# International Olympiad in Informatics
# National Olympiad in Informatics
# Hong Kong Delegation Team Formation Test 2011

## Task Overview Sheet

| | **mars** | **tetris** | **stones** | **flow** |
|---|---|---|---|---|
| Input file | `stdin` | | | |
| Output file | `stdout` | | | |
| Time limit per test case | 4 seconds | 1 second | 2 seconds | 1 second |
| Memory limit | 128MB | 64MB | 32MB | 64MB |
| Compiler options in C | `-lm -w -O2 -static -static-libgcc` | | | |
| Compiler options in C++ | `-lm -w -O2 -static -static-libgcc` | | | |
| Compiler options in Pascal | `-So -XS -O2` | | | |
| Number of test cases | 10 | 10 | 10 | 20 |
| Maximum points per test case | 10 | 10 | 10 | 5 |
| Maximum total points | 100 | 100 | 100 | 100 |

**WARNING:** **`iostream`** is sometimes too slow for input reading/output writing. **`C++`** coders are advised to use **`cstdio`** instead.

**Example:** "**`cin >> n`**" can be written as "**`scanf("%d", &n)`**" for **`int`** variable **`n`**.
"**`cout << n << endl`**" can be replaced by "**`printf("%d\n", n)`**" for **`int`** variable **`n`**.

# Mars Exploration

## PROBLEM

Dr. Jones now works at Hong Kong Outer-space Institute (HKOI). He is responsible for a research in Mars, so he must send spaceships to explore every square of land in Mars.

The whole planet can be modeled as a $N \times N$ grid. Each square in the grid has a certain height $H_{ij}$. Two squares are adjacent if they share an edge. A spaceship can travel from one square to an adjacent square if the absolute difference of their heights is not more than $K$. Moreover, when a spaceship travels to an adjacent square with difference of height of $d$, it uses $d$ liters of fuel. Luckily, when Dr. Jones was chased by Tough Fighting Tiger, he invented a rebounding system: when a spaceship travels back to a square it visited before, it consumes **no** fuel.

Since Mars is made up of mountains, possibly more than one spaceship is needed. Each spaceship will have its own responsible area, where it must travel to every square in the area using the fuel it has. Now, to complete the exploration in Mars, Dr. Jones must fulfill these two conditions:

(1) He must send the minimum number of spaceships $S$.

(2) When $S$ spaceships are sent, each spaceship must carry minimum liters of fuel $L$.

Note that as all spaceships are homogenous (the same), each spaceship will carry the same amount of fuel, even if it cannot use up the fuel.

Moreover, as this project is still in the planning stage, Dr. Jones is not sure about how capable a spaceship can travel in Mars. Therefore, $K$ may change. You must find out for some values $K$, the numbers $S$ and $L$ described above.

## INPUT

The first line contains two integers: $N$, $Q$, the dimension of Mars and the number of queries, which describes $K$ in each situation.

The following $N$ lines contain $N$ integers each. Each integer describes the height $H_{ij}$ of that square.

Lastly, there are $Q$ lines. Each line has one integer $K$, represents the maximum difference of height a spaceship can travel.

## OUTPUT

You should output ***Q*** lines. Each line contains two integers ***S*** and ***L*** for the corresponding ***K***.

Note that the answer may be out of the range of a 32-bit integer. For C/C++ users, please use `%lld` to print instead of `%I64d`.

