Alex and money

Alex plays a game where he has different names with the amount of money. He wants to give some money to his friend in pairs. But there are some conditions to do that job-

- Alex can give money if and only if the average of money is more than or equal to 9000 TK.

Your task is to find out all the pairs of names where the condition of Alex is fulfilled. The pair should be printed once. If a name is given more than once, ignore that strictly.

Input: The first line should take, N for test cases(N<=100). The next lines will take N inputs.

Output: Print all output using “and”. If there is no average print ”Not Found”. For better understand see the output test case

Input:

1
David: 9005, Beckham: 10000, Messi: 10005, Neuer: 6000, Suarez: 7000

Output:

David and Beckham
David and Messi
Beckham and Messi

Alex and Reverse Game

Alex wants to make a dictionary. He collects some words and meanings from different sources and inputs reversed as he does everything in reverse order. Before making the program he is facing a problem to handle the exceptions. The problem is-

- A word can be unique but it has some different meanings.

He needs help from you to handle the problem. You have to write a program where-

- You need to swap words and meanings but notice that a word can have multiple meanings. Store all the multiple meanings besides the word as dictionary stores like a list.

Input: The first line should take N for test cases(N<=100). The next line will take inputs.

Output: The full output should be print in one line.

Alex and Interval

Alex is facing a problem and he needs a help from you.

You must make a program that reads a float-point number and print a message saying in which of the following intervals the number belongs: [0,25] (25,50], (50,75], (75,100]. If the read

number is less than zero or greater than 100, the program must print the message “Get Out!” that means "Out of Interval".

The symbol '(' represents greater than. For example:

[0,25] indicates numbers between 0 and 25.0000, including both.

(25,50] indicates numbers greater than 25 (25.00001) up to 50.0000000.

Input

First take n(n<=100) as a test case. Then add The input file contains a floating-point number.

Output

Example

Input	**output**
4 25.01 25.00 100.00 -25.02	(25,50] [0,25] (75,100] Get Out!