



Exercícios de Fixação - Aula 07

Orientações gerais para resolução dos algoritmos:

- Ler o exercício cuidadosamente, até obter seu completo entendimento.
- Se for adequado, crie uma “massa” de dados para simular o ambiente de execução do programa. Não se esqueça de gerar os “valores extremos” dentro do universo dos possíveis valores (por exemplo, zero, o menor valor, o maior valor, valor negativo, etc.). Não se limite apenas à regra; analise também as exceções.
- Crie a solução mental do problema (parcial ou total se possível).
- Se o problema for complexo, trabalhe de maneira analítica, desmembrando-o em partes e resolvendo cada parte individualmente (refinamento sucessivo).
- Desenvolva o código correspondente ao algoritmo criado. Se você ainda tem dificuldade de entendimento, opte pelo fluxograma antes do código.
- Faça os testes de mesa com o código, simulando a resolução do problema. Se foi gerada uma massa de dados (item 2), use-a simulando seu tratamento pelo algoritmo.

ESTRUTURA DE DADOS - MATRIZ

Arrays:

1. Fazer um programa que dados um número de 1 até 7, apresente o nome do dia da semana correspondente. Considere 1 como “Domingo”. Utilize vetor.

2. Ler 10 elementos de uma matriz tipo vetor e apresentá-los depois de todos serem lidos.
3. Ler 10 elementos de uma matriz tipo vetor e apresentá-los na ordem inversa em que foram lidos e armazenados.
4. Dado um número de 1 até 12, apresentar o número de dias do mês correspondente. Se o número escolhido for 2 (fevereiro), considere 28 dias. Usar vetor.
5. Elaborar um programa que popule um vetor de 30 posições com valores de 1 até 30. Exibir o conteúdo do vetor.
6. Elaborar um programa que popule um vetor de 30 posições com valores de 30 até 1. Exibir o conteúdo do vetor.
7. Elaborar um programa que popule um vetor de 10 posições com valores pares a partir de 2 até o 20. Exibir o conteúdo do vetor.
8. Ler dois vetores A e B, de 20 elementos cada e calcular um terceiro, C, onde cada elemento será a soma dos correspondentes em A e B. Apresentar o vetor resultante.
9. Fazer uma função que apresente na tela todos os elementos de um vetor que será passado como parâmetro para a função. Este procedimento será útil para vários exercícios. Que tal colocá-lo numa biblioteca?
10. Ler dois vetores, A e B de 10 elementos cada um e calcular um vetor C onde seu primeiro elemento é o primeiro elemento de A, seu segundo elemento é o primeiro de B, seu terceiro elemento é o segundo de A, e assim sucessivamente. Notem que o vetor C deverá ter 20 elementos. Exibir o conteúdo do vetor C.
11. Ler um vetor A de 15 elementos e a partir dele construir um vetor B de mesmo tamanho, sendo o inverso de A, ou seja, o primeiro elemento de B deverá receber o último de A o segundo de B deverá receber o penúltimo de A e assim por diante.
12. Fazer um programa que carregue um vetor de 8 elementos e que inverta a ordem do seu conteúdo. Apresente no final o vetor invertido. Usar apenas uma variável composta.
13. Ler 8 elementos em uma matriz A tipo vetor. Construir uma matriz B de mesma dimensão onde cada elemento é o triplo do elemento correspondente da matriz A.
14. Ler duas matrizes A e B do tipo vetor com 20 elementos. Construir uma matriz C, onde cada elemento de C é a subtração do elemento

correspondente de A com B. Apresentar a matriz C.

15. Ler duas matrizes do tipo vetor. A com 10 elementos e B com 15 elementos. Construir uma matriz C, sendo esta, a junção das duas outras matrizes. Desta forma, C deverá ter a capacidade de armazenar 25 elementos. Apresentar a matriz C.
16. Ler uma matriz A do tipo vetor com 5 elementos. Construir uma matriz B de mesmo tipo e dimensão, onde cada elemento da matriz B deverá ser o valor inverso (sinal trocado) do elemento correspondente da matriz A. Apresentar os elementos da matriz.
17. Ler uma matriz A do tipo vetor com 10 elementos. Construir uma matriz B de mesmo tipo, sendo que cada elemento de B seja o somatório do primeiro até o elemento correspondente de A. Apresentar os elementos da matriz B.
18. Dada uma data informada pelo usuário, indicar ordinalmente o dia do ano que ela representa do 1º dia ao 365º/366º (conforme o ano for ou não bissexto). O usuário informará separadamente dos valores do dia, do mês e do ano pesquisado.

