

Disciplina: Técnicas de Programação para Engenharia I	Código: TI0044	Prática: 04
Assunto: Arrays Unidimensionais, Bidimensionais e Strings		

Arrays Unidimensionais e Bidimensionais:

- 1) Dada uma sequência de n (≤ 50) números, imprimi-la na ordem inversa à da leitura.
- 2) Ler um elemento K.
Ler um vetor A de N elementos.
Verificar se o elemento K está presente no vetor. Se estiver, imprimir a posição onde ele foi encontrado. Caso contrário, imprimir mensagem "elemento K não encontrado".
- 3) Ler N ($N \leq 20$) valores inteiros e armazená-los em um vetor. Trocar, a seguir, o 1º elemento com o último, o 2º com o penúltimo, etc., e escrever o vetor assim modificado.
- 4) Ler N valores inteiros ($N \leq 100$) até que seja digitado o valor zero. A seguir, inverter o vetor, trocando o 1º elemento com o último, o 2º com o penúltimo e assim sucessivamente. Ao final, imprimir o vetor invertido.
- 5) Escreva um programa que aceite até 10 números de ponto flutuante digitados pelo usuário. Seu programa deve mostrar a média.
- 6) Ler vinte valores inteiros e armazená-los em um vetor. Trocar, a seguir, os elementos de ordem ímpar com os de ordem par imediatamente seguintes e escrever o vetor modificado.
- 7) Ler dois vetores K e N, ambos compostos de 10 elementos inteiros. Criar, a seguir, um vetor M que seja a diferença entre K e N ($M = K - N$). Finalmente, escrever o vetor M.
- 8) Tentando descobrir se um dado era viciado, um dono de cassino honesto (ha! ha! ha! ha!) o lançou n vezes. Dados os n resultados dos lançamentos, determinar o número de ocorrências de cada face.
- 9) Ler um vetor V de dimensão N ($N \leq 50$). Compactar, a seguir, o vetor V, retirando dele todos os valores nulos ou negativos e escrever o vetor compactado.
- 10) Ler dois vetores A e B com N e M (N e $M \leq 50$) elementos respectivamente. Gerar um novo vetor C, que corresponde à intersecção dos vetores A e B, e um novo vetor D que corresponde à união dos vetores A e B. Imprimir os vetores A, B, C e D. Considerar que não existem elementos repetidos em cada um dos vetores A e B
- 11) Ler um vetor A de dimensão N ($N \leq 50$).
Calcular e imprimir o valor de S, sendo

$$S = \sum_{i=1}^N \frac{i}{a_i}$$
, onde a_i é o i-ésimo valor armazenado na variável A
Calcular e imprimir quantos termos da série têm o numerador inferior ao denominador
- 12) Um armazém trabalha com 100 mercadorias diferentes identificadas pelos números inteiros de 1 a 100. O dono do armazém anota a quantidade de cada mercadoria vendida durante o mês. Ele tem uma tabela que indica para cada mercadoria o preço de venda.
Calcular o faturamento mensal do armazém, isto é

$$\text{FATURAMENTO} = \sum_{i=1}^{100} (\text{QUANTIDADE}(i) * \text{PREÇO}(i))$$

A tabela de preços e de quantidade são fornecidas em dois vetores, sendo que um vetor contém a quantidade vendida e o outro o preço de cada mercadoria.

- 13) Dada uma sequência de n números reais, determinar os números que compõem a sequência e o número de vezes que cada um deles ocorre na mesma.

Exemplo: $n = 8$

Sequência: -1.7, 3.0, 0.0, 1.5, 0.0, -1.7, 2.3, -1.7

Saída: -1.7 ocorre 3 vezes

3.0 ocorre 1 vez

0.0 ocorre 2 vezes

1.5 ocorre 1 vez

2.3 ocorre 1 vez

- 14) Ler um vetor A de N elementos ($N \leq 50$). Determinar e imprimir a maior diferença entre dois elementos consecutivos desse vetor.

- 15) Ler um vetor A de dimensão 20.

Calcular e imprimir o valor de S, onde

$$S = (A_1 - A_{20})^2 + (A_2 - A_{19})^2 + \dots + (A_{10} - A_{11})^2$$

- 16) Dadas duas sequências com n números inteiros entre 0 e 9, interpretadas como dois números inteiros de n algarismos, calcular a sequência de números que representa a soma dos dois inteiros.

Exemplo: $n = 8$,

1ª sequência		8	2	4	3	4	2	5	1
2ª sequência	+	3	3	7	5	2	3	3	7
		1	1	6	1	8	6	5	8

- 17) Ler um vetor V de elementos inteiros de dimensão N ($N \leq 50$). Separar este vetor em dois vetores A e B. O vetor A deve conter os elementos positivos de V e o vetor B os elementos negativos de V. Os elementos nulos de V não devem ser gravados mas devem ser contados. Escrever os vetores V, A e B. Escrever quantos elementos nulos foram encontrados em V.

- 18) Escreva um programa que leia um conjunto de 10 números inteiros não nulos, armazenando cada um deles em uma posição de um vetor. Para cada número digitado o programa deve verificar se o número foi digitado anteriormente. Caso algum valor já tenha sido digitado, substituir o valor anterior por zero. Ao final do processamento, imprima os 10 valores armazenados.

- 19) Ler um vetor V de N elementos. Supor que o vetor V está ordenado em ordem crescente. Calcular e escrever o valor da mediana dos elementos do vetor.

Observação:

para N ímpar, a mediana é definida como o valor central do vetor ordenado.

para N par, a mediana é definida como a média aritmética dos dois elementos centrais do vetor.

