Problem H. Hard Test

Input file:	`test.in`
Output file:	`test.out`
Time limit:	2 seconds
Memory limit:	256 megabytes

Andrew is having a hard time preparing his 239-th contest for Petrozavodsk. This time the solution to the problem is based on Dijkstra algorithm and Andrew wants to prepare the hard test for the algorithm.

The Dijkstra algorithm is used to find the shortest path from a source vertex to all other vertices in a graph. The algorithm acts as follows. Let G be a weight directed graph with vertex set V, edge set E and weight function $w : E \to \mathbb{R}^+$. Let all vertices be reachable from vertex s. The algorithm uses a set of vertices U, first initialized as empty. Each vertex is labeled with either an integer number, or with $+\infty$. Initially all vertices are labeled with $+\infty$, and the vertex s is labeled with 0. Denote the label of vertex v as $d[v]$.

A step of the algorithm is the following: the vertex with the minimal label that doesn't belong to U is selected. Let this vertex be u. The vertex u is added to the set U, and each edge $uv \in E$ is *relaxed*. The relaxation replaces $d[v]$ with $\min(d[v], d[u] + w(uv))$. The algorithm is over when all vertices belong to U. If the label of the vertex v has changed, the relaxation is said to be *active*.

Now Andrew would like to create a graph with n vertices and m edges, such that the Dijkstra algorithm makes as many active relaxations as possible. Help him to create such graph. To avoid nondeterminism, each time when selecting a vertex with minimal label among vertices that are not in U there must be exactly one vertex with the minimal label.

Input

The first line of the input file contains two integer numbers: n and m — the number of vertices and the number of edges in the graph Andrew would like to create ($4 \le n \le 1000$, $n - 1 \le m \le n^2/5$).

Output

Output m lines — the edges of the graph. Each line must contain three integer numbers: the beginning of the edge, the end of the edge and the weight of the edge. All weights must be non-negative and must not exceed 10^6. All vertices must be reachable from vertex 1. If Dijkstra algorithm is run with $s = 1$ there must be maximal possible number of active relaxations among all graphs with n vertices and m edges. There must be no loops and no parallel edges.

Example

test.in	test.out

`4 3`	`1 2 0`
	`1 3 1`
	`1 4 2`

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