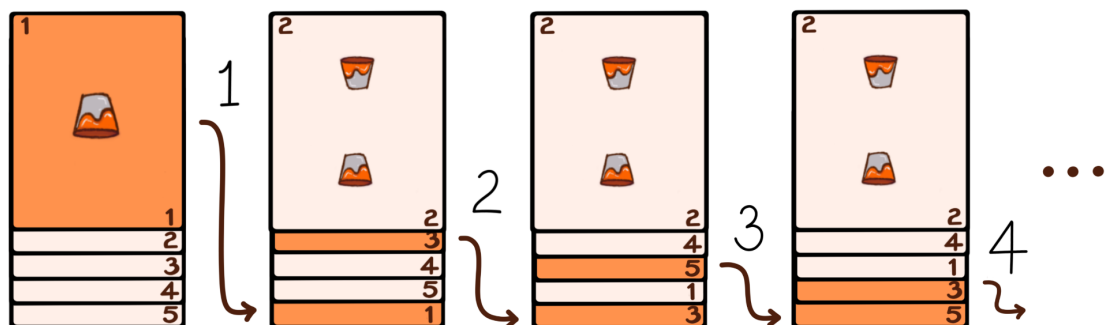
In the bustling streets of Madrid stood a remarkable card dealer named El Gran Benjamín. Each of Benjamín's cards were imbued with magic, and his game was **N Card Monte**, a mystifying generalization of the classic [Three Card Monte](#). With a glint in his eye, Benjamín shuffled his deck, sliding cards with swift, practiced movements. His spectators were left in awe, trying to decipher the sequence in which the cards appeared. Eager to crack the secrets behind **N Card Monte**, Bessie the Cow embarks on a quest to unravel the patterns within Benjamín's shuffles.

Problem Statement

A deck of **N** cards is labeled with integers from 1 to **N**. Initially, they are sorted such that the top card is labeled 1 and the bottom card is labeled **N**. A shuffle is then performed on the deck:

1. Move the topmost card to the bottom of the deck. The deck now has a new ordering.
2. In the new ordering, move the second from topmost card to the bottom of the deck.
3. Then, move the third from topmost card to the bottom of the deck.
4. Continue this process until the bottom of the deck (and the order stops changing).

After the shuffle, find the position of the card labeled **K**. The topmost card is considered to be in position 1, the second from topmost in position 2, and so on. Here's an example with **N = 5**:



Input Format

The first line of the input contains a single integer **T** denoting the number of test cases that follow.

Each test case is described in a single line containing two space-separated integers **N K** denoting the number of cards in the deck and the card label we're looking for.

Output Format

For each test case, output a single line containing an integer denoting the position of the card labeled **K**, where the topmost card is in position 1, the second from topmost card is position 2, and so on.

Constraints

$$1 \leq T \leq 100$$

Main Test Set

$$1 \leq K \leq N \leq 100$$

Bonus Test Set 1

$$1 \leq K \leq N \leq 10^6$$

For this test set only, the sum of **N** across all test cases in a test file does not exceed 10^5 .

