Problem 1 - Aladdin's Gifts

Aladdin, rich and powerful with the help of the Genie, is now preparing to marry the princess Jasmine. He asks Genie for help to prepare the wedding presents.

First, you will receive a sequence of **integers** representing the materials for every wedding present. After that, you will be given another sequence of **integers** – Genie magic level for every aim to make a gift.

Your task is to **mix materials** with **magic levels** so you can make presents, listed in the table below.

Gift	Magic needed
Gemstone	100 to 199
Porcelain Sculpture	200 to 299
Gold	300 to 399
Diamond Jewellery	400 to 499

To make a present, you should take the **last integer of materials** and **sum** it with the **first magic level value**. If the **result is between or equal to** the numbers described in the table above, you **make the corresponding gift** and **remove both** materials and magic value. **Otherwise**:

- If the **product of the operation** is **under 100**:
  - And if it is an **even** number, **double the materials**, and **triple the magic**, then **sum** it again.
  - And if it is an **odd** number, **double the sum** of the materials and the magic level. Then, **check again** if it is between or equal to any of the numbers in the table above.
- If the **product of the operation** is **more than 499**, **divide** the sum of the material and the magic level **by 2**. Then, **check again** if it is between or equal to any of the numbers in the table above.
- If, however, **the result is not between or equal to any of the numbers in the table above after performing the calculation**, **remove both** the materials and the magic level.

Stop crafting gifts when you are out of materials or magic level.

You have succeeded in crafting the presents when you've crafted either one of the pairs - a gemstone and a sculpture or gold and jewellery.

Input

- The **first** line of input will represent the values of **materials** - **integers**, separated by a **single space**
- On the **second** line, you will be given the **magic levels** - **integers**, separated by a **single space**

Output

- On the **first line** - print **whether you have succeeded** in crafting the presents:
  - **`"The wedding presents are made!"`**

- **"Aladdin does not have enough wedding presents."**

- On the **next two lines** print the **materials** and **magic** that are **left**, **if there are any**, **otherwise skip the line:**
  - "**Materials left: {material1}, {material2}, …**"
  - "**Magic left: {magicValue1}, {magicValue2}, …**

- On the **next lines**, print the **gifts alphabetically** that the Genie has crafted **at least once**:

```
"{present1}: {amount}
{present2}: {amount}
…
{presentN}: {amount}"
```

Constraints

- All the materials values will be **integers** in the range **[1, 1000]**
- Magic level values will be **integers** in the range **[1, 1000]**

Examples

Input	Output	Comment
105 20 30 25 120 60 11 400 10 1	The wedding presents are made! Magic left: 10, 1 Gemstone: 1 Porcelain Sculpture: 2	First, we have 25 + 120 = 145, which is the needed product for a gemstone. Remove both. 30 + 60 = 90 (under 100 and even) => 30 * 2 + 60 * 3 = 240 which is the needed product for a porcelain sculpture. Remove both. 20 + 11 = 31 (under 100 and odd) => 31 * 2 = 62 which is under 100 again so we remove both. 105 + 400 = 505 (more than 450) => 505 / 2 = 252.5 which is the needed product for a diamond porcelain sculpture. Remove both. We do not have any material left. The program ends. Print the desired text.
30 5 21 6 0 91 15 9 5 15 8	Aladdin does not have enough wedding presents. Materials left: 30 Gemstone: 1	
200 5 15 32 20 10 5	Aladdin does not have enough wedding presents. Magic left: 15, 32, 20, 10, 5 Porcelain Sculpture: 1	

Problem 2 - Ball in the Bucket

You are at the funfair to play different games and test your skills. Now you are playing ball in the bucket game.

You will be given a **matrix with 6 rows and 6 columns** representing the board. On the board, there will be points (integers) and **buckets** marked with the **letter "B"**. Rules of the game:

- You can **throw** a ball only **3 times**.
- When you **hit a bucket** (position marked with `'B'`), you score the **sum of the points** in the **same column**.
- You can hit a bucket **only once**. If you hit the **same bucket again**, you do **not** score any points.
- If you hit **outside a bucket** (hit a number on the board) or **outside the board**, you do **not** score any points.

