# Vet Clinic

## Preparation

Download the skeleton provided in Judge. **Do not** change the **StartUp** class or its **namespace**.

## Problem description

Your task is to create a repository, which stores items by creating the classes described below.

First, write a C# class **Pet** with the following properties:

* **Name: string**
* **Age: int**
* **Owner: string**

The class **constructor** should receive **name, age** and **owner.** The class should override the **ToString()** method in the following format:

**"Name: {Name} Age: {Age} Owner: {Owner}"**

**Next**, write a C# class **Clinic** that has **data** (a collection, which stores the Pets). All entities inside the repository have the **same properties**. Also, the **Clinic** class should have those properties:

* **Capacity: int**

The class **constructor** should receive **capacity**, also it should initialize the **data** with a new instance of the collection**.** Implement the following features:

* Field **data** – **collection** that holds added pets
* Method Add(Pet pet) – **adds** an **entity** to the data **if** **there** **is** an **empty cell** for the pet.
* Method Remove(string name) – removes the pet by **given name,** if such **exists**, and **returns bool**.
* Method **GetPet(string name, string owner)** – returns the pet with the **given name** and **owner** or **null if no such pet exists**.
* Method GetOldestPet() – returns the oldest Pet.
* Getter Count – **returns** the **number** of pets.
* **GetStatistics()** – **returns** a **string** in the following **format**:
  + **"The clinic has the following patients:  
    Pet {Name} with owner: {Owner}  
    Pet {Name} with owner: {Owner}**

**(…)**"

## Constraints

* The **combinations** of **names** and **owners** will **always be unique**.
* The **age** of the pets will always be **positive**.  
  Examples

This is an example of how the **Clinic** class is **intended to be used**.

|  |
| --- |
| Sample code usage |
| // Initialize the repository  Clinic clinic = new Clinic(20);  // Initialize entity  Pet dog = new Pet("Ellias", 5, "Tim");  // Print Pet  Console.WriteLine(dog); // Ellias 5 (Tim)  // Add Pet  clinic.Add(dog);  // Remove Pet  Console.WriteLine(clinic.Remove("Ellias")); // True  Console.WriteLine(clinic.Remove("Pufa")); // False  Pet cat = new Pet("Bella", 2, "Mia");  Pet bunny = new Pet("Zak", 4, "Jon");  clinic.Add(cat);  clinic.Add(bunny);  // Get Oldest Pet  Pet oldestPet = clinic.GetOldestPet();  Console.WriteLine(oldestPet); // Zak 4 (Jon)  // Get Pet  Pet pet = clinic.GetPet("Bella", "Mia");  Console.WriteLine(pet); // Bella 2 (Mia)  // Count  Console.WriteLine(clinic.Count); // 2  // Get Statistics  Console.WriteLine(clinic.GetStatistics());  //The clinic has the following patients:  //Bella Mia  //Zak Jon |

## Submission

Zip all the files in the project folder except **bin** and **obj** folders.