

Exercícios de Fixação: Manipulação de Strings [github.com/ermogenes/aulas-programacao-csharp]

Nome: _____ Turma: _____ Data: _____

Observação: `␣` representa um caractere *espaço em branco* (ASCII 32). Use quando não ficar claro que há um espaço.

[A] Forme strings com os conteúdos indicados.

#	Conteúdo	String
1	O nome do professor	<code>"Ermogenes Palacio"</code>
2	Seu nome	
3	Sua cor preferida	
4	O título de um livro	
5	O nome de uma música	

[B] Exiba uma string contendo o texto `Ⓞ` no console.

#	Ⓞ	Comando
1	Olá, mundo!	<code>Console.WriteLine("Olá, mundo!");</code>
2	Lobo	
3	Homem de Ferro	
4	Biscoito	
5	Cozinheira	

[C] Armazene uma string contendo o texto `💣` em uma nova variável `🍷`.

1) `💣` = programação, `🍷` = hobby
`string hobby = "programação";`

2) `💣` = programação, `🍷` = hobby

3) `💣` = Castelo, `🍷` = tipoDeEdificacao

4) `💣` = sorvete, `🍷` = sobremesa

5) `💣` = Girafa, `🍷` = animal

[D] Altere o valor da variável string preexistente `👤` para uma string contendo o texto `☠️`.

1) `👤` = nome, `☠️` = Ermogenes
`nome = "Ermogenes";`

2) `👤` = comida, `☠️` = melancia

3) 🗿 = objeto, 🚫 = Sino

4) 🗿 = direcao, 🚫 = acima

5) 🗿 = tipo, 🚫 = criança

[E] Use sequências de escape para formar strings contendo o texto 🌻.

1) 🌻 = programação é "fácil" de aprender
"programação é \"fácil\" de aprender"

2) 🌻 = Era uma casa muito "engraçada".

3) 🌻 = Adoro as músicas do Guns N' Roses.

4) 🌻 = It's a wonderful world...

5) 🌻 = Posso sair para tomar um copo d'água?

6) 🌻 =
- Oi! Tudo bem?
- Sim, e você?
"- Oi! Tudo bem?\n- Sim, e você?"

7) 🌻 =
Compilar é fácil
Até chegar no Syntax Error
Aí é um pesadelo

8) 🌻 =
Escrevi um programa
Mas ele não funciona bem
Devo ser poeta

9) 🌻 =
C# é a linguagem
Que escolhi pra aprender
Desafio constante.

10) 🌻 =

Laços e condições
São fundamentais
No mundo da programação

[F] Use o identificador textual para formar strings contendo o texto 🍃.

1) 🍃 =

Aprender C# é um desafio
Mas com dedicação, você sai do frio
Variáveis, métodos e classes
São a base para quem avança nas fases

*@"Aprender C# é um desafio
Mas com dedicação, você sai do frio
Variáveis, métodos e classes
São a base para quem avança nas fases"*

2) 🍃 =

Escrever código é quase poesia
Cada linha é uma nova fantasia
Com C# a imaginação é o limite
Aprender é só começar e persistir

3) 🍃 =

No universo da programação
C# é a linguagem em ascensão
Aprender é como um grande jogo
Com recompensas em forma de código

4) 🍃 =

Declarar variáveis é só o começo
Em C#, a lógica é o apreço
Condições, loops e funções
São o caminho para novas soluções

5) 🍌 =

Programar é uma arte
Que não é feita só com um clique
Em C#, cada passo é importante
Para fazer código de qualidade, sem deslize.

[G] Forme a string indicada por concatenação, conforme o exemplo.

Usando o operador +, forme "🍌🍌".

1) 🍌 = para, 🍌 = quedas

"para" + "quedas"

2) 🍌 = manda 🍌 = chuva

3) 🍌 = passa 🍌 = tempo

4) 🍌 = ponta 🍌 = pé

5) 🍌 = vai 🍌 = vêm

Usando o operador +, forme "🍌-🍌".

6) 🍌 = saca, 🍌 = rolhas

"saca" + "-" + "rolhas"

7) 🍌 = peixe 🍌 = espada

8) 🍌 = guarda 🍌 = roupa

9) 🍌 = beija 🍌 = flor

10) 🍌 = arco 🍌 = íris

Usando o operador +, forme "🍌🍌🍌".

11) 🍌 = Ermogenes, 🍌 = Palacio

"Ermogenes" + "b" + "Palacio"

12) 🍌 = sete 🍌 = quedas

13) 🍊 = dois 🍌 = pés

14) 🍊 = professor 🍌 = fantasma

15) 🍊 = gaivota 🍌 = acima

Usando o operador +, forme "🍊🍌", sendo 🍊 e 🍌 variáveis string preexistentes.

