F. A Heap of Heaps

time limit per test: 3 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Andrew skipped lessons on the subject 'Algorithms and Data Structures' for the entire term. When he came to the final test, the teacher decided to give him a difficult task as a punishment.

The teacher gave Andrew an array of n numbers $a_1, ..., a_n$. After that he asked Andrew for each k from 1 to n - 1 to build a k-ary heap on the array and count the number of elements for which the property of the minimum-rooted heap is violated, i.e. the value of an element is less than the value of its parent.

Andrew looked up on the Wikipedia that a k-ary heap is a rooted tree with vertices in elements of the array.
If the elements of the array are indexed from 1 to n, then the children of element v are elements with indices $k(v - 1) + 2, ..., kv + 1$ (if some of these elements lie outside the borders of the array, the corresponding children are absent). In any k-ary heap every element except for the first one has exactly one parent; for the element 1 the parent is absent (this element is the *root* of the heap). Denote $p(v)$ as the number of the parent of the element with the number v. Let's say that for a non-root element v the *property of the heap is violated* if $a_v < a_{p(v)}$.

Help Andrew cope with the task!

Input

The first line contains a single integer n ($2 \le n \le 2 \cdot 10^5$).

The second line contains n space-separated integers $a_1, ..., a_n$ (- $10^9 \le a_i \le 10^9$).

Output

in a single line print n - 1 integers, separate the consecutive numbers with a single space — the number of elements for which the property of the k-ary heap is violated, for $k = 1, 2, ..., n - 1$.

Examples

input
5 1 5 4 3 2
output
3 2 1 0

input
6 2 2 2 2 2 2
output
0 0 0 0 0

Note

Pictures with the heaps for the first sample are given below; elements for which the property of the heap is violated are marked with red.

In the second sample all elements are equal, so the property holds for all pairs.