

1. Considere a seguinte lista de frases (*strings*):

```
string[] Lista = {"C#", "Aprender C#", "ASP.NET Core com C#", "Entity Framework", "Bootstrap", "Identity", "Lambda", "Delegates", "Linq", "POO com C#"};
```

Para se obter uma lista das frases que contêm o texto "C#" pode-se proceder de três formas:

- solução sem usar o LINQ;
- solução através do LINQ com base em *query syntax*;
- solução através do LINQ com base em *extension methods*:

```
var ListA = new List<string>();  
foreach (var x in Lista) if (x.Contains("C#")) ListA.Add(x);  
  
var ListB = from s in Lista where s.Contains("C#") select s;  
  
var ListC = Lista.Where(l => l.Contains("C#"));
```

2. Considere as seguintes definições:

```
string[] ListaUm = { "C#", "Aprender C#", "ASP.NET Core com C#",  
"Entity Framework", "Bootstrap", "Identity", "Lambda", "Delegates",  
"Linq", "POO com C# "};  
  
string[] ListaDois = { "C#", "ASP.NET Core", "Linq", "Lambda e C# "};  
  
int[] Numeros = { 10, 23, 54, 77, 85, 12, 1, 4, 53 };
```

Implemente, com recurso ao LINQ (com *query syntax* ou *extension methods*), as seguintes alíneas:

- Obtenha a lista de todos os elementos da **ListaUm**, ordenados por ordem alfabética.
- Obtenha a lista de todos os elementos da **ListaUm** que tenham menos de seis caracteres.
- Apresente o número total de elementos da **ListaUm** que incluem o texto "C#".
- Construa uma lista em que cada elemento da lista representa o número de palavras de cada um dos elementos da **ListaUm**. Considere que o carácter espaço é o separador entre as palavras e que não existe mais do que um espaço entre as palavras.
- Calcule a média dos elementos de **Numeros**.

- f) Obtenha o valor máximo existente no conjunto **Numeros**.
- g) Obtenha o conjunto de elementos de **Numeros** que pertençam ao intervalo [1, 25], ordenados por ordem decrescente.
- h) Obtenha o conjunto definido pelos elementos comuns às duas listas, **ListaUm** e **ListaDois**. O conjunto definido é a interseção da **ListaUm** com a **ListaDois**.
- i) Obtenha o conjunto definido por todos os elementos das listas **ListaUm** e **ListaDois** e no qual não devem existir elementos repetidos. O conjunto definido é a reunião da **ListaUm** com a **ListaDois**.
- j) Agrupe os elementos existentes em **Numeros** em grupo de números pares e ímpares (Figura 1).

```

Numeros Pares: 10 54 12 4
Numeros Impares: 23 77 85 1 53

```

Figura 1

- k) Obtenha o produto de todos os números (pertencentes a **Numeros**) inferiores a trinta.
- l) Obtenha e apresente a lista de todos os elementos que tenham a palavra C# e, para cada elemento, apresente a primeira e a última palavra (Figura 2). Se no elemento existir apenas uma palavra (como no primeiro caso de Figura 2) deve mostrar “NÃO TEM” na “última palavra”. Considere o seguinte ciclo *foreach* para apresentar o resultado da instrução a implementar:

```

foreach (var s in solucao)
    Console.WriteLine("Frase: " + s.frase + "\n\tPrimeira Palavra: " +
        s.primeira + "\n\tUltima Palavra: " + s.ultima);

```

```

Microsoft Visual Studio Debug Console
Frase: C#
Primeira Palavra: C#
Ultima Palavra: NAO TEM
Frase: Aprender C#
Primeira Palavra: Aprender
Ultima Palavra: C#
Frase: ASP.NET Core com C#
Primeira Palavra: ASP.NET
Ultima Palavra: C#
Frase: POO com C#
Primeira Palavra: POO
Ultima Palavra: C#

```

Figura 2

3. Considere o programa para gerir as tarefas de um utilizador desenvolvido no primeiro exercício da Ficha 1. Na classe “**Utilizador**”:

- Defina um método que apresente as tarefas por ordem (crescente ou decrescente) do nome, da prioridade, da categoria, do estado, da data-limite ou da data de registo da tarefa.
- Defina um método que apresente as tarefas por ordem crescente da data-limite e, para cada conjunto de tarefas com a mesma data-limite, pela ordem de prioridade (alta, normal, baixa) da tarefa (Figura 3).

NOME	PRIORIDADE	CATEGORIA	ESTADO	DATA REGISTO	DATA LIMITE
Baixa	baixa	trabalho	executar	10/17/2021 4:07:28 PM	10/18/2021 4:07:28 PM
Exame PWEB	normal	trabalho	executar	10/17/2021 4:07:28 PM	2/9/2021 9:30:00 AM
Recurso POO	normal	trabalho	executar	10/17/2021 4:07:28 PM	10/18/2021 4:07:28 PM
Tarefa	alta	trabalho	executar	10/17/2021 4:07:28 PM	10/18/2021 4:07:28 PM
Compras	alta	peessoal	executar	10/17/2021 4:07:28 PM	10/18/2021 4:07:28 PM
Reuniao	alta	trabalho	executar	10/17/2021 4:07:28 PM	10/18/2021 4:07:28 PM
Férias	alta	trabalho	concluida	10/17/2021 4:07:28 PM	1/8/2021 5:30:00 PM
Concluida	alta	trabalho	concluida	10/17/2021 4:07:28 PM	10/18/2021 4:07:28 PM
Pessoal	alta	peessoal	executar	10/17/2021 4:07:28 PM	10/18/2021 4:07:28 PM

Figura 3