A Plus B

Borcsa has two arrays, each of them containing N non-negative integers.

The numbers in the first array are $A[0], A[1], \ldots, A[N-1]$ and the numbers in the second array are $B[0], B[1], \ldots, B[N-1]$. The numbers in both arrays are in increasing order, that is,

- $A[0] \leq A[1] \leq \ldots \leq A[N-1]$, and
- $B[0] \leq B[1] \leq \ldots \leq B[N-1]$.

Borcsa really likes arithmetical addition, so for each i from 0 to $N-1$ and for each j from 0 to $N-1$, inclusive, she computed the sum $A[i] + B[j]$.

Let array C contain all N^2 sums computed by Borcsa, sorted in *increasing* order. Your task is to find the first N values in C.

Implementation Details

You should implement the following procedure:

```
int[] smallest_sums(int N, int[] A, int[] B)
```

- N: the number of elements in each array.
- A, B: arrays of length N containing non-negative integers sorted in increasing order.
- This procedure should return an array of length N containing the N smallest sums obtained by adding an element from A and an element from B. The array should contain the sums in **increasing** order.
- This procedure is called exactly once for each test case.

Examples

Example 1

Consider the following call:

```
smallest_sums(2, [0, 2], [1, 4])
```

In this case, $N = 2$. There are $N^2 = 4$ sums computed by Borcsa:

- $0 + 1 = 1$
- $0 + 4 = 4$
- $2 + 1 = 3$
- $2 + 4 = 6$

Array C contains these sums sorted in increasing order, that is, $C = [1, 3, 4, 6]$. The $N = 2$ smallest elements in C are 1 and 3. Therefore, the procedure should return the array $[1, 3]$.

Example 2

Consider the following call:

```
smallest_sums(3, [0, 2, 2], [3, 5, 6])
```

The 9 pairwise sums (in increasing order) are $3, 5, 5, 5, 6, 7, 7, 8, 8$. The procedure should return the 3 smallest sums, that is, the array $[3, 5, 5]$.

Constraints

- $1 \leq N \leq 100\,000$
- $0 \leq A[i] \leq 10^9$ (for each i such that $0 \leq i < N$)
- $0 \leq B[j] \leq 10^9$ (for each j such that $0 \leq j < N$)
- A and B are sorted in increasing order.

Subtasks

1. (10 points) $N = 1$
2. (20 points) $N \leq 100$
3. (30 points) $N \leq 2\,500$
4. (40 points) No additional constraints.

Sample Grader

The sample grader reads the input in the following format:

- line 1: N
- line 2: $A[0]$ $A[1]$ $\ldots$ $A[N-1]$
- line 3: $B[0]$ $B[1]$ $\ldots$ $B[N-1]$

Let the elements of the array returned by `smallest_sums` be $c[0], c[1], \ldots, c[n-1]$ for some non-negative n. The output of the sample grader is in the following format:

- line 1: $c[0]$ $c[1]$ $\ldots$ $c[n-1]$