Everyone Loves Permutations

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

A permutation of length n is an array of length n containing all integers from 1 to n, and all its elements are pairwise distinct.

Having grown up and played with arrays, Anton moved on to studying more interesting arrays — permutations. While writing his thesis, he faced a very difficult task.

He has a permutation p of length n and an integer k. He decided to construct a two-dimensional array a with sizes $(k+1) \times n$.

1. $a_{0j} = j$ for all j $(1 \le j \le n)$;
2. $a_{ij} = a_{(i-1)p_j}$ for all i $(1 \le i \le k)$ and j $(1 \le j \le n)$.

Let $p = [5, 3, 1, 4, 2]$ and $k = 3$, then we have the following array.

a_{ij}	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$
$i=0$	1	2	3	4	5
$i=1$	5	3	1	4	2
$i=2$	2	1	5	4	3
$i=3$	3	5	2	4	1

For each x $(1 \le x \le n)$, he wants to know the sum of all j such that $a_{ij} = x$, where $1 \le i \le k$. In other words, he wants to find the sum of k numbers — the indices of the number x in each a_i.

Consider the last example. If $x = 1$, the answer will be $3 + 2 + 5 = 10$.

After some deliberation and simple ideas, Anton managed to solve this problem quickly. Now he wants to check if you can solve it too.

Input

The first line of the input contains two integers n, k $(1 \le n \le 10^6,\ 1 \le k \le 10^9)$ — the length of the permutation and the number of repetitions of operations, respectively.

The second line contains the permutation p $(1 \le p_i \le n)$.

Output

Print n integers, where the i-th number is the answer for $x = i$.

Scoring

1. (8 points): $k = 1$;
2. (17 points): $p_i = i$;
3. (26 points): $n \le 2000, k \le 2000$;
4. (28 points): $n \le 2000$, for any i and j, there exists a k such that $p[p[p \dots p[i] \dots]] = j$, where the nesting is taken k times;
5. (9 points): for any i and j, there exists a k such that $p[p[p \dots p[i] \dots]] = j$, where the nesting is taken k times;

6. (12 points): without additional restrictions.

Examples

standard input	standard output
3 2 2 1 3	3 3 6
5 3 5 3 1 4 2	10 9 8 12 6