Given a function  f(x, y) and a value z, return all positive integer pairs x and y where f(x,y) == z.

The function is constantly increasing, i.e.:

* f(x, y) < f(x + 1, y)
* f(x, y) < f(x, y + 1)

The function interface is defined like this:

interface CustomFunction {

public:

  // Returns positive integer f(x, y) for any given positive integer x and y.

  int f(int x, int y);

};

For custom testing purposes you're given an integer function\_id and a target z as input, where function\_id represent one function from an secret internal list, on the examples you'll know only two functions from the list.

You may return the solutions in any order.

**Example 1:**

**Input:** function\_id = 1, z = 5

**Output:** [[1,4],[2,3],[3,2],[4,1]]

**Explanation:** function\_id = 1 means that f(x, y) = x + y

**Example 2:**

**Input:** function\_id = 2, z = 5

**Output:** [[1,5],[5,1]]

**Explanation:** function\_id = 2 means that f(x, y) = x \* y

**Constraints:**

* 1 <= function\_id <= 9
* 1 <= z <= 100
* It's guaranteed that the solutions of f(x, y) == z will be on the range 1 <= x, y <= 1000
* It's also guaranteed that f(x, y) will fit in 32 bit signed integer if 1 <= x, y <= 1000