

2780 – Homework

Source : <http://acm.tju.edu.cn/toj/showp2780.html>

Doing homework often makes students understand knowledge deeply. As a student of UESTC, WCM usually has much homework to do. Every day he gets a set of problems from teachers. Problem i will take t_i time to complete. Given a schedule (i.e., an ordering of the problems), let C_i denote the finishing time of problem i . For example, if problem j is the first to be done, we would have $C_j=t_j$. Each problem i also has a given weight w_i that represents its importance to the student's mastering of the correlative knowledge. He wants to order the problems to minimize the weighted sum of the completion times, namely $w_1C_1 + w_2C_2 + w_3C_3 + \dots + w_nC_n$.

You should design an efficient algorithm to solve this problem. That is, you are given a set of n problems with a processing time t_i and a weight w_i for each problem. You want to order the problems so as to minimize the weighted sum of the completion times, $\sum_{(i=1..n)}w_iC_i$.

Example: Suppose there are two problems: the first takes time $t_1=2$ and has weight $w_1=12$, while the second problem takes time $t_2=3$ and has weight $w_2=4$. Then doing problem 1 first would yield a weighted completion time of $12*2+4*5=44$, while doing the second problem first would yield a larger weighted completion time of $4*3+12*5=72$. Apparently, 44 is the minimum of the weighted sum of completion times, $\sum_{(i=1..n)}w_iC_i$.

Input

The input contains an integer T on the first line, which indicates the number of test cases. Each test case consists of three lines. The first line contains one integer n , ($0 < n \leq 1000$), which is the number of problems. The second line contains n numbers, t_1, t_2, \dots, t_n , ($0 < t_i \leq 20$), t_i being the time problem i takes. The third line contains n numbers, w_1, w_2, \dots, w_n , ($0 < w_i \leq 20$), w_i being the weight of problem i .

Output

For each test case output one line containing a single integer, which represents the minimum of the weighted sum of the completion times, $\sum_{(i=1..n)}w_iC_i$.

Example

Input :

```
1
2
2 3
12 4
```

Output :

```
44
```