

Prova Pratica - Exercícios de Strings em C#

1. Retorne o tamanho de uma string.

Requisito: Use a propriedade Length da linguagem.

Exemplo: "hello" -> 5

2. Conte quantas vogais existem em uma string.

Requisito: Fazer manualmente percorrendo caractere por caractere.

Exemplo: "casa" -> 2

3. Inverta uma string.

Requisito: Nao pode usar metodos prontos como Reverse().

Exemplo: "abc" -> "cba"

4. Verifique se uma string e palindromo.

Requisito: Resolver usando a logica do exercicio anterior.

Exemplo: "ovo" -> true

5. Conte quantas vezes um caractere especifico aparece.

Requisito: Percorrer usando foreach.

Exemplo: ('a', "banana") -> 3

6. Remova todos os espacos de uma string.

Requisito: Fazer manualmente sem usar Replace().

7. Converta a primeira letra de cada palavra para maiuscula.

Requisito: Pode usar Split() e manipulacao de array.

8. Verifique se duas strings sao anagramas.

Requisito: Nao pode usar bibliotecas externas. Dica: Ordene os caracteres.

9. Retorne a palavra mais longa de uma frase.

Requisito: Usar Split() e percorrer array.

10. Conte quantas palavras existem em uma frase.

Requisito: Considerar espacos extras.

11. Implemente um metodo que compacte uma string.

Exemplo: "aaabbc" -> "a3b2c1"

Dica: Controle repeticoes consecutivas.

12. Verifique se uma string contem apenas numeros.

Requisito: Nao pode usar int.TryParse().

13. Crie um metodo que substitui todas as ocorrencias de uma substring sem usar Replace().

14. Retorne o segundo maior caractere (em valor ASCII) de uma string.

Dica: Percorra comparando valores.

15. Dada uma string, retorne o caractere que mais aparece.

Requisito: Use Dictionary<char, int>.

16. Implemente um metodo que verifica se duas strings sao rotacoes uma da outra.

Exemplo: "abcde" e "cdeab" -> true

Dica: Concatenacao pode ajudar.

17. Dada uma string, retorne todas as substrings possiveis.

Dica: Use dois loops aninhados.

18. Implemente uma função que valida se parenteses estão平衡ados.
Exemplo: "()" -> true
Requisito: Use Stack<char>.

19. Dada uma string, retorne o menor trecho que contém todos os caracteres únicos dela.
Dica: Sliding window.

20. Implemente um algoritmo que encontre a maior substring sem caracteres repetidos.
Nível junior. Dica: Sliding window com HashSet.