

## Problem 1. pay

Input file:        `pay.in`  
Output file:      `pay.out`  
Time limit:       1 second  
Memory limit:    256 MB

Mr.Hu 开了个饭店，来了两位客人：Alice 和 Bob，他们吃完饭要结账时，发现他们需要支付  $c$  元钱，但是 Alice 只有面值为  $a$  的钱，Bob 只有面值为  $b$  的钱（他们每个人的钱的和都大于  $c$ ，即可以认为他们有无数张对应面值的钱）。现在，Mr.Hu 想知道，他们可能刚好支付完饭钱吗？如果可能，那么有多少种方式？你还需要计算出他们所有可能的支付方式的支付的钱的张数的和。

### Input

第 1 行包含 1 个整数：  $T$   $opt$ ，其中  $T$  表示数据组数， $opt$  为数据类型。

接下来  $T$  行，每行 3 个整数：  $a$   $b$   $c$ 。

### Output

对于每组数据：

- 如果  $opt = 1$ ，输出一行，包含一个整数：  $A$ ，其中  $A$  表示刚好支付的方案数。
- 如果  $opt = 2$ ，输出一行，包含两个整数：  $A$   $B$ ，其中  $A$  表示刚好支付的方案数， $B$  表示所有可能支付方式的张数和。

### Sample

<code>pay.in</code>	<code>pay.out</code>
2 2	2 13
3 4 21	4 18
2 4 12	

样例解释：

对于 3 4 21，一共有两种可能的支付方式，分别是：  $(3,3)$ ,  $(7,0)$ <sup>1</sup>，所以  $A$  为 2， $B$  为  $3+3+7+0=13$ 。

对于 2 4 12，一共有四种可能的支付方式，分别是：  $(6,0)$ ,  $(4,1)$ ,  $(2,2)$ ,  $(0,3)$ ，所以  $A$  为 4， $B$  为  $6+0+4+1+2+2+0+3=18$ 。

<code>pay.in</code>	<code>pay.out</code>
2 1	2
3 4 21	4
2 4 12	

### Note

- 对于 20% 的数据，  $1 \leq a, b, c \leq 10000$ ，  $1 \leq T \leq 1000$ ；
- 对于另外 40% 的数据，  $1 \leq a, b, c \leq 10^9$ ，其中  $opt = 1$ ；
- 对于另外 40% 的数据，  $1 \leq a, b, c \leq 10^9$ ，其中  $opt = 2$ ；
- 对于 100% 的数据，  $1 \leq T \leq 10^5$ ，  $1 \leq opt \leq 2$ 。

<sup>1</sup>其中  $(x, y)$  表示 Alice 支付  $x$  张面值为  $a$  的钱，Bob 支付  $y$  张面值为  $b$  的钱

## Problem 2. sumcomb

Input file: `sumcomb.in`  
Output file: `sumcomb.out`  
Time limit: 1 second  
Memory limit: 256 MB

Mr.Hu 被传送到了一个无限大的表格上，现在这个表格的第  $i$  行第  $j$  列的值是  $a_{i,j}$  ( $0 \leq i, j$ ):

$$a_{i,j} = \begin{cases} 1 & j = 0 \text{ 或 } i = j \\ a_{i-1,j} + a_{i-1,j-1} & 0 < j < i \\ 0 & j > i \end{cases}$$

现在，Mr.Hu 站在  $(n, m)$  这个位置，他想知道，他向上或向左上方 45 度望去，看到的数的和是多少。  
从  $(n, m)$  向上望去，他会看到  $(n, m), (n-1, m), (n-2, m), \dots, (0, m)$  这些位置。  
从  $(n, m)$  向左上方 45 度望去，他会看到  $(n, m), (n-1, m-1), \dots$ ，直到某一维的下标变为 0。  
这个数可能很大，你只需将答案对  $10^9 + 7$  取模即可。

### Input

第 1 行一个整数:  $T$ ，表示数据组数。

接下来  $T$  行，每行格式为:  $dir\ n\ m$ ，其中  $dir$  为 1 表示向上看，2 表示向左上方看， $(n, m)$  为 Mr.Hu 现在的位置。

### Output

对于每组数据，输出一行表示答案。

### Sample

sumcomb.in	sumcomb.out
2	4
1 3 2	6
2 3 2	

表格左上角长成这样（行列都是 0 base 的）:

```
1 0 0 0
1 1 0 0
1 2 1 0
1 3 3 1
```

这样从  $(3, 2)$  向上看，会看到: 3 1 0 0，和为 4。

向左上角看，会看到: 3 2 1，和为 6。

### Note

- 对于 30% 的数据， $1 \leq n, m \leq 5000$ ， $1 \leq T \leq 1000$ ；
- 对于 100% 的数据， $1 \leq n, m \leq 10^6$ ， $1 \leq T \leq 50000$ 。

## Problem 3. kor

Input file: kor.in  
Output file: kor.out  
Time limit: 1 second  
Memory limit: 256 MB

Mr.Hu 觉得在学习过程中, 需要举一反三, 做一题要理解透, 然后遇到相似的问题时能类似地转化。所以想了一道和以前类似的题目, 相信聪明如你, 肯定能轻而易举地解决。

Mr.Hu 会给你  $n$  个非负整数, 然后从中选  $k$  个出来, 然后把这  $k$  个数按位或起来, Mr.Hu 想知道有多少种选法, 使得或起来的结果为  $r$ 。

### Input

第 1 行一个整数  $T$ , 表示测试组数。

接下来  $T$  组数据, 对于每组数据:

第 1 行两个整数  $n\ k\ r$ 。

接下来 1 行包含  $n$  个非负整数:  $a_1\ a_2\ \dots\ a_n$ 。

### Output

对于每组数据, 输出一行, 包含一个整数, 即方案数, 因为结果可能很大, 只需要对  $10^9 + 7$  取模即可。

### Sample

kor.in	kor.out
2	3
4 2 3	1
1 2 3 4	
4 1 1	
1 2 3 4	

对于第一组数据, 一共有 3 种选法:  $(1, 2), (1, 3), (2, 3)$ 。

对于第二组数据, 一共有 1 种选法:  $(1)$ 。

### Note

- 对于 10% 的数据,  $1 \leq n \leq 10$ ,  $0 \leq a_i < 2^{10}$ ;
- 对于 30% 的数据,  $1 \leq n \leq 100$ ,  $0 \leq a_i < 2^{10}$ ;
- 对于 50% 的数据,  $1 \leq n \leq 10^5$ ,  $0 \leq a_i < 2^{15}$ ;
- 对于 100% 的数据,  $1 \leq n \leq 10^5$ ,  $0 \leq a_i < 2^{20}$ ,  $1 \leq k \leq n$ ,  $1 \leq T \leq 5$ 。