

Problem J. Jitterbug

Input file: jitterbug.in
Output file: jitterbug.out
Time limit: 2 seconds
Memory limit: 256 mebibytes

You have caught a fascinating insect — a jitterbug. For your own amusement, you have drawn a labyrinth (which can be represented as an undirected graph which doesn't contain self-loops or multiple edges) and have taught your bug to move in it. You noticed that after you place the bug at some vertex of the labyrinth, it uses the following moving strategy: it equiprobably chooses an edge that is incident to the current vertex and moves to the other end of this edge; then it chooses a random edge again, and so on.

Usually, you place the bug at some starting vertex in the labyrinth and observe it until it arrives at the chosen finish vertex (of course, there must be at least one path between the starting vertex and the finish vertex). You like to watch as the bug runs around the labyrinth, so you want to construct a labyrinth such that on average it takes the bug the most time to arrive to the finish point.

For example, suppose $n = 3$. It can be shown that the labyrinth with the longest expected number of moves between vertices 1 and 3 is the chain 1–2–3: the average number of moves is four.

Construct a labyrinth on 200 vertices such that it takes the bug at least 10^6 moves on average to reach the vertex 200 starting from the vertex 1.

Input

The first line of input contains two space-separated integer numbers n and b — the number of vertices and the lower limit on average number of moves (that is, your answer will be accepted if the average number of moves is at least b).

This problem has only two testcases:

1. $n = 3, b = 4$.
2. $n = 200, b = 10^6$.

Output

On the first line print a single integer m — the number of edges in your labyrinth.

Next m lines must describe edges of the labyrinth. On i -th of these lines print two integers — numbers of vertices connected by i -th edge. Remember that the edges are bidirectional.

Your graph must not have self-loops or multiple edges; vertices 1 and n must be connected by at least one path.

Example

jitterbug.in	jitterbug.out
3 4	2 1 2 2 3