02. Bomb Field

One of the mission that new agents have to complete is called BombField. Your task is to implement the mission into a simple program.

We get as input **the size** of the **field** in which our sapper moves. The field is **always a square**. After that we receive the commands wich represent the directions in wich the sapper should move.The sapper **starts** from **s**-position The commands will be: **left/right/up/down.** If the sapper 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.
- **B** – bomb
- **s** - the place where the **sapper starts**

Each time when sapper finds a bomb, he deactivates it, and **replace "B" with "+"**. Keep track of the **count of the bombs**. Each time you find a bomb, you have to print the following message: **"You found a bomb!".** If sapper **steps at the end of the route game is over (the program stops)** and you have to print the output as shown in the output section. After executing all of the commands there are only 2 possible outcomes (there are not going to be more cases):

- if you found all bombs – you win and the game ends
- if you reach the end point ("e"), you have to stop

Print the corresponding output depending on the case.

Input

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

Output

- There are three types of output:
  - If all of the bombs have cleared print the following output: "**Congratulations! You found all bombs!**"
  - If you reached the end, you have to stop moving and print the following line: **"END! {bombs left} bombs left on the field"**
  - If there are no more commands and none of the above cases happens, you have to print the following message: **"{bombs left} bombs left on the field. Sapper position: ({row},{col})"**

Constraints

- The **input numbers** will be a 32-bit integer in the range [0 … 2 147 483 647].
- Allowed working time for your program: 0.1 seconds.
- Allowed memory: 16 MB.

Examples

Input	Output	Comments
5 **up,right,right,up,right** + + + B + + + + e + + + B + + s + + B + + + B + +	You found a bomb! END! 3 bombs left on the field	After executing all of the commands,sapper move to the **end point.**But there are some boms left, so we print the information.
4 **up,right,right,right,down** + + + e + + B + + s + B + + + +	You found a bomb! You found a bomb! Congratulations! You found all bombs!	The sapper reached the **end of the field**, so he remains hes position and we print the message. After all the bombs are found, we should stop the program and print the appropriate message.
6 **left,left,down,right,up,left,left,down,down,down** + + + + + + e + + + B + + + B s + + + + + + + + B + + + B + + + B + + +	You found a bomb! You found a bomb! 3 bombs left on the field. Sapper position: (5,0)	The sapper finds a bomb, then he steps on it **again**, but it **does not count**. After executing all comands sapper did not reach the end and did not found all bombs, so we print the appropriate message.