

Problem 4 – Master Herbalist

Penka is a young and aspiring herbalist, every day she gets up early and goes collecting herbs in the mountains, however the mountains are big and she only has a limited amount of time for gathering each day. At the end of the day she sells the tinctures she prepares with the herbs at the market, she repeats this process until the end of the season at which point she has to calculate the average money she made per day. Since Penka is too busy gathering herbs and doesn't want to lose time calculating she asked you to make a program which would do the calculations for her.

You will receive input from the console, on the first line you will receive a number representing Penka's **daily expenses**. On each of the next lines you will receive input in the format "{hours} {path} {price}" until you receive the command "**Season Over**". The **hours** in the format represents the hours Penka has for gathering herbs that day, **path** will be a string containing only capital letters representing the path through the mountains she will take and **price** represents the money she gets for each herb sold that day (assume Penka always sells all the herbs she gathers on the same day). Penka will traverse the **path linearly from left to right** and each letter represents a location in the mountain which Penka will search for herbs, if the letter is "**H**" it means Penka found **1 herb**, any other letter means she found **nothing**. Regardless if she finds or doesn't find anything a location will **take exactly 1 hour** to get searched. If Penka has more hours for gathering than the length of the path, **she will start again from the beginning and all the herbs will have regrown**. Upon receiving the command "**Season Over**" you have to calculate Penka's **average earnings** per day of the season, the average being the **total amount of money** Penka made that seasons **divided by the number of days** that season (**number of days** being the number of input lines excluding the first and the last line (the line with her **daily expenses** and the line with "**Season Over**" respectively)). If Penka's **average earnings** per day are **more or equal to her daily expenses** print on the console "**Times are good. Extra money per day: {extra money}.**" where **extra money** is the difference between her **average earning** per day and her **daily expenses**. Alternatively if her **average earnings** are **less than her daily expenses** print "**We are in the red. Money needed: {money needed}.**" where **money needed** is difference between her **total expenses** (her **daily expenses** multiplied by the **number of days** in the season) and the **total amount of money** she made that season. For more details see the examples below.

Input

The input data should be read from the console. The input will contain a random number of lines. On the first input line you will receive a number representing Penka's **daily expenses**. Each of the next lines will contain information in the following format "{hours} {path} {price}"*. The last line of the input will always be "**Season Over**" – signaling the end of the input.

The input data will always be valid and in the format described. There is no need to check it explicitly.

***HINT:** Use `string.Split()` to separate {hours}, {path} and {price}.

Output

The output should consist of only one line:

- If Penka's average earnings are more or equal than her daily expenses print:
 - "Times are good. Extra money per day: {extra money}."
- Alternatively if her daily expenses are more than her average earnings print:
 - "We are in the red. Money needed: {money needed}."
- The **extra money** must be rounded to two digits after the decimal point; the **money needed** must be a whole number.

Constraints

- The **number of input lines** will be between [3...100].
- The elements in the input lines will be separated by a **single space**, there will be no leading or trailing spaces.
- The **daily expenses** will be a valid integer between [1...1000].
- The **hours** will be between [1...20].
- The **path** will be a valid string containing **only capital letters** with a length between [1...20].
- The **price** will be a valid integer between [1...1000].
- Allowed working time for your program: 0.1 seconds. Allowed memory: 16 MB.

Examples

Input	Output	Comments
250 5 MMZHQQQ 37 11 ZZHHHQ 80 Season Over	Times are good. Extra money per day: 8.50.	1st day: 5 hours -> We can traverse only the first 5 letters "MMZHQ". The string contains only 1 H thus we get only 1 herb . Money gained on day 1 => $1 * 37 = 37$ 2nd day: 11 hours -> We can traverse 11 letters, but path contains only 6 so we loop back from the beginning and we get "ZZHHHQZZHHH" as the letters we traversed we have 6 H so we get 6 herbs . Money gained on day 2 => $6 * 80 = 480$ Total money made = $37 + 480 = 517$ Average earnings per day => $517 / 2 = 258.50$ 258.50 is more than 250 so we calculate the difference $258.50 - 250 = 8.50$
Input	Output	
477 9 QQQQQQQ 999 2 HH 15 6 HKKKKKKK 5 Season Over	We are in the red. Money needed: 1396.	
Input	Output	
133 12 QQHWWHEEH 50 6 HAHA 33 Season Over	Times are good. Extra money per day: 16.50.	