

# Spy Office

Tyrion Lannister is very smart man, and he knows importance of being informed, especially when you are important man behind the Iron Throne. So Tyrion placed one spy in every major city of Seven Kingdoms.

But long before internet existed, information was carried by foot, so it was very important to organise your spies optimally.

Seven Kingdoms can be imagined as straight road with  $N$  cities on it. Cities are numbered from  $1$  to  $N$  inclusive, with cities  $i$  and  $i+1$  being neighbouring. Tyrion is located in city numbered  $1$ , called King's Landing. We know distance between two neighbouring cities, and it is given in array  $D$ . Distance between city  $i$  and  $i+1$  is  $D[i]$  kilometers.

After spy in city  $i$  hears something important he starts preparing for departure. He needs exactly  $T[i]$  days to prepare. After that he starts traveling to King's Landing with constant speed  $V[i]$  kilometers per day. Spies never travel away from King's landing, that is, at every point in time, they try to reduce their distance to King's Landing. After he enters some city in between, he can either tell his news to spy located in that city, after which that spy repeats same steps, or he can continue traveling without telling anything to that spy.

Tyrion needs to know what is smallest amount of time needed for information from every city to reach King's Landing (city  $1$  where Iron Throne is located).

$N \leq 100000$  ( in 60% of tests  $N \leq 1000$  )  
 $0 < D[i], T[i], V[i] \leq 100$

## Input

In first line of the input, there is integer  $N$  (number of cities). Second line consists of  $N-1$  integers describing array  $D$ .  $N-1$  lines follow each containing  $T[i]$  and  $V[i]$ .

## Output

Output  $N-1$  numbers, smallest number of days for each city. Also note, that solutions are in fact decimal numbers, but you just output integers.

## Example

**Input:**  
7  
2 9 7 2 2 10  
6 6  
9 2  
9 10  
2 3  
1 1  
6 6

**Output:**  
6  
14  
11  
9  
12  
11

Note : Actual results are decimal numbers, like shown below. But to avoid precision errors, output rounded numbers with 0 decimal places.

Use `printf("%.0lf")` formatting to output your results

6.33333

14.50000  
10.80000  
8.66667  
11.66667  
11.33333