After the board state, you are going to receive the information for every throw on a **separate line**. The **coordinates'** information of **a hit** will be in the format: **`"({row}, {column})"`**.

Depending on **how many points you have collected**, you win one of the following:

`Football`	**100 to 199 points**
`Teddy Bear`	**200 to 299 points**
`Lego Construction Set`	**300 and more points**

Your job is to keep track of the scored points and to check if you won a prize.

For more clarifications, see the examples below.

Input

- **6 lines** – **matrix** representing the board (each position **separated by a single space**)
- On the next 3 **lines** - the **coordinates** of the throw in the format: **`"({row}, {column})"`**

Output

- On the first line:
  - If you won a prize, print:
    `"Good job! You scored {points} points, and you've won {prize}."`
  - If you did not win any prize, print the points you need to get at least the first prize:
    `"Sorry! You need {points} points more to win a prize."`

Constraints

- All of the given **points** will be **integers** in the range **[1, 30]**
- All the given **indexes** will be **integers** in the range **[0, 30]**
- There **always** will be **exactly 6 buckets - 1 on each column**

Examples

Input	Output
10 30 B 4 20 24 7 8 27 23 11 19 13 3 14 B 17 B 12 5 21 22 9 6 B 26 1 28 29 2 25 B 16 15 B 18 (1, 1) (20, 15) (4, 0)	Sorry! You need 33 points more to win a prize.
B 30 14 23 20 24 29 8 27 18 11 19 13 3 B B 17 6 28 5 21 22 9 B 10 B 26 12 B 2 25 1 16 15 7 4 (0, 0) (2, 2) (2, 3)	Good job! You scored 299 points, and you've won Teddy Bear.

Problem 3 - Shopping List

Write a function called **`shopping_list`** which will receive an integer **number** representing **the budget in leva** and **a different number of keywords.** Each **key** represents the **product** (string), and each **value** will be a **tuple** with **the product's price** (integer or float number) at the **first** position and **quantity** (integer) at the **second** position.

Your job is to **return which products you bought** with the given budget. You only buy a product if you can buy **all of its quantity**.

You could **only start** shopping if you have **at least 100 leva** budget. Otherwise, you should **stop the program** and **return `"You do not have enough budget."`**.

Also, you are **carrying a basket** that cannot hold **more than 5 types of products**. You should **stop buying** products**:**

- if you **reach** the allowed **amount of types of products** (no matter their quantity)
- if you **did not reach** the allowed amount, but you do **not have more products** on the shopping list

You should **always buy products successively**!

For each product (all its quantity) you succeeded to buy, return a string on a new line in the following format:

`"You bought {product_name} for {total_price} leva."`

Note: Submit only the function in the judge system

Input

- There will be **no input,** and just parameters passed to your function

Output

- The function should **return strings on separate lines** containing the **bought products** and **their price** in the format described above.
- The **total price** should be formatted to the **second decimal point**.

Examples

Test Code	Output

`print(shopping_list(100,`
`microwave=(70, 2),`
`skirts=(15, 4),`
`coffee=(1.50, 10),`
`))`
	`You bought skirts for 60.00 leva.`
	`You bought coffee for 15.00 leva.`

`print(shopping_list(20,`
`jeans=(19.99, 1),`
`))`
	`You do not have enough budget.`

`print(shopping_list(104,`
`cola=(1.20, 2),`
`candies=(0.25, 15),`
`bread=(1.80, 1),`
`pie=(10.50, 5),`
`tomatoes=(4.20, 1),`
`milk=(2.50, 2),`
	`You bought cola for 2.40 leva.`
	`You bought candies for 3.75 leva.`
	`You bought bread for 1.80 leva.`
	`You bought pie for 52.50 leva.`
	`You bought tomatoes for 4.20 leva.`

juice=(2, 3), eggs=(3, 1),))	