# Multidimensional Arrays – Exercise

This document defines the exercises for the ["C++ Advanced" course @ Software University](). Please submit your solutions (source code) to all below-described problems in [Judge]().

Write C++ code for solving the tasks on the following pages.

Code should compile under the C++03 or the C++11 standard.

## 1. Compare Matrices

Write a program that reads two integer matrices (2D arrays) from the console and compares them element by element. For better code reusability, you could do the comparison in a function, which returns **true** if they are equal and **false** if not.

Each matrix definition on the console will contain a line with a positive integer number **R** – the number of rows in the matrix – followed by R lines containing the numbers in the matrix, separated by spaces (each line will have an equal amount of numbers.

The matrices will have at most **10** rows and most **10** columns.

Print **equal** if the matrices match, and **not equal** if they do not match.

### Examples

| Input | Output |
|---|---|
| 1<br>1 2 3<br>1<br>1 2 3 | equal |
| 2<br>1 2 3<br>2 1 3<br>2<br>1 2 3<br>2 1 3 | equal |
| 4<br>1<br>11<br>21<br>31<br>4<br>1<br>11<br>21<br>31 | equal |
| 2<br>1 2 3<br>4 5 6<br>2<br>1 3 2<br>4 5 6 | not equal |
| 2 | not equal |

| | |
|---|---|
| 1 2 3<br>4 5 6<br>2<br>1 2<br>3 4 | |

# 2. Positions Of

Write a program that reads a matrix of integers from the console, then a number, and prints all the positions at which that number appears in the matrix.

The matrix definition on the console will contain a line with two positive integer numbers **R** and **C** – the number of rows and columns in the matrix – followed by **R** lines, each containing **C** numbers (separated by spaces), representing each row of the matrix.

The number you will need to find the positions of will be entered on a single line, after the matrix.

You should print each position on a single line – first print the row, then the column at which the number appears.

If the number does not appear in the matrix, print **not found.**

## Examples

| Input | Output |
|---|---|
| 2 3<br>1 2 3<br>4 5 6<br>3 | 0 2 |
| 2 3<br>1 2 3<br>4 2 2<br>2 | 0 1<br>1 1<br>1 2 |
| 2 3<br>1 -2 -3<br>4 -5 6<br>5 | not found |

# 3. 2X2 Squares in Matrix

Find the count of **2 x 2 squares of equal chars** in a matrix.

## Input

- On the **first line**, you are given the integers **rows** and **cols** – the matrix's dimensions.
- Matrix characters come at the next **rows** lines (space separated).

## Output

- Print the number of all the squares matrixes you have found.

Follow us:

## Examples

| Input | Output | Comments |
|---|---|---|
| 3 4<br>A B B D<br>E B B B<br>I J B B | 2 | Two 2 x 2 squares of equal cells:<br>A **B B** D    A B B D<br>E **B B** B    E B **B B**<br>I J B B    I J **B B** |
| 2 2<br>a b<br>c d | 0 | No 2 x 2 squares of equal cells exist. |

# 4. Snake Moves

You are walking in the park and you encounter a snake! You are terrified, and you start running zig-zag, so the snake starts following you.

You have a task to visualize the snake's path in a square form. A **snake** is represented by **a string**. The **isle** is a **rectangular matrix of size NxM**. A snake starts going down from the **top-left corner** and slithers its way down. The first cell is filled with the first symbol of the snake, the second cell is filled with the second symbol, etc. The snake is as long as it takes to **fill the stairs completely** – if you reach the end of the string representing the snake, start again at the beginning. After you fill the matrix with the snake's path, you should print it.

## Input

- The input data should be read from the console. It consists of exactly two lines.
- On the first line, you'll receive the **dimensions** of the stairs in the format: **"N M"**, where **N** is the number of **rows**, and **M** is the number of **columns**. They'll be separated by a single space.
- On the second line, you'll receive the string representing the **snake.**

## Output

- The output should be printed on the console. It should consist of **N lines.**
- Each line should contain a string representing the respective row of the matrix.

## Constraints

- The **dimensions** N and M of the matrix will be integers in the range [1 … 12].
- The **snake** will be a string with a length in the range [1 … 20] and **will not contain any whitespace characters.**

## Examples

| Input | Output |
|---|---|
| 5 6<br>SoftUni | SoftUn<br>UtfoSi<br>niSoft<br>foSinU<br>tUniSo |