# Python Advanced: Exam Preparation

# Fireworks Show

**Link to Judge:** <https://judge.softuni.org/Contests/Practice/Index/2812#0>

*Maria wants to make a firework show for the wedding of her best friend. We should help her to make the perfect firework show.*

First, you will be given **a sequence of integers representing firework effects**. Afterwards you will be given another **sequence of integers representing explosive power**.

You need to start from the **first firework effect** and try to mix it with the **last explosive power**. If the **sum** of their values is:

* **divisible by 3, but it is not divisible by 5** – **create Palm firework** and **remove** **both** materials
* **divisible by 5, but it is not divisible by 3** – **create Willow firework** and **remove** **both** materials
* **divisible by both 3 and 5** – **create Crossette firework** and **remove** **both** materials

Otherwise, **decrease** the value of the **firework effect by 1** and **move** it at the **end** of the sequence. Then, try to mix the same **explosive power** with the next **firework effect.**

If any value is **equal to or below 0,** you should **remove it** from the sequence **before trying to mix it with the other.**

When you have **successfully prepared enough fireworks for the show** or you have **no more firework punches** or **explosive power,** you need tostopmixing**.**

To make the perfect firework show, Maria needs **3 of each** of the **firework types.**

### Input

* On the **first line**, you will receive the integers representing the **firework effects**, **separated** by ", ".
* On the **second line**, you will receive the integers representing the **explosive power**, **separated** by ", ".

### Output

* On the **first** line, print:
  + if Maria **successfully prepared** the firework show: "**Congrats! You made the perfect firework show!**"
  + if Maria **failed** to prepare it: "Sorry. You can't make the perfect firework show."
* On the **second** line, print all firework effects left **if there are any**:
* "**Firework Effects left: {effect1}, {effect2},** **(…)**"
* On the **third** line, print all explosive fillings left **if there are any**:
* " **Explosive Power left: {filling1}, {filling2},** **(…)"**
* Then**,** you need to print **all** fireworks and the **amount you have of them**:
  + **"Palm Fireworks: {count}"**
  + **"Willow Fireworks: {count}"**
  + **"Crossette Fireworks: {count}"**

### Constraints

* All the given numbers will be integers in the range **[-100, 100]**.
* There will be no cases with empty sequences.

### Examples

|  |  |
| --- | --- |
| ****Input**** | ****Output**** |
| **5, 6, 4, 16, 11, 5, 30, 2, 3, 27**  **1, 13, 5, 3, -7, 32, 19, 3, 5, 7, 22** | **Congrats! You made the perfect firework show!**  **Palm Fireworks: 4**  **Willow Fireworks: 3**  **Crossette Fireworks: 3** |
| ****Comment**** | |
| 1) 5 + 22 = 27 is devisible by 3 -> Palm Firework. Remove both.  2) 6 + 7 = 13 -> can't create firework. Firework effect should be decreased with 1 -> 5 and moved at the end  3) 4 + 7= 11 -> can't create firework. Firework effect should be decreased with 1 -> 3 and moved at the end  3) 16 + 7 = 23 -> can't create firework. Firework effect should be decreased with 1 -> 15 and moved at the end  4) 11 + 7 = 18 is devisible by 3 -> Palm Firework. Remove both.  5) 5 + 5 = 10 is devisible by 5 -> Willow Firework. Remove both.  6) 30 + 3 = 33 is devisible by 3 -> Palm Firework. Remove both.  7) 2 + 19 = 21 is devisible by 3 -> Palm Firework. Remove both.  8) 3 + 32 = 35 is devisible by 5 -> Willow Firework. Remove both.  9) (-7) is negative, so we remove it before mixing.  10) 27 + 3 = 30 is devisible by 5 and 3 -> Crossette Firework. Remove both.  11) 5 + 5 = 10 is devisible by 5 -> Willow Firework. Remove both.  12) 3 + 13 = 16 -> can't create firework. Firework effect should be decreased with 1 -> 2 and moved at the end  13) 15 + 13 = 28 -> can't create firework. Firework effect should be decreased with 1 -> 14 and moved at the end  14) 2 + 13 = 15 is devisible by 5 and 3 -> Crossette Firework. Remove both.  15) 1 + 14 = 15 is devisible by 5 and 3 -> Crossette Firework. Remove both.  We have enough fireworks to make a firework show. | |

|  |  |
| --- | --- |
| ****Input**** | ****Output**** |
| **-15, -8, 0, -16, 0, -22**  **10, 5** | **Sorry. You can't make the perfect firework show.**  **Explosive Power left: 10, 5**  **Palm Fireworks: 0**  **Willow Fireworks: 0**  **Crossette Fireworks: 0** |
| ****Comment**** | |
| After removing all the invalid integers, the firework effects's sequence is empty and the program ends. | |

# Darts

**Link to Judge:** <https://judge.softuni.org/Contests/Practice/Index/2828#1>

*Two players bare-handedly throw small sharp-pointed missiles known as darts at a round target known as a dartboard. Who is going to win this game?*

You will be given a **matrix with 7 rows and 7 columns** representing the dartboard. For example:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 24 | D | D | D | D | D | 8 |
| 23 | D | T | T | T | D | 9 |
| 22 | D | T | B | T | D | 10 |
| 21 | D | T | T | T | D | 11 |
| 20 | D | D | D | D | D | 12 |
| 19 | 18 | 17 | 16 | 15 | 14 | 13 |

Each of the **two players** starts with a **score of 501** and they **take turns** to throw **a dart – one throw for each player**. The score for each turn is **deducted** from the **player’s total** score. The **first** player who reduces their **score to zero or less** **wins** the game.

