*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**.