## SAMPLE INPUT

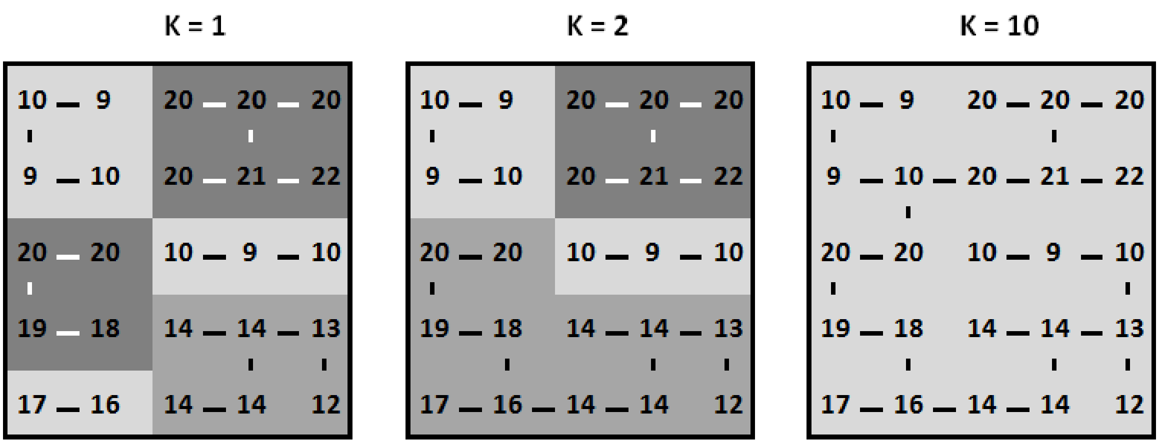
```
5 3
10 9 20 20 20
9 10 20 21 22
20 20 10 9 10
19 18 14 14 13
17 16 14 14 12
1
10
2
```

## SAMPLE OUTPUT

```
6 3
1 39
4 9
```

## EXPLANATION

Shading represents a spaceship's responsible area. Lines within an area are the route of a spaceship. For ***K=1***, the spaceship at the upper left corner uses 3 liters of fuel. For ***K=2***, the spaceship at the bottom uses 9 liters of fuel.



## CONSTRAINTS

In all of the test cases,

- $1 \le N \le 800$
- $1 \le Q \le 100{,}000$
- $0 \le K, H_{ij} \le 1{,}000{,}000$

In test cases that worth 50% of the points,

- $1 \le N \le 50$
- $1 \le Q \le 100$
- $0 \le K, H_{ij} \le 1000$

In test cases that worth 20% of the points,

- $Q = 1$

## SCORING

For each test case, if the first integer is correct on every line, or the second integer is correct every line, you get half of the maximum points in that test case. For example, for the sample input,

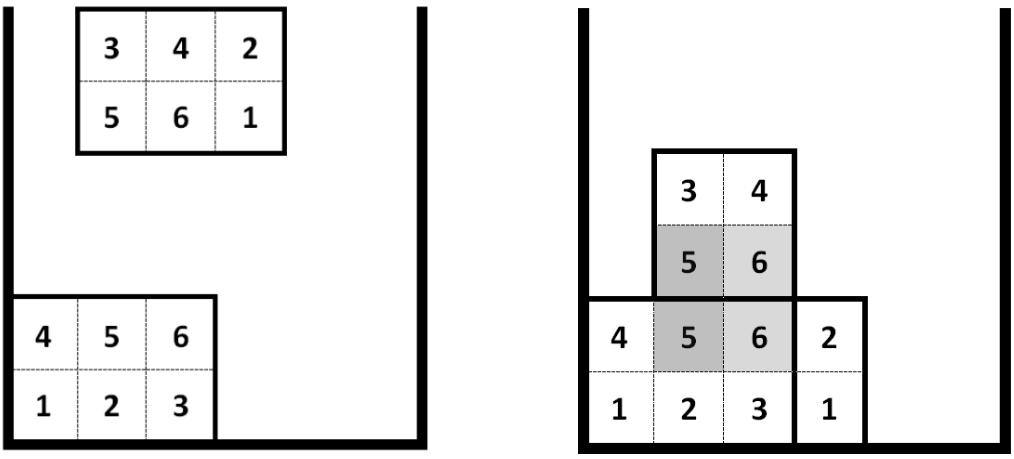
| Sample Output | Points | Explanation |
|---|---|---|
| 6 3<br>1 39<br>4 9 | 10 | Both integers are correct on every line |
| 6 100<br>1 0<br>4 0 | 5 | Only the first integer is correct on every line |
| 6 3<br>1 39<br>0 9 | 5 | Only the second integer is correct on every line |
| 6 3<br>1 39<br>4 | 0 | Although the first integer is correct on every line, the output is of wrong format (two integers per line) |

# Tetrisudoku

## PROBLEM

Dr. Jones is really bored of watching those spaceships travelling in Mars. He is also bored of playing Tetris (俄羅斯方塊) and Sudoku (數獨). That's why he invented Tetrisudoku, which is a combination of both, to waste his time.

