Path

SEASON 8 – FINAL

Let's consider a table consisting of *N* rows and *M* columns. The rows and columns are numbered starting from 1. There is a number written in every cell of the table and we will call those numbers the ***values*** of the cells. Some of the cells are ***blocked*** and so the number **-1** is written in them. The values of all other cells are positive integers. Two cells are considered ***adjacent*** if they have a common border, i.e. cell (*x*,*y*) is adjacent to (*x* + 1, *y*), (*x* – 1, *y*), (*x*, *y* – 1) and (*x*, *y* + 1), when the corresponding cells exist.

Path in the table is defined as a non-empty sequence of ***different non-blocked cells***, such that they are consecutively adjacent. Unfortunately, we are afraid of long paths, so the length of every path shouldn't exceed *MAXL*.

A ***subsequence*** of a path is a sequence that can be derived from the given path by deleting zero or more elements without changing the order of the remaining elements. ***The longest increasing subsequence*** of a path is the longest sequence $\{A_1, A_2, \ldots, A_M\}$, such that for every $1 < i < M$ we have $A_{i-1} < A_i < A_{i+1}$. Similarly, ***the longest decreasing subsequence*** is the sequence $\{A_1, A_2, \ldots, A_M\}$, such that for every $1 < i < M$ we have $A_{i-1} > A_i > A_{i+1}$. We will denote the lengths of these two sequences for a path ***P*** as *LIS*(***P***) and *LDS*(***P***).

The ***value*** of a path ***P*** is defined as *LIS*(***P***).*LDS*(***P***).

Write a program **`path`**, that finds a path *P* such that its value is as large as possible.

Input:

The first line of the input file **`path.in`** will contain the integers *N* and *M*. Each of the next *N* lines will contain *M* numbers – the values of the cells стойностите от поредния ред на таблицата, започвайки от реда с номер 1. На последния ред на входния файл ще бъде зададено цялото число *MAXL*.

Output:

On the first line of the output file **`path.out`** the program should print the length *K* of the path found by you. On each of the following *K* lines print the coordinates - number of row and number of column - of the corresponding cell in the path.

Scoring:

If the output does not fulfill the restrictions above, you will receive 0 points for the test. Otherwise, you will receive $score \times \left(\frac{yours}{best}\right)^2$ points, where $score$ is the number of points the test is worth, $yours$ is the value of the path, obtained by you, and $best$ is the maximum value of a path among all participants for the given test.

Path

Constraints:

The values of the non-blocked cells are integers in the interval $[1; 10^9]$

Portion of tests	Constraints on N, M and MAXL
10%	$N \leq 20, M \leq 20, MAXL \in [1; 100]$
40%	$N \leq 100, M \leq 100, MAXL \in [1; 40000]$
50%	$N \leq 1000, M \leq 1000, MAXL \in [1; 300000]$

Time limit: 5 sec
Memory limit: 256 MB

Example test

Input (path.in)	Output (path.out)
3 3 **1 3** -1 -1 **2 1** -1 1 1 4	4 1 1 1 2 2 2 2 3

Note:

The found path **P** contains the values {1, 3, 2, 1} in this order. This means that *LIS(**P**)* = 2, *LDS(**P**)* = 3 and its cost is equal to 6.