

Mirko is hungry as a bear, scratch that, programmer and has stumbled upon a local restaurant. The restaurant offers  $N$  meals and has an interesting pricing policy: each meal  $i$  has two assigned prices,  $A_i$  and  $B_i$ . Mirko pays  $A$  only for the **first ordered meal**, while  $B$  prices apply for **all other** meals.

Mirko can't decide how many meals to order. In order to make his decision easier, he has asked you to compute, for each  $k$  between 1 and  $N$  (inclusive), the minimum total price for  $k$  ordered meals. Mirko doesn't care which particular meals he orders or in which order he orders them, however he won't order the same meal twice. Order, order, order.

### INPUT

The first line of input contains the positive integer  $N$  ( $2 \leq N \leq 500\,000$ ), the number of different meals offered by the restaurant.

Each of the following  $N$  lines contains two positive integers,  $A_i$  and  $B_i$  ( $1 \leq A_i, B_i \leq 1\,000\,000\,000$ ), the prices for meal  $i$  as described above.

### OUTPUT

Output must consist of  $N$  lines, where line  $k$  contains the minimum price for ordering exactly  $k$  different meals.

### SAMPLE TESTS

<b>input</b> 3 10 5 9 3 10 5	<b>input</b> 2 100 1 1 100	<b>input</b> 5 1000000000 1000000000 1000000000 1000000000 1000000000 1000000000 1000000000 1000000000 1000000000 1000000000
<b>output</b> 9 13 18	<b>output</b> 1 2	<b>output</b> 1000000000 2000000000 3000000000 4000000000 5000000000

#### Clarification of the first example:

$k = 1$ : Mirko pays  $A_2 = 9$  for the starting meal 2.

$k = 2$ : Mirko pays  $A_1 = 10$  for the starting meal 1, then  $B_2 = 3$  for meal 2.

$k = 3$ : Mirko pays  $A_1 = 10$  for the starting meal 1, then  $B_2 = 3$  for meal 2, and finally  $B_3 = 5$  for meal 3.