Q2.

某個自動化系統中有一個存取物品的子系統，該系統是將 N 個物品堆在一個垂直的貨架上,每個物品各佔一層。系統運作的方式如下:每次只會取用一個物品，取用時必須先將在其上方的物品貨架升高，取用後必須將該物品放回，然後將剛才升起的貨架降回原始位置，之後才會進行下一個物品的取用。

每一次升高某些物品所需要消耗的能量是以這些物品的總重量來計算,在此我們忽略貨架的重量以及其他可能的能量消耗。現在有 N 個物品，第 i 個物品的重量為 W(i) 而需要取用的次數為 f(i)，我們需要決定如何擺放這些物品的順序來讓消耗的能量越少越好。舉例來說，有兩個物品 w(1)=1 w(2)=2。f(1)=3 f(2)=4。也就是說物品 1 的重量是 1 需要取 3 次，物品 2 的重量是 2 需要取 4 次。我們有兩個可能的擺放順序（由上而下）:

(1,2)，也就是物品 1 放在上方，2 在下方。那麼，取用 物品 1 的時候不需要能量,而每次取用 物品 2 的能量消耗是 w(1)=1,因為 物品 2 需要取用 f(2)=4 次，所以消耗能量數為 w(1)*f(2)=4。

(2,1)，也就是物品 2 放在 物品 1 的上方。那麼，取用 物品 2 的時候不需要能量,而每次取用 物品 1 的能量消耗是 w(2)=2,因為 1 需要取用 f(1)=3 次，所以消耗能量數為 w(2)*f(1)=6。

在所有可能的兩種擺放順序中，最少的能量是 4，所以答案是 4。再舉一例，若有三物品而 w(1)=3、w(2)=4、w(3)=5、f(1)=1、f(2)=2、f(3)=3。假設由上而下以 (3,2,1) 的順序， 此時能量計算方式如下：取用物品 3 不需要能量，取用物品 2 消耗 w(3)*f(2)=10，取用物品 1 消耗 (w(3)+w(2))f(1)=9，總計能量為 19。如果以 (1,2,3) 的順序，則消耗能量為 32+ (3+4)*3=27。事實上，我們共有 3!=6 種可能的擺放順序，其中順序 (3,2,1) 可得到最小消耗能量 19。

In a certain automated system, there is a subsystem for storing and retrieving items. This system stacks N items on a vertical shelf, each item occupying one layer. The system operates as follows: each time, only one item will be retrieved. When retrieving, the shelves above the item must be lifted. After retrieval, the item must be placed back, and the previously lifted shelves must be lowered back to their original position before proceeding to retrieve the next item.

The energy required to lift certain items each time is calculated based on the total weight of these items. Here, we ignore the weight of the shelf and any other possible energy consumption. Now, there are N items, and the weight of the i-th item is W(i) with the

number of times it needs to be retrieved being f(i). We need to determine the order of these items to minimize energy consumption. For example, there are two items w(1)=1 w(2)=2, f(1)=3 f(2)=4. This means item 1 weighs 1 and needs to be retrieved 3 times, and item 2 weighs 2 and needs to be retrieved 4 times. We have two possible stacking orders (from top to bottom):

(1,2), which means item 1 is on top and 2 is below. In this case, retrieving 1 does not require energy, while the energy consumed for each retrieval of 2 is w(1)=1 because 2 needs to be retrieved f(2)=4 times. So the total energy consumed is w(1)*f(2)=4.

(2,1), which means item 2 is on top of 1. In this case, retrieving 2 does not require energy, while the energy consumed for each retrieval of 1 is w(2)=2 because 1 needs to be retrieved f(1)=3 times. So the total energy consumed is w(2)*f(1)=6.

Among all possible stacking orders, the minimum energy is 4, so the answer is 4. Another example: if there are three items with w(1)=3, w(2)=4, w(3)=5, f(1)=1, f(2)=2, f(3)=3. Assuming the order (3,2,1) from top to bottom, the energy calculation is as follows: retrieving item 3 requires no energy, retrieving item 2 consumes w(3)*f(2)=10, and retrieving item 1 consumes (w(3)+w(2))f(1)=9. The total energy is 19. If the order is (1,2,3), the energy consumption is 32 + (3+4)*3=27. In fact, we have 3!=6 possible stacking orders, and among them, the order (3,2,1) yields the minimum energy consumption of 19.

Input Format

```
<Total number of objects> # 物品數量
w(1) w(2) ... w(N)        # N個值小於1000之正整數，依序為各個物品對應的重量
f(1) f(2) ... f(N)        # N個值小於1000之正整數，依序為各個物品對應的取用次數
```

Example 1:

```
Input: 2
       20 10
       1 1
Output: 10
```

Example 2:

```
Input: 3
       3 4 5
       1 2 3
Output: 19
```

What You Should Do

Please finish your code inside the file Q2.cpp. After finishing, you can use following commands to check your answer.

Execution

```
$ make
$ make run  # Use make run to run specific testcase, remember to change variable *CASE* inside Makefile
```

Verifier

You can use *verifier* to check a specific pattern to make sure whether your answer is correct.

```
$ make check # Use make check to check specific testcase, remember to change variable *CASE* inside Mak
```

AutoChecker

You can also use 01_autoCheck to test all the patterns.

- Tips: You can change PATNUM inside 01_autoCheck to test different number of pattern.

```
$ ./01_autoCheck
```

這題寫完後可以先編譯後，必須打上./Q2 inputfile outputfile 來執行程式，testcase 裡有提供兩個 input 檔使用，就是上面的 example，例如./Q2 ./testcase/1.in out.txt，可以用來確認 out.txt 裡的輸出是否跟上面的 example 一致，而 output 只需要輸出一個正整數即可，若測試結果都一樣再使用 autoCheck 來測試所有測資。
使用 autoCheck 前記得先在終端機打上 chmod 755 verifier 以及 chmod 755 01_autoCheck 以及 chmod 755 ./testcase/caseGen !!!