Ajob Subsequence

You are currently studying the language $Ajob$ (which means strange in English) from a renowned professor. The language has infinite number of letters in its alphabet (now you know, why it is called ajob).

The professor taught you N words, one by one. The number of letters in a word is equal to it's place in the teaching order. Thus, the 1st word taught by the professor has 1 letter, 2nd word has 2 letters, 3rd word has 3 letters, $\cdots$, the Nth word has N letters.

All the letters within a word are distinct to each other.

Now, you are handed an assignment. You have to choose any one of the N words and form a subsequence from it. The length of the subsequence should be **exactly** K less than the length of original word. For example, if the length of the chosen word is L, then the length of the subsequence should be $L - K$. If for any word, L is smaller than K $(L < K)$, then you must not choose that word. Two subsequences are different to each other if, the lengths of them are different or they contain different characters in the same position.

Find the number of ways you can complete the assignment $\mathrm{modulo}\ p$ (p will always be a $prime$).

1.
The first line contains T, the number of testcases. The next T lines contain three space separated integers N, K and p.

Output Format
For each testcase, print one integer in a single line, the number possible ways you can complete the assignment, $\mathrm{modulo}\ p$.

Constraints
$1 \le T \le 100$
$2 \le N \le 10^{18}$
$0 < K < N$
$2 \le p \le 10^5$
p is a $Prime$

Sample Input #00

```
3
2 1 2
2 1 5
5 3 13
```

Sample Output #00

```
1
3
2
```

Sample Input #01

```
5
```

```
5 2 3
6 5 11
7 6 3
6 5 7
6 5 5
```

Sample Output #01

```
2
7
2
0
2
```