# 02. Wall Destroyer

You will be given an integer **n** for the size of the wall (square shape). On the next **n** lines, you will receive the rows of the wall. Vanko will start at a **random** position, marked with the letter '**V**'. After he makes his **first** **move**, his **initial** position will be **considered a successfully created hole** and you must **mark his starting position with a '\*'**. The **steel rods** and the **cables** will also be on **random** positions. The **rods** will be marked with the letter '**R**' and the **cables** will be marked with '**C**'. All of the **other** positions will be marked with '**-**' (dash).

Until you receive the "**End**" command, on each turn you will be guiding Vanko and telling him the **direction**, which he should move to and make a hole at. The commands will be "**up**", "**down**", "**left**" and "**right**".

If Vanko **manages to create a hole** at the desired location, mark the position with a '**\***'.

If he **hits a rod**, Vanko **returns to his previous** position and **continues** with the **next** directions. Print "**Vanko hit a rod!**" and consider that he did **not** make a **hole**.

If he **hits a cable**, he gets **electrocuted**, the position is marked with an '**E**' and the program **ends**. The **position** that holds the '**E**' letter is **considered a successfully created hole**.

If Vanko lands on a position that **already has a hole on it**, print "**The wall is already destroyed at position [row, col]!**". In case the directions lead Vanko outside of the wall, Vanko **doesn't move at all** and you must **do nothing.**

Keep track of the holes that Vanko manages to create and of the times that he has hit a steel rod.

**The program will end when Vanko gets electrocuted оr the "End" command is given.**

### Input

* On the first line, you are given the integer **n** – the size of the matrix (wall).
* The **next n lines** hold the values for every **row**.
* On each of the next lines, until you receive the "**End**" command, you will get a move command.

### Output

* On the first line:
  + If Vanko manages to make all of the holes, print "**Vanko managed to make {countOfHoles} hole(s) and he hit only {countOfRods} rod(s).**" .
  + If Vanko gets electrocuted, print "**Vanko got electrocuted, but he managed to make** **{countOfHoles} hole(s).**"
* If Vanko lands on a position that already has a hole on it, print "**The wall is already destroyed at position [row, col]!**"
* If Vanko hits a rod, print "**Vanko hit a rod!**".
* At the end, print the **final state** of the matrix (wall) **with Vanko's position on it**.

### Constraints

* The size of the **square** matrix (wall) will be between **[2…10].**
* Vanko's starting position will always be marked with '**V**'.
* There may be cases where the given directions will be outside of the wall.
* There will be always two output scenarios:
  + Vanko manages to make all of the holes, until the "**End**" command;
  + Vanko gets electrocuted.

### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 5  ---C-  -C-R-  -----  -V---  R----  down  left  right  up  left  up  up  right  down | Vanko hit a rod! The wall is already destroyed at position [3, 1]! Vanko got electrocuted, but he managed to make 6 hole(s).  ---C-  -E-R-  -\*---  -\*\*--  R\*\*-- | The first command is "**down**". Vanko moves to the position, its value is '**-**', so he makes a hole there and increases the counter.  ---C-  -C-R-  -----  -\*---  RV---  **Next command** is "**left**", but there is a rod there, so we only print the corresponding message.  Next command is "**right**" and this position holds a '**-**', so Vanko makes a hole there and we increase the counter.  ---C-  -C-R-  -----  -\*---  R\*V--  The next command is "**up**". Vanko moves to the position and we increase the counter.  Next command us "**left**", but **there's already a hole there**, so we print the corresponding message, without increasing the counter.  The next command is "**up**". Vanko moves to the position and we increase the counter.  Next command is "**up**" again and the position holds a cable, so Vanko gets electrocuted. We increase the counter of the created holes and we print the corresponding message and the final state of the wall, before ending the program. |
| 5  --V--  ---R-  CC---  -----  -----  up  down  right  down  right  right  End | Vanko hit a rod!  Vanko managed to make 5 hole(s) and he hit only 1 rod(s).  --\*--  --\*R-  CC\*\*V  -----  ----- | The first command is "**up**", but that means that Vanko will go out of the wall, so we do nothing.  Next command is "**down**" and we increase the counter.  Next command is "**right**", but there is a rod, so Vanko doesn't change his position and we only print the corresponding message and increase the rod hits counter.  Next command is "**down**" and we increase the counter.  Next command is "**right**" and we increase the counter.  Next command is "**right**" and we increase the counter again.  The last command is "**End**", so we end the program and print the corresponing message and the final state of the wall. |
| 4  ----  C--R  -V--  ----  up  right  End | Vanko managed to make 3 hole(s) and he hit only 0 rod(s).  ----  C\*VR  -\*--  ---- |  |