Bonus Test Set 2

$$1 \leq K \leq N \leq 10^{18}$$

Sample Test Cases

Main Sample Input

[Download](#)

```
6
1 1
5 1
5 2
6 3
6 4
98 57
```

Main Sample Output

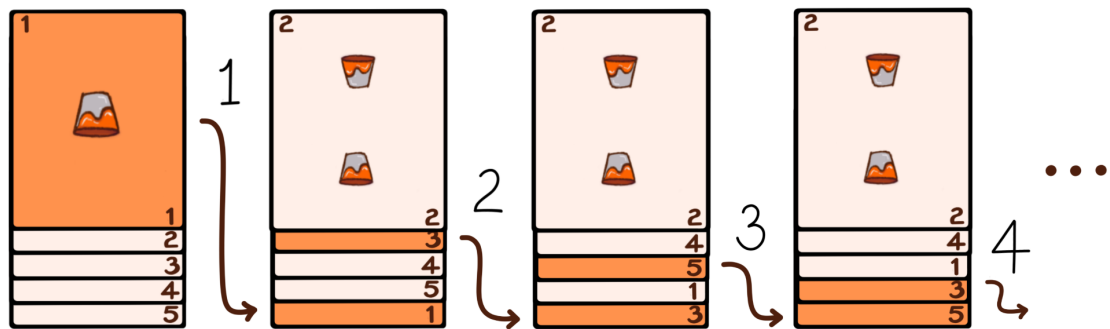
[Download](#)

```
1
3
1
4
2
81
```

Main Sample Explanations

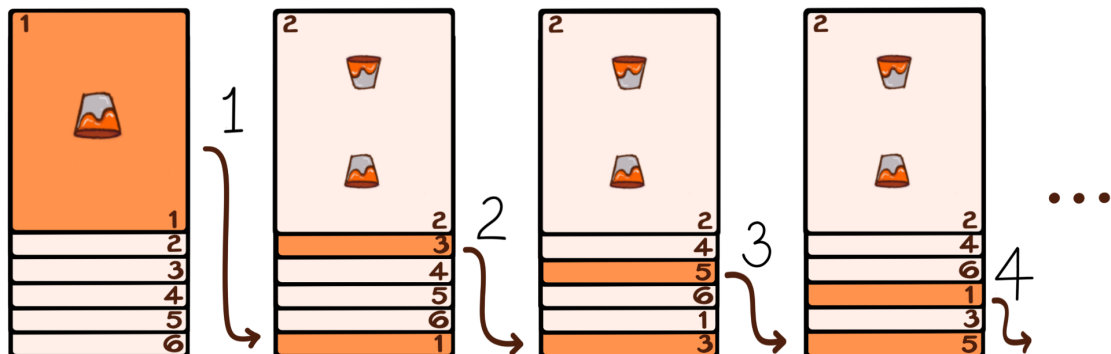
For test case #1, we only have 1 card. Shuffling does nothing, so the answer is 1 (the top card).

For test case #2, we have $N = 5$ cards and we're looking for the card labeled $K = 1$. After the first few steps of the shuffle process shown below, we see that the card labeled 1 is the third from topmost card.



For test case #3, we have 5 cards and we're looking for 2. As 2 is the top card, the answer is 1.

For test cases #4 and #5, the result of the few steps of the shuffle for 6 cards is shown below.



Bonus 1 Sample Input[Download](#)

```
2
1337 420
6666 999
```

Bonus 1 Sample Output[Download](#)

