

Problem 6. Tea Sort

Input file: Output file: output .txt
Time limit: 3 seconds
Memory limit: 256 mebibytes

After a long day of laborious coding, Stepan decided to have five o'clock tea. Programmers love tea of all kinds: kocha, sencha, matcha, mocha and the like. Stepan went into the kitchen and discovered that all packs with tea in the drawer had been mixed. What a mess! Stepan likes order. He wants to have the packs arranged in the usual order.

There are N packs of tea in the drawer, arranged in K rows. In each row, the packs are lined up, from the pack closest to Stepan to the most distant. But the drawer is very low, and you can only pick the closest pack of tea in each row. So, the only possible operation here is to pick the closest pack from a row i and put it into the row j , where it becomes the closest pack. In the programming mumbo-jumbo, each row is a stack, or "LIFO".

Each pack is marked with an integer, which denotes the type of tea. Stepan likes it when:

1. the number of packs in every row is the same,
2. the type number of the pack from the row i is less than or equals the type number of the pack from the row j (for any $i < j$),
3. in every row, packs are arranged by type in the ascending order from the closest to the most distant pack.

Suggest a plan with no more than $13 \cdot N$ operations of moving packs, resulting in everything being in place according to Stepan's taste. You do not need to minimize the number of operations.

Input

The first line of the input file contains two integers N and K — the number of packs of tea and the number of rows ($1 \leq N \leq 10^5$, $11 \leq K \leq 111$).

It is followed by K lines, with each r -th line defining the r -th row of tea packs. For each row, an integer T_r is provided, denoting the number of packs in the row, followed by T_r integers — the types of tea in the packs of this row ($0 \leq T_r \leq N$). Packs in a row are listed from the closest to the most distant.

It is guaranteed that the sum of all T_r equals N , and that N is divisible by K . Integers denoting tea types are not greater than 10^9 in absolute value.

Output

In the first line of the output file, print one integer M — the number of operations in your plan ($0 \leq M \leq 13 \cdot N$).

In the remaining M lines, print the operations in the order of their execution. For each operation, print two space-separated integers: i — the number of the row, from which Stepan must pick the closest pack, and j — the number of the row where he must put it ($1 \leq i, j \leq K$).

Examples

input.txt	output.txt
33 11 3 -1 1 1 3 1 1 1 3 1 2 2 3 3 3 3 3 3 4 4 3 5 7 7 3 7 7 7 3 7 8 8 3 8 9 9 3 9 9 9 3 9 9 9	0
33 11 3 -1 1 1 3 1 1 1 3 1 2 2 0 5 3 4 3 3 4 4 5 7 7 3 3 7 7 7 3 7 8 8 3 8 9 9 3 9 9 9 3 9 9 9	13 6 7 6 7 6 7 6 4 7 6 7 6 7 6 5 4 5 11 5 4 5 4 11 5 4 5

Example explanation

In the first example, all packs are already placed in Stepan's favorite order, so there's nothing to do here.

In the second example, the row 4 is empty, and the packs that should be there have been placed in the rows 5 and 6. The first seven operations of the plan pick the packs of the type 3 from the row 6. Note that this pack is the most distant, so to pick it, the preceding three packs must be picked first. These packs are placed to the row 7 in reverse order; upon the removal of the desired pack of the type 3, they are placed back in the right order. The remaining six operations move two packs of the type 3 from the row 5 to the row 4. This shifts a pack of the type 4 further in this row.