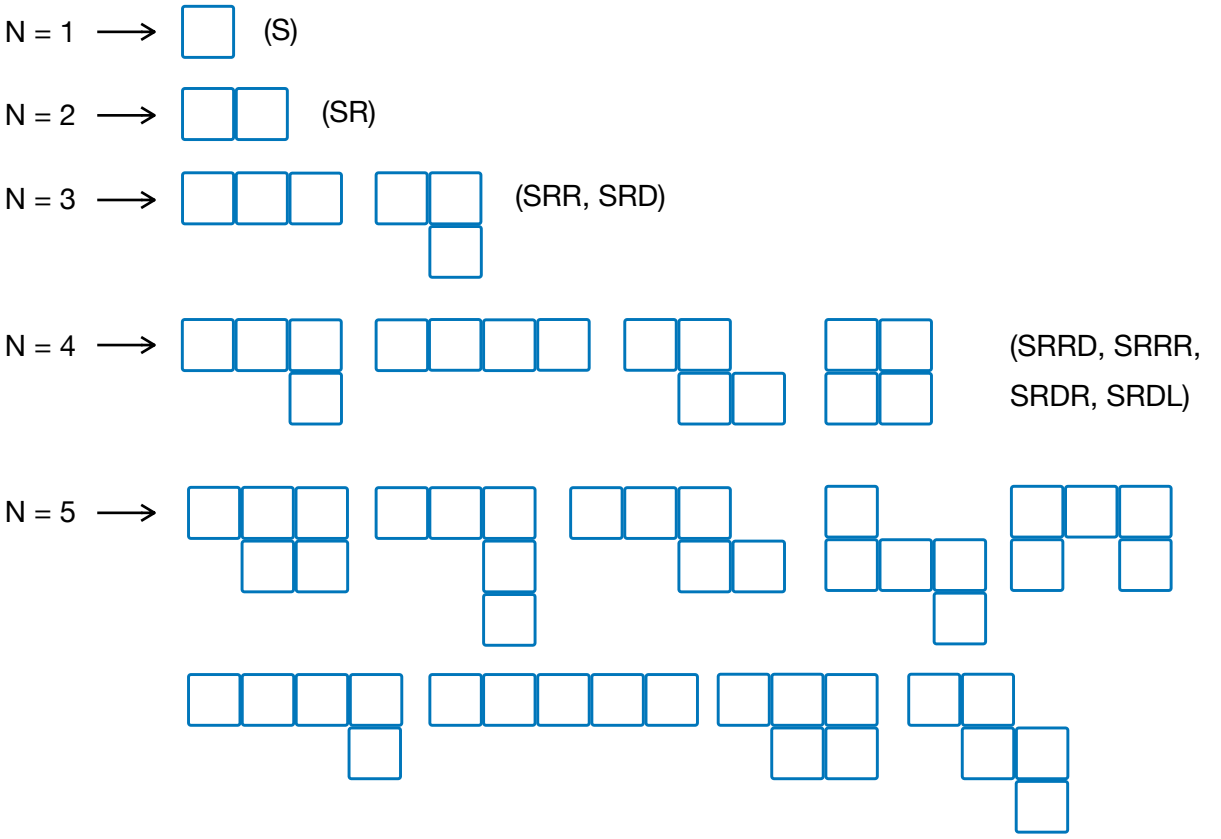
Generate Snakes

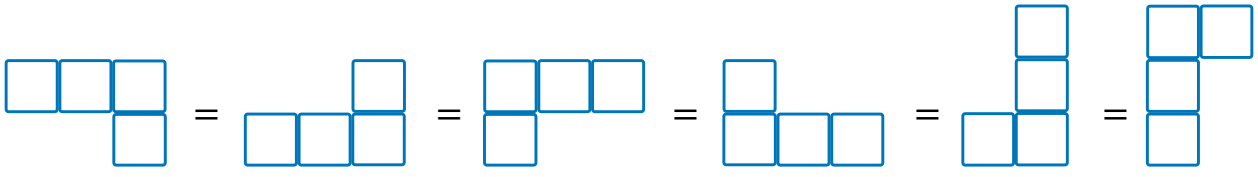
A *snake* is a sequence of several square blocks, attached one after another. A snake starts with a block at some position and continues with another block to the left, right, up or down, then again with another block to the left, right, up or down, etc. A snake of size N consists of a sequence of N blocks and is not allowed to cross itself.

You are given a number N and you should find all possible snakes of N blocks, represented as sequences of moves denoted as: S (start), L (move left), R (move right), U (move up) and D (move down). Examples (for N = 1, 2, 3, 4, and 5):

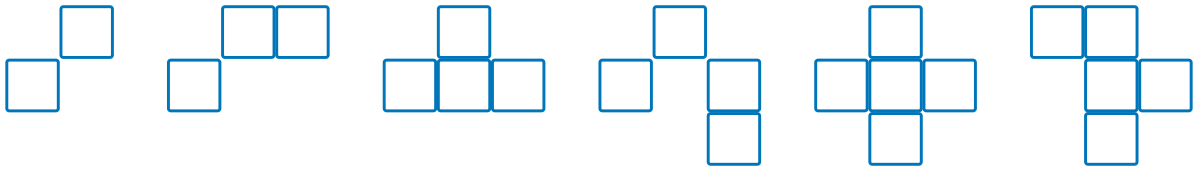


Note: some figures could look visually the same but represent different snakes, e.g. **SRRDL** and **SRDRU**.

Some snakes (sequences of blocks) are the same and should be printed only once. If after a number of rotations and/or flips two snakes are equal they are considered the same and should be printed only once. For example the snakes **SRRD**, **SRRU**, **SLLD**, **SLLU**, **SRUU** and **SUUR** are the same:



Not all forms consisting of N blocks are snakes of size N. Examples of non-snake forms:



Note: When generating the snakes, there may be different correct answers. When testing your solution, priority should be as follows: R → D → L → U. The visual example above for n = 5 does NOT follow this priority.

Input
- The input should be read from the console.
- It will contain an integer number N in the range [1...15].
- The input data will always be valid and in the format described. There is no need to check it explicitly.

Output
- The output should be printed on the console. It should consist of a variable number of lines:
- Each line should hold a snake represented as a sequence of moves.
- On the last line, print the number of snakes in format: "Snakes count = 0".

Constraints
- Allowed working time for your program: 10 seconds. Allowed memory: 512 MB.

Examples

Input	Output	Comments
2	SR Snakes count = 1	Note that SU, SL and SD are also correct outputs. However, SR takes precedence because R has priority over all other directions.
4	SRRR SRRD SRDR SRDL Snakes count = 4	Note that there are many other correct outputs for N = 4, but this is the expected output according to the priority of directions (right, down, left, up)
5	SRRRR SRRRD SRRDR SRRDD SRRDL SRDRD SRDRU SRDDR SRDDL Snakes count = 9	
10	1047	
15	148752	
16	401629	