# Problem 1 – Dragons Hatching

Teamwork is importatn but team points is what matters the most especially if we have a team full of dragons. Your task is to write down a program that calculates the points in a local dragon hatching competition. Never heard of dragon hatching before? Nevermind the rules are simple anyway.

You will receive **N**, an integer – the **count** of **teams participant in the competition**.

Since drgaons are ancient and their mood changes often you will receive also the **current dragon hatching value {dragonHatchingValue}, represented as** **a floating-point number**.

For **each team**, you will receive an input line in the following format:

“{dragonsHatched} {teamMembersCount}”

The **dragonsHatched** and the **teamMembersCount** will be integers.

You will have to calculate **the team points** by dividing the number of dragonsHatched by teamMembersCount {teamPoints} = {dragonsHatched /teamMembersCount}.

Your task is to **sum all team points for all the participant teams**, and then **divide** that sum by the **dragonHatchingValue**, and **print** the **result**.

If a **division** is **not possible**, just print the **sum** of **all team points**.

## Input

* On the **first input line** you will receive **N** – the **amount** of **teams**.
* On the **second input line** you will receive the **dragonHatchingValue**.
* On the **next N input lines** you will receive **information** about each team in the format specified above.

## Output

* As output you must print the sum of all **team points** **divided** by the **dragonHatchingValue**.
* **If a division is not possible you must print the total sum of team points**.
* The output should be **rounded** and **printed** to **3 places** after the **decimal point**.

## Constraints

* The **amount** of **teams** – **N** will be an **integer** in **range [0, 100]**.
* The **dragonHatchingValue** will be a **floating-point number** in **range [0, 9]**.
* The **dragonsHatched** will be an **integer** in **range [-231, 231]**.
* The **teamMembersCount** will be an **integer** in **range [1, 10000]**.
* Allowed working **time** / **memory**: **100ms** / **16MB**.

## Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comment** |
| 4  4  2000 10  1000 5  5000 2000  3000 30 | 125.625 | 2000 / 10 = 200  1000 / 5 = 200  5000 / 2000 = 2.5  3000 / 30 = 100  200 + 200 + 2.5 + 100 = 502.5  502.5 / 4 = 125.625 |
| 2  2  100000 50  200000 25 | 5000.000 | 100000 / 50 = 2000  200000 / 25 = 8000  2000 + 8000 = 10000  10000 / 2 = 5000  (rounded to 3rd symbol) = 5000.000 |