**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>

* **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 number 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 **Citizen**, "**Robot** **<model> <id>**" for **Robot** or "**Pet <name> <birthdate**" for **Pet**. 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 **Citizen** and **Pet**) in that yearin the format **day/month/year** in the **order** of **input**.

**Examples**

|  |  |
| --- | --- |
| **Input** | **Output** |
| Citizen Peter 22 9010101122 10/10/1990  Pet Sharo 13/11/2005  Robot MK-13 558833251  End  1990 | 10/10/1990 |
| Citizen Stam 16 0041018380 01/01/2000  Robot MK-10 12345678  Robot PP-09 00000001  Pet Topcho 24/12/2000  Pet Rex 12/06/2002  End  2000 | 01/01/2000  24/12/2000 |
| Robot VV-XYZ 11213141  Citizen Corso 35 7903210713 21/03/1979  Citizen Kane 40 7409073566 07/09/1974  End  1975 | <empty output> |

* **Food Shortage**

Your totalitarian dystopian society suffers a shortage of food, so many rebels appear. Extend the code from your previous task with new functionality to solve this task.

Define a class **Rebel** which has a **name**, **age,**and**group**(**string**)**,**names are**unique -**there will never be 2 **Rebel**s/**Citizen**s or a **Rebel** and **Citizen** with the same name**.** Define an interface **IBuyer** which defines a method **BuyFood()**and an integer property**Food**. Implement the **IBuyer** interface in the **Citizen** and **Rebel** class, both **Rebel**s and **Citizen**s **start with 0 food**, when a **Rebel** buys food his **Food** increases by **5**, when a **Citizen** buys food his **Food** increases by **10**.

On the first line of the input you will receive an integer **N** - the number of people, on each of the next **N** lines you will receive information in one of the following formats "**<name> <age> <id> <birthdate>**" for a **Citizen** or "**<name> <age><group>**" for a **Rebel**. After the **N** lines, until the command "**End**" is received, you will receive names of people who bought food, each on a new line. Note that not all names may be valid, in case of an incorrect name - nothing should happen.

**Output**

The **output** consists of only **one line** on which you should print the **total** amount of food purchased.

**Examples**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  Peter 25 8904041303 04/04/1989  Stan 27 WildMonkeys  Peter  George  Peter  End | 20 |
| 4  Stam 23 TheSwarm  Ton 44 7308185527 18/08/1973  George 31 Terrorists  Pen 27 881222212 22/12/1988  John  Geo rge  John  Joro  Stam  Pen  End | 15 |