

PHIẾU BÀI TẬP 1

Tạo 1 ứng dụng Console đặt tên như sau: **Bai11_tên-sinh-viên-3-số-cuối-mã-sinh-viên_P1**, tên không dấu (ví dụ: **Bai11_Anh123_P1**)

Thực hiện các yêu cầu sau:

- Làm các bài tập trong phiếu giao
- Nén toàn bộ solution và nộp file nén lên hệ thống LMS

Bài tập 1.

Trong solution đã tạo ở trên, chuẩn bị nguồn dữ liệu cho các truy vấn LINQ

- Xây dựng lớp Product như sau:

```
public class Product
{
    public int ID { set; get; }
    public string Name { set; get; }           // tên
    public double Price { set; get; }          // giá
    public string[] Colors { set; get; }       // các màu
    public int Brand { set; get; }             // ID Nhãn hiệu, hãng

    public Product(int id, string name, double price,
                  string[] colors, int brand)
    {
        ID = id;
        Name = name;
        Price = price;
        Colors = colors;
        Brand = brand;
    }

    // Lấy chuỗi thông tin sản phẩm gồm ID, Name, Price
    override public string ToString()
    => $"{ID,3} {Name,12} {Price,5} {Brand,2} {string.Join(", ",
        Colors)}";
}
```

- Xây dựng lớp Brand để biểu diễn nhãn hiệu hàng hóa

```
public class Brand
{
    public string Name { set; get; }
    public int ID { set; get; }
}
```

- Trong lớp Program
using System.Collections.Generic;
using System.Linq;

- Trong phương thức Main, khởi tạo ra 2 danh sách làm nguồn dữ liệu, dùng để thực hiện các truy vấn linq: danh sách sản phẩm **products**, danh sách nhãn hiệu **brands**

```
var brands = new List<Brand>()
{
    new Brand{ID = 1, Name = "Vingroup"},
    new Brand{ID = 2, Name = "Samsung"},
    new Brand{ID = 3, Name = "FPT"},
};

var products = new List<Product>()
{
    new Product(1,"O to", 400, new string[] {"Do","Trang","Den"}, 1),
    new Product(2,"Dien thoai",400, new string[] {"Den", "Xanh"}, 2),
    new Product(3,"May giat", 500, new string[] {"Trang"}, 2),
    new Product(4,"Tu lanh", 200, new string[] {"Trang", "Xam"}, 2),
    new Product(5,"Laptop", 300, new string[] {"Xam", "Den", "Do"}, 3),
    new Product(6,"Dieu hoa", 500, new string[] {"Trang"}, 2),
    new Product(7,"Xe may", 600, new string[] {"Trang"}, 1),
};
```

Bài tập 2.

Sử dụng nguồn dữ liệu đã tạo trong bài tập 1, thực hiện các câu truy vấn LINQ sau để viết được các truy vấn LINQ theo cú pháp truy vấn và cú pháp phương thức

1. Mệnh đề from và select.

Yêu cầu: In ra danh sách các sản phẩm

```
//với mỗi product trong danh sách products, lấy ra product đó
var ketqua = from product in products
              select product;
//thực thi truy vấn để lấy dữ liệu
foreach (var product in ketqua)
    Console.WriteLine(product.ToString());
```

 Kết quả:

| | | | | |
|---|------------|-----|---|--------------|
| 1 | O to | 400 | 1 | Do,Trang,Den |
| 2 | Dien thoai | 400 | 2 | Den,Xanh |
| 3 | May giat | 500 | 2 | Trang |
| 4 | Tu lanh | 200 | 2 | Trang,Xam |
| 5 | Laptop | 300 | 3 | Xam,Den,Do |
| 6 | Dieu hoa | 500 | 2 | Trang |
| 7 | Xe may | 600 | 1 | Trang |

```
//Cú pháp phương thức cho kết quả tương tự
var ketqua = products.Select(product => product);
```

2. Chỉ lấy ra một loại dữ liệu của phần tử, ví dụ lấy tên của sản phẩm

```
var ketqua = from product in products
              select product.Name;
```

```
foreach (var name in ketqua) Console.WriteLine(name);
```

 Kết quả:

```
0 to
Điện thoại
Máy giặt
Tủ lạnh
Laptop
Điều hòa
Xe máy
```

//Cú pháp phương thức cho kết quả tương tự

```
var ketqua = products.Select(product => product.Name);
```

3. Trả về kiểu vô danh với toán tử new { }

```
var ketqua = from product in products
              select new
              {
                  ten = product.Name.ToUpper(),
                  mausac = string.Join(',', product.Colors)
              };
```

```
foreach (var item in ketqua)
    Console.WriteLine(item.ten + " - " + item.mausac);
```

