# Miner

We get as input **the size** of the **field** in which our miner moves. The field is **always a square**. After that we receive the commands wich represent the directions in wich the miner should move.The miner **starts** from **s**-position. The commands will be: **left**, **right**, **up** and **down**.

If the miner reaches the side edge of the field (left, right, up or down), it **remains on his current possition**. The possible characters that may appear on the screeen are:

* **\*** – regular position on the field.
* **e** – end of the route.
* **c -** coal
* **s** - the place where the **miner starts**

Each time when miner finds a coal, he collects it, and **replace it with '\*'**. Keep track of the **count of the collected coals**. If the miner collects all coals in the field, the program stops and you have to print the following message: **"You collected all coals! ({rowIndex}, {colIndex})"**.

If the miner **steps at 'e' game is over (the program stops)** and you have to print the following message: **"Game over! ({rowIndex}, {colIndex})"**.

If there are no more commands and none of the above cases happens, you have to print the following message: **"{remainingCoals} coals left. ({rowIndex}, {colIndex})"**.

## Input

* **Field size** – an integer number.
* **Commands to move** the miner – an array of strings separated by **" "**.
* **The field: some of the following characters (\*, e, c, s),** separated by whitespace (" ");

## Output

* There are three types of output:
  + If all the coals have been collected, print the following output: **"You collected all coals! ({rowIndex}, {colIndex})"**
  + If you reached the end, you have to stop moving and print the following line: **"Game over! ({rowIndex}, {colIndex})"**
  + If there are no more commands and none of the above cases happens, you have to print the following message: "{totalCoals} coals left. ({rowIndex}, {colIndex})"

## Constraints

* The **field size** will be a 32-bit integer in the range [0 … 2 147 483 647].
* The field will always have only one 's'.
* Allowed working time for your program: 0.1 seconds.
* Allowed memory: 16 MB.

## Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5  up right right up right  \* \* \* c \*  \* \* \* e \*  \* \* c \* \*  s \* \* c \*  \* \* c \* \* | Game over! (1, 3) |
| 4  up right right right down  \* \* \* e  \* \* c \*  \* s \* c  \* \* \* \* | You collected all coals! (2, 3) |
| 6  left left down right up left left down down down  \* \* \* \* \* \*  e \* \* \* c \*  \* \* c s \* \*  \* \* \* \* \* \*  c \* \* \* c \*  \* \* c \* \* \* | 3 coals left. (5, 0) |