Tax Collection

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Dictator S wants to collect taxes from his citizens. He is given a list of N towns, each with A_i citizens, representing the order for tax collector K to visit these N towns. Dictator S can reorder the list by swapping the positions of certain towns on the list.

On the i^{th} day ($1 \leq i \leq N$), tax collector K will visit the towns from 1 to i on the list and collect taxes from the citizens. In other words, on the i^{th} day, tax collector K will collect taxes from the **prefix sum** of citizens in the first i towns on the list. For example, if the list is (5, 10, 9, 22), then the total number of citizens taxed is $5 + 15 + 24 + 46 = 90$.

However, dictator S has political enemies and can only reorder certain towns. Help dictator S collect taxes from the **maximum number of citizens** possible!

Input

The first line of input contains one integer, N ($1 \leq N \leq 10^5$).
The second line of input contains N integers, with the i^{th} integer representing A_i ($1 \leq A_i \leq 10^5$).
The third line of input contains N integers, with the i^{th} integer being either 0 or 1.

- 0 indicates that the order of the i^{th} town cannot be changed, meaning it must remain on the i^{th} position on the list.
- 1 indicates that the order of the i^{th} town can be changed, meaning it can be swapped with another town on a different position.

Output

Output one integer, the maximum number of citizens Dictator S can tax.

Scoring

Subtask	**Score**	**N**	**Additional constraints**
1	20	-	$A_i = 1$
2	20	$N \leq 10^3$	The order of all towns can be changed
3	60	-	-
4	0	Sample Testcases	

Examples

`standard input`	`standard output`

`4`
`5 10 9 22`
`0 1 1 1`
	`115`

`6`
`1 2 3 4 5 6`
`1 1 1 1 1 0`
	`76`

Note

For sample testcase 1:
Initial list: $\underline{5}$, 10, 9, 22 (Position of 5 cannot be changed)
Optimal list: $\underline{5}$, 22, 10, 9
$(5) + (5 + 22) + (5 + 22 + 10) + (5 + 22 + 10 + 9) = 115$

For sample testcase 2:
Initial list: 1, 2, 3, 4, 5, $\underline{6}$ (Position of 6 cannot be changed)
Optimal list: 5, 4, 3, 2, 1, $\underline{6}$
$(5) + (5 + 4) + (5 + 4 + 3) + (5 + 4 + 3 + 2) + (5 + 4 + 3 + 2 + 1) + (5 + 4 + 3 + 2 + 1 + 6) = 76$