

ACM ICPC World Finals Warmup 2

At UVa Online Judge

7th May 2011
You get 14 Pages
10 Problems
&
300 Minutes

A

Unlock

Input: Standard Input
Output: Standard Output



You are about to finish your favorite game ... (put the name of your favorite game here). Now there is just one hard quest before you can unlock the last level. Or a shortcut. As the quest is too hard you were wondering whether you can solve the shortcut. The shortcut is to solve a puzzle. The puzzle is given as an $n*m$ grid of integer from 1 to $m*n$. You have to rearrange it so that the grid is sorted. That is the first row contains 1 to n . Second row contains $n+1$ to $2n$. And so on. Thus last row contains $mn-n+1$ to mn .

The only allowed operation is pressing a switch. After some trial and error you figured out how the switch works. It rearranges the element by reading them by one diagonal after another. And put them back row wise. For example the following grid will be read as 1-2-10-11-9-15-16-3-14-12-13-4-8-17-7-5-18-19-6-20 and it will be transformed into the grid shown in below.

1	→	2	↗	15	→	16	↗	17
	↘		↖		↘		↖	
10	↓	9	↗	3	↗	8	↗	7
	↘		↖		↘		↖	
11	↓	14	↗	4	↗	5	↗	6
	↘		↖		↘		↖	
12	→	13	↗	18	→	19	↗	20
	↘		↖		↘		↖	

1	2	10	11	9
15	16	3	14	12
13	4	8	17	7
5	18	19	6	20

Now you are wondering given the initial configuration how long it will take to solve the puzzle or it is impossible to solve.

Input

Input starts with an integer $T \leq 70$. T test cases follow.

Each test case starts with two positive integer $m, n \leq 200$. You may assume that the m, n will be such that any solvable puzzle of grid size $m \times n$ can be solved within 2×10^{18} steps. Then follows m lines, each of these lines contains n integers. j -th integer of i -th is (i, j) th entry of the grid. You may assume that the grid consists of all number between 1 to $(n \times m)$ exactly once.

Output

For each case print one line containing number of steps needed to reach the solution or -1 if solution cannot be reached.

Sample Input

```
4
3 4
1 2 3 4
5 6 7 8
9 10 11 12
3 4
1 2 6 7
3 5 8 11
4 9 10 12
3 4
1 2 3 11
5 6 10 9
8 7 4 12
3 4
1 2 5 9
6 3 4 7
10 11 8 12
```

Output for Sample Input

```
0
1
3
5
```

Problem Setter: Tanaeem M. Moosa, Special Thanks: Derek Kisman

B

K-Transformed Permutations

Input: Standard Input
Output: Standard Output



Consider a sequence of n integers $\langle 1\ 2\ 3\ 4\ \dots\ n \rangle$. Since all the values are distinct, we know that there are n factorial permutations. A permutation is called *K-transformed* if the absolute difference between the original position and the new position of every element is at most K .

Given n and K , you have to find out the total number of *K-transformed* permutations.

Example: $n = 4, K = 2$

	<u>1</u> <u>2</u> <u>3</u> <u>4</u>	<u>Valid</u>	<u>Annotation</u>
	(position)		
P ₁	1 2 3 4	Yes	The original sequence. All the elements are in their original position
P ₂	1 2 4 3	Yes	3 and 4 are reordered, but each is shifted by 1 position only.
P ₃	1 3 2 4	Yes	
P ₄	1 3 4 2	Yes	2 is shifted by 2 positions. $2 \leq K$, so it's a valid one.
P ₅	1 4 2 3	Yes	
P ₆	1 4 3 2	Yes	
P ₇	2 1 3 4	Yes	
P ₈	2 1 4 3	Yes	
P ₉	2 3 1 4	Yes	
P ₁₀	2 3 4 1	No	1 is shifted by 3 positions. $3 > K$ and so this is an invalid permutation
P ₁₁	2 4 1 3	Yes	
P ₁₂	2 4 3 1	No	
P ₁₃	3 1 2 4	Yes	
P ₁₄	3 1 4 2	Yes	
P ₁₅	3 2 1 4	Yes	
P ₁₆	3 2 4 1	No	
P ₁₇	3 4 1 2	Yes	
P ₁₈	3 4 2 1	No	
P ₁₉	4 1 2 3	No	4 is shifted by 3 positions. $3 > K$ and so this is also invalid
P ₂₀	4 1 3 2	No	
P ₂₁	4 2 1 3	No	
P ₂₂	4 2 3 1	No	
P ₂₃	4 3 1 2	No	
P ₂₄	4 3 2 1	No	Here, both 4 and 1 are breaking the property.

