

Análise e Desenvolvimento de Sistemas - ADS **Linguagem de Programação - LP**

LISTA DE EXERCÍCIOS

Prof. Cristóvão Cunha

Assunto: Strings, mas envolve laços de repetição e casting.

Esta lista contém 30 exercícios que devem ser entregues ao professor, resolvidos em linguagem de programação Java, dentro de uma pasta com o Seu Nome Completo, não sendo aceito o envio pela Internet. Dentro de cada exercício (arquivo fonte) deve haver um comentário com o Seu Nome Completo. Estes exercícios devem ser feitos em casa ou na monitoria. Para facilitar a resolução desses exercícios, é de suma importância o estudo e entendimento da tabela ASCII. As referências estão mencionadas no exercício 1.

1) Exibir na tela os caracteres da tabela de códigos ASCII (American Standard Code for Information Interchange).

Referências: - Wikipédia: <http://pt.wikipedia.org/wiki/ASCII>

- Arquivo TabelaASCII.pdf na pasta de Apostilas do Material didático de LP.

2) Declarar uma variável do tipo String com seu valor inicial igual a branco ("") e a seguir atribuir as letras **maiúsculas** do alfabeto (ABCDE...Z), uma-a-uma, as primeiras vinte e seis posições da variável String. Imprimir o resultado.

3) Declarar uma variável do tipo String com seu valor inicial igual a branco ("") e a seguir atribuir as letras **minúsculas** do alfabeto (abcde...z), uma-a-uma, as primeiras vinte e seis posições da variável String. Imprimir o resultado.

4) Declarar uma variável do tipo String com seu valor inicial igual a branco ("") e a seguir atribuir os **caracteres numéricos** (0123456789), um-a-um, as primeiras dez posições da variável String. Imprimir o resultado.

5) Escreva um programa que calcule e escreva a quantidade de vezes que o **caractere 'a'** aparece em uma frase fornecida pelo usuário. Nota Importante: para a contagem, o caractere pode ser maiúsculo ou minúsculo.

6) Escreva um programa que calcule e escreva a quantidade de vezes que um **caractere fornecido pelo usuário** aparece em uma frase, também, fornecida pelo mesmo. Nota Importante: para a contagem, o caractere pode ser maiúsculo ou minúsculo.

7) Escreva um programa para ler uma frase fornecida pelo usuário e contar a quantidade de cada caractere do alfabeto, imprimindo ao final uma listagem, onde consta o caractere e na frente a quantidade de ocorrências. Caso não tenha ocorrência apresentar o valor zero. Não diferenciar maiúsculo de minúsculo.

- 8) Escreva um programa para ler uma frase fornecida pelo usuário e contar a quantidade de cada caractere do alfabeto, imprimindo ao final uma listagem onde consta o caractere e na frente a quantidade de ocorrências. Caso a quantidade de ocorrências de um caractere seja zero, não apresentar o caractere na listagem. Não diferenciar maiúsculo de minúsculo.
- 9) Fazer um programa para contar quantos espaços em branco existem em uma frase fornecida pelo usuário.
- 10) Escreva um programa que calcule e escreva a quantidade de letras maiúsculas de uma frase fornecida pelo usuário.
- 11) Escreva um programa que conte e exiba, separadamente, a quantidade de caracteres alfabéticos e de dígitos numéricos existentes em uma frase fornecida pelo usuário.
- 12) Escreva um programa que conta e exibe a quantidade de vezes que cada vogal ocorre em uma frase fornecida pelo usuário.
- 13) Escreva um programa que imprima o número de consoantes de uma frase fornecida pelo usuário.
- 14) Escreva um programa para ler uma frase e um caractere. Sempre que o caractere lido aparecer na frase ele deve ser substituído por asterisco. Por exemplo, se os valores fornecidos fossem para a frase: **o dia esta nublado**, e para o caractere: **a**, o programa deverá fornecer o seguinte resultado: **o di* est* nubl*do**.
- 15) Escreva um programa que informe se em uma frase fornecida pelo usuário existem caracteres numéricos.
- 16) Escreva um programa que informe se todos os caracteres alfabéticos de uma frase fornecida pelo usuário são maiúsculos.
- 17) Escreva um programa que leia duas frases e a seguir informe se as frases fornecidas são iguais. *Observação:* realizar esta verificação caractere por caractere.
- 18) Desenvolva um programa que leia uma cadeia de caracteres e através de uma mensagem indique se esta cadeia representa uma sequência binária de 8 bit's ou não.
Ex:
- não é uma sequência de 8bit's: 01101, 10192898, 10001jaA
- é uma sequência de 8bit's: 01101011, 10110101, 10000000
- 19) Escreva um programa que leia uma cadeia de caracteres qualquer e determine se a palavra lida representa um palíndromo ou não (exemplo de palavras palíndromas: ovo, natan, sos, arara, etc).
- 20) Escreva um programa que receba uma frase, conte e imprima o número de **palavras** desta frase.
- 21) Escreva um programa que receba uma frase, imprima cada palavras desta frase em uma linha.
- 22) Dados uma frase e uma palavra, escreva um programa que verifique o número de vezes que a palavra ocorre na frase. Exemplo: Na frase **ANA E MARIANA GOSTAM DE BANANA**, a palavra **ANA** ocorre 4 vezes.

