# Problem 2 - Automobile Pairing



*Ensuring each car is matched perfectly with its correct registration number is crucial for maintaining accuracy and reliability in this exercise. This meticulous process guarantees seamless identification and linkage of cars with their respective numbers!*

You need to **link** **cars** with their perfect **matching** car **numbers**. On the following lines until the command "**Done**" you will be receiving **strings** consisting of the car's **name** and the car's **registration** **number**.

**Gather** all the car names, then identify all the **valid** registration numbers by running a few **validations**:

* **Car Name**:
  + Should consist of only **letters** and **spaces**.
  + The first letter is always **uppercase**.
  + Is surrounded by "**<**" (**on the left**) and "**>**" (**on the right**).
* **Registration Number**:
  + Always positioned **after the car name**.
  + Consists of only **letters** (**all uppercase**) and **digits.**
  + The number is **exactly** **8** characters long.
  + The number **starts** with **2** **letters**, followed by **4** **digits,** and **finishes** with **2** **letters**.
  + Is surrounded by "**&&**" on both sides.
* **Additions**:
  + In between and/or sides, **digits**, **letters**, and **special** **characters** could be included.
* **Message validity:** If either the car **name** or the **registration** **number** **doesn't** meet the **requirements**, the entire match is **invalid**!

Examples of **valid** messages:

* **!!<Mercedes CLS AMG><.>&&TX1234AM&&312!!, ++<Bugatti Chiron>lds'&&PB7777MM&&???**

Examples of **invalid** messages:

* **<Audi-RS-7>->?-&&CB0001KM&&312, <Ford Mustang|&&TX0044XA&&**

Go through every string and find all **valid** matches.

Print each found message in the format:

* **"Linked car {car\_name} with number {registration\_number}!"**

### Input / Constraints

* You will receive strings on new lines, until the command "**Done**".
* Print the proper output messages in the proper cases as described in the problem description.
* Every string should consist of only **one** **car** and only **one** **registration** **number**.

### Examples

|  |
| --- |
| **Input** |
| <Mercedes S Class>&&CA0022PP&&---  <Audi-Q-Series>->?-&&CB0001KM&&111  |BMW M Series|&&EB8000OM&&312  Done |
| **Output** |
| Linked car Mercedes S Class with number CA0022PP! |
| **Input** |
| <Tesla Model S>-&&PB4567KK&&  ?<Dodge Challenger>&&EH3214KK&&?  Done |
| **Output** |
| Linked car Tesla Model S with number PB4567KK!  Linked car Dodge Challenger with number EH3214KK! |

### JS Examples

|  |
| --- |
| **Input** |
| (["<Mercedes S Class>&&CA0022PP&&",  "<Audi-Q-Series>->?-&&CB0001KM&&111",  "|BMW M Series|&&EB8000OM&&312",  "Done"]) |
| **Output** |
| Linked car Mercedes S Class with number CA0022PP! |
| **Input** |
| (["<Tesla Model S>-&&PB4567KK&&",  "?<Dodge Challenger>&&EH3214KK&&?",  "Done"]) |
| **Output** |
| Linked car Tesla Model S with number PB4567KK!  Linked car Dodge Challenger with number EH3214KK! |