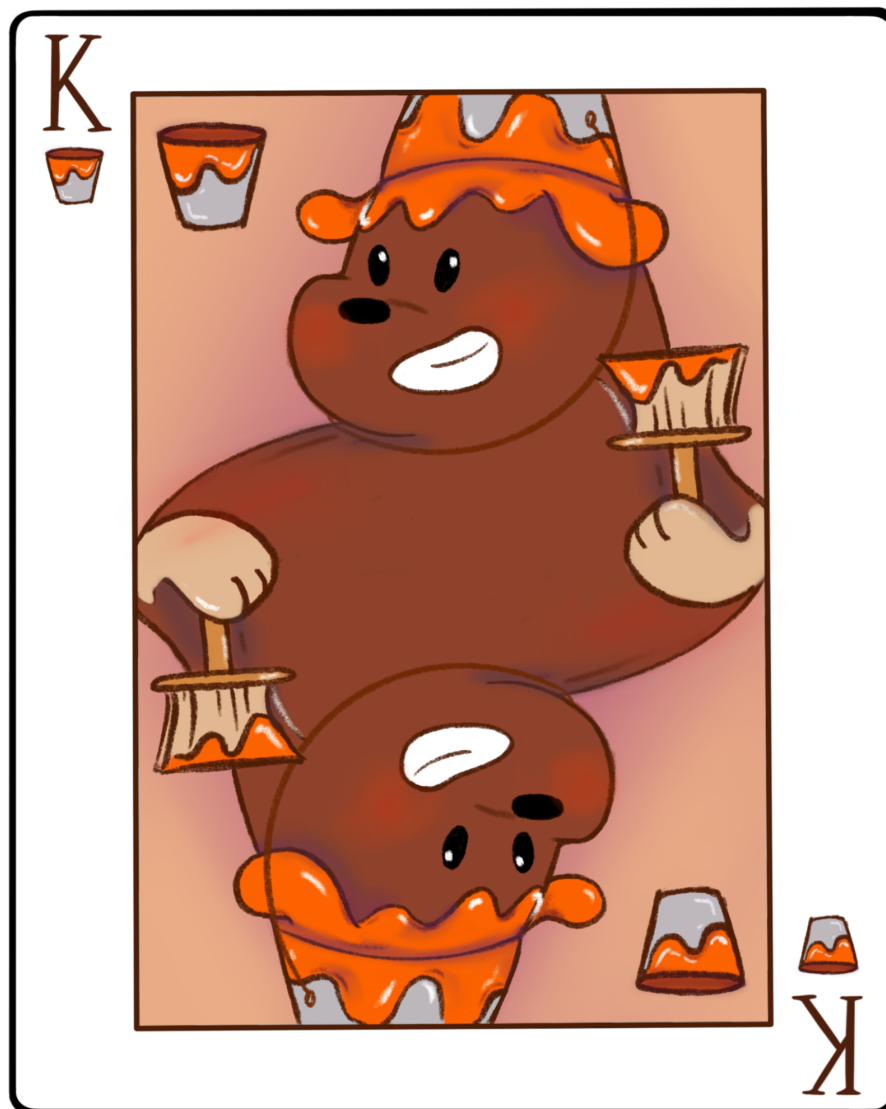
```
210
3583
```

Bonus 2 Sample Input[Download](#)

```
1
31415926535897932 3846264338327950
```

Bonus 2 Sample Output[Download](#)

```
1923132169163975
```



第6题: N 张牌游戏的奥秘

4+2+3=9 分

问题标识符: rotate

难度等级: 2+2+3

奖励: 首个通过此问题任意测试集并填写[此表格](#)的团队的每位成员可赢得 100 美元 [X-Camp](#) 折扣码!

问题背景

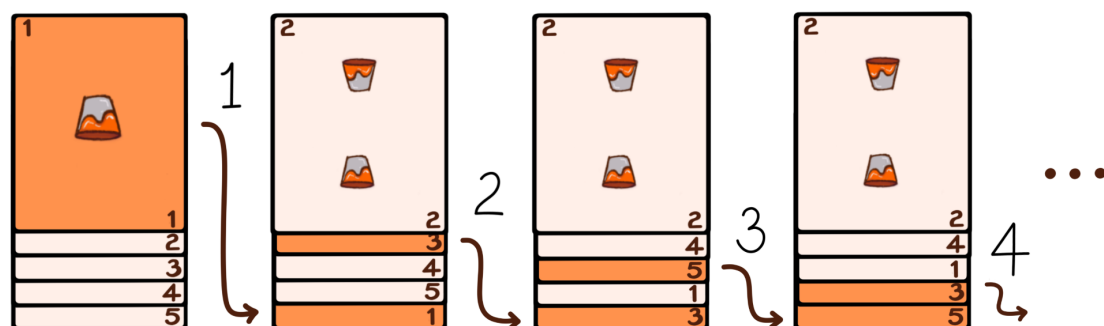
在马德里繁华的街头有一个名叫本杰明的知名卡牌商。本杰明并非普通商贩，他手持一副蕴含魔力的扑克牌。他的游戏被称为“N张牌游戏”，这是一种“改良”后的[三张牌](#)游戏。本杰明眼中闪烁着光彩，他熟练地洗着牌，使卡牌在他手中迅速翻动。观众们目不转睛，试图破解卡牌出现的顺序。贝茜为了解开N张牌游戏背后的奥秘，开启了解开本杰明洗牌模式的探索之旅。

问题描述

一副由整数 1 到 N 标记的 N 张牌。起初牌按顺序排列，最上面是标记为 1 的牌，最下面是标记为 N 的牌。然后按照以下方式进行洗牌：

1. 将最上面的牌移动到底部，使牌的顺序改变。
2. 在新顺序的基础上，将从最上面开始数的第二张牌移动到底部。
3. 然后，将从最上面开始数的第三张牌移动到底部。
4. 继续这个过程，直到将最下面的牌移到底部（这不会对牌的顺序作出改变）。

洗牌后，需找到初始标号为 **K** 的牌现在的位置，其中最上面的牌位于第一位，第二张牌位于第二位，以此类推。



输入格式

输入的第一行包含一个整数 **T**，表示后面测试用例的数量。

每个测试用例包含一行，包含用空格分隔的两个整数 **N** 和 **K**，**N** 代表卡牌数量，**K** 代表我们要找的卡牌标号。

输出格式

对于每个测试用例，请输出一行，包含一个整数，表示初始标号为 **K** 的牌在洗牌后的位置。洗牌后的牌堆里，最上面的牌位于第一位，第二张牌位于第二位，依此类推。

数据范围

$$1 \leq T \leq 100$$

主测试集

$$1 \leq K \leq N \leq 100$$

附加测试集 1

$$1 \leq K \leq N \leq 10^6$$

For this test set only, the sum of **N** across all test cases in a test file does not exceed 10^5 .

