Programming contests are being arranged so frequently these days. While this might be a good news for the contestants, the scenario is completely opposite for the problemsetters. So far, the problemsetters somehow managed to produce some sorts of a set & say "No problem!". But it is doubtful how long will it be possible if the trend of arranging contests in a short notice continues.

You are given the number of problems created in every month of a year and number of problems required in each month. If N problems are required in a month & there are not enough problems at that time, all contests of that month is canceled. Write a program to determine if there are enough problems for the contests. Please keep in mind that, if a problem is created in month X, it can only be used in month $X + 1$ & the later months.

Input

The first line of every test case has an integer S ($0 \leq S \leq 100$). Number of problems that is ready at the beginning of the year. The 2-nd line has 12 space separated integers, denoting the number of problems created in each of the 12 months of that year. The months are in the same order as they appear in a year. The 3-rd line has another 12 space separated integers, the number of problems required to use in contests in those 12 months (With the same order as above). These integers will be between 0 & 20 (inclusive). The end of input will be denoted by a negative integer.

Output

For each test case, print a line of the form, '`Case` X`:`', where X is the case number. Then print 12 lines. If there are enough problems to meet the requirements in month i ($1 \leq i \leq 12$), print '`No problem! :D`' in the i-th line, otherwise print '`No problem.  :(`'

Sample Input

```
5
3 0 3 5 8 2 1 0 3 5 6 9
0 0 10 2 6 4 1 0 1 1 2 2
-1
```

Sample Output

```
Case 1:
No problem! :D
No problem! :D
No problem. :(
No problem! :D
No problem! :D
No problem! :D
No problem! :D
No problem! :D
No problem! :D
No problem! :D
No problem! :D
No problem! :D
```