# Problem 1 – Travel Plans

*Mariyka really wishes to go on a world trip, but so far she’s never had the chance and so she finally decided to give it a try. But as beginning she need to collect some money, right?*

Write a JavaScript program that calculates the **total amount** of **gold** Mariyka managed to collect. Mariyka is **specialized** in some professions, **average** in others and **clumsy** at yet others. Your program will receive an **array with professions and the amount of gold** for each activity. Mariyka is so good at her **Specialized** professions that **every second** customer of Specialized profession gives her **200 additional** **gold right after she spends the gold mentioned below for candies**. She’s as well so bad at her **Clumsy** professions that every **second** customer from the Clumsy professions gives her **5% less** **gold** and **every third – 10%.**

Also, Mariyka is **spending 20% of every** activity she does that she’s **specialized** in, of course she spends them for candies.

Have in mind that Mariyka **does not** accept to work for **less** than **200 gold** in her **specialized** professions.

As **input** you will receive an array of strings.

Each string will consist of the following information with format:

“Professions : Gold offered”

Professions will be as follows:

Specialized:  
Programming, Hardware maintenance, Cooking, Translating, Designing.  
Average:  
Driving, Managing, Fishing, Gardening.

Clumsy:  
Singing, Accounting, Teaching, Exam-Making, Acting, Writing, Lecturing, Modeling, Nursing.

As output you need to print the total cash Mariyka has made.

If the amount is less than 1000 gold , she need to collect more gold until she get enough. Else she will be able to move to the next task and start planning her travel!

### Input

You will receive **one argument–** an **array strings** as shown above.

### Output

Print on the **console** the total amount of **gold** and if Mariyka has collected **enough or not**.

The output should be **formatted** to the second delimiter.

### Constraints

* The **number** of **elements** in the **input argument** will be in range **[1..100] inclusive**
* There **will** be **no invalid** **input**
* There **will** be **no negative Gold**

### Examples

|  |
| --- |
| **Input** |
| ["Programming : 500", "Driving : 243", "Singing : 100", "Cooking : 199"] |
| **Output** |
| Final sum: 743.00  Mariyka need to earn 257.00 gold more to continue in the next task. |

|  |
| --- |
| **Input** |
| ["Programming : 500", "Driving : 243.55", "Acting : 200", "Singing : 100", "Cooking : 199’, "Hardware maintenance : 800", "Gardening : 700", "Programming : 500"] |
| **Output** |
| Final sum: 2878.55  Mariyka earned 1878.55 gold more. |