We have two rows. There are a dots on the top row and b dots on the bottom row. We draw line segments connecting every dot on the top row with every dot on the bottom row. The dots are arranged in such a way that the number of internal intersections among the line segments is maximized. To achieve this goal we must not allow more than two line segments to intersect in a point. The intersection points on the top row and the bottom are not included in our count; we can allow more than two line segments to intersect on those two rows. Given the value of a and b, your task is to compute $P(a, b)$, the number of intersections in between the two rows. For example, in the following figure $a = 2$ and $b = 3$. This figure illustrates that $P(2, 3) = 3$.

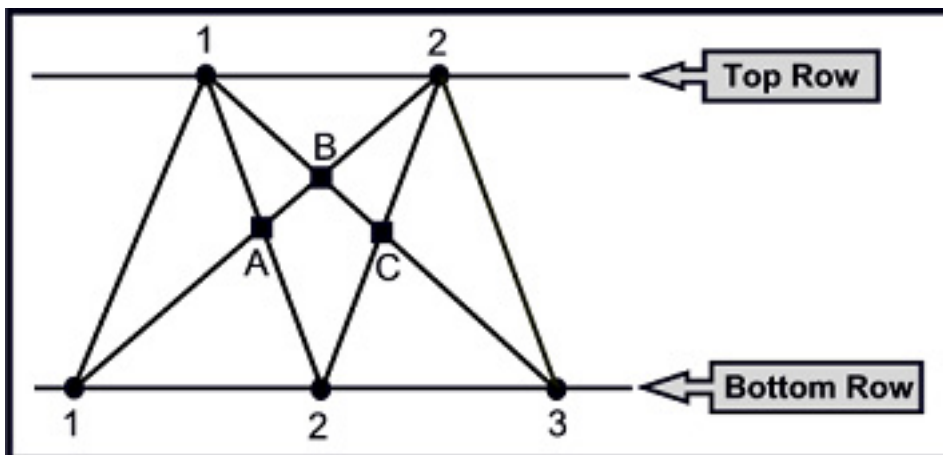

Figure: 2 dots on the top row and 3 dots on the bottom row. And total no of internal intersections is P(2,3)=3. In this picture these intersections are marked as A, B and C.

Input

Each line in the input will contain two positive integers a ($0 < a \leq 20000$) and b ($0 < b \leq 20000$). Input is terminated by a line where both a and b are zero. This case should not be processed. You will need to process at most 1200 sets of inputs.

Output

For each line of input, print in a line the serial of output followed by the value of $P(a, b)$. Look at the output for sample input for details. You can assume that the output for the test cases will fit in **64-bit** signed integers.

Sample Input

```
2 2
2 3
3 3
0 0
```

Sample Output

```
Case 1: 1
Case 2: 3
Case 3: 9
```