

Problem D. Pruefsumme

Input file: `pruefsumme.in`
Output file: `pruefsumme.out`
Time limit: 2 seconds
Memory limit: 256 mebibytes

We are surrounded by long numeric identifiers: bank account numbers, book ISBN codes, product barcodes, and so on. Most of those identifiers are or were often written and typed in by people, and in order to avoid mistakes have some sort of error detection built in.

The error detection usually comes in form of a checksum. For example, the ISBN-10 book codes are 10-digit sequences a_1, a_2, \dots, a_{10} with the following checksum:

$$10 \times a_1 + 9 \times a_2 + 8 \times a_3 + \dots + 2 \times a_9 + 1 \times a_{10} \pmod{11}$$

The checksum is a number between 0 and 10, and for valid book codes it is always 0. As it was designed for error detection, it has the following important properties:

1. If we change any single digit while keeping the others the same, the checksum changes.
2. If we swap two different adjacent digits while keeping the rest the same, the checksum changes.

However, it is not very beautiful because it has 11 possible values, while the individual digits have only 10 possible values, and so it seems a bit excessive.

We introduce the concept of a *beautiful checksum* as follows: suppose our identifiers are sequences of n integers, each being between 0 and $m - 1$, inclusive: a_1, a_2, \dots, a_n . A checksum algorithm is defined by two matrices p_{ij} ($0 \leq i, j \leq m - 1$) and q_{ki} ($1 \leq k \leq n, 0 \leq i \leq m - 1$) with values between 0 and $m - 1$, inclusive. The checksum is then computed like this:

1. Set $s_0 = 0$.
2. For each k from 1 to n :
 - (a) Set $t_k = q_{k, a_k}$.
 - (b) Set $s_k = p_{s_{k-1}, t_k}$.
3. Return the value of s_n as the checksum.

In other words, the matrix q defines a transformation of numbers for each position, and the matrix p tells how to combine transformed numbers. The two matrices p and q define a *beautiful checksum* for the given n and m if the checksum value satisfies the two properties listed above for any sequence a : it must be sensitive to changing a single number, and to swapping two different adjacent numbers ($a_k \neq a_{k+1}$ for some k).

For example, ISBN-10 defined as $p_{ij} = i + j \pmod{11}$, $q_{ki} = (11 - k) \times i \pmod{11}$ is a beautiful checksum for $n = 10, m = 11$.

Can you come up with a beautiful checksum for the given n and m ?

Input

The only line of the input file contains two numbers: n and m , $2 \leq n \leq 100, 2 \leq m \leq 9$.

Output

If there is a beautiful checksum for the given values of n and m , print **Ja** on the first line of the output file, otherwise print **Nein**.

In case you printed **Ja**, print the description of the checksum afterwards. On the next m lines print the matrix p , with m space-separated integers per line: the j -th (0-based) value in the i -th (0-based) of those m lines should contain the value p_{ij} . On the next n lines print the matrix q , with m space-separated integers per line: the i -th (0-based) value in the k -th (1-based) of those n lines should contain the value q_{ki} .

All numbers you print must be between 0 and $m - 1$, inclusive. In case there are multiple possible solutions, you may output any of them.

Example

pruefsumme.in	pruefsumme.out
4 3	Ja 0 1 2 1 2 0 2 0 1 0 1 2 0 2 1 0 1 2 0 2 1