

# C# Exercises - Lists (Beginner to Junior Level)

1. Crie uma List e adicione 5 números informados pelo usuário. Exiba todos os números usando foreach.
2. Dada uma List com alguns nomes, imprima o primeiro e o último elemento usando indexação.
3. Crie uma função que receba uma List e retorne a soma de todos os elementos. (Sem usar LINQ)
4. Dada uma List, retorne o maior número da lista. Não use LINQ. Faça manualmente.
5. Dada uma List, retorne a quantidade de elementos. Use a propriedade da lista (não faça contador manual).
6. Crie uma função que receba uma List e retorne uma nova lista apenas com números pares.
7. Dada uma List, remova todos os elementos que tenham menos de 4 caracteres.
8. Verifique se um determinado valor existe dentro de uma List. Use método da linguagem apropriado.
9. Inverter manualmente uma List sem usar Reverse(). Crie uma nova lista com os valores invertidos.
10. Dada uma List, retorne uma nova lista com os valores duplicados (ex: [1,2] -> [1,1,2,2]).
11. Conte quantas vezes um número específico aparece em uma List. Não use LINQ.
12. Ordene uma List em ordem crescente sem usar Sort(). Implemente um algoritmo simples (ex: Bubble Sort).
13. Dada uma List, retorne apenas as strings que começam com determinada letra. Use StartsWith().
14. Crie uma função que receba duas List e retorne uma nova lista contendo apenas os elementos em comum.
15. Remova elementos duplicados de uma List sem usar Distinct().
16. Dada uma List, encontre o segundo maior número. Não use LINQ.
17. Dada uma List, ordene por tamanho da string (menor para maior).

18. Implemente uma função que simule uma fila (Queue) usando List. Métodos: Enqueue, Dequeue, Peek.
19. Implemente uma função que simule uma pilha (Stack) usando List. Métodos: Push, Pop, Peek.
20. Desafio final: Dada uma List, retorne a maior sequência contínua de números crescentes. Ex: [1,2,3,1,2] -> [1,2,3]. Dica: pesquise sobre 'sliding window'.