ICPC (International Collegiate Programming Contest), as you might know, is a team programming contest for college students. Each team consists of exactly three students and they will work on a number of programming problems.

Andi, Budi and Chandra plan to participate in this year ICPC as a team. As for their team strategy, they come up with a simple one:

- In the first 20 minutes of 5 hours contest, they will read through all problems. Then each of them will assign a number to each problem. This number indicates how many minute(s) it will take for him to get the problem accepted (correct solution). Then they will start to code, meaning that they only have 280 minutes to really work on the problems. You may assume that they always get accepted on time whenever they work on a problem.
- There's only one computer for each team, so it's not possible for them to code two different problems simultaneously.
- To avoid any brain fatigue and adrenaline rush (because they attend competitions so frequently), they decided to switch role after each problem, such that none of them will work at the computer for more than one problem consecutively.
- They want to solve as many problems as they can. The order of problem to be solved does not matter.

Input

The first line of input contains an integer T, number of test cases to follow. Each case begins with an integer N ($1 \le N \le 12$) in one line, denoting the number of problems. The second line contains N integers, $A_{1..N}$ ($1 \le A_i \le 300$), denoting the total time (in minutes) needed by Andi to solve i^{th} problem. The third and fourth line will correspond to the total time needed by Budi and Chandra respectively and will have the same input format as the second line.

Output

For each case, print in a single line containing the maximum total number of problem that can be solved by that team.

Explanation for 1st sample case:

Actually Andi could solve all the three problems alone, but the team has decided that none of them should work at the computer for more than one problem consecutively.

Explanation for 2nd sample case:

The team can solve all the problems. Here is one solution:

- Let Budi work on Problem-2 (100 minutes).

- Switch to Andi and let him work on Problem-1 (50 minutes).
- Switch to Budi again and let him work on Problem-3 (30 minutes).
- Finally, switch to Chandra and let him work on Problem-4 which (100 minutes).

Overall, they need 100 + 50 + 30 + 100 = 280 minutes.

Sample Input

```
2
3
100 100 80
190 120 90
120 150 100
4
50 20 300 300
200 100 30 250
140 120 100 100
```

Sample Output

```
2
4
```

Jakarta 2008-2009