In the game, there is a playing field of width **6** and infinite height. ***N*** 2x3 Sudoku pieces will fall down one by one in order. A piece actually consists of number 1 to 6 each appears exactly once. Dr. Jones can move a piece horizontally before it touches other pieces (we call it 'landing'). Once the piece is landed, each column of the piece will move down until it touches other pieces, or reach the bottom.




For each column of the new Sudoku piece, if the number at the lower row is the **same** as the number lying right below it (which must belong to another Sudoku piece), Dr. Jones gets 1 point. Dr. Jones is smart, so it is natural for him to get the maximum points. You are therefore required to calculate how many points he gets.

## INPUT

The first line contains one integer ***N***, the number of Sudoku pieces.

Every two of the following ***2N*** lines contain 3 integers each. These two lines represent one Sudoku piece. The 1st piece is at the beginning of the input. The ith coming piece is described on the ***2i*** and ***2i+1*** lines.

## OUTPUT

One line with one integer, which is the maximum score of the game.

## SAMPLE INPUT

```
2
4 5 6
1 2 3
3 4 2
5 6 1
```

## SAMPLE OUTPUT

```
2
```

## EXPLANATION

The above sample input corresponds to the pictures in the problem statement.

## CONSTRAINTS

In all of the test cases,

- $1 \leq \boldsymbol{N} \leq 1000$

In test cases that worth 60% of the points,

- $1 \leq \boldsymbol{N} \leq 40$

In test cases that worth 20% of the points,

- $1 \leq \boldsymbol{N} \leq 10$

# Stones Rearrangement

## PROBLEM

Mars is really a fascinating place. It was even more exciting when the spaceships have collected some stone samples from Mars. Dr. Jones decides to stop playing Tetrisudoku and focus on the analysis of the stone samples picked up by the spaceships in Mars.

A spaceship has collected ***N*** stones. Dr. Jones wishes to put the stones in a particular order. He controls the spaceship to do so by transmitting the instructions. However, transmission of data from Earth to Mars is expensive, but from Mars to Earth is cheap. Therefore, Dr. Jones invents a smart way to instruct the spaceship.

For simplicity, the stones are indexed from ***1*** to ***N***. At first, the spaceship sends an arrangement scheme of stones to Dr. Jones, in form of an array of integers ***1*** to ***N*** each appears exactly once (a permutation), and receives a number $A_i$ from Dr. Jones. Then the spaceship sends another permutation and receives another number. This process continues until at last, when the spaceship receives a number ***0***, the process terminates as the stones are arranged in Dr. Jones' order.

The number $A_i$ is actually the number of inversions between the current permutation and the correct permutation. The number of inversions is the number of pairs that is out of order relative to the correct sequence. In other words, let **X** and **Y** be two integers. If **X** comes **before** **Y** in the current permutation while **X** comes **after** **Y** in the correct permutation, then the pair **X** and **Y** is counted as one inversion.

You are asked to write a program for the spaceship so that data transmission is minimized, i.e. guess the correct permutation of stones in the minimum number of times.

## INPUT

The first line of input contains an integers ***N***.

## INTERACTION

Your program should output a permutation of stones with their ID ***1*** to ***N*** separated by a space. Then, your program should read an integer ***A***, representing the number of inversions of your guess.

Your program should terminate when it receives ***A=0***, i.e. the correct permutation is guessed.

