# Problem 3 – Rubik’s Matrix

Rubik’s cube – everyone’s favorite head-scratcher. Writing a program to solve it will be quite a difficult task for an exam, though. Instead, we have a Rubik’s matrix prepared for you. You will be given a pair of dimensions: **R** and **C.** To prepare the matrix, fill each row with increasing integers, starting from one. For example, **2 x 4** matrix must look like this:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |

Next, you will receive series of commands, indicating which row or column you must move, in which direction, and how many times. For example, **1 up 1** means: column 1, direction: up, 1 move. After executing it, the matrix should look like:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 6 | 3 | 4 |
| 5 | 2 | 7 | 8 |

Directions **left** and **right** mean you must move the corresponding **row**, while **up** and **down** and related to the **columns.** After shuffling the Rubik’s matrix, you have to **rearrange** it (meaning that the **values in each cell** should be in **increasing order**, such as the ones in the original matrix). The rearranging process should start at **top-left** and end at **bottom-right**. Find the **position** of the value you need, and print the **swap** **command** on the console, in the format described below.

### Input

* On the first line, you are given the integers **R** and **C**, separated by a space.
* On the second line, you are given an integer **N**, indicating the number of commands to follow
* On the next N lines, you are given commands in format **{row/col} {direction} {moves}**

### Output

* On **R** \* **C** number of lines, print the **swap commands** needed to rearrange the matrix, either:
  + **Swap ({row}, {col}) with ({row}, {col})** or
  + **No swap required**

### Constraints

* **R, C, N** are integers in range [1 … 100]
* {**row**} and {**col**} will always be inside the matrix
* {**moves**} is in range [0 … 231-1]
* Allowed time and memory: 0.25s / 16 MB

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 3  2  1 down 1  1 left 1 | No swap required  Swap (0, 1) with (1, 0)  No swap required  Swap (1, 0) with (1, 2)  Swap (1, 1) with (2, 1)  Swap (1, 2) with (2, 1)  No swap required  No swap required  No swap required |

123 183 183

456 426 264

789 759 759

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 3  2  0 down 3  0 left 3 | No swap required  No swap required  No swap required  No swap required  No swap required  No swap required  No swap required  No swap required  No swap required |

3 3

2

0 up 1

0 **down** 1