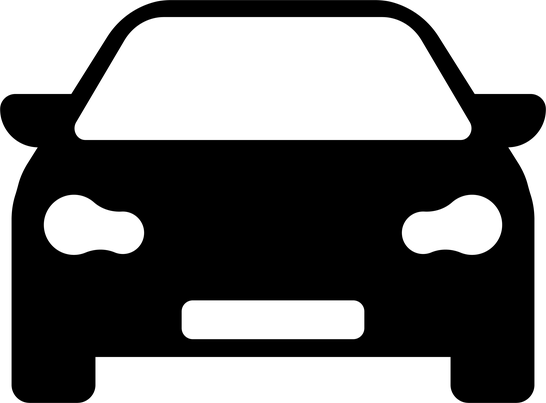
# KAT



*Your task is to optimize the performance of the KAT. By helping them reduce the queue of cars waiting for registration.*

You will be given **two sequences of integers** – one with **license plates** and one with **cars**. Two license plates are required for each car. Your goal is to find the **number of registered cars** and **how many days** they are installed. If there are cars or license plates that have not been used, you should **count** them.

The installation starts from the **first** license plate with the **last** cars. If, after the end of the day, **license plates remain**, they are added at the **end of the sequence** for a new day. If after the end of the first day there are **cars without license plates**, they are added at the **beginning of the sequence** for a new day. The procedure is performed until at least one **sequence** is over.

### Input

* On the **first line**, you will receive the integers representing the number of **license plates, separated** by **", "**.
* On the **second line**, you will receive the integers representing the number of **cars**, **separated** by "**,** ".

### Output

* Print count of registered cars and how many days were needed:
  + **“{count of registered cars} cars were registered for {count of days} days!”**
* If there are any remaining license plates:
  + **“{count of plates} license plates remain!”**
* If any cars are remaining without license plates:
  + **“{count of cars without license plates} cars remain without license plates!”**
* If all cars and license plates were used:
  + **"Good job! There is no queue in front of the KAT!"**

### Constraints

* License plates will be **even** numbers.
* All of the given numbers for license plates will be **valid** integers in the range **[4, 2147483646]**.
* All of the given numbers for cars will be **valid** integers in the range **[2, 2147483647]**.

### Examples

|  |  |
| --- | --- |
| ****Input**** | ****Output**** |
| **30, 50**  **25, 15** | **40 cars were registered for 2 days!**  **Good job! There is no queue in front of the KAT!** |
| ****Comment**** | |
| (Day 1) -> We start with the last cars (15) and the first license plates (30) -> 2 license plates for 1 car -> 2 \* 15 cars = 30 license plates.  Count of registered cars: 15  Sequences:  [50]  [25]  (Day 2) -> 25 cars and 50 license plates -> 2 license plates for 1 car -> 2 \* 25 cars = 50 license plates. **25** cars were registered.  Count of registered cars: 40. | |

|  |  |
| --- | --- |
| ****Input**** | ****Output**** |
| **20, 100, 40, 68**  **33, 50** | **83 cars were registered for 4 days!**  **62 license plates remain!** |
| ****Comment**** | |
| (Day 1) -> We start with the last cars (50) and the first license plates (20) -> 20 license plates for **10** cars -> 10 cars were registered -> The **rest (40)** cars were moved at **beginning** of the sequence.  Count of registered cars: **10**  Sequances:  [100, 40, 68]  [40, 33]  (Day 2) -> 33 cars and 100 license plates -> 66 license plates for 33 cars -> Add **33** cars to count of registered cars -> The **rest (34)** license plates are moved to the **end** of sequence.  Count of registered cars: **43**  Sequences:  [40, 68, **34**]  [40]  (Day 3) -> 40 cars and 40 plates -> 40 plates for 20 cars -> Add 20 cars to count of registered cars -> The **rest (20)** cars were moved in **beginning** of sequance.  Count of registered cars: **63**  Sequences:  [68, 34]  [**20**]  (Day 4) -> 20 cars and 68 plates -> 40 plates for 20 cars -> Add **20** cars to count of registered cars -> The **rest (28)** plates are moved to the **end** of sequence.  Count of registered cars: **83**  The sequence of license plates: [34, 28] | |