附加测试集 2

$$1 \leq K \leq N \leq 10^{18}$$

测试样例

主样例输入

[Download](#)

主样例输出

[Download](#)

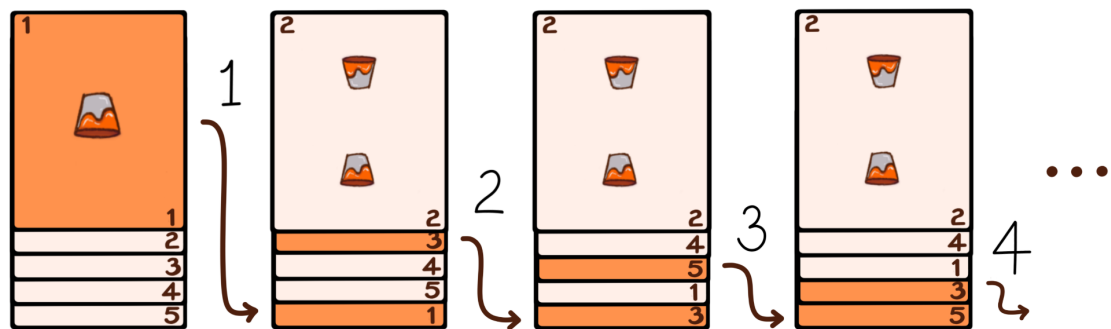
```
6
1 1
5 1
5 2
6 3
6 4
98 57
```

```
1
3
1
4
2
81
```

主样例解释

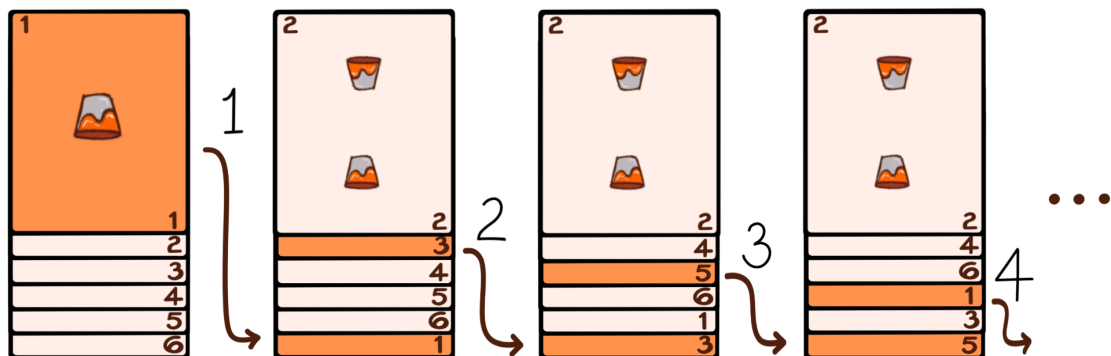
对于测试用例 #1，只有 1 张卡牌。洗牌不会改变顺序，所以答案是 1 (即最上面的牌)。

对于测试用例#2，有 $N = 5$ 张牌，要找的牌的标号为 $K = 1$ 。在洗牌过程后，标记为 1 的牌是从上数的第三张牌。



对于测试用例#3，有 5 张牌，要找的牌标号为 2。因为 2 是最上面的牌，答案是 1。

对于测试用例 #4 和 #5，下图显示了 6 张牌的前几次洗牌结果。



附加测试集 1 样例输入

[Download](#)

```
2
1337 420
6666 999
```

附加测试集 1 样例输出

[Download](#)

```
210
3583
```

附加测试集 2 样例输入

[Download](#)

```
1
31415926535897932 3846264338327950
```

附加测试集 2 样例输出

[Download](#)

```
1923132169163975
```

