# Exercises: Interfaces

This document defines the **exercise assignments** for the ["PHP Fundamentals" course @ Software University](https://softuni.bg/php-basics).

### WARNING!

In all tasks use **point** as a decimal separator and do not use comma or any other styling as thousands separator.

## Define an Interface IPerson

Define an interface **Person** with properties for **Name** and **Age**. Define a class **Citizen** which implements **Person** and has a constructor which takes a **string** name and an **int** age.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Pesho  25 | Pesho  25 |

## Multiple Implementation

Using the code from the previous task, define an interface **Identifiable** with a **string** property **Id** and an interface **Birthable** with a **string** property **Birthdate** and implement them in the **Citizen** class. Rewrite the Citizen constructor to accept the new parameters.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Pesho  25  9105152287  15/05/1991 | Pesho  25  9105152287  15/05/1991 |

## Ferrari

Model an application which contains a **class Ferrari** and an **interface**. Your task is simple, you have a **car - Ferrari**, its model is **"****488-Spider"** and it has a **driver**. Your Ferrari should have functionality to **use brakes** and **push the gas pedal**. When the **brakes** are pushed down **print "****Brakes!"**, and when the **gas pedal** is pushed down - **"****Zadu6avam sA!"**. As you may have guessed this functionality is typical for all cars, so you should **implement an interface** to describe it.

Your task is to **create a Ferrari** and **set the driver's name** to the passed one in the input. After that, print the info. Take a look at the Examples to understand the task better.

### Input

On the **single input line**, you will be given the **driver's name**.

### Output

On the **single output line**, print the model, the messages from the brakes and gas pedal methods and the driver's name. In the following format:

<**model**>/<**brakes**>/<**gas** **pedal**>/<**driver's** **name**>

### Constraints

The input will always be valid, no need to check it explicitly! The Driver's name may contain any ASCII characters.

### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| Bat Giorgi | 488-Spider/Brakes!/Zadu6avam sA!/Bat Giorgi |
| Dinko | 488-Spider/Brakes!/Zadu6avam sA!/Dinko |

## Telephony

You have a business - **manufacturing cell phones**. But you have no software developers, so you call your friends and ask them to help you create a cell phone software. They agree and you start working on the project. The project consists of one main **model - a Smartphone**. Each of your smartphones should have functionalities of **calling other phones** and **browsing in the world wide web.**

Your friends are very busy, so you decide to write the code on your own. Here is the mandatory assignment:

You should have a **model** - **Smartphone** and two separate functionalities which your smartphone has - to **call other phones** and to **browse in the world wide web**. You should end up with **one class** and **two interfaces**.

### Input

The input comes from the console. It will hold two lines:

* **First line**: **phone numbers** to call (String), separated by spaces.
* **Second line: sites** to visit (String), separated by spaces.

### Output

* First **call all numbers** in the order of input then **browse all sites** in order of input
* The functionality of calling phones is printing on the console the number which are being called in the format:

**Calling... <number>**

* The functionality of the browser should print on the console the site in format:

**Browsing: <site>!**

* If there is a number in the input of the URLs, print: **"****Invalid URL!"** and continue printing the rest of the URLs.
* If there is a character different from a digit in a number, print: **"Invalid number!"** and continue to the next number.

### Constraints

* Each site's URL should consist only of letters and symbols (**No digits are allowed** in the URL address)

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 0882134215 0882134333 08992134215 0558123 3333 1  http://softuni.bg http://youtube.com http://www.g00gle.com | Calling... 0882134215  Calling... 0882134333  Calling... 08992134215  Calling... 0558123  Calling... 3333  Calling... 1  Browsing: http://softuni.bg!  Browsing: http://youtube.com!  Invalid URL! |

## Border Control

It’s the future, you’re the ruler of a totalitarian dystopian society inhabited by **citizens** and **robots**, since you’re afraid of rebellions you decide to implement strict control of who enters your city. Your soldiers check the **Id**s of everyone who enters and leaves.

### Input

You will receive from the console an unknown amount of lines until the command “**End**” is received, on each line there will be the information for either a citizen or a robot who tries to enter your city in the format **“<name> <age> <id>**” for citizens and “**<model> <id>**” for robots. After the end command on the next line you will receive a single number representing **the last digits of fake ids**, all citizens or robots whose **Id** ends with the specified digits must be detained.

### Output

The output of your program should consist of all detained **Id**s each on a separate line (print in the same order of appearance).

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Pesho 22 9010101122  MK-13 558833251  MK-12 33283122  End  122 | 9010101122  33283122 |
| Toncho 31 7801211340  Penka 29 8007181534  IV-228 999999  Stamat 54 3401018380  KKK-666 80808080  End  340 | 7801211340 |

## Birthday Celebrations

It is a well known fact that people celebrate birthdays, it is also known that some people also celebrate their pets birthdays. Extend the program from your last task to add **birthdates** to citizens and include a class **Pet**, pets have a **name** and a **birthdate**. Encompass repeated functionality into interfaces and implement them in your classes.

You will receive from the console an unknown amount of lines until the command “**End**” is received, each line will contain information in one of the following formats **“Citizen <name> <age> <id> <birthdate>**” for citizens, “**Robot** **<model> <id>**” for robots or “**Pet <name> <birthdate>**” for pets. After the end command on the next line you will receive a single number representing **a specific year**, your task is to print all birthdates (of both citizens and pets) in that year in the format **day/month/year** (print in order of appearance).

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Citizen Pesho 22 9010101122 10/10/1990  Pet Sharo 13/11/2005  Robot MK-13 558833251  End  1990 | 10/10/1990 |
| Citizen Stamat 16 0041018380 01/01/2000  Robot MK-10 12345678  Robot PP-09 00000001  Pet Topcho 24/12/2000  Pet Kosmat 12/06/2002  End  2000 | 01/01/2000  24/12/2000 |
| Robot VV-XYZ 11213141  Citizen Penka 35 7903210713 21/03/1979  Citizen Kane 40 7409073566 07/09/1974  End  1975 | <no output> |