So, for the above case, there are 14 *2-transformed* permutations.

Input

The first line of input is an integer $T(T < 20)$ that indicates the number of test cases. Each case consists of a line containing two integers n and K . ($1 \leq n \leq 10^9$) and ($0 \leq K \leq 3$).

Output

For each case, output the case number first followed by the required result. Since the result could be huge, output result modulo 73405.

Sample Input

```
3
4 2
100 0
10 1
```

Output for Sample Input

```
Case 1: 14
Case 2: 1
Case 3: 89
```

Problem Setter: Sohel Hafiz, Special Thanks: Mahbubul Hasan

C

UVa Panel Discussion

Input: Standard Input
Output: Standard Output



The UVa online judge team is arranging a panel discussion for the next ACM-ICPC World Finals event in Orlando, Florida. They want that three or four of the contestants take part in the panel and as they have about 300 persons for selecting such a little group, they have decided to put some restrictions in order to reduce the number of possibilities.

After thinking about several options, they finally propose that in case the number of contestants to choice be 3, all of them must be of the same country or from three different countries; and in case the number be 4, at least three of them will be of the same country or must be from at least three different countries.

Could you help them to calculate the number of different selections they can make following the restrictions above.

Input

The input file contains several test cases; each of them consists of two lines.

The first contains two integers N and M separated by one space. N ($3 \leq N \leq 300$) is the number of contestants and M ($1 \leq M \leq 50$) the total number of different countries. The second line consists of N integers between 1 and M , separated by a space, representing the country each contestant is from (It is not necessary that contestants will be from M countries).

Last line of the input will contain two zeroes and it won't be processed.

Output

For each input case write, in a line by itself, two integers separated by a space.

The first integer being be the number of ways to select a group of three people, and the second the number of ways to do it of four people.

Sample Input

```
3 5
5 4 2
5 3
3 1 3 2 2
10 10
1 8 9 1 6 7 3 4 10 4
0 0
```

Output for Sample Input

```
1 0
4 4
104 209
```

Problem Setter: Miguel Revilla, Special Thanks: Luis Hernandez Corbato

D

Happy Birthday

Input: Standard Input
Output: Standard Output



Today it's February 13th. It's a very special day: Miguel's birthday! Like every year, he's organised a big celebration for all his friends. He prepared a succulent dinner at his house. Everyone had a lot of fun and the celebration was a complete success.

Now it's time for the not-so-funny cleaning up. He wants to start with moving all the dishes from the table to the kitchen. Since he's been going to the gym lately, he feels strong enough to pile and carry at once as many dishes as he wants. Time doesn't go unnoticed though: he's not as agile as he used to be, so he can only carry the stack of dishes if it's completely stable. A stack of dishes is stable if each dish on the stack is bigger or equal in size than all the dishes above it. If the stack wasn't stable he would drop the dishes and would have even more things to clean!

At the beginning of the scene, Miguel is empty-handed in one side of the table, walks along the table finding and maybe collecting dishes of different sizes until he reaches the other side, and then brings the collected dishes to the kitchen. When he finds a dish, he can:

- ignore the dish.
- if he has empty hands, pick up the dish.
- if he has a stack of dishes on his hands, pile the dish on top of the stack.
- if he has a stack of dishes on his hands, put the stack on top of the dish, then pick up the new stack (including the dish).

Miguel is tired, so he wants to clean up the table as soon as possible. He'd like to take as many dishes as he can in each run, but he's exhausted from the party and can't think clearly. He's asked you for help to find out what the maximum number of dishes he can collect in a single run is.

Input

Input contains several datasets. Each dataset consists on two lines. The first line of each dataset contains an integer N ($1 \leq N \leq 500$), the number of dishes on the table. The second line of each dataset contains N integers, $k_1 \dots k_N$ ($1 \leq k_i \leq 1000$), specifying the size of each dish he finds, in the same order he finds them.