23) Elabore um programa para obter o nome completo de uma pessoa e a seguir forneça o sobrenome do nome completo informado. Por exemplo, para o nome completo: **Omero Francisco Bertol**, o programa deverá fornecer como resultado: **Bertol**.

24) Desenvolva um programa que leia uma String (que representa uma sequência de dígitos **0's** e **1's**) e que determine o tamanho da maior sequência consecutiva de números **1** existente na String. No exemplo seguinte a maior cadeia consecutiva de números **1** tem tamanho 8 (oito).

001010100110110100001011101111111101101001011000

25) Escreva um programa validador de senhas. A regra para a criação de uma senha forte é que a senha deve ser de tamanho mínimo 10. A senha deve conter letras maiúsculas, letras minúsculas, números e caracteres especiais/símbolos (!@#\$%&*()+). Outra regra, deve ter pelo menos 3 caracteres (maiúsculo ou minúsculo), pelo menos 3 números e pelo menos 2 símbolos. O programa deve solicitar uma senha forte ao usuário até ele conseguir uma senha que respeite esses critérios. Após cada senha digitada, apresentar as regras e se a senha passou pelos quesitos ou não.

Ex:

Senha: senha123

Regras

Tamanho 10: 8 (x)

Letras 3: 5 (ok)

Números 3: 3 (ok)

Símbolos 2: 0 (x)

Senha reprovada! Tente novamente.

Senha: Bgp2lp4*@1

Regras

Tamanho 10: 10 (ok)

Letras 3: 5 (ok)

Números 3: 3 (ok)

Símbolos 2: 2 (ok)

Senha aprovada! Parabéns!

Desafios!!!

26) Escreva um programa para gerar aleatoriamente uma String com 100 letras minúsculas. Apresentar, caso haja:

- as letras do alfabeto que NÃO constam na sequência;
- a maior sequência de letras repetidas;
- a maior sequência de vogais; e
- a maior sequência alfabética.

27) Escreva um programa para solicitar ao usuário duas Strings. Se forem de mesmo tamanho, concatenar seus caracteres, caso contrário, informar ao usuário.

Ex:

Entrada 1: banana

Entrada 2: uva

Saída: Erro! Entradas de tamanhos diferentes.

Entrada 1: bananas

Entrada 2: pitanga

Saída: bpaintaannagsa

28) Escreva um programa que leia uma String e “dobre” ela ao meio concatenado seus caracteres.

