Join the Ranks

Problem

You recently acquired a new deck of cards. Each card displays a rank, which is an integer between 1 and **R**, and a suit, which is an integer between 1 and **S**. For each combination of a rank and a suit, there is exactly one card that displays it, meaning that the deck has **R** × **S** cards in total. We will denote a card with rank r and suit s as (r, s).

Being brand new, the deck is sorted from top to bottom by suit in increasing order, with ties being broken by ranks in increasing order. That is, (1, 1) comes first, then (2, 1), ..., (**R**, 1), then (1, 2), (2, 2), ..., (**R**, 2), and so on up to (**R**, **S**). For example, with **R** = 4 ranks and **S** = 2 suits, the initial ordering would be: (1, 1), (2, 1), (3, 1), (4, 1), (1, 2), (2, 2), (3, 2), (4, 2).

You want to reorder the deck to be sorted by rank. That is, you want to put all the cards of the same rank together, and have the ranks be in increasing order. You do not care, however, about the order of the suits within each rank. For example, with **R** = 4 and **S** = 2, one of the various possible valid new orderings would be: (1, 2), (1, 1), (2, 1), (2, 2), (3, 1), (3, 2), (4, 2), (4, 1).

You have been learning how to cook, so you want to reorder the deck without putting your spatulas down. You decided to sort the deck using only the following multi-part operation:

- First, take one or more cards from the top of the deck, and set that selection aside as pile A.
- Next, take one or more cards from the new top of the deck, and set that selection aside as pile B.
- Finally, put pile A on top of the deck, and then put pile B on top of the new deck.

Notice that the operation exchanges the pile A part of the deck and the pile B part of the deck, without affecting any other cards deeper in the deck (if there are any).

Continuing with our **R** = 4, **S** = 2 example, if your first move is to choose 3 cards from the top for pile A and 2 cards for pile B, then these are the cards you get:
A: (1, 1), (2, 1), (3, 1),
B: (4, 1), (1, 2), and
Remainder of deck: (2, 2), (3, 2), (4, 2).
After putting A on the deck and then B on top of that, the new deck is ordered like this:
(4, 1), (1, 2), (1, 1), (2, 1), (3, 1), (2, 2), (3, 2), (4, 2).

Given **R** and **S**, find a sequence of operations that reorders the deck to be sorted by rank, as described above, and uses the minimum possible number of operations to do so.

Input

The first line of the input gives the number of test cases, **T**. **T** lines follow. Each of these lines describes a single test case with two integers **R** and **S**, the number of ranks and suits in the deck, respectively.

Output

For each test case, output one line containing `Case #x: y`, where `x` is the test case number (starting from 1) and `y` is the minimum number of operations needed to reorder the deck as

explained above. Then, print y more lines containing a_i b_i, meaning that in the i-th operation in a sequence of moves that reorders the deck, you take a_i cards first to form pile A and then b_i cards after that to form pile B.

Limits

Memory limit: 1GB.

Test set 1 (Visible Verdict)

Time limit: 30 seconds.
T = 12.
$2 \leq$ **R** ≤ 5.
$2 \leq$ **S** ≤ 7.
R × **S** ≤ 14.

Test set 2 (Hidden Verdict)

Time limit: 60 seconds.
$1 \leq$ **T** ≤ 100.
$2 \leq$ **R** ≤ 40.
$2 \leq$ **S** ≤ 40.

Sample

Sample Input	Sample Output
3 2 2 3 2 2 3	Case #1: 1 2 1 Case #2: 2 3 2 2 1 Case #3: 2 2 3 2 2

In Sample Case #1, the initial order is (1, 1), (2, 1), (1, 2), (2, 2). After swapping A = (1, 1), (2, 1) and B = (1, 2) the deck is left as (1, 2), (1, 1), (2, 1), (2, 2), which is sorted by rank as needed. Notice that the suits are in different orders within each rank, which is allowed.

In Sample Case #2, the initial order is (1, 1), (2, 1), (3, 1), (1, 2), (2, 2), (3, 2). After swapping A = (1, 1), (2, 1), (3, 1) and B = (1, 2), (2, 2) the deck is left as (1, 2), (2, 2), (1, 1), (2, 1), (3, 1), (3, 2). In a second move, we can do A = (1, 2), (2, 2) and B = (1, 1) to get (1, 1), (1, 2), (2, 2), (2, 1), (3, 1), (3, 2).

In Sample Case #3, another valid solution is $a_1 = 4$, $b_1 = 1$ first, and then $a_2 = 3$ and $b_2 = 1$.