Input ends with a dataset with $N=0$. This case shouldn't be processed.

Output

For each dataset, print the maximum number of dishes Miguel can collect in a single run.

Sample Input

```
6
3 1 5 6 2 4
6
3 4 2 5 2 6
0
```

Output for Sample Input

```
4
6
```

E

Array Transformer

Input: Standard Input
Output: Standard Output



Write a program to transform an array $A[1], A[2], \dots, A[n]$ according to m instructions. Each instruction (L, R, v, p) means: First, calculate how many numbers from $A[L]$ to $A[R]$ (inclusive) are strictly less than v , call this answer k . Then, change the value of $A[p]$ to $u * k / (R - L + 1)$, here we use integer division (i.e. ignoring fractional part).

Input

The first line of input contains three integer n, m, u ($1 \leq n \leq 10,000, 1 \leq m \leq 20,000, 1 \leq u \leq 1,000,000,000$). Each of the next n lines contains an integer $A[i]$ ($1 \leq A[i] \leq u$). Each of the next m lines contains an instruction consisting of four integers L, R, v, p ($1 \leq L \leq R \leq n, 1 \leq v \leq u, 1 \leq p \leq n$).

Output

Print n lines, one for each integer, the final array.

Sample Input

```
10 1 11
1
2
3
4
5
6
7
8
9
10
2 8 6 10
```

Output for Sample Input

```
1
2
3
4
5
6
7
8
9
6
```

Explanation

There is only one instruction: $L=2, R=8, v=6, p=10$. There are 4 numbers (2,3,4,5) less than 6, so $k=4$. The new number in $A[10]$ is $11 * 4 / (8 - 2 + 1) = 44 / 7 = 6$.

Problem Setter: Rujia Liu, Special Thanks: Jane Alam Jan

F

Bubble Sort

Input: Standard Input
Output: Standard Output



Check the following code which counts the number of swaps of bubble sort.

```
int findSwaps( int n, int a[] )
{
    int count = 0, i, j, temp, b[100000];

    for( i = 0; i < n; i++ ) {
        b[i] = a[i];
    }
    for( i = 0; i < n; i++ ) {
        for( j = 0; j < n - 1; j++ ) {
            if( b[j] > b[j+1] ) {
                temp = b[j];
                b[j] = b[j+1];
                b[j+1] = temp;

                count++;
            }
        }
    }
    return count;
}
```

You have to find the average value of 'count' in the given code if we run **findSwaps()** infinitely many times using constant 'n' and each time some random integers (from 1 to n) are given in array **a[]**. You can assume that the input integers in array **a[]** are distinct.

Input

Input starts with an integer **T** (≤ 1000), denoting the number of test cases.
Each test case contains an integer **n** ($1 \leq n \leq 10^5$) in a single line.

Output

For each case, print the case number and the desired result. If the result is an integer, print it. Otherwise print it in '**p/q**' form, where **p** and **q** are relative prime.

Sample Input

```
2
1
2
```

Output for Sample Input

```
Case 1: 0
Case 2: 1/2
```

Problem Setter: Jane Alam Jan, Special Thanks: Sohel Hafiz

G

Find Solutions

Input: Standard Input
Output: Standard Output



Look at the following equation:

$$c = ab - \frac{a + b}{2} + 1$$

Now given the value of c , how many possible values of a and b are there (a and b must be integers)? That is you will have to find the number of pairs (a, b) which satisfies the above equation.

Input

The input file contains around 3000 line of input. Each line contains an integers n ($0 < n \leq 10^{14}$). This n actually denotes the value of c . A line containing a single zero terminates the input. This line should not be processed.

Output

For each line of input produce one line of output. This line contains two integers. First integer denotes the value of c and the second integer denotes the number of pair of values of a and b that satisfies the above equation, given the value of c .

Sample Input

```
1020
400
0
```

Output for Sample Input

```
1020 8
400 2
```

Comments:

The 8 solution pairs for the first sample input are $(1, 2039)$, $(2, 680)$, $(5, 227)$, $(14, 76)$, $(76, 14)$, $(227, 5)$, $(680, 2)$ and $(2039, 1)$.

