Programming Fundamentals with Python: Exam Preparation

01. Secret Chat

Submit your solutions in the SoftUni judge system at https://judge.softuni.org/Contests/Practice/Index/2307#0.

You have plenty of free time, so you decide to write a program that conceals and reveals your received messages. Go ahead and type it in!

On the first line of the input, you will receive the **concealed message**. After that, until the **"Reveal"** command is given, **you will receive strings with instructions for different operations that need to be performed upon the concealed message** to **interpret it** and reveal its actual content. There are several types of instructions, split by **":|:"**

- **"InsertSpace:|:{index}"**:
  - Inserts a single **space at the given index**. The given index will always be valid.
- **"Reverse:|:{substring}"**:
  - If the message contains the given **substring**, **cut it out**, **reverse** it and **add** it at the **end** of the message.
  - If not, print **"error"**.
  - This operation should replace only the first occurrence of the given **substring if there are two or more occurrences**.
- **"ChangeAll:|:{substring}:|:{replacement}"**:
  - Changes **all occurrences** of the given **substring** with the **replacement text**.

Input / Constraints

- On the first line, you will receive a string with a message.
- On the following lines, you will be receiving commands, split by **":|:"**.

Output

- After each set of instructions, print the resulting string.
- After the **"Reveal"** command is received, print this message:
  "You have a new text message: {message}"

Examples

Input	Output
heVVodar!gniV ChangeAll:|:V:|:l Reverse:|:!gnil InsertSpace:|:5 Reveal	hellodar!gnil hellodarling! hello darling! You have a new text message: hello darling!
Comments	
ChangeAll:|:V:|:l heVVodar!gniV -> hellodar!gnil (We replace all occurrences of "V" with "l")	

Reverse:|:!gnil

hellodar!gnil -> !gnil -> ling! -> hellodarling! (We reverse !gnil to ling! And put it at the end of the string)

InsertSpace:|:5

hellodarling! -> hello darling! (We insert a space at index 5)

Finally, after receiving the **"Reveal"** command, we print the resulting message.

Input	Output
Hiware?uiy ChangeAll:|:i:|:o Reverse:|:?uoy Reverse:|:jd InsertSpace:|:3 InsertSpace:|:7 Reveal	Howare?uoy Howareyou? error How areyou? How are you? You have a new text message: How are you?

02. Destination Mapper

Submit your solutions in the SoftUni judge system at https://judge.softuni.org/Contests/Practice/Index/2518#1.

Now that you have planned out your tour, you are ready to go! Your next task is to mark all the points on the map that you are going to visit.

You will be given a **string** representing some **places** on the map. You have to **filter** only the **valid ones**. A valid location is:

- Surrounded by **"="** or **"/"** on **both sides** (the **first** and the **last** symbols must **match**)
- After the **first "="** or **"/"** there should be **only letters** (the **first** must be **upper-case,** other letters could be upper or lower-case)
- The **letters** must be **at least 3**

Example: In the string **"=Hawai=/Cyprus/=Invalid/invalid==i5valid=/I5valid/=i="** only the **first two** locations are valid.

After you have **matched** all the **valid locations**, you have to **calculate travel points**. They are calculated by **summing** the **lengths** of all the **valid destinations** that you have found on the map.

In the end, on the **first line,** print: **"Destinations: {destinations joined by ', '}"**.

On the **second line,** print **"Travel Points: {travel_points}"**.

Input / Constraints

- You will receive a string representing the **locations on the map**
- **JavaScript**: you will receive a **single parameter: string**

Output

- Print the **messages described above**

Examples

Input	Output
=Hawai=/Cyprus/=Invalid/invalid==i5valid=/I5valid/=i=	Destinations: Hawai, Cyprus Travel Points: 11
ThisIs some InvalidInput	Destinations: Travel Points: 0

03. The Pianist

Submit your solutions in the SoftUni judge system at https://judge.softuni.org/Contests/Practice/Index/2525#2.

You are a pianist, and you like to keep a list of your favorite piano pieces. Create a program to help you organize it and add, change, remove pieces from it!

