# Divide Polygon (HARD)

This is hard version of DTPOLY.

Determine the number of ways to cut a convex polygon with **N** vertices if the only cuts allowed are from vertex to vertex, each cut divides exactly one polygon into exactly two polygons, and you must end up with exactly **K** polygons. Consider each vertex distinct. For example, there are three ways to cut a square - the two diagonals and not cutting at all - but only two ways to cut it to form 2 polygons, and only one way to cut it to form 1 polygon. The order of cuts does not matter. Since the number of ways can be very large, you should return the number taken modulo **M**.

## Input

Input contains several test cases, i-th line consists of 3 integers: $\mathbf{N_i}$ ($3 \le \mathbf{N_i}$, $\mathbf{\Sigma N_i} \le 10^8$ over all test cases),

$\mathbf{K_i}$ ($1 \le \mathbf{K_i} \le \mathbf{N_i} - 2$) and $\mathbf{M_i}$ ($1 < \mathbf{M_i} < 2^{60}$), all pairs ($\mathbf{N_i}$, $\mathbf{K_i}$) are different.

## Output

On the i-th line print the number of different ways to cut the polygon with $\mathbf{N_i}$ vertices into $\mathbf{K_i}$ pieces modulo $\mathbf{M_i}$.

## Example

**Input:**

```
4 2 100
6 3 100
10000000 2 1000000007
10000000 5000000 1000000014000000049
```

**Output:**

```
2
21
984650007
780127215155143528
```