

## D. Restore Permutation

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

An array of integers  $p_1, p_2, \dots, p_n$  is called a permutation if it contains each number from 1 to  $n$  exactly once. For example, the following arrays are permutations:  $[3, 1, 2]$ ,  $[1]$ ,  $[1, 2, 3, 4, 5]$  and  $[4, 3, 1, 2]$ . The following arrays are not permutations:  $[2]$ ,  $[1, 1]$ ,  $[2, 3, 4]$ .

There is a hidden permutation of length  $n$ .

For each index  $i$ , you are given  $s_i$ , which equals to the sum of all  $p_j$  such that  $j < i$  and  $p_j < p_i$ . In other words,  $s_i$  is the sum of elements before the  $i$ -th element that are smaller than the  $i$ -th element.

Your task is to restore the permutation.

### Input

The first line contains a single integer  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) — the size of the permutation.

The second line contains  $n$  integers  $s_1, s_2, \dots, s_n$  ( $0 \leq s_i \leq \frac{n(n-1)}{2}$ ).

It is guaranteed that the array  $s$  corresponds to a valid permutation of length  $n$ .

### Output

Print  $n$  integers  $p_1, p_2, \dots, p_n$  — the elements of the restored permutation. We can show that the answer is always unique.

### Examples

<b>input</b>	<a href="#">Copy</a>
3 0 0 0	
<b>output</b>	<a href="#">Copy</a>
3 2 1	

  

<b>input</b>	<a href="#">Copy</a>
2 0 1	
<b>output</b>	<a href="#">Copy</a>
1 2	

  

<b>input</b>	<a href="#">Copy</a>
5 0 1 1 1 10	
<b>output</b>	<a href="#">Copy</a>
1 4 3 2 5	

### Note

In the first example for each  $i$  there is no index  $j$  satisfying both conditions, hence  $s_i$  are always 0.

In the second example for  $i = 2$  it happens that  $j = 1$  satisfies the conditions, so  $s_2 = p_1$ .

In the third example for  $i = 2, 3, 4$  only  $j = 1$  satisfies the conditions, so  $s_2 = s_3 = s_4 = 1$ . For  $i = 5$  all  $j = 1, 2, 3, 4$  are possible, so  $s_5 = p_1 + p_2 + p_3 + p_4 = 10$ .

**Manthan, Codefest 19 (open for everyone, rated, Div. 1 + Div. 2)**

Finished

Practice



→ Virtual participation

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ACM-ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

Start virtual contest

→ Practice

You are registered for practice. You can solve problems unofficially. Results can be found in the contest status and in the bottom of standings.

→ Clone Contest to Mashup

You can clone this contest to a mashup.

Clone Contest

→ Submit?

Language: GNU G++11 5.1.0

Choose file: 选择文件 未选择任何文件

Be careful: there is 50 points penalty for submission which fails the pretests or resubmission (except failure on the first test, denial of judgement or similar verdicts). "Passed pretests" submission verdict doesn't guarantee that the solution is absolutely correct and it will pass system tests.

Submit

→ Problem tags

binary search data structures greedy implementation

No tag edit access

→ Contest materials

- [Announcement \(en\)](#) ☐
- [Tutorial \(en\)](#) ☐

---

[Codeforces](#) (c) Copyright 2010-2019 Mike Mirzayanov  
The only programming contests Web 2.0 platform  
Server time: Aug/26/2019 11:12:28<sup>UTC+8</sup> (h3).  
Desktop version, switch to [mobile version](#).  
[Privacy Policy](#).

Supported by



ITMO UNIVERSITY