Programming Fundamentals Final Exam Preparation 2

Problem 1 - Password Reset

Link: https://judge.softuni.org/Contests/Practice/Index/2303#0

Yet again, you have forgotten your password. Naturally, it's not the first time this has happened. Actually, you got so tired of it that you decided to help yourself with an intelligent solution.

Write a password reset program that performs a series of commands upon a predefined string. First, you will receive a string, and afterward, until the command **"Done"** is given, you will be receiving strings with commands split by a single space. The commands will be the following:

- **"TakeOdd"**
  - Takes only the characters at **odd indices** and **concatenates** them to obtain the **new raw password** and then **prints** it.
- **"Cut {index} {length}"**
  - Gets the substring with the **given length** starting from the **given index** from the password and removes its **first occurrence**, then **prints** the password on the console.
  - The given index and the length will **always** be **valid**.
- **"Substitute {substring} {substitute}"**
  - If the raw password contains the given substring, replaces all of its occurrences with the substitute text given and prints the result.
  - If it doesn't, prints **"Nothing to replace!"**.

Input

- You will be receiving strings until the **"Done"** command is given.

Output

- After the **"Done"** command is received, print:
  - **"Your password is: {password}"**

Constraints

- The indexes from the "**Cut {index} {length}**" command will always be valid.

Examples

Input	Output
["Siiceercaroetavm!:?:ahsott.:i:nstupmomceqr", "TakeOdd", "Cut 15 3", "Substitute :: -", "Substitute | ^", "Done"])	icecream::hot::summer icecream::hot::mer icecream-hot-mer Nothing to replace! Your password is: icecream-hot-mer

Comments
TakeOdd -> We only take the chars at odd indices 1, 3, 5 etc. Siiceercaroetavm!:?:ahsott.:i:nstupmomceqr -> icecream::hot::summer **Cut 15 3 ->** We cut a substring starting at index 15 with length 3, then remove it from the raw password: icecream::hot::summer -> sum **Substitute :: - ->** We replace "::" with "-": icecream::hot::summer -> icream-hot-summer **Substitute** | ^ **->** "|" is not found anywhere in the raw password, so we print **"Nothing to replace!"** Finally, after receiving the **"Done"** command, we print the resulting password in the proper format.

Input	Output

`(["up8rgoyg3r1atmlmpiunagt!-irs7!1fgulnnnqy",`
`"TakeOdd",`
`"Cut 18 2",`
`"Substitute ! ***",`
`"Substitute ? .!.",`
`"Done"])`
	`programming!is!funny`
	`programming!is!fun`
	`programming***is***fun`
	`Nothing to replace!`
	`Your password is:`
	`programming***is***fun`

Problem 2 - Destination Mapper

Link: 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`

Problem 3 - Need for Speed III

Link: https://judge.softuni.org/Contests/Practice/Index/2307#2

You have just bought the latest and greatest computer game – Need for Seed III. Pick your favorite cars and drive them all you want! We know that you can't wait to start playing.

On the first line of the standard input, you will receive an integer **n** – the **number of cars** that you can obtain. On the next **n** lines, the **cars themselves** will follow with their **mileage** and **fuel available**, separated by **"|"** in the following format:

`"{car}|{mileage}|{fuel}"`

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

- **`"Drive : {car} : {distance} : {fuel}"`**:
  - You need to **drive the given distance**, and you will **need the given** fuel to do that. If the car **doesn't have enough fuel**, print: **`"Not enough fuel to make that ride"`**
  - If the car has the required fuel available in the tank, **increase its mileage** with **the given distance**, **decrease its fuel with the given fuel,** and **print**:
    `"{car} driven for {distance} kilometers. {fuel} liters of fuel consumed."`
  - You like driving new cars only, so if a car's mileage reaches **100 000** km, remove it from the collection(s) and print: **`"Time to sell the {car}!"`**
- **`"Refuel : {car} : {fuel}"`**:
  - **Refill** the tank of your car.
  - Each tank can hold a **maximum of 75 liters of fuel**, so if the given amount of fuel is more than you can fit in the tank, take only what is required to fill it up.
  - Print a message in the following format: **`"{car} refueled with {fuel} liters"`**
- **`"Revert : {car} : {kilometers}"`**:

- Decrease the **mileage** of the given **car with the given kilometers** and print the kilometers you have decreased it with in the following format:
  `"{car} mileage decreased by {amount reverted} kilometers"`
- If the mileage becomes **less than 10 000km after** it is decreased, **just set it to 10 000km** and **DO NOT print anything.**

Upon receiving the **`"Stop"`** command, you need to print all cars in your possession in the following format:
`"{car} -> Mileage: {mileage} kms, Fuel in the tank: {fuel} lt."`

Input/Constraints

- The **mileage** and **fuel** of the cars will be valid, 32-bit integers, and will never be negative.
- The **fuel** and **distance** amounts **in the commands will never be negative**.
- The **car names** in the **commands** will always be **valid cars in your possession**.

Output

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

Examples

Input	Output
['3', 'Audi A6|38000|62', 'Mercedes CLS|11000|35', 'Volkswagen Passat CC|45678|5', 'Drive : Audi A6 : 543 : 47', 'Drive : Mercedes CLS : 94 : 11', 'Drive : Volkswagen Passat CC : 69 : 8', 'Refuel : Audi A6 : 50', 'Revert : Mercedes CLS : 500', 'Revert : Audi A6 : 30000', 'Stop']	Audi A6 driven for 543 kilometers. 47 liters of fuel consumed. Mercedes CLS driven for 94 kilometers. 11 liters of fuel consumed. Not enough fuel to make that ride Audi A6 refueled with 50 liters Mercedes CLS mileage decreased by 500 kilometers Audi A6 -> Mileage: 10000 kms, Fuel in the tank: 65 lt. Mercedes CLS -> Mileage: 10594 kms, Fuel in the tank: 24 lt. Volkswagen Passat CC -> Mileage: 45678 kms, Fuel in the tank: 5 lt.
Comments	
After we receive the cars with their mileage and fuel, we start driving them. When we get to "**Drive : Volkswagen Passat CC : 69 : 8**" command, our program calculates that there is not enough fuel,0 and we print the appropriate message. Then we refuel the Audi A6 with 50 l of fuel and Revert the Mercedes with 500 kilometers.	

When we receive the "Revert : Audi A6 : 30000", we set its mileage to **10000** km, because if the current mileage of the Audi is **38543** kms and if we subtract **30000** from it, we receive **8543** kms, which is less than 10000 kms.

After all the commands, we print our current collection of cars with their current mileage and current fuel.

Input	Output
['4', 'Lamborghini Veneno|11111|74', 'Bugatti Veyron|12345|67', 'Koenigsegg CCXR|67890|12', 'Aston Martin Valkryie|99900|50', 'Drive : Koenigsegg CCXR : 382 : 82', 'Drive : Aston Martin Valkryie : 99 : 23', 'Drive : Aston Martin Valkryie : 2 : 1', 'Refuel : Lamborghini Veneno : 40', 'Revert : Bugatti Veyron : 2000', 'Stop']	Not enough fuel to make that ride Aston Martin Valkryie driven for 99 kilometers. 23 liters of fuel consumed. Aston Martin Valkryie driven for 2 kilometers. 1 liters of fuel consumed. Time to sell the Aston Martin Valkryie! Lamborghini Veneno refueled with 1 liters Bugatti Veyron mileage decreased by 2000 kilometers Lamborghini Veneno -> Mileage: 11111 kms, Fuel in the tank: 75 lt. Bugatti Veyron -> Mileage: 10345 kms, Fuel in the tank: 67 lt. Koenigsegg CCXR -> Mileage: 67890 kms, Fuel in the tank: 12 lt.