

C++ Fundamentals: Judge Assignment 2 (JA2)

The following tasks should be submitted to the SoftUni Judge system, which will be open starting **Saturday, 9 December 2017, 10:00** (in the morning) and will close on **Saturday, 23 December 2017, 23:59**. Submit your solutions here: <https://judge.softuni.bg/Contests/Compete/Index/878>.

After the system closes, you will be able to “Practice” on the tasks – however the “Practice” results are NOT considered in the homework evaluation.

For this assignment, the code for each task should be a single C++ file, the contents of which you copy-paste into the Judge system.

Please be mindful of the strict input and output requirements for each task, as well as any additional requirements on running time, used memory, etc., as the tasks are evaluated automatically and not following the requirements strictly may result in your program’s output being evaluated as incorrect, even if the program’s logic is mostly correct.

You can use C++03 and C++11 features in your code.

Unless explicitly stated, any integer input fits into **int** and any floating-point input can be stored in **double**.

NOTE: the tasks here are NOT ordered by difficulty level.

Task 4 – Fill Matrix (JA2-Task-4-The-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 **Rth** row of the matrix, starting from the **0th** 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

$0 < R, C < 20$

$0 \leq \text{startRow} < R$

$0 \leq \text{startCol} < C$

All symbols in the input matrix will be lowercase alphanumeric (**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	xxx
a a a	xxx
a a a	xbx
a b a	xbx
a b a	xbx

a b a x 0 0	
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