Task 5: `Fruits`

Supermarkets usually have fruits for sale in sections, where each section is dedicated to a single type of fruit. The supermarket that Benson the Rabbit is visiting has N sections and N types of fruits. The sections are numbered from 1 to N and each fruit is numbered from 1 to N.

The ith fruit has tastiness i and cost C_i. **It is guaranteed that $C_i \leq C_j$ for all $1 \leq i < j \leq N$.**

Each of the N sections is to be assigned a **distinct** type of fruit. At the moment, the type of fruit assigned to section j is A_j. If $A_j = -1$, then section j is empty. Otherwise, fruit A_j is already assigned to section j. Once all N fruits have been assigned, the supermarket will open and Benson will enter the supermarket to buy the fruits.

Benson is very picky but also in a rush, so he will visit the sections in increasing order. Benson's basket is initially empty, and when he reaches a section, he will compare the tastiness of the fruit in that section to the tastiness of all of the fruits in his basket. If his basket is empty, or if the tastiness of the fruit at the current section is **greater than the tastiness of every other fruit in his basket**, Benson will add that fruit to his basket.

To maximise revenue, you have been tasked with assigning the fruits to the sections such that the sum of cost of fruits that Benson adds to his basket is maximised. As Benson is rushing for time, he might only visit the first few sections before going straight to the cashier. Help compute, for each k from 1 to N, the maximum possible revenue that can be achieved if Benson only visits the first k sections **given that the arrangement of fruits can change for different k.**

Input

Your program must read from standard input.

The input starts with a line with one positive integer N.

The second line contains N integers where the ith integer represents A_i.

The third line contains N integers where the ith integer represents C_i.

Output

Your program must print to standard output.

The output should contain a N integers on a single line. The kth one should be the maximum sum of costs that Benson will pay if the fruits are assigned optimially if he visits only the first

k sections.

Subtasks

The maximum execution time on each instance is 1.0s. For all testcases, the input will satisfy the following bounds:

- $1 \leq N \leq 400\,000$
- $1 \leq A_j \leq N$ or $A_j = -1$
- $1 \leq C_i \leq 10^9$
- $C_i \leq C_j$ for all $1 \leq i < j \leq N$

Your program will be tested on input instances that satisfy the following restrictions:

Subtask	Marks	Additional Constraints
1	6	$N \leq 8$
2	5	$A_j = -1$ for all $0 \leq j < n$
3	11	$N \leq 200$
4	13	$N \leq 2000$
5	23	$C_i = 1$ for all $0 \leq i < n$
6	42	-

Sample Testcase 1

This testcase is valid for all subtasks.

Input	Output
5 -1 -1 -1 -1 -1 1 1 1 1 1	1 2 3 4 5

Sample Testcase 1 Explanation

We can arrange the fruits in increasing order (1,2,3,4,5). Benson will take all fruits that he passes by, so for each k from 1 to 5, Benson will take k fruits which have a total cost of k.

Sample Testcase 2

This testcase is valid for subtasks 1, 3, 4 and 6.

Input	Output
5 -1 3 -1 -1 -1 1 2 2 2 3	3 4 7 9 9

Sample Testcase 2 Explanation

If Benson only visits the first section, it is optimal to put fruit 5 in section 1. This gives a cost of 3.

If Benson only visits the first 2 sections, it is optimal to put fruit 2 in section 1 and fruit 3 in section 2. This gives a cost of $2 + 2 = 4$.

If Benson only visits the first 3 sections, it is optimal to put fruit 2 in section 1 and fruit 3 in section 2 and fruit 5 in section 3. This gives a cost of $2 + 2 + 3 = 7$.

If Benson visits either the first 4 or all 5 sections, it is optimal to put the fruits in the order $2, 3, 4, 5, 1$. This gives a cost of $2 + 2 + 2 + 3 = 9$.

Sample Testcase 3

This testcase is valid for subtasks $3, 4, 5$ and 6. Note that the second line of input has been split into 2 lines for clarity's sake.

Input	Output
13 -1 -1 5 6 -1 -1 7 11 -1 -1 10 -1 -1 1 1 1 1 1 1 1 1 1 1 1 1 1	1 2 3 4 5 6 6 7 8 9 9 9 9

Sample Testcase 4

This testcase is valid for subtasks 3, 4 and 6. Note that the first line of output has been split into 2 lines for clarity's sake.

Input	Output
10 -1 -1 -1 -1 5 -1 -1 -1 9 -1 5 11 24 27 35 60 72 81 91 92	92 173 245 305 305 332 356 367 406 498