

Problem F. Random Maze

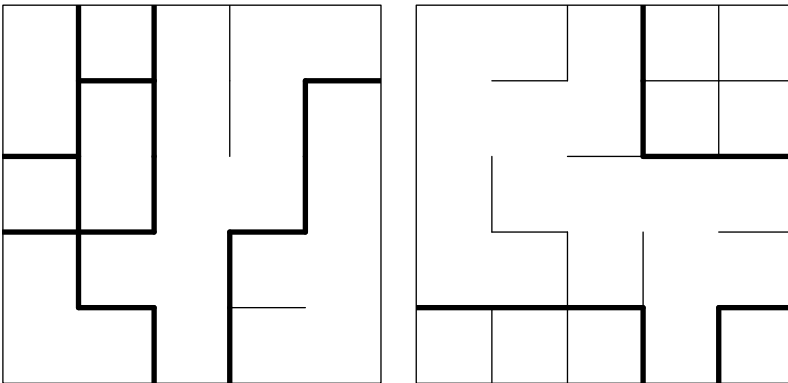
Input file: maze.in
Output file: maze.out
Time limit: 3 seconds
Memory limit: 256 megabytes

There are several algorithmically nice ways to generate random mazes. This problem considers two of them.

We will define a *maze* as a 100×100 grid, with some edges between grid cells being walls, and the rest being two-way passages, such that all outside edges are walls, and passages are chosen in such a way that there's exactly one simple path through passages from any grid cell to any other grid cell (so the grid cells form a tree).

The first way to generate a random maze is similar to Kruskal's minimum spanning tree algorithm. We start with a grid that has only walls and no passages, so all grid cells are disconnected. Then, we repeat the following: pick any wall that connects two disconnected parts randomly with equal probabilities, and convert it into a passage, until we end up with the entire maze being just one part, at which point we're done.

The second way to generate a random maze is similar to Prim's minimum spanning tree algorithm. Again, we start with a grid that has only walls and no passages. Then, we repeat the following: pick any wall that connects the part that contains the top-left corner cell with a yet-untouched cell, and convert it into a passage, until we end up with the entire maze being connected.



The picture to the left shows an intermediate step of the Kruskal algorithm, while the picture to the right shows an intermediate step of the Prim algorithm. The edges that could be removed on the next step are highlighted (19 edges for Kruskal, 10 edges for Prim).

You are given a maze that was generated using one of those two algorithms. You need to figure out which algorithm was used.

Input

The input file will contain 100 lines with 100 tokens in each line. Each token will describe the passages going out from the corresponding grid cell. It will contain one or more distinct letters from the following list. “N” stands for passage to the previous row, “S” stands for passage to the next row, “W” stands for passage to the previous column, “E” stands for passage to the next column.

The maze will be generated randomly according to one of the two above algorithms.

Output

Output a single word: either “KRUSKAL” (without quotes) if the first algorithm was used, or “PRIM”

(without quotes) if the second algorithm was used.

Examples

maze.in	maze.out
ES WE SWE W S N E WSEN EW NSW ES WES NSW S NS N SN N NE WNS E NEW W E WN	KRUSKAL

Note

Note that the first test is bigger than the example above, as it is a 100×100 grid, not 5×5 . You can download the first test at <http://forest.acm.petrsvu.ru/tests/maze.in>. There are 200 tests in this problem.