You are advised to flush the output after writing every line to it. C/C++ users can use `fflush(stdout);` Pascal users can use `Flush(Output);`

**SAMPLE INPUT AND INTERACTION**

| Standard Input | Standard Output | Explanation |
|---|---|---|
| 4 | | $N = 4$ |
| | 1 2 3 4 | Permutation guessed = {1,2,3,4} |
| 4 | | Number of inversions = 4 |
| | 2 3 4 1 | Permutation guessed = {2,3,4,1} |
| 3 | | Number of inversions = 3 |
| | 3 4 2 1 | Permutation guessed = {3,4,2,1} |
| 1 | | Number of inversions = 1 |
| | 3 4 1 2 | Permutation guessed = {3,4,1,2} |
| 0 | | This is the correct permutation. Your program should terminate. You have guessed 4 times in total. |

**SCORING**

There are 10 set of test cases. Each set contains one or more independent test cases. For each test case in the set, if your program terminates correctly within the time limit, and the number of guesses (including the last correct guess) does not exceed ***M***, you will score full marks in the set. Otherwise, you get no mark.

| Test Case Set | $N$ | $M$ |
|---|---|---|
| 1 | 5 | 10000 |
| 2 | 10 | 10000 |
| 3 | 20 | 1000 |
| 4 | 30 | 1000 |
| 5 | 50 | 600 |
| 6 | 50 | 600 |
| 7 | 50 | 300 |
| 8 | 50 | 100 |
| 9 | 100 | 100 |
| 10 | 100 | 100 |

# Current Flow

## PROBLEM

Dr. Jones dreams to travel to Mars one day, but he must first make sure the atmosphere is not too bad for people to live in. One important indicator is the current in the air.

In a particular $N \times M$ area in Mars, the air pressure of each square is known. Air particles on Earth tend to move from high pressure to low pressure. On Mars, however, the particles Potassium Iohydroxide (HKOI) move from low pressure to high pressure. If there is an adjacent square (which shares an edge) with higher pressure than the current one, the particle moves towards it. If there are several such squares, the particle moves to the one with highest pressure. If there is no adjacent square with higher pressure, the particle stays at its original square. Every particle moves at 1 square per second.

For some HKOI particles, Dr. Jones places a micro-tracer in each of the nucleus at time 0. Then, he makes some observations regarding the micro-tracers at a certain time $T$. You need to find out where the particles are at that particular time.

## INPUT

The first line contains 4 integers $N$, $M$, $Q$ and $T$, the dimensions $N \times M$ of the area, the number of observations made and the time when the observations are made.

Following $N$ lines contain $M$ integers each. Each integer represents the air pressure of that square, the larger the number, the higher the temperature. The coordinates of the grid are in form of *(row, col)*. The top row is $1$ and the bottom is $N$; the leftmost column is $1$ at the left and the rightmost is $M$. Assume there is exactly one particle at every square initially at time 0, and the air pressure will never change.

Then $Q$ lines follow. Each line has 2 integers $R_i$ and $C_i$. It is a query in this form: “At time $T$, where the particle initially at $(R_i, C_i)$ will be at?”.

## OUTPUT

For each of the $Q$ line, output two integers, representing the coordinates $(R_i', C_i')$ to the respective observation.

## SAMPLE INPUT

```
2 4 2 2
2 4 6 7
1 3 5 8
2 1
1 4
```

## SAMPLE OUTPUT

```
2 3
2 4
```

## EXPLANATION

The particle at *(2,1)* initially will move to *(2,2)* then *(2,3)* at time 1 and 2 respectively.
The particle at *(1,4)* initially will move to *(2,4)* and then stay there.

## CONSTRAINTS

In all of the test cases,

- $1 \le N, M \le 1000$
- $1 \le AP_{ij} \le 10^9$, where $AP_{ij}$ is the air pressure at every square
- $1 \le Q \le 100{,}000$
- $0 \le T \le 10^9$
- $1 \le R_i \le N$
- $1 \le C_i \le M$
- All $AP_{ij}$ have distinct values

In test cases that worth 50% of the points,

- $1 \le N, M \le 50$