using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

namespace ConsoleApp4

{

class Car

{

public int Year { get; set; }

public string Model { get; set; }

public string Vendor { get; set; }

public override string ToString()

{

return $"{Model.PadRight(15)}-{Vendor}-{Year}";

}

}

public class Program

{

static void Main(string[] args)

{

//int[] scores = { 97, 92, 78, 81, 60, 55 };

//var scoreQuery =

// from score in scores

// where score > 80

// select score;

//var scoreQuery2=

// scores

// .Where(score => score >80)

// .ToList();

//foreach (var item in scoreQuery)

//{

// Console.WriteLine(item);

//}

var cars = new List<Car>

{

new Car

{

Model="Mustang",

Vendor="Ford",

Year=1964

},

new Car

{

Model="Charger",

Vendor="Dodge",

Year=2000,

},

new Car

{

Model="Veyron",

Vendor="Bugatti",

Year=2020

},

new Car

{

Model="Malibu",

Vendor="Chevrolet",

Year=2020

},

new Car

{

Model="M5",

Vendor="BMW",

Year=2021,

},

new Car

{

Model="Escalade",

Vendor="Cadillac",

Year=2021

},

new Car

{

Model="Fusion",

Vendor="Ford",

Year=2000

}

};

//var newcars =

// from car in cars

// where car.Year >= 2006

// orderby car.Year ascending

// select car;

//foreach (var car in newcars)

//{

// Console.WriteLine(car);

// Console.WriteLine();

//}

////AUTOMATIC SEARCH

//string search=string.Empty;

//while (true)

//{

// var letter = Console.ReadKey();

// Console.Clear();

// search+=letter.KeyChar;

// search=search.ToLower();

// if(letter.Key == ConsoleKey.Backspace)

// {

// search = "";

// }

// Console.WriteLine(search);

// var selectedCars =

// from car in cars

// where car.Vendor.ToLower().Contains(search) || car.Model.ToLower().Contains(search)

// orderby car.Year descending

// select car;

// foreach (var car in selectedCars)

// {

// Console.WriteLine(car);

// }

// Console.WriteLine("\n\n");

//}

//4.Query

//var sameYears =

// from car in cars

// where car.Year > 1800

// orderby car.Year descending

// group car by car.Year into cargroup

// select cargroup;

//foreach (var item in sameYears)

//{

// Console.WriteLine(item.Key);

// foreach (var car in item)

// {

// Console.WriteLine($"\t\t{car}");

// }

//}

//var sameYears = cars

// .Where(car => car.Year >= 1800)

// .OrderByDescending(car => car.Year)

// .GroupBy(car => car.Year)

// .ToList();

//foreach (var item in sameYears)

//{

// Console.WriteLine(item.Key);

// foreach (var car in item)

// {

// Console.WriteLine($"\t\t{car}");

// }

//}

//Regex r = new Regex(@"^\+994-\d{2,2}\-\d{3,3}\-\d{2,2}\-\d{2,2}");

//string[] str = { "+994-51-584-87-62", "9678967101", "+91-9678-967101", "+91-96789-67101", "+919678967101" };

////Input strings for Match valid mobile number.

//foreach (string s in str)

//{

// if (r.IsMatch(s))

// {

// Console.ForegroundColor = ConsoleColor.Green;

// }

// else

// {

// Console.ForegroundColor = ConsoleColor.Red;

// }

// Console.WriteLine(s);

// Console.ResetColor(); ;

// //The IsMatch method is used to validate a string or

// //to ensure that a string conforms to a particular pattern.

//}

//while (true)

//{

//string emailString = Console.ReadLine();

//bool isEmail = Regex.IsMatch(emailString, @"\A(?:[a-z0-9!#$%&'\*+/=?^\_`{|}~-]+(?:\.[a-z0-9!#$%&'\*+/=?^\_`{|}~-]+)\*@(?:[a-z0-9](?:[a-z0-9-]\*[a-z0-9])?\.)+[a-z0-9](?:[a-z0-9-]\*[a-z0-9])?)\Z", RegexOptions.IgnoreCase);

// if (isEmail)

// {

// Console.ForegroundColor = ConsoleColor.Green;

// Console.WriteLine("OKAY");

// }

// else

// {

// Console.ForegroundColor = ConsoleColor.Red;

// Console.WriteLine("ERROR");

// }

//}

}

}

}