

Escola de Ciências e Tecnologia – UFRN
Informática Fundamental
Prof.: Hugo Melo

Lista de exercícios

Vetores – strings

Exercícios de fixação

1. Implemente um programa que recebe o primeiro nome de uma pessoa (com no máximo 20 caracteres) e calcule quantos caracteres o nome dela tem.
2. Implemente um programa que recebe um texto de até 50 caracteres (o texto pode conter espaços, mas deve ser digitado em uma única linha) e uma vogal. O programa deve imprimir quantas vezes e em quais posições aquela vogal aparece no texto.
3. Implemente um programa que recebe um texto qualquer de até 1000 caracteres e imprime quantas palavras o texto possui. O seu programa pode considerar que as palavras serão separadas por um espaço.
4. Implemente um programa que recebe uma palavra e faz a cópia invertida dela. Por exemplo: se foi digitado “exemplo”, seu programa deve criar uma outra string que armazena “olpmexe”, e então imprimí-la.
5. Implemente um programa que recebe um texto e inverte a caixa do texto (maiúsculas são transformadas em minúsculas e vice-versa). Caracteres não-alfabéticos não devem ser afetados.

Exercícios complementares

6. Implemente um programa que recebe duas palavras de no máximo 20 caracteres cada e determina se elas são iguais ou não.
7. Implemente um programa que recebe um texto e determina a porcentagem de ocorrência de cada letra do alfabeto no texto. O programa não deve distinguir minúsculas e maiúsculas.
8. Implemente um programa que recebe o nome completo de uma pessoa e seu sexo (um caractere 'F' ou 'M'). O programa deve imprimir “Bom dia Sra. <ultimo_nome>” ou “Bom dia Sr. <ultimo_nome>”, a depender do sexo. Dica: crie um vetor auxiliar para armazenar o último nome da pessoa.

9. Implemente um programa que recebe um texto e uma palavra. O programa deve determinar se a palavra ocorre no texto ou não.
10. Implemente um programa que recebe um número inteiro positivo e em seguida essa quantidade de palavras (com no máximo 10 caracteres cada). O programa deve concatenar todas as palavras em uma única string.
11. Implemente um programa que verifica se um texto é um palíndromo. Uma palavra ou frase é um palíndromo se ela é a mesma lida normal ou ao contrário. Exemplos: “Radar”, “Reviver” e “A pateta ama ate tapa.”. O programa deve ignorar pontuação, espaços e não deve distinguir minúsculas e maiúsculas.

Exercícios avançados

12. Implemente um programa que recebe duas palavras (*palavra1* e *palavra2*) e um texto. O programa deve substituir todas as ocorrências de *palavra1* por *palavra2* no texto.
13. Implemente o jogo da forca. Recomendações:
 - Use o caractere '_' para representar as letras que ainda não foram encontradas
 - Use 4 como número máximo de erros antes do jogador perder.
 - Apresente a cada rodada as letras que já foram chutadas mas que não pertencem à palavra-chave.
 - Não diferencie maiúsculas de minúsculas.
 - Use o comando “system(“cls”)” (incluir biblioteca windows.h) logo após o primeiro jogador inserir a palavra-chave e a cada nova rodada. Esse comando “reseta” a tela, ocultando tudo o que havia nela antes.
 - Desenhe o boneco enforcado usando os caracteres '_' e '|'. Cada parte do boneco será desenhado a depender do número de erros.
14. Implemente um programa que recebe um texto de até 1000 caracteres em português o converte para o “miguxês”. O texto de saída deve:
 - Substituir todas as ocorrências de 'a' ou 'A' por '4'.
 - Substituir todas as ocorrências de 'u' ou 'U' por 'W'.
 - Substituir todas as ocorrências de 's' ou 'S' por “xX”
 - Substituir todas as ocorrências de 'C' por 'k'.
 - Garantir que pelo menos metade (arredondado para baixo) das letras de cada palavra é maiúscula (caracteres não-alfabéticos não contam)
 - Garantir que nenhuma palavra tenha consoantes minúsculas adjacentes. Essa regra tem prioridade sobre a anterior.