Ex:

Entrada: Macaco Caco quer banana (tamanho ímpar)

Saída: Maancaancaob rCeaucqo

Entrada: Batatinha quando nasce (tamanho par)

Saída: Beactsaatniondhnaa uq

Criptografia (em grego: kryptós, "escondido", e gráphein, "escrita") é o estudo dos princípios e técnicas pelas quais a informação pode ser transformada da sua forma original para outra ilegível, de forma que possa ser conhecida apenas por seu destinatário (detentor da "chave secreta"), o que a torna difícil de ser lida por alguém não autorizado. Assim sendo, só o receptor da mensagem pode ler a informação com facilidade. É um ramo da Matemática, parte da Criptologia. Há dois tipos de chaves criptográficas: chaves simétricas (criptografia de chave única) e chaves assimétricas (criptografia de chave pública).

Uma informação não-cifrada que é enviada de uma pessoa (ou organização) para outra é chamada de "texto claro" (plaintext). Cifragem é o processo de conversão de um texto claro para um código cifrado e decifragem é o processo contrário, de recuperar o texto original a partir de um texto cifrado. De fato, o estudo da criptografia cobre bem mais do que apenas cifragem e decifragem. É um ramo especializado da teoria da informação com muitas contribuições de outros campos da matemática e do conhecimento, incluindo autores como Maquiavel, Sun Tzu e Karl von Clausewitz. A criptografia moderna é basicamente formada pelo estudo dos algoritmos criptográficos que podem ser implementados em computadores.

Zenit Polar é um sistema simples de criptografia, que consiste na substituição das letras de uma palavra pela sua correspondente no nome ZENIT POLAR.

1. O **Z** substitui o **P** e vice e versa.
2. O **E** substitui o **O** e vice e versa.
3. O **N** substitui o **L** e vice e versa.
4. O **I** substitui o **A** e vice e versa.
5. O **T** substitui o **R** e vice e versa.

Sendo assim, as seguintes frases:

"Uma leve justiça leva a varias compreensoes" Anonimous

"Todo mundo age nao apenas por compulsao externa, mas tambem por necessidade intima"

Albert Einstein

"Se for para o bem da Nação e felicidade geral, diga ao povo que fico" Dom Pedro I

Ficariam assim:

"Umi novo jusraçi novi i vitais cemztoolseos" ILeLame

"Rede mulde igo lie izolis zet cemzunsie oxrotli, mis rimbom zet locossadido alrami" Inbotr
Oalsroal

"So fet ziti e bom di Licie o fonacadido gotin, dagi ie zeve quo face" Dem Zodte A

As demais letras, que não compõem tais palavras, permanecem inalteradas, e ignoram-se os acentos.

De tão simples, era usado nas escolas pelos alunos para passarem bilhetes ou escreverem em código, sem que os adultos descobrissem.

29) Escreva um programa que apresente um menu com 3 opções:

- 1 - Criptografar
- 2 - Descriptografar
- 3 - Sair

Caso o usuário digite uma opção inválida, uma mensagem de erro deve ser apresentada e novamente, solicitado uma estrada.

Se selecionado a opção 1, criptografar, solicitar a entrada do texto plano e como saída, exibir o resultado do Zenit Polar e retornar ao menu principal.

Se selecionado a opção 2, descriptografar, solicitar a mensagem cifrada com Zenit Polar e como saída, apresentar a mensagem original e retornar ao menu principal.

Por último, se selecionado a opção 3, finalizar o programa.

ROT-13 (ou rot13, rot-13, Rot13, etc) é o nome que se costuma usar para um procedimento simples mas eficaz para garantir que textos eletrônicos não sejam lidos por distração ou acidente. ROT-13 vem do inglês, ROTate by 13 places, "ROTacionar 13 posições".

Especificamente, a cifra ROT-13 tem se mostrado útil principalmente para proteger endereços de correio eletrônico (evitando SPAM, o envio de mensagens não solicitadas) e para "proteger" mensagens que o remetente pode preferir não.

Como era de se esperar, o ROT-13 é usado principalmente em grupos de discussão (como na usenet) e em listas de discussão como as dos Yahoo.

Um detalhe importante: não há qualquer diferença entre o procedimento para codificar um texto em ROT-13 e o procedimento para decodificá-lo; simplesmente aplicamos o mesmo procedimento uma segunda vez.

30) Da mesma forma que o exercício anterior, implemente um programa para criptografar e descriptografar utilizando a cifra ROT-13.