**Exercise: Interfaces and Abstraction**

Problems for exercise and homework for the ["C# OOP" course @ SoftUni"](https://softuni.bg/trainings/3585/csharp-oop-february-2022).

You can check your solutions here: <https://judge.softuni.org/Contests/1502/Interfaces-and-Abstraction-Exercise>

* **Telephony**

You have a small business - **manufacturing phones** and to run your business you need to create phone software. The software should support two main phone **models with the following functionality:**

* **Smartphone:**
* Can **calling other phones.**
* Can **browsing in the world wide web.**
* **Stationary phone**:
* Can **only call other phones.**

You should start the project by implementing two **classes**:

* **The Smartphone** can **call other phones** and **browse the world wide web**.
* **The StationaryPhone** can only **call other phones**.

You should also implement **interfaces for each class with the appropriate methods**.

**Input**

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

* **The First line** consists of **phone numbers**: a **string**, separated by spaces.
* **The Second** line consists **of websites: a string**, separated by spaces.

**Output**

* First, **call all valid numbers** in the order of input:
* If there is a character different from a digit in a number, print: **"Invalid number!"** and continue with the next number.
* If the number is **10 digits long**, you are making a call from your smartphone and print: "**Calling... {number}"**
* If the number is 7 **digits long**, you are making a call from your stationary phone and print: " **Dialing... {number}"**
* Next, **browser all valid websites** in the order of input:
* If there is a number in the input of the URLs, print: **"Invalid URL!"** and continue with the next URLs.
* If the URL is valid, print on the console the website in the format: **"Browsing: {site}!"**

**Constraints**

* Each site's URL should consist only of letters and symbols (**No digits are allowed** in the URL address).
* The phone numbers will always be 7 or 10 digits long.

**Examples**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 0882134215 0882134333 0899213421 0558123 3333123  [http://softuni.bg](http://softuni.bg/) [http://youtube.com](http://youtube.com/) [http://www.g00gle.com](http://www.g00gle.com/) | Calling... 0882134215  Calling... 0882134333  Calling... 0899213421  Dialing... 0558123  Dialing... 3333123  Browsing: [http://softuni.bg](http://softuni.bg/)!  Browsing: [http://youtube.com](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.

You will receive an unknown amount of lines from the console until the command "**End**" is received, on each line, there will be a piece of 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.

The output of your program should consist of all detained **Id**s each on a separate line in the **order** of **input**.

**Input**

The input comes from the console. Every commands’ parameters before the command "**End**" will be separated by a **single space**.

**Examples**

|  |  |
| --- | --- |
| **Input** | **Output** |
| Peter 22 9010101122  MK-13 558833251  MK-12 33283122  End  122 | 9010101122  33283122 |
| Teo 31 7801211340  Peter 29 8007181534  IV-228 999999  Sam 54 3401018380  KKK-666 80808080  End  340 | 7801211340 |
| George 954614  Ron 124610  VI-228 999999  Mike 13 7604128614  Peter 90 5602142414  T500 131313130  End  14 | 954614  7604128614  5602142414 |