# Bombs

*Ezio still is learning to make bombs. With their help, he will save civilization. We should help Ezio to make his perfect bombs.*

You will be given **a two sequence of integers, representing bomb effects** and **bomb casings**.

You need to start from the **first bomb effect** and try to mix it with the **last bomb casing**. If the **sum** of their values is **equal** to **any of the materials in the table below** – **create the bomb** **corresponding** to the **value** and **remove** the **both** bomb materials. Otherwise, just **decrease** the value of the **bomb casing** **by 5**. You need to **stop** combining when you have **no more bomb effects** or **bomb casings,** or you successfully filled the bomb pouch.

**Bombs:**

* **Datura Bombs: 40**
* **Cherry Bombs: 60**
* **Smoke Decoy Bombs: 120**

To fill the bomb pouch, Ezio needs **three of each** of the **bomb types**.

### Input

* On the **first line**, you will receive the integers representing the **bomb effects**, **separated** by **", "**.
* On the **second line**, you will receive the integers representing the **bomb casing**, **separated** by ", ".

### Output

* On the **first** line of output – print one of these rows according whether Ezio succeeded fulfill the bomb pouch:
  + "**Bene! You have successfully filled the bomb pouch!**"
  + "You don't have enough materials to fill the bomb pouch."
* On the **second** line - print all bomb effects left:
  + If there are no bomb effects: "**Bomb Effects: empty**"
  + If there are effects: "**Bomb Effects: {bombEffect1}, {bombEffect2},** **(…)**"
* On the **third** line - print all bomb casings left:
  + If there are no bomb casings: "**Bomb** **Casings: empty**"
  + If there are casings: "**Bomb Casings: {bombCasing1}, {bombCasing2},** **(…)"**
* Then**,** you need to print **all** created bombs and the **count you have of them**, ordered **alphabetically**:
  + **"Cherry Bombs: {count}"**
  + **"Datura Bombs: {count}"**
  + **"Smoke Decoy Bombs: {count}"**

### Constraints

* All of the given numbers will be valid integers in the range **[0, 120]**.
* Don't have situation with negative material.

### Examples

|  |  |
| --- | --- |
| ****Input**** | ****Output**** |
| **5, 25, 50, 115**  **5, 15, 25, 35** | **You don't have enough materials to fill the bomb pouch.**  **Bomb Effects: empty**  **Bomb Casings: empty**  **Cherry Bombs: 1**  **Datura Bombs: 2**  **Smoke Decoy Bombs: 1** |
| ****Comment**** | |
| 1) 5 + 35 = 40 -> Datura Bomb. Remove both.  2) 25 + 25 = 50 -> can't create bomb. Bomb casing should be decreased with 5 -> 20  3) 25 + 20 = 45 -> can't create bomb. Bomb casing should be decreased with 5 -> 15  4) 25 + 15 = 40 -> Datura Bomb. Remove both  … | |

|  |  |
| --- | --- |
| ****Input**** | ****Output**** |
| **30, 40, 5, 55, 50, 100, 110, 35, 40, 35, 100, 80**  **20, 25, 20, 5, 20, 20, 70, 5, 35, 0, 10** | **Bene! You have successfully filled the bomb pouch!**  **Bomb Effects: 100, 80**  **Bomb Casings: 20**  **Cherry Bombs: 3**  **Datura Bombs: 4**  **Smoke Decoy Bombs: 3** |
| ****Comment**** | |
| **…**  After creating a bomb with bomb effect 35 and bomb casing 25, have created 3 Cherry bombs, 4 Datura bombs, and 3 Smoke Decoy bombs. From all of the bomb types have 3 bombs and the program ends. | |

*"Nothing is true; everything is permitted"*