

Lista4 de Exercícios – Manipulação de strings

Utilizando Linguagem C, resolva os problemas abaixo:

- 1) Solicitar uma palavra e apresentar a quantidade de caracteres na mesma;
- 2) Solicite uma palavra e apresente os 4 primeiros caracteres;
- 3) Solicitar uma palavra, copiar e apresentar apenas 3 caracteres a partir do segundo caractere;
- 4) Solicite uma palavra com no mínimo 5 caracteres e apresente apenas os 3 últimos;
- 5) Solicitar uma palavra e apresentá-la na vertical (cada caractere será uma linha);
- 6) Solicitar duas palavras, concatena-las em uma terceira, apresentando-a no final;
- 7) Solicitar uma palavra e apresentá-la toda em MAIUSCULA;
- 8) Solicitar uma palavra e apresentá-la toda em MINUSCULA;
- 9) Solicitar uma palavra e apresentá-la em ordem inversa. (ex. AMOR = ROMA);
- 10) Solicitar uma palavra (5 a 10 caracteres) e apresentar cada caractere alternando em MAIUSCULO e MINUSCULO cada um dos caracteres;
- 11) Solicitar uma palavra (4 a 15 caracteres) e apresentar o código ASCII de cada caractere;
- 12) Solicitar uma palavra (3 a 20 caracteres) e apresentar cada um dos caracteres adicionando o valor 10 ao código ASCII;
- 13) Solicitar uma palavra (4 a 12 caracteres) e apresentar cada um dos caracteres adicionando o valor 20 ao código ASCII e também a posição do caractere na palavra;
- 14) Solicite uma *string* e apresente a frase contendo apenas o primeiro e o último caractere em maiúsculo, os demais deverão ser apresentados em minúsculo.
- 15) Solicite duas palavras e apresente se elas fazem parte de um ANAGRAMA
- 16) Solicite uma palavra (mínimo 3 caracteres) e apresente se a mesma é um palíndromo:

```

32 33 ! 34 " 35 # 36 $ 37 % 38 & 39 ' 40 ( 41 ) 42 *
43 + 44 , 45 - 46 . 47 / 48 0 49 1 50 2 51 3 52 4 53 5
54 6 55 7 56 8 57 9 58 : 59 ; 60 < 61 = 62 > 63 ? 64 @
65 A 66 B 67 C 68 D 69 E 70 F 71 G 72 H 73 I 74 J 75 K
76 L 77 M 78 N 79 O 80 P 81 Q 82 R 83 S 84 T 85 U 86 V
87 W 88 X 89 Y 90 Z 91 [ 92 \ 93 ] 94 ^ 95 _ 96 ` 97 a
98 b 99 c 100 d 101 e 102 f 103 g 104 h 105 i 106 j 107 k 108 l
109 m 110 n 111 o 112 p 113 q 114 r 115 s 116 t 117 u 118 v 119 w
120 x 121 y 122 z 123 { 124 | 125 } 126 ~ 127 ¨ 128 € 129 ¸ 130 /
131 f 132 ¨ 133 ... 134 † 135 ‡ 136 ^ 137 ¤ 138 § 139 < 140 ¤ 141
142 ¤ 143 ¤ 144 ¤ 145 ¤ 146 ¤ 147 ¤ 148 ¤ 149 ¤ 150 ¤ 151 ¤ 152 ¤
153 ¤ 154 ¤ 155 ¤ 156 ¤ 157 ¤ 158 ¤ 159 ¤ 160 ¤ 161 ¤ 162 ¤ 163 ¤
164 ¤ 165 ¤ 166 ¤ 167 ¤ 168 ¤ 169 ¤ 170 ¤ 171 ¤ 172 ¤ 173 ¤ 174 ¤
175 ¤ 176 ¤ 177 ¤ 178 ¤ 179 ¤ 180 ¤ 181 ¤ 182 ¤ 183 ¤ 184 ¤ 185 ¤
186 ¤ 187 ¤ 188 ¤ 189 ¤ 190 ¤ 191 ¤ 192 ¤ 193 ¤ 194 ¤ 195 ¤ 196 ¤
197 ¤ 198 ¤ 199 ¤ 200 ¤ 201 ¤ 202 ¤ 203 ¤ 204 ¤ 205 ¤ 206 ¤ 207 ¤
208 ¤ 209 ¤ 210 ¤ 211 ¤ 212 ¤ 213 ¤ 214 ¤ 215 ¤ 216 ¤ 217 ¤ 218 ¤
219 ¤ 220 ¤ 221 ¤ 222 ¤ 223 ¤ 224 ¤ 225 ¤ 226 ¤ 227 ¤ 228 ¤ 229 ¤
230 ¤ 231 ¤ 232 ¤ 233 ¤ 234 ¤ 235 ¤ 236 ¤ 237 ¤ 238 ¤ 239 ¤ 240 ¤
241 ¤ 242 ¤ 243 ¤ 244 ¤ 245 ¤ 246 ¤ 247 ¤ 248 ¤ 249 ¤ 250 ¤ 251 ¤
252 ¤ 253 ¤ 254 ¤ 255 ¤

*** Fim da execução.
*** Feche esta janela para retornar ao Visualg.
    
```