Problem Setter: Shahriar Manzoor, Special Thanks: Jane Alam Jan

H

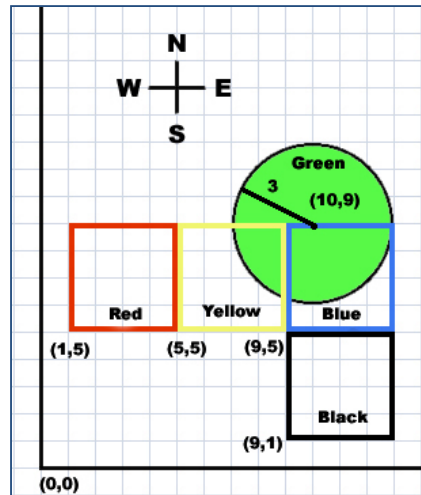
Paint Brush

Input: Standard Input
Output: Standard Output



Jim just had a party in his home and that's why he painted some non overlapping beautiful circles in his floor. But after the party, Jim wants to repaint the floor, mostly the portions where the circles were drawn. As you know that painting for a party is exciting, but the repainting after the party is definitely boring; he bought an automatic Paint Brush from the departmental store for this task.

The Paint Brush is a square shaped electronic brush and the size of its one side is k . It considers the room as a **2D** plane. Initially it's placed in the room. When turned on, it follows some pre-written moves that are programmed into its memory. And in each move, it moves k units North, South, West or East. The brush starts painting right after being turned on and continues painting until it stops. And it stops after finishing all the moves. The moves are programmed in such a way that it never paints the same part it has painted already. Now you are given the co-ordinates and radiuses of the circles and the lower left co-ordinates of the brush, its prewritten moves and k , you have to find the total percentage of the area of the circles that are repainted by the brush. All the co-ordinates are given according to the view of the brush.



For example, in the above picture, the brush is initially placed on **(1, 5)** (red) and k is **4**. One circle (green) was drawn whose center is on **(10, 9)** and radius is **3**. And the instructions for the brush are **(east 2, south 1)**. It means that it has **2** moves in east and **1** move in south. After **1** move in east it goes to **(5, 5)** (yellow). After **2nd** move to east it goes to **(9, 5)** (blue). And after **1** move to south it goes to **(9, 1)** (black).

Input

Input starts with an integer T (≤ 4000), denoting the number of test cases.

Each test case starts with a line containing five integers n ($1 \leq n \leq 16$), B_x , B_y , k ($1 \leq k \leq 20$), and m ($0 \leq m \leq 20000$) where n denotes the number of non-overlapping circles, (B_x, B_y) denotes the lower-left corner of the initial position of the brush and m denotes the number of instructions. Each of the next n lines contains **3** integers x_i , y_i , r_i ($0 < r_i \leq 20000$), where (x_i, y_i) denotes the center of the i^{th} circle and r_i denotes the radius. Each of the next m lines will contain an instruction which is denoted by a string which shows the direction and is one of {"north", "south", "east", "west"} and an integer p ($1 \leq p \leq 20000$) denoting the number of moves in this direction. $|B_x|, |B_y|, |x_i|, |y_i| \leq 20000$ and the instructions are given such that it never paints any part more than once.

You can assume that the total number of lines in the input file is less than **100000**.

Output

For each case, print the case number and percentage of the area of the circles that are repainted by the Paint Brush. Print the percentage with at least 6 digits after the decimal point. Errors less than 10^{-5} will be ignored. See the samples for more details.

Sample Input

```
2
1 1 5 4 2
10 9 3
east 2
south 1
1 1 5 4 3
10 9 3
east 2
north 1
west 1
```

Output for Sample Input

```
Case 1: 50.000000
Case 2: 100.000000
```

Problem Setter: Jane Alam Jan, Special Thanks: Arifuzzaman Arif

I	<h1 style="margin: 0;">Chess board Coloring</h1> <p style="margin: 0;">Input: Standard Input Output: Standard Output</p>	
----------	--	---

You have a chessboard with N rows and N columns. You want to color each of the cells with exactly N colors (colors are numbered from 0 to $N-1$). A coloring is valid if and only if no row or no columns contains a duplicate color. That means in the final coloring each cell should be colored by a color from 0 to $N-1$. No row will contain two cells of the same color and same for the no column.

