

Task: KLO

Building blocks



Stage I. Source file `klo.*`

22.10–19.11.2007

Available memory: 32 MB.

Byteasar loved to play with building blocks as a child. He used to arrange the blocks into n columns of random height and then organize them in the following manner: Byteasar would choose a number k and try to arrange the blocks in such a way that some k consecutive columns would be of equal height. Furthermore he always tried to achieve this goal in a minimum number of moves possible, where a single move consists in:

- laying one block on top of any column (Byteasar had a huge box with spare blocks, ensuring this move could always be performed), or
- removing the uppermost block from the top of any column.

However, Byteasar was never quite sure if his sequence of moves was indeed optimal, therefore he has asked you to write a programme that will help him solve the problem.

Task

Write a programme that:

- reads the number k along with the initial setup of the blocks from the standard input,
- determines the optimal solution (shortest possible sequence of moves),
- writes out the solution to the standard output.

Input

In the first line of the standard input there are two integers, n and k ($1 \leq k \leq n \leq 100\,000$), separated by a single space. Each of the following n lines contains the height of some column; the line no. $i + 1$ contains the integer $0 \leq h_i \leq 1\,000\,000$ — the height of the i^{th} column, ie. the number of blocks in consists of.

Output

The optimal solution should be written out to the standard output, ie. such arrangement of blocks that:

- it contains k consecutive columns of equal height,
- it can be obtained from the initial setup in a minimum possible number of moves.

The output should consist of $n + 1$ lines, each one containing a single integer. The number in the first line should be the minimum number of moves needed to get the desired arrangement. The line no. $i + 1$ (for $1 \leq i \leq n$) should contain the number h'_i — the final height of the i^{th} column. If more than one optimal solution exists, write out one chosen arbitrarily.

Example

For the input data:

5 3
3
9
2
3
1

the correct result is:

2
3
9
2
2
2