Problem 5 – Knight Path

You are given a chess board of **size 8 x 8**, divided into 64 cells. Initially, the board is empty (all cells have a value of **0**). You place a knight on the board which inverts the bits on all positions it lands on (0 -> 1; 1 -> 0). The initial position of the knight is [0, 0]. Each move will be defined by a combination of two directions, one vertical and one horizontal (example: "left up"). The knight moves according to the standard rules of chess – two positions in the first direction and then one position in the second direction (see the example below for reference). When the given command leads you out of the board don't move the knight. The valid commands are:

"**left up**", "**left down**", "**right up**", "**right down**", "**up left**", "**up right**", "**down left**", "**down right**".

When you receive the string "**stop**" from the console, you should stop moving the knight. There would be some 1s and 0s on the board. Each row of the board represents a binary integer number. Your task is to print all the rows, which are different from 0, to the console. To understand the task better check the example below.

Input

The input data is read from the console.
- It consists of a **random number of lines**. The input **ends with the string "stop"**.
- Each line will hold **a string -** representing the **direction** of the **knight's** movement; the vertical and horizontal directions will be separated from each other by a single space.

The input data will always be valid and in the format described. There is no need to check it explicitly.

Output

The output data must be printed on the console.
- On the only output line you must print the non-zero integers from the board**.**
- Each row of the table represents an integer number in binary format.
- If the **whole board** is with zeroes, you should print out "[Board is empty]".

Constraints

- The **number moves** will be in the range [1…25].
- The **direction** will consist of a combination of the following strings: **"left", "right", "up", "down"**.
- Time limit: 0.1 seconds. Allowed memory: 16 MB.

Examples

	7	6	5	4	3	2	1	0	Number
0	0	0	0	0	0	0	0	1	1
1	0	0	0	0	0	1	0	0	4
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	1	0	2
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	1	1
6	0	0	0	0	0	1	0	0	4
7	0	0	0	1	0	0	0	0	16

Input	Output
left down	1
down right	4
right up	2
down right	1
left down	4
left down	16
stop	

Follow us: