# Problem 1. Easter Cozonacs

*Since it’s Easter you have decided to make some cozonacs and exchange them for eggs.*

Create a program that **calculates** how much **cozonacs** you can make with the **budget** you **have**. **First**, you will **receive** your **budget**. Then, you will **receive** the **price** for **1 kg flour**. Here is the **recipe** for **one** cozonac:

|  |  |
| --- | --- |
| **Eggs** | **1 pack** |
| **Flour** | **1 kg** |
| **Milk** | **0.250 l** |

The **price for 1 pack of eggs** is **75%** of the **price** **for 1 kg flour**. The **price** for **1l** **milk** is **25%** **more** than price for **1 kg flour**. Notice, that you need **0.250l milk** for **one** cozonac and the calculated price is for **1l**.

**Start** cooking the cozonacs and **keep making** them until you have **enough budget**. Keep in mind that:

* For **every** cozonac that you make, you will receive **3 colored eggs**.
* For **every** **3rd** cozonac that you make, you will lose some of your **colored** eggs **after** you have **received** the usual **3 colored eggs** for your cozonac. The count of eggs you will lose is calculated when you **subtract** **2** from your **current** **count** of **cozonacs** – **({currentCozonacsCount} – 2)**

In the end, print the cozonacs you made, the eggs you have gathered and the money you have **left**, **formatted** to the **2nd decimal place**, in the following format:

**"You made {countOfCozonacs} cozonacs! Now you have {coloredEggs} eggs and {moneyLeft}BGN left."**

## Input / Constraints

* On the **1st line** you will receive the budget – a **real number** in the range [0.0…100000.0]
* On the **2nd line** you will receive the price for **1 kg floor** – a **real number** in the range [0.0…100000.0]
* The input will always be in the right format.
* You will **always** have a **remaining** **budget**.
* There will **not** be a case in which the **eggs** become a **negative** **count**.

## Output

* In the end print the **count** of **cozonacs** you have made, the colored **eggs** you have gathered and the **money** **formatted** to **the 2nd** decimal place in the format described above.

## Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 20.50  1.25 | You made 7 cozonacs! Now you have 16 eggs and 2.45BGN left. |
| **Comments** | |
| We start by calculating the price for a **pack of eggs**, which is **75%** of the price for **1 kg** floor, which in this case is **1.25**. The pack of eggs price is **0.9375**. The price for **1l milk** is **25%** more than the price for **1kg** floor and in this case it is – **1.5625**, but we need the price for 0.250ml, which is - **0.390625**. The total price for one cozonac is:  **1.25** + **0.9375 + 0.390625** = **2.578125**.  And we start subtracting the **price** for a **single** cozonac **from the budget**, and **for every cozonac** we receive **3** eggs. So after the first **subtraction** we will have **17.921875** budget, **1** cozonac and **3** eggs.After the second **- 15.34375** budget, **6** eggs, and on the **third** - **12.765625 budget** and **9 eggs** and since it’s the **third**, we need to **subtract** the **lost eggs**, which will be 3 – 2 = **1**, so we subtract 1 from 9 and our **eggs** become **8**. We continue **subtracting** money from the **budget** until the money **aren't enough** for us to make a cozonac. In the end we have 2.45BGN left. | |
|  | |
| 15.75  1.4 | You made 5 cozonacs! Now you have 14 eggs and 1.31BGN left. |

# Problem 2. Easter Gifts

*As a good friend, you decide to buy presents for your friends.*

Create a program that helps you plan the gifts for your friends and family. First, you are going to **receive** **the gifts** you plan on buying оn a **single line,** **separated** **by** **space**, in the following **format**:

**"{gift1} {gift2} {gift3}… {giftn}"**

Then you will start receiving **commands** until you read the "**No Money**" message. There are **three** possible commands:

* **"OutOfStock {gift}"**
  + Find **the gifts** with **this name** in your collection, **if there are any**, and change their values to "**None**".
* "**Required {gift} {index}**"
  + **Replace** the value of the **current gift** on the given index **with this** **gift,** if the **index** is **valid**.
* "**JustInCase {gift}"**
  + **Replace** the value of your **last** gift **with this** **one**.

In the end, print the **gifts** on a **single** **line**, **except the ones** with value **"None",** separated by a **single** **space** in the following format:

**"{gift1} {gift2} {gift3}… {giftn}"**

## Input / Constraints

* On the **1st line** you are going to receive the **names of the gifts**, separated by a single space.
* On the next **lines**, until the **"No Money"** command is received, you will be receiving commands.
* The **input** will **always** be **valid**.

## Output

* Print the gifts in the **format** **described** **above**.

## Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Eggs StuffedAnimal Cozonac Sweets EasterBunny Eggs Clothes  OutOfStock Eggs  Required Spoon 2  JustInCase ChocolateEgg  No Money | StuffedAnimal Spoon Sweets EasterBunny ChocolateEgg |
| **Comments** | |
| First, we receive the command "**OutOfStock**" and we need to replace the values of "**Eggs**" with "**None**". After this command the list should look like this:  **None StuffedAnimal Cozonac Sweets EasterBunny None Clothes**.  Afterwards, we receive the "**Required**" command and we need to replace the value on the 2nd index of our list with the value "**Spoon**". The list should look like this:  **None StuffedAnimal Spoon Sweets EasterBunny None Clothes**  After, we receive the "**JustInCase**" command, which means we need to replace the last value in our list with "**ChocolateEggs**". The list should look like this:  **None StuffedAnimal Spoon Sweets EasterBunny None ChocolateEggs**  In the end, we print all of the gifts, except the ones with values **"None"**. This is the result list:  **StuffedAnimal Spoon Sweets EasterBunny ChocolateEggs** | |
|  | |
| Sweets Cozonac Clothes Flowers Wine Clothes Eggs Clothes  Required Paper 8  OutOfStock Clothes  Required Chocolate 2  JustInCase Hat  OutOfStock Cable  No Money | Sweets Cozonac Chocolate Flowers Wine Eggs Hat |

# Problem 3. Easter Shopping

*You have decided to go on an Easter shopping spree to take advantage of the promotions.*

Create a program that helps you keep track of the **shops** that you want to visit. You will **receive** the **list** of **shops** you have planned on checking out on a **single line,** **separated** by a **single** **space** in the following format:

**"{shop1} {shop2} {shop3}… {shopn}"**

Then you will receive a number – **n** - a **count** of **commands** you need to execute over your list. There are **four** **possible** **commands**:

* **"Include {shop}":**
  + **Add** the shop **at the end of your list.**
* **"Visit {first/last} {numberOfShops}"**
  + **Remove** either the "**first"** or the "**last"** **number of shops from your list**, **depending** on the **input**. If you have **less** **shops** on your list than the **given** **number**, **skip** this command.
* **"Prefer {shopIndex1} {shopIndex2}":**
  + **If** **both** of the **shop indexes** **exist** in your list, take the shops that are on them and **change** **their places**.
* **"Place {shop} {shopIndex}"**
  + **Insert** the **shop** **after** the given **index**, only **if** the **resulted index** **exists**.

In the end **print** the **manipulated list** in the following format:

**"Shops left:**

**{shop1} {shop2}… {shopn}"**

## Input / Constraints

* On the **1st line**, you will receive the **starting list** with the **names of the shops** **separated** by a **single space**.
* On the **2nd line**, you will receive the number of commands - **n – an integer in range [1…100]**
* On the next **n** lines you will be receiving commands in the **format** **described** above.

## Output

* Print the **list after** the **manipulations** in the **format** **described** above.

## Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Bershka CandyStore ThriftShop Armani Groceries ToyStore PeakStore  5  Include HM  Visit first 2  Visit last 1  Prefer 3 1  Place Library 2 | Shops left:  ThriftShop ToyStore Groceries Library Armani PeakStore |
| **Comments** | |
| First we receive the "**Include**" and the name of the store and we **add** the store to our **list**. The list should look like this: **Bershka CandyStore ThriftShop Armani Groceries ToyStore PeakStore HM**  After, we receive the "**Visit**" command and "**first**", which means we have to visit **the first 2 stores**, so we **remove** them from our list and the collection should look like this: **ThriftShop Armani Groceries ToyStore PeakStore HM**. After that, we receive the "**Visit**" command again, but this time we need to visit the "**last**" 1 store, so we **remove** it and the collection should look like this: **ThriftShop Armani Groceries ToyStore PeakStore**. After that we receive the "**Prefer**" command, which means we need to find the shop on the first given index – **3** and change it with the one that is on index – **1**, and the collection should look like this: **ThriftShop ToyStore Groceries Armani PeakStore**. At last, we receive the "**Place**" command and we need to **insert** the shop at the **next** index **after** **2**. And our final list looks like this:  **ThriftShop ToyStore Groceries Library Armani PeakStore** | |
|  | |
| Boutique Flowers CandyStore ThriftShop Versace Groceries ToyStore PeakStore  6  Visit first 9  Visit last 4  Prefer 3 8  Prefer 0 1  Place Store 7  Place ShoeAquarium 2 | Shops left:  Flowers Boutique CandyStore ShoeAquarium ThriftShop |