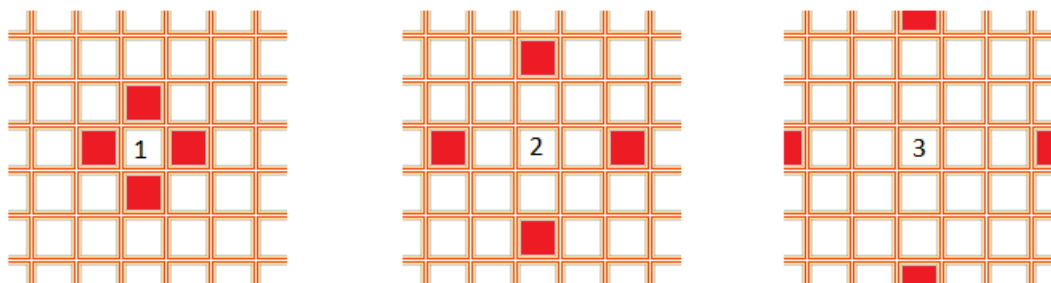


## Problem 2 – Portals

You are given a **2D labyrinth** (because three-dimensional arrays are not cool, God bless the trainers (or at least the one who wrote this task (or at least if you can solve it))). The labyrinth is divided into **cubes**, each with dimensions 1x1. You **cannot move** from one cube to another (Wut?! No moving? ;( ), because there is solid concrete between all of the cubes (this means one cannot escape the labyrinth either, but scientifically looking on the problem who cares? Watch the movie Cube<sup>2</sup> and you will understand (or not)).

Luckily for the person inside the labyrinth, there are portals almost in all cubes. Each portal have **power number**, showing how far it can teleport someone from the current cell. The portal can teleport a person in all four directions – **up, down, right, left** (no diagonals). Each **portal can be used only once**, afterwards it becomes inactive. Also, the portal **cannot teleport someone outside of the labyrinth**. Here are some examples of valid moves depending on the power level:



In the labyrinth the number of rows in each level is **R** and the number of columns in each level is **C**.

Each **cube can be defined by its position** – the row and the column on which the cube is in. Rows and columns are numbered beginning from 0. The corner cubes in clockwise order are as follows – (0,0), (0,C-1), (R-1,C-1), (R-1,0).

**Some of the cubes** in the labyrinth are **impassable**. They are filled with frogs and no one would like to teleport in the middle of a room full of small green amphibians. Besides that there is not enough space for a human to appear right in the middle of the messy and disgusting room and the portal will throw an unhandled **InvalidOutOfSpaceOverflowException** before it starts teleporting.

Find the **maximum sum of teleportation power one can use** in the labyrinth, given a starting location and the cubes in the labyrinth.

Now find a person by your choice, put him/her inside the labyrinth and call his/her phone saying "Hello {0}, I wanna play a game!"

### Input

The input data should be read from the console.

On the first line there will be the numbers **X, Y** separated by whitespace. These numbers represent the **starting location** – **X** is the starting row, **Y** is the starting column.

On the next line, of the standard input, there will be the numbers **R** and **C**, separated by whitespace.

On the following lines, until the end of the input, there will be **a descriptions** of an **RxC matrix**. The matrix is represented on exactly **R lines**, each containing exactly **C cubes, separated by whitespace**.

Each symbol in a matrix description can be:

- **Non-negative integer** – meaning an cube with portal having power equals to the integer, '#' – meaning an impassable cube, full of frogs

The input data will always be valid and in the format described. There is no need to check it explicitly.

### Output

The output data should be printed on the console.

On the only output line, print the **maximum sum of teleportation power** one can use in the labyrinth.

### Constraints

- **R and C** will be between 1 and 10, inclusive. **X and Y** will be between 0 and 9, inclusive.
- The start position will never be an impassable cube ('#').
- Allowed working time for your program: 0.3 seconds.
- Allowed memory: 32 MB.

### Examples

Example input	Example output	Explanation
0 0 5 6 1 # 5 4 6 4 3 2 # 2 6 2 9 1 7 6 3 1 8 2 7 3 8 6 3 6 1 3 1 2	23	<u>1</u> # 5 <u>4</u> 6 4 <u>3</u> <u>2</u> # <u>2</u> 6 2 9 1 7 6 3 1 8 <u>2</u> 7 <u>3</u> 8 6 <u>3</u> 6 1 <u>3</u> 1 2