

# Robot Experiment

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            1 second  
Memory limit:         512 megabytes

You are developing a type of lunar exploring robot. Now the robot is under experiment on a large field. The field can be regarded as an infinite 2D plane. Initially, the robot is at  $(0,0)$ . Then it will execute  $n$  commands sequentially. Each command can be represented as a single upper-case English letter:

- “L”: Move from  $(x, y)$  to  $(x - 1, y)$ .
- “R”: Move from  $(x, y)$  to  $(x + 1, y)$ .
- “D”: Move from  $(x, y)$  to  $(x, y - 1)$ .
- “U”: Move from  $(x, y)$  to  $(x, y + 1)$ .

Note that there are some obstacles on the field except the origin point  $(0,0)$ . When the robot is trying to move to the next position, it will check whether there is an obstacle at the target coordinate. If there is an obstacle, the robot will stay at the current position, mark the current command as executed, and continue executing the following commands.

You have sent  $n$  commands to the robot, but unfortunately, due to some bugs, the map of the field and the position of the robot can not be transmitted to you. Assume the robot has finished executing all the  $n$  commands. Before going to the field, you are wondering which places may you find the robot. Please write a program to find all the possible positions the robot will finally locate at.

## Input

The first line of the input contains a single integer  $n$  ( $1 \leq n \leq 20$ ), denoting the number of commands.

The second line contains a string  $s$  of length  $n$  ( $s_i \in \{\text{“L”}, \text{“R”}, \text{“D”}, \text{“U”}\}$ ), the  $i$ -th letter denoting the  $i$ -th command sent to the robot.

## Output

First output a single line containing an integer  $k$ , denoting the number of possible positions. Then output  $k$  lines, each line contains two integers  $x$  and  $y$ , denoting the robot can finally locate at  $(x, y)$ . When  $k \geq 2$ , you should print the answers in ascending order of  $x$ , and then in ascending order of  $y$  in case of a tie.

## Examples

standard input	standard output
2 RU	4 0 0 0 1 1 0 1 1
4 LRUD	4 0 -1 0 0 1 -1 1 0