E. Till I Collapse

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Rick and Morty want to find MR. PBH and they can't do it alone. So they need of Mr. Meeseeks. They Have generated n Mr. Meeseeks, standing in a line numbered from 1 to n. Each of them has his own color. i-th Mr. Meeseeks' color is a_i.

Rick and Morty are gathering their army and they want to divide Mr. Meeseeks into some squads. They don't want their squads to be too colorful, so each squad should have Mr. Meeseeks of at most k different colors. Also each squad should be a continuous subarray of Mr. Meeseeks in the line. Meaning that for each $1 \le i \le e \le j \le n$, if Mr. Meeseeks number i and Mr. Meeseeks number j are in the same squad then Mr. Meeseeks number e should be in that same squad.

Also, each squad needs its own presidio, and building a presidio needs money, so they want the total number of squads to be minimized.

Rick and Morty haven't finalized the exact value of k, so in order to choose it, for each k between 1 and n (inclusive) need to know the minimum number of presidios needed.

Input

The first line of input contains a single integer n ($1 \le n \le 10^5$) — number of Mr. Meeseeks.

The second line contains n integers $a_1, a_2, ..., a_n$ separated by spaces ($1 \le a_i \le n$) — colors of Mr. Meeseeks in order they standing in a line.

Output

In the first and only line of input print n integers separated by spaces. i-th integer should be the minimum number of presidios needed if the value of k is i.

Examples

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

input
8 1 5 7 8 1 7 6 1
output
8 4 3 2 1 1 1 1

Note

For the first sample testcase, some optimal ways of dividing army into squads for each k are:

1. $[1], [3], [4], [3, 3]$
2. $[1], [3, 4, 3, 3]$
3. $[1, 3, 4, 3, 3]$
4. $[1, 3, 4, 3, 3]$
5. $[1, 3, 4, 3, 3]$

For the second testcase, some optimal ways of dividing army into squads for each k are:

1. $[1], [5], [7], [8], [1], [7], [6], [1]$
2. $[1, 5], [7, 8], [1, 7], [6, 1]$
3. $[1, 5, 7], [8], [1, 7, 6, 1]$
4. $[1, 5, 7, 8], [1, 7, 6, 1]$
5. $[1, 5, 7, 8, 1, 7, 6, 1]$
6. $[1, 5, 7, 8, 1, 7, 6, 1]$
7. $[1, 5, 7, 8, 1, 7, 6, 1]$
8. $[1, 5, 7, 8, 1, 7, 6, 1]$