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

pay.in	pay.out
2 2 3 4 21 2 4 12	2 13 4 14

样例解释：

对于 2 4 21，一共有两种可能的支付方式，分别是：$(3,3),(7,0)$[1]，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 $6+0+4+1+2+2+0+3=14$。

Note

- 对于 20% 的数据，$0 \le a,b,c \le 10000$；
- 对于另外 40% 的数据，$0 \le a,b,c \le 10^9$，其中 $opt = 1$；
- 对于另外 40% 的数据，$0 \le a,b,c \le 10^9$，其中 $opt = 2$；
- 对于 100% 的数据，$1 \le T \le 1000$，$1 \le opt \le 2$。

[1]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 $a_{i,j}$（$0 \le i, j$）：

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

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 $(n, m), (n-1, m), (n-2, m), \cdots, (0, m)$ 这些位置。

从 (n, m) 向左上方 45 度望去，他会看到 $(n, m), (n-1, m-1), \cdots$，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 1 3 2 2 3 2	4 6

表格左上角长成这样（行列都是 0 base 的）：

$$\begin{matrix} 1\ 0\ 0\ 0 \\ 1\ 1\ 0\ 0 \\ 1\ 2\ 1\ 0 \\ 1\ 3\ 3\ 1 \end{matrix}$$

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，$1 \le n, m \le 5000$；
- 对于 100% 的数据，$1 \le n, m \le 10^6$，$1 \le T \le 5000$。

Problem 3. ocount

Input file:	`ocount.in`
Output file:	`ocount.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 r。

接下来 1 行包含 n 个非负整数：a_1 a_2 ... 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

ocount.in	ocount.out
2 4 3 1 2 3 4 4 1 1 2 3 4	5 1

对于第一组数据，一共有 5 种选法：$(1,2),(3),(1,3),(2,3),(1,2,3)$。

对于第二组数据，一共有 1 种选法：(1)。

Note

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

Problem 4. facsum

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

Mr.Hu 最近偶得一函数：

$$f(n) = \big(\sum_{d|n} \varphi(d)\big)^k \big(\sum_{d|n} \sigma(d)\mu(\frac{n}{d})\frac{n}{d}\big)$$

又：

$$F(n) = \sum_{i=1}^{n} f(i)$$

Mr.Hu 希望你计算 $F(n) \quad mod\, 10^9 + 7$ 的值。

Input

第一行包含两个整数：n k。

Output

输出一行包含一个数，表示答案。

Sample

facsum.in	facsum.out
2 2	13

样例解释：$f(1) = 1 \quad f(2) = 12$，故 $F(2) = f(1) + f(2) = 13$。

Note

- 对于 20% 的数据，$1 \le n \le 5000$。
- 对于 50% 的数据，$1 \le n \le 10^5$。
- 对于 100% 的数据，$1 \le n \le 10^7$，$1 \le k \le 20$。

Problem 5. group

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

Mr.Hu 最近在研究等比数列，即形如：

$$a, a^1, a^2, a^3, \ldots, a^n, \ldots$$

现在，Mr.Hu 想知道，对于给定的非负整数 a，上面这个无穷数列在摸 mod 意义下有多少项是本质不同的。(保证 $gcd(a, mod) = 1$)。

Input

第 1 行一个整数：T，表示数据组数。

接下来 T 行，每行两个整数：a mod。

Output

对于每组数据，输出一行，包含一个整数，表示模意义下本质不同的数有多少个。

Sample

group.in	group.out
2 1 3 2 5	1 4

对于第一组数据，数列是：$1, 1, 1, \ldots, 1, \ldots$

对于第二组数据，数列（取模以后）是：$2, 4, 3, 1, 2, 4, 3, 1, \ldots$，总共有 4 个本质不同的数。

Note

- 对于 30% 的数据，$0 \le a \le 10^3$，$1 \le mod \le 10^3$；
- 对于 100% 的数据，$0 \le a \le 2 \times 10^9$，$1 \le mod \le 2 \times 10^9$，且保证 $gcd(a, mod) = 1$，$1 \le T \le 1000$。

Problem 6. ccount

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

Mr.Hu 最近在学习组合数，他觉得这些数非常美丽。

于是，他写下了这样一个数列：

$$\binom{l}{n}, \binom{l+1}{n}, \binom{l+2}{n}, \ldots, \binom{r-1}{n}, \binom{r}{n}$$

Mr.Hu 想知道，这些数里面，有多少个数写成十进制的形式后，最后一位为 0。

Input

第 1 行一个整数：T，表示数据组数。

接下来 T 行，每行三个整数：$l\ r\ n$。

Output

对于每组数据，输出一行，包含一个整数，表示答案。

Sample

ccount.in	ccount.out
2 1 3 4 2 4 5	0 2

对于第一组数据，数列是：4 6 4，没有以 0 结尾的数，故答案为 0。

对于第二组数据，数列是：5 10 10 5，有 2 个数以 0 结尾，故答案为 2。

Note

- 对于 20% 的数据，$1 \le n \le 5000$。
- 对于 40% 的数据，$1 \le n \le 10^9$，$1 \le r - l + 1 \le 1000$。
- 对于 100% 的数据，$1 \le n \le 10^18$，$0 \le l \le r \le n$，$1 \le T \le 100$。