

## Club Party

*A new club has opened in town and everyone wants to go partying. The club has many halls and people may only go there with reservations.*

You will be given **n** – an integer specifying the **halls' maximum capacity**. Then you will be given input line which will contain **English alphabet letters** and **numbers**, separated by a **single space**. The input for the line should be read **from the last inserted to the first one**. The **letters** represent the **halls** and the **numbers** – the **people in a single reservation**. Companies of people should go in the halls. The **first entered** hall is the **first which people are entering**. Every reservation takes **specific capacity**, equal to its number.

When a hall **overflows** (it **cannot contain** a given number of people due to lack of enough **free capacity**), it passes the people to the **next entered hall**. If there is **no open hall** and you receive a reservation, you should **skip it**.

If a hall overflows you must **remove it**, and print it on the console, along with all of the companies of people it **currently contains**. After you've removed that hall, **the next one** becomes the **first in the order** – people will **first be passed to it**.

passed to it.

## Input

- The first line will be halls' maximum capacity.
- The second line will contain letters and digits separated by a space.

## Output

- For output, you must print a hall, every time it overflows, after removing it.
- The format is the following: **{hall}** -> **{reservation1}**, **{reservation2}**...
- Where **{hall}** is the letter that corresponds to that hall, and the **reservations** are the numbers.

## Constraints

- The halls will only be English alphabet letters.
- Each hall's letter will always be unique.
- The integer **n** will be in the range **[0, 500]**.
- The reservations will always be valid integers in the range **[0, 500]**.

- The halls will only be English alphabet letters.
- Each hall's letter will always be unique.
- The integer **n** will be in the range **[0, 500]**.
- The reservations will always be valid integers in the range **[0, 500]**.

## Examples

Input	Output
60 1 20 b 20 20 a	a -> 20, 20, 20
Comment	
<p>"a" is the first entered hall. Then we receive the reservations <b>20</b> and <b>20</b> which are passed to "a". Then we get the hall <b>"b"</b>. Then again, we receive a reservation <b>20</b>. "a" still has enough capacity to hold the people so they enter there. Then we get the reservation <b>1</b>. "a" has capacity <b>60/60</b> – it overflows, so we pass the person to the next hall. We find <b>"b"</b> and we</p>	



pass the person to "b". "a" is then removed and printed on the console. "b" becomes the first person in the queue.



1:49:14 / 2:52:51



pass the person to **"b"**. **"a"** is then removed and printed on the console. **"b"** becomes the first hall now.

Input	Output
50 15 a 40 30 20 c 15 10 b	b -> 10, 15, 20 c -> 30

Input	Output
40 20 20 20 20 20 20 D F 15 5 M 26 38	M -> 5, 15, 20 F -> 20, 20 D -> 20, 20