**Race to the Ice Cream Truck!**

**Scenerio Overview**:

Two kids and an ice cream truck are at various positions on a line. You will be given their starting positions. Your task is to determine which kid will reach the truck first, assuming the truck does not move and the kids are running at equal speed. If the kids arrive at the same time, the truck will be allowed to move to the next corner while the kids fight over the last cone.  
  
You are given ***i*** runs in a form of **k1,**k2**,** trepresenting the respective positions for ***kid\_1***, ***kid\_2***, and ***truck***.  
  
Complete the function called ***TwoKidsAndATruck*** to return the appropriate answer to each run, which will be printed on a new line.

* If kid \_1 catches the truck first, print *Kid 1*.
* If kid\_2 catches the truck first, print *Kid 2*.
* If both kids reach the truck at the same time, print Truck as the two kids fight and truck escapes.

**Example**:

**k1 = 7  
k2 = 4  
t = 5**  
  
The kids are at positions **7** (Kid 1) and **4** (Kid 2), and the truck is at position **5**. Kid 1 is **2** units away from the truck, and Kid 2 is **1** unit away; therefore, Kid 2 should arrive at the truck first. Return 'Kid 2'.

**Task Description:**

Complete the *TwoKidsAndATruck* function in the editor below.

TwoKidsAndATruck has the following parameter(s):

* *int k1*: Kid 1's position
* *int k2*: Kid 2's position
* *int t*: Ice Cream Truck's position

**Returns**

* *string:* Either 'Kid 1', 'Kid 2', or 'Truck'

**Input Format**

The first line contains a single integer,***i***, denoting the number of runs.  
Each of the ***i*** subsequent lines contains three space-separated integers describing the respective values of  ***k1*** (kid 1's location), ***k2*** (kid 2's location), and ***t*** (truck's location).

**Constraints**

* 1 <= i <= 100
* -100 <= k1, k2, t <= 100

**Sample Input**

2  
1 2 3  
1 3 2

**Sample Output**

Kid 2  
Truck

**Explanation:**

Run 1: The positions of the kids and truck are position 1, 2, and 3 respectively.  
Kid 2 will get to the truck first so we print 'Kid 2' on a new line.  
  
  
Run 2: In this run, kids 1 and 2 will reach the truck at the exact same time because each kid is 1 unit away.  
Because the truck leaves as a result of the kids fighting, we print 'Truck' on a new line.

**AutoComplete**

Implement an autocomplete system. That is, given a query string s and a set of all possible query strings, return all strings in the set that have s as a prefix (in alphabetical order).  
The query string is given on the first line of the input, followed by all the possible query strings on the subsequent lines.

**Sample Input**  
do  
dog  
cat  
bird  
donut  
cake  
dough  
  
**Sample Output**  
dog  
donut  
dough

**Word Count**  
  
You are given a string of text as input. Your job is to **count the number of times each unique word appears** in the text and then print all words and their counts ***for words that appear more than once***. For this problem, words are defined as any characters separated by spaces. Punctuation will not be included in the input text, only alphanumeric characters. Capitalization can be ignored. Output the word (lowercased), a colon, and then the count of the word,***in alphabetical order***.  
  
**Sample Input**

The big dog jumped high over the lazy red fox and the man with the red hat sat high in the chair and laughed

**Sample Output**  
and: 2  
high: 2  
red: 2  
the: 5