Problem B
Company

Input File: B.DAT
Program Source File: B.PAS or B.C or B.CPP

A company pays different hourly wages for different job types. Each week the company keeps evidence of the total number of work hours for every job type and the total amount paid to all employees for that week. In different weeks different job types can be accomplished. The hourly wage for any job type in the same company remains unchanged. The hourly wage for any job type is a positive integer and the ratio between the maximum wage and the minimum wage is less than 6.

You are asked to write a program that computes the hourly wage for every job type using the data collected a period of time (one or several weeks). The number of job types is limited to 200 and the number of weeks in one data set is limited to 50.

The input file contains several data sets in text format.

The format for the data set is:

➢ The number of lines for the data set;
➢ **information_for__one_or_more_weeks**

The format of the information for a week is the following:

➢ **job_type1   number_of _hours1**
➢ **job_type2   number_of _hours2**
➢ **job_type3   number_of _hours3**
➢ ....

➢ **.   total_paid** (the dot marks the ending of info for a week)

The job type is represented as a string of characters (limited to 20). The number of hours is a positive integer (smaller than 1E5). The total paid is a positive integer (smaller than 2E10).

The program should write to standard output (for every job type involved in the data set) a line containing the job type and the hourly wage. The output for a data set ends with a line containing a dot.

If the program cannot compute a unique hourly wage for every job type it will print **"Incomplete data"** and if it cannot compute an integer hourly wage it will print **"Inconsistent data"**

An example is given in the following table:

input	output
`5`	`Incomplete data`
`job1   6`	`.`
`job2   5`	`job3   20`
`job8   4`	`job2   10`
`job10  3`	`job1   30`
`.      100`	`.`
`13`	
`job3   1`	
`job2   2`	
`.      40`	
`job1   3`	
`job2   1`	
`.      100`	
`job1   1`	
`job3   2`	
`job2   3`	
`.      100`	
`job1   1`	
`job2   5`	
`.      80`	