**Meal Plan**

*The summer is around the corner and John wants to lose some weight. To do that he should lower his calories intake, by keeping a count of the total amount of calories intake for a day.*

Create a program that helps John to calculate his calories intake, by predefined meals.

You will be given **two lines of input.** The first line will be an **array of strings**, separated by a single space, representing **meals** that John can eat. The second line will be an **array of integers**, separated by a single space, representing the maximum **calories intake per day**.

Here is a table with the exact **meals** and **their calories**:

|  |  |
| --- | --- |
| **Meal** | **Calories** |
| salad | 350 |
| soup | 490 |
| pasta | 680 |
| steak | 790 |

Start calculating calories as follows: take the first **meal’s calories** and the **last element** in the sequence of **calorie intake for a day**. Each time John eats a meal, the **calories intake** for the current day should be **decreased** by the calories of the **given meal**, and the **current meal** should be **removed** from the collection. When the calories intake for the given day **goes to zero**, **remove** them from the collection, and continue to the **next day**. If the calories for the given day go below zero, take as **many calories as possible** from the meal, so the current daily calories intake reaches **zero**. The **rest** of the meal’s calories should be subtracted from the next day in the **calorie intake** sequence.

For more clarification, check the provided examples.

The program stops when there are no **more meals to be eaten**, or there are **no more daily calories** to be calculated.

**Input**

* On the **first line**, a sequence of **strings will be given** representing **meals**, **separated** by a single space (**" "**).
* On the **second line**, a sequence of **integers will be given,** representing the **daily calories intake**, **separated** by a single space (**" "**)**.**

**Output**

* As a result, you should print two lines for output:
* If John menage to **eat all the given meals** print:

**"John had {number of meals} meals."**

**"For the next few days, he can eat {dailyCalories1, dailyCalories2, etc.} calories."**

Separate the **daily calories intake** by comma and space **(", ")**.

* If John could **not eat every meal**:

**"John ate enough, he had {number of meals} meals."**

**"Meals left: {meal1, meal2, etc.}."**

Separate the **meals** by comma and space **(", ")**.

**Constraints**

* The **daily calories** will be in the range [1200…4000].
* No recursion is needed for solving this problem.
* The input will be always valid.
* There will be no case where **daily calories** and **meals** go to zero simultaneously.

**Examples**

|  |  |
| --- | --- |
| **Input** | **Output** |
| salad soup salad steak  2500 1800 1500 | John had 4 meals.  For the next few days, he can eat 1320, 2500 calories. |
| **Comment** | |
| We start by taking a **salad** (350 cal.) and a day with 1500 cal. We subtract the daily calories intake by the **salad's** calories:  1500 - 350 = 1150 cal.  The **salad** is removed from the sequence, and John can consume 1150 cal. more for the **current day**.  Next, we have **soup** (490 cal.), we subtract the daily calories intake by the **soup's** calories:  1150 - 490 = 660 cal.  The **soup** is removed from the sequence, and John can consume 660 cal. more for the **current day**.  Next, we have a **salad** (350 cal.), the same operation is performed:  660 - 350 = 310 cal.  The **salad** is removed from the sequence, and John can consume 310 cal. more for the **current day**.  Next, we have **steak** (790 cal.), and 310 cal. that can be consumed for the **current day**:  310 - 790 = -480 cal.  We take 310 cal. form the **steak's** total of 790 cal. and remove the current daily calory intake. Next, we check do we have another day in the collection, in this case, we do, which is 1800 cal.:  1800 - 480 = 1320 cal.  The **steak** is removed from the sequence, and John can consume 1320 cal. more for the **current day**.  There are **no more meals** in the collection, so we stop the program and print the corresponding messages. | |

|  |  |
| --- | --- |
| **Input** | **Output** |
| pasta soup salad steak pasta steak pasta  1800 1500 | John ate enough, he had 6 meals.  Meals left: pasta. |