

Problem B. Bingo

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 1024 mebibytes

Bingo is a game on a square grid. Each player gets an $n \times n$ grid and writes a unique number in each cell. The game host then draws a random number, and each player looks for that number on their grid and, if the number is present on their grid, fills in the corresponding cell. This repeats until someone finds n filled cells on a single line, which we will call a **bingo line**.

There are $2n + 2$ possible bingo lines: n horizontal lines, n vertical lines, and 2 diagonal lines.

```
---  ...  ...  |..  .|.  ..|  \..  ../
...  ---  ...  |..  .|.  ..|  .\..  ./..
...  ...  ---  |..  .|.  ..|  ..\  /..
```

For example, the following grid has four bingo lines: two horizontal lines, one vertical line, and one diagonal line.

```
#..#.
#####
..###
#####
..###
```

Exactly when is a bingo line formed? That is completely random: you can get a line quite early if you are lucky, but on the other hand, you can fill most of the grid without getting any bingo lines. In this problem, we investigate the unfortunate case of filling k cells without making any bingo lines.

Given two integers n and k , determine whether it is possible to fill exactly k cells in an $n \times n$ grid, without making any bingo lines. If it is possible, demonstrate how to do it.

Input

The first and only line of input contains two integers, n and k ($1 \leq n \leq 100$, $0 \leq k \leq n^2$).

Output

On the first line, output “YES” if it is possible to fill exactly k cells of an $n \times n$ grid without making any bingo lines. Otherwise, output “NO”.

If the answer is “YES”, then output n rows of the grid on the next n lines. Each row should be represented by a string of n characters. The i -th character is ‘#’ (ASCII 35) if the i -th cell of the row is filled, and ‘.’ (ASCII 46) if it is not filled. Exactly k cells must be filled, and there cannot be any bingo lines.

If there are multiple ways to fill the grid, then output any one of them.

Examples

<i>standard input</i>	<i>standard output</i>
4 2	YES ##..
4 16	NO