Exemplo:

1º Caso

$V = (-3, -1, 10, 12)$

$N = 4$

$$\text{MEDIANA} = \frac{-1 + 10}{2} = 4.5$$

2º Caso

$V = (-1, 0, 1, 4, 9)$

$N = 5$

MEDIANA = 1

20) Dada uma sequência de n números inteiros, determinar um segmento de soma máxima.

Exemplo: Na sequência 5, 2, -2, -7, 3, 14, 10, -3, 9, -6, 4, 1, a soma do segmento é 33.

21) Ler dois vetores (A de dimensão N e B de dimensão M, com N e $M \leq 20$) compostos de números inteiros e intercalar estes números em um terceiro vetor (C) mantendo a ordem crescente dos números.

Escrever os vetores A, B e C.

22) Dado um vetor de tamanho n , procure uma forma de ordená-lo de maneira crescente.

23) Dado um vetor ordenado, procure uma maneira de encontrar um elemento fazendo o menor número de operações (busca binária).

Strings:

1. O código identificador de uma funcionário de um empresa é formado por 8 (oito) caracteres numéricos, o caractere “hífen” e a soma verificadora de caracteres numéricos. A soma verificadora é gerada como segue; soma-se o valor numérico de cada caractere ASCII do código identificador e o resultado numérico desta soma é convertido em uma *string* correspondente. Exemplo:

número identificador: “12345678”

soma dos caracteres ASCII: $49+50+51+52+53+54+55+56 = 420$

conversão: $420 \rightarrow “420”$

resultado final: “12345678-420”

Ler uma *string* de 8 (oito) caracteres numéricos e gerar o código identificador.

DICA: Se desejar, pode utilizar a função itoa() da lib stdio.h.

2. Elaborar um programa para ler e verificar se um dado código identificador nos moldes no exercício 1 está correto ou não, exibindo as mensagens “Código válido”, ou “Código inválido”. No último caso, ler novamente o código até que ele seja um código válido.

DICA: Se desejar, pode utilizar a função atoi() da lib stdio.h.

3. Ler um caractere ASCII e exibi-lo em decimal, hexadecimal, e uma *string* correspondente em binário. Exemplos:

‘A’ \rightarrow 65, 0x41, “01000001”

‘0’ \rightarrow 48, 0x30, “00110000”

ESC \rightarrow 27, 0x1B, “00011011”

4. Ler uma frase (máximo 79+1 caracteres, incluindo espaços em branco) e exibir novamente a frase com cada palavra com seus caracteres invertidos. Exemplo:

“Adoro trabalhar com *strings*” \rightarrow “orodA rahlabart moc sgnirts”

5. Um dos protocolos muito utilizados na delimitação da quantidade de bytes a ser transmitida via rede baseia-se na inserção de caracteres especiais delimitadores STX – *Start of Text* e ETX – *End of Text* (0x02 e 0x03 na tabela ASCII). Porém, como estes caracteres também são bytes, nada impede que eles também estejam contidos na informação a ser transmitida. Resolve-se este problema através da inserção de um outro caractere especial de transparência DLE (0x10) na frente de cada um dos dois acima citados antes da transmissão, cada vez que eles aparecem dentre os bytes originais. Exemplo:

Bytes originais:

0xFF	0xAB	0x21	0x02	0x10	0x34	0x03
------	------	------	------	------	------	------

Informação transmitida:

0x02	0xFF	0xAB	0x21	0x10	0x02	0x10	0x10	0x34	0x10	0x03	0x03
------	------	------	------	------	------	------	------	------	------	------	------

Implemente um programa que realize este protocolo, carregando os bytes originais (máximo 80 bytes) como um vetor de caracteres (não uma *string*) dentro do próprio código fonte, e exibir cada byte da informação a ser transmitida no formato hexadecimal.

6. Elabore um programa que receba uma linha de texto e conte as vogais apresentando o respectivo histograma na seguinte forma:

Exemplo:

```
String: "Na próxima quarta-feira é feriado."  
a(6):  *****  
e(3):   ***  
i(3):   ***  
o(2):   **  
u(1):   *
```

7. Uma URL (*Uniform Resource Locator*) ou "endereço Internet" é uma maneira de referenciar-se a um site de forma única. Uma URL é composta pelas seguintes partes:

protocolo://domínio/diretório/arquivo

Exemplos:

a.) <http://www.cei.sp.senac.br/cursos/algoritmos/index.html>

Neste caso:

Protocolo = http;

Domínio = www.cei.sp.senac.br

Diretório = cursos/algoritmos;

Arquivo = index.html

b.) <ftp://ftp.xoom.com/pub/texto.zip>

Neste caso:

Protocolo = ftp;

Domínio = [ftp.xoom.com](ftp://ftp.xoom.com);

Diretório = pub;

Arquivo = index.html

É importante notar que:

i) O nome do arquivo, caso não encontrado na URL, assume um nome *default*, de acordo com o servidor (em geral, "index.htm").

ii) O nome do diretório também pode ser omitido, assumindo o diretório raiz do servidor web. Caso o diretório e o arquivo forem omitidos, informar "default" para o diretório e arquivo.

iii) Se o protocolo não for informado assumir "http". Construir um algoritmo que leia uma URL e informe: protocolo, domínio, diretório, arquivo.

Finalizar o algoritmo quando o usuário digitar uma URL contendo um espaço em branco em algum lugar.

8. Deseja-se construir um programa C que leia um conjunto de nomes (possuindo somente letras), realize a contagem das consoantes, exiba o nome com maior quantidade de consoantes e a quantidade de consoantes encontrada para esse nome. Como finalizador digitar '?' no nome. Obs.: se existir algum nome com quantidade igual de consoantes exibir o último encontrado.