You are going to receive the information for every throw on a **separate line**. The **coordinate** information of a hit will be in the format: **"({row}, {column})".**

* If a player **hits outside the dartboard**, he does **not** **score** any points.
* If a player **hits a number**, it is **deducted from his total**.
* If a player hits a **"D"** the **sum** of the **4 corresponding** **numbers** per **column** and **row** is **doubled** and **then deducted from his total**.
* If a player hits a **"T"** the **sum** of the **4 corresponding** **numbers** per **column** and **row** is **tripled** and **then deducted from his total**.
* **"B"** is the **bullseye**. If a player hits it, he **wins the game,** and the **program ends**.

For example, if Peter hits position with coordinates (2, 1), he wins (23 + 2 + 9 + 18) \* 2 = 104 points and they are deducted from his total.

Y**our job is to find who won the game and with how many turns.**

### Input

* The **name** of the **first** playerand the **name** of the **second** player,separated by **", "**
* **7 lines** – the dartboard (**separated by single space**)
* On the next **lines** - the **coordinates** in the format: **"({row}, {column})"**

### Output

* You should print only one line containing the winner and his count of throws:

**"{name} won the game with {count\_turns}** **throws!"**

### Constrains

* There will always be **exactly 7 lines**
* There will **always** be a winner
* The points will be in range **[1, 24]**
* The coordinates will be in range **[0, 100]**

### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comment** |
| Ivan, Peter  12 21 18 4 20 7 11  9 D D D D D 10  15 D T T T D 3  2 D T B T D 19  17 D T T T D 6  22 D D D D D 14  5 8 23 13 16 1 24  (3, 3) | Ivan won the game with 1 throws! | Ivan hits the Bullseye and wins the game. The program ends. |
| George, Hristo  17 8 21 6 13 3 24  16 D D D D D 14  7 D T T T D 15  23 D T B T D 2  9 D T T T D 22  19 D D D D D 10  12 18 4 20 5 11 1  (1, 0)  (2, 3)  (0, 0)  (4, 2)  (5, 1)  (3, 1)  (0, 0)  (2, 3) | Hristo won the game with 4 throws! | George 1st throw: 501 – 16 = 485  Hristo 1st throw: 501 – 144 = 357  George 2nd throw: 485 – 17 = 468  Hristo 2nd throw: 357 – 168 = 189  George 3rd throw: 468 – 110 = 358  Hristo 3rd throw: 189 – 102 = 87  George 4th throw: 358 – 17 = 341  Hristo 4th throw: 87 – 144 = -57  Hristo wins the game. The program ends. |

# Cupcake Shop

**Link to Judge:** <https://judge.softuni.org/Contests/Practice/Index/2812#2>

*Maria is opening a cupcake shop. Help her manage her inventory to improve stock availability.*

Write a function called **stock\_availability** which receives:

* an inventory **list** of boxes with different kinds of cupcake flavours
* **"delivery"** or **"sell"** as **second** parameter
* there **might** or **might not** be any **other parameters – numbers or strings** at the end

In case of **"delivery"** to the shop was delivered new boxes with diferent kinds of **cupcakes**:

* You should **add** the **boxes** at the **end** of the inventory list
* There will be always at least one box delivered

In case of **"sell"** Maria has a client and she is selling different **boxes** with cupcakes:

* If there is a **number** as **another parameter**, it means that Maria has sold that many **boxes with cupcakes** and you should **remove them** from the **beginning** of the inventory list
* If there is/are string/s as **another parameter/s**, it means that Maria has sold **ALL cupcake boxes of the ordered flavour/s.** Beware that **not everything the buyer has ordered might be in stock**, so you should check if the order is valid.
* If there are **no other parameters**, it means that Maria has sold only **the first box** of cupcakes and you should **remove it** of the inventory list

For more clarifications, see the examples below.

### Input

* There will be **no input**
* **Parameters** will be passed to your function

### Output

* The function should **return a new inventory list**
* All **commands** will be **valid**

### Examples

|  |
| --- |
| **Test Code** |
| print(stock\_availability(["choco", "vanilla", "banana"], "delivery", "caramel", "berry"))  print(stock\_availability(["chocolate", "vanilla", "banana"], "delivery", "cookie","banana"))  print(stock\_availability(["chocolate", "vanilla", "banana"], "sell"))  print(stock\_availability(["chocolate", "vanilla", "banana"], "sell", 3))  print(stock\_availability(["chocolate", "chocolate", "banana"], "sell", "chocolate"))  print(stock\_availability(["cookie", "chocolate", "banana"], "sell", "chocolate"))  print(stock\_availability(["chocolate", "vanilla", "banana"], "sell", "cookie")) |
| **Output** |
| ['choco', 'vanilla', 'banana', 'caramel', 'berry']  ['chocolate', 'vanilla', 'banana', 'cookie', 'banana']  ['vanilla', 'banana']  []  ['banana']  ['cookie', 'banana']  ['chocolate', 'vanilla', 'banana'] |