Упорядочивание. Группирование. Операции над множествами.

```
class Employee
{
    public int Id { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public int Age { get; set; }
    public int DepId { get; set; }
}
class Department
{
    public int Id { get; set; }
    public string Country { get; set; }
    public string City { get; set; }
}

class Program
{
      static void Main()
      {
            List<Department> departments = new List<Department>()
      {   new Department(){ Id = 1, Country = "Ukraine", City = "Odesa"},
          new Department(){ Id = 2, Country = "Ukraine", City = "Kyiv" },
          new Department(){ Id = 3, Country = "France", City = "Paris" },
          new Department(){ Id = 4, Country = " Ukraine ", City = "Lviv"}
      };
   List<Employee> employees = new List<Employee>()
   {
   new Employee()
   {
     Id = 1, FirstName = "Tamara", LastName = "Ivanova", Age = 22, DepId = 2
   },
   new Employee()
   {
     Id = 2, FirstName = "Nikita", LastName = "Larin", Age = 33, DepId = 1
   },
   new Employee()
   {
     Id = 3, FirstName = "Alica", LastName = "Ivanova", Age = 43, DepId = 3
   },
   new Employee()
   {
     Id = 4, FirstName = "Lida", LastName = "Marusyk", Age = 22, DepId = 2
   },
```

```
    new Employee()
    {
      Id = 5, FirstName = "Lida", LastName = "Voron", Age = 36, DepId = 4
    },
    new Employee()
    {
      Id = 6, FirstName = "Ivan", LastName = "Kalyta", Age = 22, DepId = 2
    },
    new Employee()
    {
      Id = 7, FirstName = "Nikita", LastName = "Krotov", Age = 27, DepId = 4
    }
    };
}
}
```

1) Упорядочить имена и фамилии сотрудников по алфавиту, которые проживают в Украине. Выполнить запрос немедленно.
2) Отсортировать сотрудников по возрастам по убыванию. Вывести Id, FirstName, LastName, Age. Выполнить запрос немедленно.
3) Сгруппировать студентов по возрасту. Вывести возраст и сколько раз он встречается в списке.