Task – Fill Matrix

You are given a matrix (2D array) of lowercase alphanumeric characters (**a-z**, **0-9**), a starting position – defined by a start row **startRow** and a start column **startCol** – and a filling symbol **fillChar**. Let’s call the symbol originally at **startRow** and **startCol** the **startChar**. Write a program, which, starting from the symbol at **startRow** and **startCol**, changes to **fillChar** every symbol in the matrix which:

* is equal to **startChar** AND
* can be reached from **startChar** by going up (**row – 1**), down (**row + 1**), left (**col – 1**) and right (**col + 1**) and “stepping” ONLY on symbols equal **startChar**

So, you basically start from **startRow** and **startCol** and can move either by changing the row OR column (not both at once, i.e. you can’t go diagonally) by **1**, and can only go to positions which have the **startChar** written on them. Once you find all those positions, you change them to **fillChar**.

In other words, you need to implement something like the Fill tool in MS Paint, but for a 2D char array instead of a bitmap.

Input

On the first line, two integers will be entered – the number **R** of rows and number **C** of columns.

On each of the next **R** lines, **C** characters separated by single spaces will be entered – the symbols of the **R**th row of the matrix, starting from the **0**th column and ending at the **C-1** column.

On the next line, a single character – the fillChar – will be entered.

On the last line, two integers – startRow and startCol – separated by a single space, will be entered.

Output

The output should consist of R lines, each consisting of exactly C characters, NOT SEPARATED by spaces, representing the matrix after the fill operation has been finished.

Restrictions

**0 < R,** **C < 20**   
**0 <= startRow < R**   
**0 <= startCol < C**

All symbols in the input matrix will be lowercase alphanumerics (**a-z**, **0-9**). The **fillChar** will also be alphanumeric and lowercase.

The total running time of your program should be no more than **0.1s**

The total memory allowed for use by your program is **5MB**

Example I/O

|  |  |
| --- | --- |
| Example Input | Expected Output |
| 5 3  a a a  a a a  a b a  a b a  a b a  x  0 0 | xxx  xxx  xbx  xbx  xbx |
| 5 3  a a a  a a a  a b a  a b a  a b a  x  2 1 | aaa  aaa  axa  axa  axa |
| 5 6  o o 1 1 o o  o 1 o o 1 o  1 o o o o 1  o 1 o o 1 o  o o 1 1 o o  3  2 1 | oo11oo  o1331o  133331  o1331o  oo11oo |
| 5 6  o o o o o o  o o o 1 o o  o o 1 o 1 1  o 1 1 w 1 o  1 o o o o o  z  4 1 | oooooo  ooo1oo  oo1o11  o11w1z  1zzzzz |
| 5 6  o 1 o o 1 o  o 1 o o 1 o  o 1 1 1 1 o  o 1 o w 1 o  o o o o o o  z  4 0 | z1oo1z  z1oo1z  z1111z  z1zw1z  zzzzzz |