16) 🍊 = nome, 🍌 = sobrenome
nome + "🍌" + sobrenome

17) 🍊 = giz 🍌 = chocolate

18) 🍊 = tambor 🍌 = espaguete

19) 🍊 = oliva 🍌 = zero

20) 🍊 = telefone 🍌 = cenoura

[H] Usando String.Concat, concatene as strings e as variáveis (representadas por emojis) necessárias para obter o texto indicado.

1) "Seu nome é 📄!", 📄 = nomeCompleto
String.Concat("Seu nome é🍌", nomeCompleto, "!!")

2) "O seu pedido de 🍕 já está em produção.", 🍕 = pedido

3) "O valor total de sua compra é 💰", 💰 = valorAPagar

4) "🍌 e 🍔 são excelentes escolhas.", 🍌 = item1, 🍔 = item2

5) "Opção selecionada: 🗑️", 🗑️ = opcaoUsuario

[I] Usando String.Format, concatene as strings e as variáveis (representadas por emojis) necessárias para obter o texto indicado.

1) "Seu nome completo é 🍰🍌🍰!", 🍰 = nome, 🍰 = sobrenome
String.Format("Seu nome completo é {0} {1}!", nome, sobrenome)

2) "Sua pizza é meia 🍕 e meia 🍕", 🍕 = sabor1, 🍕 = sabor2

3) "O valor total de sua compra é 💰", 💰 = valorAPagar

4) "🍌 e 🍔 são excelentes escolhas.", 🍌 = item1, 🍔 = item2

5) "Opção selecionada: 🗳️", 🗳️ = opcaoUsuario

[J] Usando interpolação, concatene as strings e as variáveis (representadas por emojis) necessárias para obter o texto indicado.

1) "Seu nome completo é 🍰 🏠!", 🍰 = nome, 🏠 = sobrenome
\$"Seu nome completo é {nome} {sobrenome}!"

2) "Sua pizza é meia 🍕 e meia 🍕🍕", 🍕 = sabor1, 🍕🍕 = sabor2

3) "O valor total de sua compra é 💰", 💰 = valorAPagar

4) "🍌 e 🍔 são excelentes escolhas.", 🍌 = item1, 🍔 = item2

5) "Opção selecionada: 🗳️", 🗳️ = opcaoUsuario

[K] Interpole a string 🗨️ com as variáveis indicadas por emojis, seguindo as instruções.

1) 🗨️ = "Linguagem utilizada: 🗣️", 🗣️ = linguagem (alinhado à direita, 10 caracteres)
\$"Linguagem utilizada: {linguagem,10}"

2) 🗨️ = "Real Madrid👤👤👤👤Barcelona", 👤 = golsMandante (alinhado à direita, 3 caracteres), 🗨️ = golsVisitante (alinhado à esquerda, 3 caracteres)

3) 🗨️ = "--🗣️--", 🗣️ = frase (alinhado à direita, 30 caracteres)

4) 🗨️ = "Preço👉", 👉 = precoUnitario (alinhado à direita, 8 caracteres)

5) 🗨️ = "CEP: 📍", 📍 = codigoPostal (alinhado à direita, 15 caracteres)

[L] Escreva os comandos que exibam o *template* preenchido com as variáveis indicadas por emojis.

1) 🧑 = remetente, ✉️ = destinatario, 😊 = motivo

Prezado(a) ✉️,

Gostaria de expressar minha sincera gratidão pelo 😊. Sua contribuição foi de grande valor para mim e para minha empresa.

Agradeço por seu tempo e esforço e espero que possamos continuar a trabalhar juntos no futuro.



```
string carta = $"Prezado(a) {destinatario},";  
carta = carta + $"\n\nGostaria de expressar minha sincera gratidão pelo {motivo}.";   
carta = carta + " Sua contribuição foi de grande valor para mim e para minha empresa.";   
carta = carta + "\n\nAgradeço por seu tempo e esforço e espero que ";   
carta = carta + $"possamos continuar a trabalhar juntos no futuro.\n\n{remetente}";   
Console.WriteLine(carta);
```

2) 🧑 = remetente, ✉️ = destinatario, 📄 = numeroFatura

Prezado(a) ✉️,

Escrevo para lembrar que o pagamento da fatura nº 📄 ainda não foi recebido, apesar de várias tentativas de contato.

Gostaria de lembrar que os pagamentos em atraso estão sujeitos a juros e multas, e que a falta de pagamento pode resultar em medidas legais.

Espero que possamos resolver essa situação o mais rápido possível. Se houver algum problema ou dificuldade em relação ao pagamento, por favor, entre em contato comigo imediatamente.

