



PUC Minas

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Laboratório de Algoritmos e Técnicas de Programação

Observações:

- Cópias serão desconsideradas, ou seja, a nota será igual a 0 (zero).
- Implemente os programas utilizando a linguagem C#.
- Na resolução dos exercícios só podem ser utilizados comandos vistos nas aulas.

Em cada exercício que for pedido a criação de um método é necessário implementar o método Main e apresentar, pelo menos, uma chamada do método criado para demonstração.

Lista de Exercícios 08 - String

01. Escreva um programa que leia uma string e um caractere, e retorne o número de vezes que esse caractere aparece na string.
02. Escreva um programa que leia duas strings. Se as strings forem iguais, escreva “strings iguais”. Caso contrário, imprima a string que possui o maior tamanho e, em seguida, a string resultante da concatenação das duas strings.
03. Escreva um método que receba como parâmetro uma string e retorne o número de vogais e de consoantes dessa string. Considere que a string lida terá apenas letras (Dica: consoantes não são vogais).
04. Escreva um programa que leia uma palavra e a imprima de trás-para-frente.
05. Escreva um método que receba uma string e indique se ela é um palíndromo. Seu método deve retornar um booleano. Um palíndromo é uma string que tem a propriedade de poder ser lida tanto da direita para a esquerda, como da esquerda para a direita. Ex: ovo, arara, osso.
06. Faça um programa para criptografar uma frase informada pelo usuário. A criptografia deverá trocar as vogais da frase por *

Exemplo:

Frase: EU ESTOU NA ESCOLA

Saída: ** *ST** N* *SC*L*

07. Faça um método que receba uma frase como parâmetro, calcule e retorne a quantidade de palavras da frase (considere que a string lida não terá sinais de pontuação).

Exemplos:

Frase 1: Nas quintas há aula teórica de Algoritmos e Técnicas de Programação

Número de palavras: 11

Frase 2: A lista de exercícios é sobre strings

Número de palavras: 7

08. Em um verbo regular, o radical é a parte invariante, que dá origem a todas as conjugações. Faça um programa que leia um verbo regular terminado em “AR”, e imprima a conjugação do verbo nos seguintes tempos verbais:
 - a) Presente do indicativo;
 - b) Pretérito perfeito do indicativo;
 - c) Futuro do presente do indicativo.

Se o verbo não for terminado em “AR” imprima uma mensagem de erro.
 O programa deverá ler um verbo, processar e imprimir a sua conjugação (ou a mensagem de erro), depois deverá verificar se o usuário deseja informar um novo verbo. Esse processo deverá ser repetido até que o usuário informe “**não**”.
 Dica: armazene as terminações em vetores.

Exemplo:
 Informe um verbo terminado em AR:
Falar

Indicativo		
Presente	Pretérito Perfeito	Futuro do Presente
eu falo	eu falei	eu falarei
tu falas	tu falaste	tu falarás
ele fala	ele falou	ele falará
nós falamos	nós falamos	nós falaremos
vós falais	vós falastes	vós falareis
eles falam	eles falaram	eles falarão

09. Escreva um método que receba o nome completo de um estudante e retorne o seu e-mail institucional. Na Universidade um e-mail contém apenas letras minúsculas e a regra para geração do e-mail é: nome.sobrenome@acme.br

Exemplos:

Nome: Maria Sousa Silva	Nome: Pedro José Silva Pereira
E-mail: maria.silva@acme.br	E-mail: pedro.pereira@acme.br

10. Escreva um programa que leia uma frase A com pelo menos 50 caracteres e uma string B contendo exatamente 2 caracteres. Imprima o número de vezes que a string B aparece na string A.

11. Escreva um método que receba uma string contendo dados em formato CSV (valores separados por vírgula) e retorne uma matriz com os dados.

Exemplo:

```
string csv = "nome,idade,sexo\nAna,25,F\nJoão,30,M\nMaria,22,F";
```

Exemplo de matriz gerada após a conversão e que deverá ser retornada:

```
["nome" "idade" "sexo"
 "Ana" "25" "F"
 "João" "30" "M"
 "Maria" "22" "F"]
```