Your friend already colored some of the cells in the top-left corner (cells that are in the first R row and in the first C columns). Your job is to color the rest of the cells or tell it is impossible to color.

Input

Input starts with an integer $T(1 \leq T \leq 50)$, the number of test cases. Each test case starts with a line containing 3 integers $N(1 \leq N \leq 50)$, $R(0 \leq R \leq N)$ and $C(0 \leq C \leq N)$. Each of the next R line contains C integers in each line. The j 'th integer in the i 'th line indicates the color of the cell located at row i and column j . The initial coloring will be always valid. That means No row or column will have duplicate integers. The initial colors are between 0 to $N-1$ inclusive.

Output

First line of each test case will be "YES" when it is possible to color the rest of the cells or "NO" otherwise. In case it is possible to reach a valid coloring the next N line should contain N integers in each line denoting the colors of the cells(in one of the valid final colorings). Each of these N integers should be separated by a single space. There can be multiple valid colorings; you can output any valid coloring. When the solution is "NO", you do not have to output additional line.

Sample Input

Output for Sample Input

<pre> 3 4 2 2 1 2 2 0 4 2 3 0 1 2 2 0 1 3 1 1 0 </pre>	<pre> YES 1 2 0 3 2 0 3 1 3 1 2 0 0 3 1 2 NO YES 0 1 2 2 0 1 1 2 0 </pre>
--	---

Problem Setter: Abdullah al Mahmud, Special Thanks: Jane Alam Jan

J

Emotional Bishop

Input: Standard Input
Output: Standard Output



A bishop is a piece in the board game of chess. It is only allowed to move diagonally. But it has no restriction on the distance for each move. We can denote a position of a bishop by (r, c) if it is in r^{th} row and c^{th} column. So formally we can say that, A bishop can go from $(r1, c1)$ to $(r2, c2)$ if and only if $|r1 - r2| = |c1 - c2|$, i.e. absolute difference of rows and absolute difference of columns are equal. And also a bishop cannot move outside the board. See the following pictures to make things clear, where the by ‘X’ sign the valid moves of the bishop are shown.



Bishop can move to any square diagonally inside the board

There was a bishop which is very emotional that it didn't like fighting. So it didn't involve itself into any chess game. As no other bishop is like it, it has to live alone in a board. So it created its own boards. It doesn't want to restrict itself in a 8 x 8 chessboard. So it created many two dimensional boards of different sizes. It had a board of size 100 x 100 (i.e. 100 rows and 100 columns), another board of size 1000 x 2000 (i.e. 1000 rows and 2000 columns, you see it even didn't restrict itself on row-column equality).

Everything was going perfect for that emotional bishop until the day when it discovered that it needs to move to a new cell. Since the bishop is too emotional he wants to go to the destination in fewest moves. In this problem, you have to calculate the minimum number of moves required for the emotional bishop to reach the destination. If the destination is impossible to reach, you have to say “impossible”.

Input

The input starts with an integer T ($1 \leq T \leq 10000$), the number of test cases. Each of the next T lines will describe one test case by six integers R ($1 \leq R \leq 1000000000$), C ($1 \leq C \leq 1000000000$), SR ($1 \leq SR \leq R$), SC ($1 \leq SC \leq C$), DR ($1 \leq DR \leq R$) and DC ($1 \leq DC \leq C$). Here R is the number of rows in the board, C is the number of columns in the board, (SR, SC) is the initial square of the bishop, (DR, DC) is the target location of the bishop. Initial location, target location are distinct.

Output

For each test case output a single line in the format “Case K: N”, where K is the case number and N is the minimum number of moves to reach the destination if it is possible to reach the destination and “impossible” (quotes for clarity) otherwise. See sample output for exact format.

Sample Input

```
2
10 10 2 2 4 4
10 10 2 2 4 5
```

Output for Sample Input

```
Case 1: 1
Case 2: impossible
```

Problem Setter: Arifuzzaman Arif, Special Thanks: Jane Alam Jan