Build a Computer

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

You want to build a computer to achieve a specific functionality: Given an integer x, determine whether x lies within the interval $[L, R]$. To accomplish this, you designed a directed acyclic graph (DAG) with edge weights of 0 and 1, which contains a starting node with an indegree of 0 and an ending node with an outdegree of 0. By starting from the starting node and following a path to the ending node, the sequence of the traversed edge weights forms a binary representation of an integer within the range $[L, R]$ without leading zeros. Every integer within the range $[L, R]$ must correspond to exactly one unique path in this graph. In this way, you can determine whether an integer lies within the range $[L, R]$ by checking if its binary representation can be constructed by traversing this DAG.

Clearly, you could separate the corresponding path for each integer into individual chains. However, you realized that for a large range, such a DAG would require too many nodes, and the computer you built with only 256 MiB of memory cannot store it. Therefore, you need to compress this DAG, allowing different paths to share nodes, in order to reduce the number of nodes and edges. Formally, you need to construct a DAG with no more than 100 nodes, where each node has an outdegree of at most 200. The DAG must have edge weights of 0 and 1, with exactly one starting node with an in-degree of 0 and one ending node with an out-degree of 0. Every integer in the range $[L, R]$ must correspond to **exactly** one unique path from the start to the end in this DAG, and no path should represent any integer outside the range $[L, R]$. Note that none of the binary sequences formed by any path in the graph should have leading zeros. There may be two edges with different weights between two nodes.

Input

A single line containing two positive integers L, R ($1 \le L \le R \le 10^6$).

Output

The first line should output the number of nodes n ($1 \le n \le 100$).

For the next n lines, the i-th line should start with an integer k ($0 \le k \le 200$), representing the number of outgoing edges from node i. Then output $2 \cdot k$ integers $a_{i,k}, v_{i,k}$ ($1 \le a_{i,k} \le n$, $a_{i,k} \ne i$, $v_{i,k} \in \{0, 1\}$), which means that node i has a directed edge with weight $v_{i,k}$ to node $a_{i,k}$. You must ensure that the output represents a directed acyclic graph that satisfies the requirements.

Example

`standard input`	`standard output`

`5 7`	`8`
	`3 2 1 3 1 4 1`
	`1 5 0`
	`1 6 1`
	`1 7 1`
	`1 8 1`
	`1 8 0`
	`1 8 1`
	`0`