# Chicken Snack



*Henry loves street food and he can't wait for the weekend to come because that's when he goes for a walk and enjoys his favorite food. His mother knows this and decides to surprise him by putting some change in his pants pocket.*

On the first line, you will be given **a sequence of integers** representingthe **amount of money** in Henry's **pocket**. In the next line, you will be given another **sequence of integers** representingthe **prices** of **foods** that Henry can buy.

Henry has gone to his favorite fast food place, fumbles in his pocket and pulls out some change.

You have to start with the **last element** from the **amount of money sequence** and compare it with the **first element from the prices sequence**.

* If their **values** are **equal**, Henry buys the food. After that, you should **remove both** of them from their **sequences**.
* If the **amount of money** is **bigger** than **the** **price,** he buys the food again, taking change (the **difference** between the **amount of money** and the **price)** and putting it in his pocket. You should **calculate the difference** between the values, and keep it.
  + **Remove** **the current amount of money** from its sequence and **increase** **the next amount of money** value in the sequence by the **resulting difference you have calculated.**
  + **Remove the price** from the prices sequence**.**
* If the **amount of money** is **lower** than the **price** remove **both** of them from their **sequences.**

You need to **stop** **comparing** when you have **no more amounts of money** or **prices**.

### Input / Constraints

* On the **first line**, you will receive the integers, representing the **amount of money size**, **separated** by a **single space**.
* On the **second line,** you will receive the integers, representing the **price size**, **separated** by a **single space**.
* All given numbers will be valid integers in the range **[1, 20]**.

### Output

* The output of your program **should be a single line of text**, formatted according to the following rules:
* If Henry managed to **eat four or more foods** print the following:
  + "**Gluttony of the day! Henry ate {food count} foods."**
* If Henry has eaten **some of the foods** print the following:
* **"Henry ate: {food count} foods."**
* in case Henry has eaten **only one** **food**, print: **"Henry ate: {food count} food."**
* If Henry has **not eaten** any food:
  + "**Henry remained hungry. He will try next weekend again.**"

### Examples

|  |  |  |
| --- | --- | --- |
| ****Input**** | ****Output**** | ****Comment**** |
| **9 5 8 18**  **18 12 10 5** | **Henry ate: 2 foods.** | The first pair is the **first price** with a value of **18** and the **last amount of money** with a value of **18**, their **values are equal**, so Henry can buy the food **and we remove both values** from the **sequences**. Next, we have **12** and **8** **(12 > 8)** Henry can not buy food. You should **remove both of them**, too. The next values are **10** and **5**, and you do the same - only remove them. Finally, we have **money** 9 and **price** 5 **(9 > 5)** Henry buys the food.  The **price** sequence is **empty** so the program ends. |
| **18 10 8 9**  **5 10 12 18** | **Gluttony of the day! Henry ate 4 foods.** | **Here we have **price** **5** and **money 9**. **9 > 5** Henry buys** current food and takes a change in the amount of 4 which he transfers to the next amount of money – 8 + 4 = 12, finally we remove both values. Then we take 12 and match it with 10. 12 > 10 -> he buys second food, the change is 2. 10 + 2 = 12. 12 == 12 -> third food. 18 == 18 -> fourth food. |
| **1 1 4 5 9 9 9**  **9 15 18 13 10** | **Henry ate: 1 food.** |  |
| **1 1 4 5 6 2 3 2**  **17 8 18 19 20** | **Henry remained hungry. He will try next weekend again.** |  |