

## D. Table

time limit per test: 4 seconds  
memory limit per test: 256 megabytes  
input: standard input  
output: standard output

John Doe has an  $n \times m$  table. John Doe can paint points in some table cells, not more than one point in one table cell. John Doe wants to use such operations to make each square subtable of size  $n \times n$  have exactly  $k$  points.

John Doe wondered, how many distinct ways to fill the table with points are there, provided that the condition must hold. As this number can be rather large, John Doe asks to find its remainder after dividing by 1000000007 ( $10^9 + 7$ ).

You should assume that John always paints a point exactly in the center of some cell. Two ways to fill a table are considered distinct, if there exists a table cell, that has a point in one way and doesn't have it in the other.

### Input

A single line contains space-separated integers  $n, m, k$  ( $1 \leq n \leq 100$ ;  $n \leq m \leq 10^{18}$ ;  $0 \leq k \leq n^2$ ) — the number of rows of the table, the number of columns of the table and the number of points each square must contain.

Please, do not use the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specifier.

### Output

In a single line print a single integer — the remainder from dividing the described number of ways by 1000000007 ( $10^9 + 7$ ).

### Examples

input
5 6 1
output
45

### Note

Let's consider the first test case:

The gray area belongs to both  $5 \times 5$  squares. So, if it has one point, then there shouldn't be points in any other place. If one of the white areas has a point, then the other one also must have a point. Thus, there are about 20 variants, where the point lies in the gray area and 25 variants, where each of the white areas contains a point. Overall there are 45 variants.