Problem 2 – Beer Stock

Students in SoftUni really like drinking beers, so the Game Bar has to always be stocked with beer and with all the students that will be celebrating passing the Programming Basics Exam a lot of beer is going to be needed today. Normally Stamat would be keeping track of the stock, however he was at a party last night and got wasted, since he's too drunk to go and count the beers he asked you to write him a program to do it instead.

Before the exam starts the students who will want a beer will tell you, so on the first line you'll get the amount of reserved beers, then the shipments will start coming in. Each shipment will be in the format "{amount} {type}", where type is the type of package received – either beers, sixpacks or cases and amount is the amount received, each sixpack holds exactly 6 beers and each case holds exactly 24 beers. However since the world isn't perfect in every 100 beers exactly the 100th is always broken so it gets discarded. On the last line you'll receive the text "Exam Over", at that point you have to calculate all the beers received and if they are more or equal to the amount of reserved beers print "Cheers! Beer left: {amount of cases left} cases, {amount of sixpacks left} sixpacks and {amount of beers left} beers." or if they are less print "Not enough beer. Beer needed: {amount of cases needed} cases, {amount of sixpacks needed} sixpacks and {amount of beers needed} beers." Where {amount of cases left/needed}, {amount of sixpacks left/needed} and {amount of beers left/needed} represent respectively the amount of beer left/needed, represented so that the most amount of cases are used, after which the most amount of sixpacks are used (see the Examples to get a better idea).

Input

The input data should be read from the console.

- On the first line you'll receive the amount of reserved beers.
- On each of the next lines you will be given a string representing a shipment in the format "{amount} {type}"*, until you receive the command "Exam Over".
- The names will always be given in plural regardless of amount "beers", "sixpacks", "cases".
- The amount and the type will be separated by exactly **one space**, there will be no leading or trailing spaces in the input.

The input data will always be valid and in the format described. There is no need to check it explicitly.

*HINT: Use string.Split() to separate {amount} from {type}.

Output

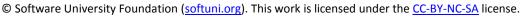
The output should be printed on the console.

- If the amount of beer is more or equal to the amount of reserved beer:
 - "Cheers! Beer left: {amount of cases left} cases, {amount of sixpacks left} sixpacks and {amount of beers left} beers."
- If the amount of beer is less than the amount of reserved beer:
 - "Not enough beer. Beer needed: {amount of cases needed} cases, {amount of sixpacks needed} sixpacks and {amount of beers needed} beers."
- The names must always be printed in in plural regardless of amount "beers", "sixpacks", "cases".

Constraints

- The input lines will be in the range: [1... 100].
- The amount of reserved beer, and the amount in a shipment will be valid integers in the range [0... 2,147,483,647].
- Allowed working time for your program: 0.25 seconds.
- Allowed memory: 16 MB.



















Examples

Input	Output	Comments
197 1 beers	Not enough beer. Beer needed: 4 cases, 1 sixpacks and 1 beers.	Amount of beers we have -> 2(cases) * 24 + 7(sixpacks) * 6 + 1 + 3 = 94 beers
2 cases		Since 94 is less than 197 (the reserved beer), we need 197-94 = 103 more beers
7 sixpacks 3 beers		103 / 24 = 4 cases + 7 beers left 7 / 6 = 1 sixpack + 1 beer left
Exam Over		So in the end we need 4 cases, 1 sixpack and 1 beer.

Input	Output	Comments
23	Cheers! Beer left: 7 cases, 2 sixpacks and 1 beers.	Amount of beers we have -> 11 + 3 + 8(cases) * 24 =
11 beers		206 beers
3 beers		However since every 100 th beer is always broken we actually have 206 - (206 / 100) = 206 - 2 = 204 beers
8 cases		
Exam Over		Since 204 is more than 23 (the reserved beer), we have 204-23 = 181 beers left
		181 / 24 = 7 cases + 13 beers left
		13 / 6 = 2 sixpack + 1 beer left
		So in the end we have 7 cases, 2 sixpacks and 1 beers left.



















