UVa Online Judge

11269 Setting Problems

As you see, setting problems for a programming contest is a tough job. There are so many things to do like creating problems, solutions, data files, verification of problem statements, writing alternate judge solutions etc. etc. Given the responsibility of creating the problemset for If you can not crash judges by solutions, crash contestants by problems programming contest, Sultan and GolapiBaba have realized that to the backbone. Finally they agree that they will set N problems for the contest. For each of the problems, first Sultan will create the problem statement, solution and i/o data. After he finishes his work, GolapiBaba does the verification and alternate solution writing part for that particular problem. Each of them needs a particular amount of time to complete their tasks for a certain problem. Also, none of them works on more than one problem at a time. Note that, GolapiBaba can start working on a problem immediately after Sultan finishes that problem or he may wish to start that problem later.

You will be given the times that Sultan and GolapiBaba requires to complete their respective tasks for every single problem. Determine the minimum possible time required to complete the whole problemset.

Input

There are around 50 test cases. Each test case starts with a single integer N $(1 \leq N \leq 20)$, the number of problems in the contest. The next line contains N integers S_i $(1 \leq S_i \leq 100, 1 \leq i \leq N)$ where S_i denotes the time required for Sultan to complete his tasks for problem i. The next line has N more integers G_i $(1 \leq G_i \leq 100, 1 \leq i \leq N)$ where G_i denotes the time required for Golapibaba to complete his tasks on problem i.

Output

For each test case, print the minimum time required to complete the problemset.

Sample Input

```
3
8 1 6
1 6 3
3
4 5 6
1 1 6
```

Sample Output

```
16
16
```