New Year Train

On the New Year Eve, a government of one country decided to send a train with gifts to each of its towns. For each of the N towns exactly one wagon with gifts was sent. The route was organized in such way that at each place the last wagon would be detached and train would continue on its way, until all gifts were delivered. Just before the departure it turned out that the loading workers did not pay attention to numeration of the wagons and loaded the gifts in random order. It was impossible to detach a wagon from the middle of the train and there was no time to rearrange gifts. Luckily, there was a depot with parallel tracks. At the entrance of the depot each wagon could be directed to any of the tracks and wagons could leave the depot from the other side in the right sequences 1, 2, 3, 4, and so on. Note that we will then be leaving presents in towns in the reversed order (..., 4, 3, 2, 1).

For example, let's say we have a train with wagons in the following order: 2, 5, 1, 4, 6, 3. Wagons 2, 5, 6 could be directed to the first track; wagons 1, 4 to the second one and wagon 3 to the third. In this case wagons could leave the depot in the right order. Fortunately, there were enough tracks in the depot to rearrange the train.

Input

First line of the input contains two integers N and M: the number of wagons in the train and the number of tracks in the depot respectively (1 <= N <= 800 000, 1 <= M <= 100 000, M <= N). Second line contains N integers: sequence of wagons before the entrance to the depot. It's guaranteed that solution always exists.

Output

First line of the output must contain N integers: number of track that should be chosen for each wagon from input sequence (tracts are numbered from 1 to M). On the second line print the number of tracks in order the wagons should leave the depot to result in the sequence 1, 2, 3, and so on. If multiple solutions exists, print the one that results in lexicographically smallest sequence in the first line of the output.

Example

Input
6 3
2 5 1 4 6 3

Output
1 1 2 2 1 3
2 1 3 2 1 1