On the first line of the standard input, you will receive an integer **n** – the **number of pieces** you will initially have. On the next **n** lines, the **pieces themselves** will follow with their **composer** and **key**, separated by **"|"** in the following format: **"{piece}|{composer}|{key}"**.

Then, you will be receiving different **commands**, each on a new line, separated by **"|",** until the **"Stop"** command is given:

- **"Add|{piece}|{composer}|{key}"**:
  - You need to **add the given piece** with the information about it to the other pieces and print:
    "{piece} by {composer} in {key} added to the collection!"
  - If the piece **is already in the collection**, print:
    "{piece} is already in the collection!"
- **"Remove|{piece}"**:
  - If the piece is in the collection, **remove it** and print:
    "Successfully removed {piece}!"
  - Otherwise, print:
    "Invalid operation! {piece} does not exist in the collection."
- **"ChangeKey|{piece}|{new key}"**:
  - If the piece is in the collection, **change its key with the given one** and print:
    "Changed the key of {piece} to {new key}!"
  - Otherwise, print:
    "Invalid operation! {piece} does not exist in the collection."

Upon receiving the **"Stop"** command, you need to print all pieces in your collection in the following format:
"{Piece} -> Composer: {composer}, Key: {key}"

Input/Constraints

- You will receive **a single integer** at first – **the initial number of pieces in the collection**
- For each piece, you will receive a single line of text with information about it.
- Then you will receive multiple commands in the way described above until the command **"Stop"**.

Output

- All the output messages with the appropriate formats are described in the problem description.

Examples

Input	Output
3 Fur Elise|Beethoven|A Minor Moonlight Sonata|Beethoven|C# Minor Clair de Lune|Debussy|C# Minor Add|Sonata No.2|Chopin|B Minor Add|Hungarian Rhapsody No.2|Liszt|C# Minor Add|Fur Elise|Beethoven|C# Minor Remove|Clair de Lune ChangeKey|Moonlight Sonata|C# Major Stop	Sonata No.2 by Chopin in B Minor added to the collection! Hungarian Rhapsody No.2 by Liszt in C# Minor added to the collection! Fur Elise is already in the collection! Successfully removed Clair de Lune! Changed the key of Moonlight Sonata to C# Major! Fur Elise -> Composer: Beethoven, Key: A Minor Moonlight Sonata -> Composer: Beethoven, Key: C# Major Sonata No.2 -> Composer: Chopin, Key: B Minor Hungarian Rhapsody No.2 -> Composer: Liszt, Key: C# Minor

Comments

After we receive the initial pieces with their info, we start receiving commands. The first two commands are to add a piece to the collection, and since the pieces are not already added, we manage to add them. The third add command, however, **attempts to add a piece, which is already in the collection**, so we print a special message and don't add the piece. After that, we receive the remove command, and since the piece is in the collection, we remove it successfully.
Finally, the last command says to change the key of a piece. Since the key is present in the collection, we modify its key.
We receive the Stop command, print the information about the pieces, and the program ends.

Input	Output
4 Eine kleine Nachtmusik|Mozart|G Major La Campanella|Liszt|G# Minor The Marriage of Figaro|Mozart|G Major Hungarian Dance No.5|Brahms|G Minor Add|Spring|Vivaldi|E Major	Spring by Vivaldi in E Major added to the collection! Successfully removed The Marriage of Figaro! Invalid operation! Turkish March does not exist in the collection.

Remove|The Marriage of Figaro Remove|Turkish March ChangeKey|Spring|C Major Add|Nocturne|Chopin|C# Minor Stop	Changed the key of Spring to C Major! Nocturne by Chopin in C# Minor added to the collection! Eine kleine Nachtmusik -> Composer: Mozart, Key: G Major La Campanella -> Composer: Liszt, Key: G# Minor Hungarian Dance No.5 -> Composer: Brahms, Key: G Minor Spring -> Composer: Vivaldi, Key: C Major Nocturne -> Composer: Chopin, Key: C# Minor

Suggestion: Choose a piece from this collection and listen to it while solving the problem!