

Factorial factorisation

Factorial $n!$ of an integer number $n \geq 0$ is defined recursively as follows:

$$\begin{aligned}0! &= 1 \\ n! &= n * (n - 1)!\end{aligned}$$

While playing with factorials chEEtah noticed that some of them can be represented as a product of other factorials, e.g. $6! = 3! * 5! = 720$. Such factorisation helps chEEtah to simplify a certain class of equations he is working on during his research.

So he needs a program that finds a compact factorisation of a given factorial or determines that it is impossible if that is the case. If there are more than one factorisation the program has to find such that contains the minimum number of terms. For example, $10!$ can be factorised in two different ways: $10! = 6! * 7! = 3! * 5! * 7!$. The first factorisation contains only two terms and should be preferred to the second one. If there are several factorisations with the same minimum number of terms then any optimal solution is acceptable.

Input

Input contains the only integer $2 \leq n \leq 1000$ which factorial $n!$ should be factorised.

Output

Output should contain the optimal factorisation in the format shown in the samples. The factorisation terms should go in non-decreasing order. If no factorisation can be found print **No solution**.

Example

Input:

9

Output:

$9! = 2! * 3! * 3! * 7!$