Atenciosamente,



3) 📧 = remetente, ✉️ = destinatário, 💬 = assunto, 🧠 = ocorrência

Prezado(a) ✉️,

Escrevo para obter esclarecimentos sobre 💬. Recentemente, 🧠.

Gostaria de saber mais informações sobre o assunto e se há alguma medida que possamos tomar para resolver a situação.

Agradeço antecipadamente por sua atenção e espero sua resposta o mais rápido possível.

Atenciosamente,



[M] Usando strings literais textuais (@""), forme as strings necessárias para exibir os textos indicados. Guarde na variável 📦 e a exiba.

1) 📦 = verbete

Programador (substantivo masculino)

A pessoa que resolve um problema que não sabes que existe, de uma forma que tu não consegues entender.

```
string verbete = @"Programador (substantivo masculino)
```

```
A pessoa que resolve um problema que não sabes que existe, de uma forma que tu não consegues entender.";
```

```
Console.WriteLine(verbete);
```

2) 📦 = verbete

Programador (substantivo masculino)

Uma máquina que transforma café em código.

3) 📦 = fato

Quando eu escrevi esse código, só Deus e eu sabíamos o que ele fazia.

Agora...

Só Deus sabe.

4) 📦 = piadinha

Existem apenas 10 tipos de pessoas no mundo:
aquelas que entendem binário e aquelas que não.

5) 📦 = piadinha

Programar depende apenas de beleza e paciência.
Se funcionar, beleza. Senão, paciência.

[N] Forme strings que contenham o caminho (*path*) indicado.

1) c:\etec\aulapc1

`"c:\etec\aulapc1"`

2) C:\Windows\System32\calc.exe

3) D:\Documentos\Pessoais\Fotos\Férias2022\Havaí\praia.jpg

4) G:\Downloads\ArquivosCompartilhados\Documentação\ManualdoUsuário.pdf

5) C:\Projetos\Cliente

6) c:\etec\aulapc1

`@"c:\etec\aulapc1"`

7) C:\Windows\System32\calc.exe

8) D:\Documentos\Pessoais\Fotos\Férias2022\Havaí\praia.jpg

9) G:\Downloads\ArquivosCompartilhados\Documentação\ManualdoUsuário.pdf

10) C:\Projetos\Cliente

[M] Siga os exemplos de uso dos métodos de System.String. Considere a existência das seguintes variáveis:

```
string nome = "Ermogenes";  
string sobrenome = "Palacio";  
string groselha = "açsfdKJ FNBl dsfo";  
string vogais = "aeiou";
```

```
string escola = "Etec";  
string frase = "O rato roeu a roupa do rei de Roma";  
string bagunca = "1234567890abcdefghijklmnopqrstuvwxyz";
```

#	Comando	String
1	nome.ToUpper()	"ERMOGENES"
2	vogais.ToUpper()	
3	escola.ToLower()	
4	groselha.Insert(5, escola)	"açsfdEteckJ FNBI dsfo"
5	vogais.Insert(2, "kkk")	
6	nome.Insert(4, vogais)	
7	"123".PadLeft(10, '0')	"0000000123"
8	nome.PadLeft(15, 'b')	
9	"123".PadRight(10, 'b')	"123bbbbbb"
10	escola.PadRight(20, '.')	
11	"12345678".Remove(3)	"123"
12	frase.Remove(6)	
13	"12345678".Remove(4, 3)	"12348"
14	"desincompatibilização".Remove(3, 2)	
15	nome.Replace("e", "3")	"Ermog3n3s"
16	"etec a. berezin".Replace("e", "*")	
17	"Concerto".Replace("c", "s")	
18	frase.Replace("a roupa", "o queijo")	"O rato roeu o queijo do rei de Roma"
19	groselha.Replace("J FNBI", "#")	
20	frase.Replace("de Roma", "")	"O rato roeu a roupa do rei"
21	sobrenome.Replace("a", "")	
22	"banana".("a", "4").Replace("n", "N")	"b4N4N4"
23	vogais.Replace("i", "\$").Replace("u", "+")	
24	frase.Substring(12, 14)	"a roupa do rei"
25	groselha.Substring(4, 8)	
26	escola.Substring(2)	"Et"
27	"multidisciplinaridade".Substring(5)	
28	bagunca.Trim()	"abc123xyz"
29	"Mongaguá".Trim()	
30	bagunca.TrimEnd()	"abc123xyz"
31	"Eu amo programar.".TrimEnd()	
32	bagunca.TrimStart()	"abc123xyz"
33	"Eu amo programar.".TrimStart()	