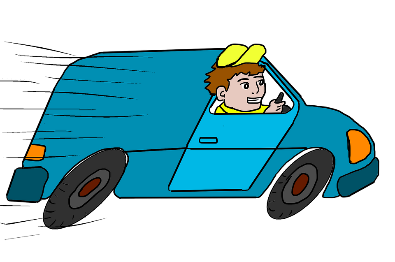
1. **Rapid Courier**



*You have created your own delivery service called "Rapid Courier". You want to analyze how well your couriers are doing in delivering all the packages.*

On the **first line**, you will be given a **sequence of integers** representing **packages** that need to be delivered, with the values of their **weight**. On the **next line**, you will be given **another sequence of integers** representing **couriers** with their loading **capacities**.

Until **there are any packages** to deliver **and any couriers** available, the program continues running.

Track the **total weight of packages** **delivered** by your couriers. You must **compare** the [**last package**](#LastPackage) to the [**first courier**](#FirstCourier).

* If the courier can deliver the package (the **capacity of the courier is equal to or greater than the weight of the package**), he does the delivery:
  + If the **capacity of the courier is greater than the weight of the package**, **reduce** the **courier's capacity** by **twice the package's weight**:
    - If the **new courier's capacity** is **positive**, the **courier moves at the back** of the sequence with the **updated capacity**.
    - If the new capacity is **zero or negative**, **remove** the **courier.**
  + **Аdd the weight of the package** to the **total delivered weight.**
  + **Remove the package** from the sequence.
* If the courier cannot deliver the package (the **capacity of the courier is less than the weight of the package**), **subtract the courier's capacity** from the **package's weight**
  + Return the **remaining weight** to the **sequence,** and **remove the courier**.
  + **Add** the delivered **portion of the package's weight** to the **total delivered weight**.

### Input / Constraints

* On the **first line**, you will receive **integers** representing the **weight of the packages** to be delivered, separated by a **single space**.
* On the **second line**, you will receive **integers** representing the **capacities of the couriers**, separated by a **single space**.
* All given numbers will be valid **integers** in the range **[0; 100]**.

### Output

The output of your program should be printed on the Console, on separate lines, formatted according to the following rules:

* At the end of the program, print the weight of the packages delivered:
  + "**Total weight: {total\_weight} kg"**
* If all of the **packages are delivered** and there are **no couriers left**:
  + "**Congratulations, all packages were delivered successfully by the couriers today.**"
* If **there are packages** left but **no more couriers available**:
* **"Unfortunately, there are no more available couriers to deliver the following packages: {package1}, {package2}, (…),{packagen}"**
* If **there are couriers** left but there are **no more packages to deliver**:
* **"Couriers are still on duty: {****couriers1}, {couriers2}, (…),{couriersn} but there are no more packages to deliver."**

### Examples

|  |  |  |
| --- | --- | --- |
| ****Input**** | ****Output**** | ****Comment**** |
| **2 4 6** **8**  **8 6 4 2** | **Total weight: 20 kg**  **Congratulations, all packages were delivered successfully by the couriers today.** | The first pair consists of the **last package** with a weight of 8 and the **first courier's** capacity with a value of 8. Since the two **values are equal**, the courier delivers the package successfully and **both values are removed** from the collections.  8 kilograms of packages are delivered.  Now, the sequences are as follows:  2 4 6  6 4 2  We repeat the same operations until all packages are delivered and no courier available is left.  2 4 6  6 4 2  8 + 6 = 14 kg of packages are delivered  2 4  4 2  8 + 6 + 4 = 18 kg of packages are delivered  2  2  8 + 6 + 4 + 2 = 20 kg of packages are delivered  Finally, since there are **no more elements in both sequences**, the program ends. The correct output is printed on the Console. |
| **2 4 6** 8  18 **6 4 2 6** | **Couriers are still on duty: 6, 2 but there are no more packages to deliver.** | The first pair consists of the **last package** with a weight of 8 and the **first courier's** capacity with a value of 18. Since the **value of the courier's capacity is greater than the value of the package's weight**, the courier delivers the package successfully, but the capcity is decreased by twice the package's weight ->  to 2 (18 – 8 \* 2), and is being **moved at the back** of the sequence. The **weight value is removed** from the collection.  8 kilograms of packages are delivered.  Now, the sequences are as follows:  2 4 6  6 4 2 6 2  We repeat the same operations until all packages are delivered.  2 4 6  6 4 2  8 + 6 = 14 kg of packages are delivered  2 4  4 2 6 2  8 + 6 + 4 = 18 kg of packages are delivered  2  2 6 2  8 + 6 + 4 + 2 = 20 kg of packages are delivered  Finally, since there are **no more packeges to be delivered**, and the program ends. The correct output is printed on the Console. |
| **13 11 5**  **5** 11 | **Total weight: 16 kg**  **Unfortunately, there are no more available couriers to deliver the following packages: 13** | The first pair consists of the **last package** with a weight of **5** and the **first courier's** capacity with a value of **5**. Since the two **values are equal**, the courier delivers the package successfully and **both values are removed** from the collections.  **5** kilograms of packages are delivered.  Now, the sequences are as follows:  13 11  11  We repeat the same operations until all packages are delivered and no courier available is left.  13 **11**  **5 + 11** = 16 kg of packages are delivered  Now, the sequences are as follows:  13  [the second sequence is empty]  Finally, since there are **no more elements in the courier sequence**, the program ends. The correct output is printed on the Console. |
| **7 14**  **2 2 2 1 7** | **Total weight: 14 kg**  **Unfortunately, there are no more available couriers to deliver the following packages: 7** | The first pair consists of the **last package** with a weight of **14** and the **first courier's** capacity with a value of **2**. Since the **value of the package weight is greater than the value of the courier loading capacity**, we take the value of the package, decrease it by the value of the courier put it back on the top of the sequence. We remove the value of the courier from its sequence.  **2** kilograms of packages are delivered.  Now, the sequences are as follows:  7 **12**  2 2 1 7  **2** kilograms of packages are delivered.  We repeat the same operations until all packages are delivered and no courier available is left.  7 **10**  2 1 7  **4** kilograms of packages are delivered.  7 **8**  1 7  **6** kilograms of packages are delivered.  7 **7**  7  **7** kilograms of packages are delivered.  Now, the sequences are as follows:  7  [the second sequence is empty]  **14** kilograms of packages are delivered.  Finally, since there are no more elements in the courier sequence, the program ends. The correct output is printed on the Console. |