 Kết quả:

```
0 TO - Đỏ,Trang,Đen
DIEN THOAI - Đen,Xanh
MAY GIAT - Trang
TU LANH - Trang,Xam
LAPTOP - Xam,Đen,Đỏ
DIEU HOA - Trang
XE MAY - Trang
```

//Cú pháp phương thức cho kết quả tương tự

```
var ketqua = products.Select(product =>
                        new {
                            ten = product.Name.ToUpper(),
                            mausac = string.Join(',', product.Colors)
                        });
```

4. Mệnh đề where

Yêu cầu: In ra các sản phẩm có giá 400

```
var ketqua = from product in products
              where product.Price == 400
              select product;

foreach (var product in ketqua)
    Console.WriteLine(product.ToString());
```

 Kết quả:

```
1      0 to   400  1 Do,Trang,Den
2  Dien thoai  400  2 Den,Xanh
```

//Cú pháp phương thức cho kết quả tương tự

```
var ketqua = products.Where(product => product.Price == 400);
```

5. In ra các sản phẩm có giá >=500 và <600 hoặc tên sản phẩm có ký tự e

```
var ketqua = from product in products
              where (product.Price >= 500 && product.Price < 600) || product.Name.Contains('e')
              select product;
```

```
foreach (var product in ketqua)
    Console.WriteLine(product.ToString());
```

 Kết quả:

```
2  Dien thoai  400  2 Den,Xanh
3    May giat  500  2 Trang
6  Dieu hoa   500  2 Trang
7    Xe may   600  1 Trang
```

//Cú pháp phương thức cho kết quả tương tự

```
var ketQua = products.Where(product => (product.Price >= 500 &&
                                         product.Price < 600) || product.Name.Contains('e'));
```

6. Mệnh đề orderby.

Yêu cầu: In ra tên và giá của các sản phẩm, danh sách được sắp xếp theo giá giảm dần

```
var ketqua = from product in products
              where product.Price <= 400
              orderby product.Price descending
              select product;

foreach (var product in ketqua)
    Console.WriteLine($"{product.Name} - {product.Price}");
```

 Kết quả:

```
0 to - 400
Dien thoai - 400
Laptop - 300
Tu lanh - 200
```

//Cú pháp phương thức cho kết quả tương tự

```
var ketQua = (products.Where(product => product.Price <= 400))
              .OrderByDescending(product => product.Price);
```

- Sắp xếp theo nhiều tiêu chí, các tiêu chí cách nhau dấu ,

```
var ketqua = from product in products
```

```

    where product.Price <= 400
    orderby product.Price, product.Name descending
    select product;
foreach (var product in ketqua)
    Console.WriteLine($"{product.Name} - {product.Price}");

```

 Kết quả:

```

Tu lanh - 200
Laptop - 300
0 to - 400
Dien thoai - 400

```

7. Mệnh đề group và by.

Yêu cầu: Lấy sản phẩm có giá <=500, nhóm lại theo thương hiệu (cùng Brand cho vào một nhóm)

```

var ketqua = from product in products
              where product.Price <= 500
              group product by product.Brand;
foreach (var group in ketqua)
{
    Console.WriteLine(group.Key);
    foreach (var product in group)
    {
        Console.WriteLine($"    {product.Name} - {product.Price}");
    }
}

```

 Kết quả:

```

1    0 to - 400
2    Dien thoai - 400
    May giat - 500
    Tu lanh - 200
    Dieu hoa - 500
3    Laptop - 300

```

//Cú pháp phương thức cho kết quả tương tự

```

var ketQua = (products.Where(product => product.Price <= 500))
               .GroupBy(product => product.Brand);

```

8. Mệnh đề join.

Yêu cầu: Truy vấn sản phẩm, mỗi sản phẩm căn cứ vào Brand ID của nó - lấy tên Brand tương ứng

```

var ketqua = from product in products
              join brand in brands on product.Brand equals brand.ID

```

```

    select new
    {
        name = product.Name,
        brand = brand.Name,
        price = product.Price
    };

foreach (var item in ketqua)
{
    Console.WriteLine($"{item.name,10} {item.price,4} {item.brand,12}");
}

```



Kết quả:

| | | |
|------------|-----|----------|
| 0 to | 400 | Vingroup |
| Điện thoại | 400 | Samsung |
| Máy giặt | 500 | Samsung |
| Tủ lạnh | 200 | Samsung |
| Laptop | 300 | FPT |
| Điều hòa | 500 | Samsung |
| Xe máy | 600 | Vingroup |

//Cú pháp phương thức cho kết quả tương tự

```

var ketqua = products.Join(brands, p => p.Brand, b => b.ID,
    (p, b) => { return new {
        name = p.Name,
        price= p.Price,
        brand= b.Name
    }
    });

```