Sequence

As you know, there number of permutation of the increasing vector {1, 2, 3…n} is exactly n! ;For example, if n = 3, then, {1,2,3} , {1,3,2}, {2,1,3}, {2,3,1}, {3,1,2}, {3,2,1} are all the permutation of the vector {1,2,3 };

We define D({A1,A2...An **}) = the number of element that satisfy Ai = i.**

For example, D({1,2,3}) = 3 ,D({1,3,2}) = 1 (only '1' is at 1), D({3,1,2}) = 0 ….

Now we want to calculate the number of permutation that satisfy D({A1,A2...An }) = K.

For example, if n = 3 and k = 3, then of course there is only one permutation {1,2,3} that satisfy D({1,2,3}) = 3. But if n = 3 and k = 0, then there are two permutations {3,1,2} and {2,3,1} satisfy D({3,2,1}) = D({2,3,1}) = 0;

But when n is very large, it's hard to calculate by brute force algorithm. Optimal is one required here.

Because the answer may be very large, so just output the remainder of the answer after divided by m.

Input

In the first line there is an integer T, indicates the number of test cases. (T <= 500)

In each case, the first line contains three integers n,k and m. (0 <= k<= n <=10^9, 1 <= m <= 10^5, n != 0)

Output

Output "Case d: "first where d is the case number counted from one. Then output the remainder of the answer after divided by m.

Example

Input:

```
3
3 0 7
3 3 3
331105236 82934100 3711
```

Output:

```
Case 1: 2
Case 2: 1
Case 3: 2622
```