

# NM Chars

Input file:            **standard input**  
Output file:           **standard output**  
Time limit:            3 seconds  
Memory limit:         256 megabytes

Busy Beaver is making a new language for his linguistics class! He has already decided that his language will have an alphabet whose letters are the integers  $1, \dots, NM$  in that order; now he wants to make some words for the language.

Because Busy Beaver is interested in statistics, he wants to control the frequencies of letters appearing in words, so he has chosen a multiset  $a$  of size  $NM$  of letters from his alphabet. He will now form  $N$  words each with  $M$  letters such that each letter from  $a$  is used exactly once (i.e. if a given letter  $x$  appears exactly  $k$  times in  $a$ , then it is used exactly  $k$  times across all  $N$  words).

After forming the words, Busy Beaver plans to organize them into a dictionary, so he will lexicographically sort his  $N$  words to produce the sequence of words  $s_1, \dots, s_N$ . Busy Beaver likes strings to be lexicographically small, so for each  $k$  from 1 to  $N$  he wants to know the lexicographically minimum possible value of  $s_k$  over all choices of words.

Note that the answer for each  $s_k$  is independent; for example, the choice of words so that  $s_1$  is minimized may be different from the choice of words so that  $s_2$  is minimized.

## Input

The first line contains two positive integers  $N, M$  ( $1 \leq NM \leq 10^6$ ).

The second line contains  $NM$  integers  $a_1, \dots, a_{NM}$  composing the multiset  $a$  ( $1 \leq a_j \leq NM$ ).

## Output

For each  $k$  from 1 to  $N$ , output the minimum possible value of  $s_k$  on its own line as a sequence of space-separated integers.

## Examples

standard input	standard output
4 3 1 1 2 2 3 3 4 4 5 5 6 6	1 1 2 1 2 3 2 2 3 2 3 4
3 4 12 4 4 4 1 1 4 4 6 4 1 4	1 1 1 4 1 4 4 4 1 4 4 12
4 5 12 7 17 3 6 11 9 2 17 7 1 6 9 12 6 3 9	1 2 2 3 3 12 2 3 5 3 6 2 3 6 7 7 3 3 6 6 6
1 1 1	1

## Note

In the first sample, the following choices of words minimize each of  $s_k$ :

By choosing the words 112, 233, 445, 566, we get that  $s = 112, 233, 445, 566$ , so  $s_1 = 112$  as desired.

By choosing the words 123, 456, 123, 456, we get that  $s = 123, 123, 456, 456$ , so  $s_2 = 123$  as desired.

By choosing the words 166, 155, 344, 223, we get that  $s = 155, 166, 223, 344$ , so  $s_3 = 223$  as desired.

By choosing the words 234, 234, 156, 156, we get that  $s = 156, 156, 234, 234$ , so  $s_4 = 234$  as desired.

It can be shown that in all of these cases there is no way to choose the words so that the respective  $s_k$  is even lexicographically smaller.