- Solicite 4 nomes ao usuário e apresente o menor nome da lista (ordem alfabética);
- 17) Solicite um nome e apresente a quantidade de vogais da mesma;
 - 18) Solicite um nome e apresente a quantidade de vogais (apenas maiúsculas) da mesma;
 - 19) Solicite um nome e apresente a quantidade de caracteres numéricos da mesma;
 - 20) Solicite um nome completo (mínimo 13 caracteres) e apresente apenas o seu sobrenome;
 - 21) Solicite uma frase e informe ao usuário se a palavra “AMOR” está ou não na frase;
 - 22) Construa um algoritmo que apresente cada uma das vogais e seu respectivo código decimal ASCII correspondente (cada vogal será uma), tanto maiúsculo quanto minúsculo. Exemplo: A=65 a=97
 - 23) Faça o mesmo que o exercício anterior, entretanto apresentando apenas as consoantes;
 - 24) Construa um algoritmo que solicite uma frase e apresente a quantidade de espaços da mesma;
 - 25) Solicite uma *string* e informe ao usuário trata-se ou não de palíndromo.
 - 26) Solicite uma frase (mínimo 5 e máximo 50 caracteres) e imagine que possam haver caracteres em branco tanto no início quanto no fim da frase. Caso haja, elimine-os (espaços das extremidades);
 - 27) Construa um algoritmo que apresente todo o alfabeto maiúsculo em uma única linha, utilizando a tabela ASCII e laço de repetição.
 - 28) Com base ao exercício anterior, apresente o alfabeto (minúsculo) em ordem decrescente.
 - 29) Solicite ao usuário duas palavras e concatene-as em uma terceira variável, alternando entre os caracteres da primeira e da segunda palavra. (ex: ANA / CLAUDIO = ACNLAAUDIO);
 - 30) Solicite duas palavras e apresente a quantidade total de vogais das mesmas;

- 31) Solicite uma palavra e apresente o caractere que possui o maior código ASCII, bem como seu valor;
- 32) Faça um algoritmo que sorteie uma cadeia de 10 caracteres maiúsculos e apresente ao usuário. Utilize a função RANDI.
- 33) Solicite uma palavra ao usuário (entre 3 e 15 caracteres) e posteriormente apresente-a em ordem inversa, linha a linha, ou seja, na vertical. Obs. Antes de apresentar, transforme todos os caracteres em maiúsculos;
- 34) Solicite uma palavra (3 a 20 caracteres) e intercale entre cada caracteres, os caracteres “@” e “#”, alternando-os. Ex. MORENO = M@O#R@E#N@O. Apresente o resultado final ao usuário;
- 35) Com base ao exercício anterior, intercale com sequência numérica (entre 1 e 9) para uma palavra (entre 3 e 10 caracteres. Ex: RAQUEL = R1A2Q3U4E5L. Apresente o resultado final ao usuário;
- 36) Solicite 5 números (entre 65 e 119) e apresente seus respectivos caracteres segundo a tabela ASCII;
- 37) Construa um algoritmo que gere uma cadeia de 10 caracteres a partir de sorteios entre 65 e 119, entretanto contendo apenas consoantes não repetidas.
- 38) Solicite 3 nomes e apresente-os em ordem crescente;
- 39) Solicite 3 nomes e apresente-os em ordem decrescente;
- 40) Solicite 6 palavras e apresente a palavra que contiver o maior número de vogais. Caso haja palavras com o mesmo valor de vogais, apresente apenas a primeira;
- 41) Solicite uma *string* qualquer (obrigatoriamente com quantidade par de caracteres) e apresente-a intercalando os caracteres das extremidades para o centro. (ex: AMOR = ARMO);
- 42) Solicite uma *string* qualquer (obrigatoriamente com quantidade par de caracteres) e apresente-a intercalando os caracteres do meio para as extremidades (ex: DANIELLA = IENLALDA);
- 43) Solicite um nome completo (mínimo 10 caracteres) cada um dos nomes em linhas separadas;
- 44) Solicite uma frase (mínimo 10 caracteres), converta-a em maiúsculo e apresente-a posteriormente substituindo alguns caracteres (A – 4, B – 8, E – 3, I – 1, O – 0, S – 5, T – 7 e Z – 2);
- 45) Construa um algoritmo que criptografe uma sequência de caracteres (máximo 8) informado pelo usuário, seguindo as regras que identificar conforme os exemplos abaixo:
 - a. ALEXANDRE MORENO = U_WiQjR_Q+WXZLTT
 - b. ZENAIDE DIAS = jT\NUOO)LPGX
- 46) Construa um algoritmo que execute a engenharia reversa conforme exercício anterior, ou seja, o usuário informará o valor criptografia e o programa deverá apresentar o valor DESCRIPTOGRAFADO;
- 47) Construa um jogo da forca (mínimo 5 e máximo 10 caracteres). As demais regras fica a seu critério, desde que instruído o usuário ao jogar.