Code Jam 2020 - Qualification Round

Indicium

Problem

Indicium means "trace" in Latin. In this problem we work with Latin squares and matrix traces.

A *Latin square* is an **N**-by-**N** square matrix in which each cell contains one of **N** different values, such that no value is repeated within a row or a column. In this problem, we will deal only with "natural Latin squares" in which the **N** values are the integers between 1 and **N**.

The *trace* of a square matrix is the sum of the values on the main diagonal (which runs from the upper left to the lower right).

Given values **N** and **K**, produce any **N**-by-**N** "natural Latin square" with trace **K**, or say it is impossible. For example, here are two possible answers for **N** = 3, **K** = 6. In each case, the values that contribute to the trace are underlined.

```
2 1 3    3 1 2
3 2 1    1 2 3
1 3 2    2 3 1
```

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each consists of one line containing two integers **N** and **K**: the desired size of the matrix and the desired trace.

Output

For each test case, output one line containing `Case #x: y`, where `x` is the test case number (starting from 1) and `y` is `IMPOSSIBLE` if there is no answer for the given parameters or `POSSIBLE` otherwise. In the latter case, output **N** more lines of **N** integers each, representing a valid "natural Latin square" with a trace of **K**, as described above.

Limits

Time limit: 20 seconds per test set.
Memory limit: 1GB.
$\mathbf{N} \le \mathbf{K} \le \mathbf{N}^2$.

Test set 1 (Visible Verdict)

T = 44.
$2 \le \mathbf{N} \le 5$.

Test set 2 (Hidden Verdict)

$1 \le \mathbf{T} \le 100$.
$2 \le \mathbf{N} \le 50$.

Sample

Sample Input

```
2
3 6
2 3
```

Sample Output

```
Case #1: POSSIBLE
2 1 3
3 2 1
1 3 2
Case #2: IMPOSSIBLE
```

Sample Case #1 is the one described in the problem statement.

Sample Case #2 has no answer. The only possible 2-by-2 "natural Latin squares" are as follows:

```
1 2   2 1
2 1   1 2
```

These have traces of 2 and 4, respectively. There is no way to get a trace of 3.