Ordenação e pesquisa:

19. Carregar um vetor com 50 caracteres gerados aleatoriamente (randômicos). Serão válidos apenas caracteres que representem letras maiúsculas de "A" até "Z". A carga do vetor deverá ser feita através de uma sub-rotina que é chamada automaticamente no programa principal. Os caracteres podem se repetir.
20. Alterar o programa anterior de modo a incluir 4 menus que terão as seguintes finalidades:
 - a. Listar o conteúdo completo do vetor.
 - b. Solicitar ao usuário um caractere, a ser pesquisado no vetor e informar se o caractere pesquisado existe ou não no vetor informando a posição em que ele está (todas as ocorrências visto que o caractere pode se repetir).
 - c. Pesquisar se existe no vetor um caractere informado pelo usuário e informar qual a posição da última ocorrência desse caractere (visto que um mesmo caractere poderá se repetir).
 - d. Pesquisar se existe no vetor um caractere informado pelo usuário e informar qual a posição da primeira ocorrência desse caractere (visto que um mesmo caractere poderá se repetir).
21. Carregar um vetor com 12 caracteres gerados aleatoriamente (randômicos). Serão válidos apenas caracteres que representem letras minúsculas de "a" até "z". A carga do vetor deverá ser feita através de uma sub-rotina que é chamada automaticamente no

- programa principal. Os caracteres NÃO podem se repetir. Exibir no final o vetor gerado. A exibição deverá ser feita via uma sub-rotina.
22. Ler 12 elementos de uma matriz tipo vetor, colocá-los em ordem crescente.
 23. Fazer um programa que trate de um vetor de 8 elementos numéricos. Ele deverá apresentar um menu que permitirá as seguintes operações:
 - a. Carregar o vetor através da leitura dos valores.
 - b. Exibir o conteúdo do vetor.
 - c. Classificar os valores em ordem crescente. Caso o vetor esteja vazio, informar.
 - d. Classificar os valores em ordem decrescente. Caso o vetor esteja vazio, informar.
 - e. Pesquisar a existência de um valor no vetor. Caso o vetor esteja vazio, informar.
 - f. Informar o somatório dos valores do vetor.
 - g. Informar a média dos valores do vetor.
 - h. Informar o maior e o menor valor do vetor.
 - i. Embaralhar o conteúdo do vetor.
 - j. Informar os valores que se repetem e a quantidade de repetições.
 24. Estudar através da execução passo a passo o algoritmo de pesquisa binária demonstrado em aula.
 25. Carregar um vetor com valores de 0 a 100. Implementar um menu que permita realizar neste vetor uma pesquisa sequencial e uma pesquisa binária. Incluir um contador nas rotinas de pesquisa para contar quantas inteirações foram realizadas para se encontrar (ou não) o valor pesquisado.

String:

26. Fazer um programa que dada uma palavra qualquer informada pelo usuário apresente na tela o seu primeiro caractere.
27. Fazer um programa que apresente invertida uma palavra informada pelo usuário. Informar também se é ou não um palíndromo.
28. Faça um programa que possibilite realizar a conversão de um número binário em decimal e um número decimal em binário.
29. Faça um programa que valide um CPF informado pelo usuário, através do cálculo do dígito verificador (check-digit).
30. Através de teste de mesa, indique a finalidade do trecho de código a

seguir:

```
ANAG ← "AMOR";  
Para i de 1 até 4 faça  
    AUX ← ANAG[i];  
    ANAG[i] ← ANAG[5-i];  
    ANAG[5-i] ← AUX;  
Fim-Para
```

31. Imagine que você está contribuindo num projeto de criação de uma nova linguagem de programação e sua parte consiste em criar funções genéricas de manipulação de strings, que serão utilizadas pelos programadores dessa linguagem. Assim sendo, crie uma biblioteca chamada "StrUtil" com as seguintes funções:
- a. Tam: retorna o tamanho de uma string informada.
 - b. Posic: retorna a posição de um dado caractere informado na string informada.
 - c. Copia: retorna uma parte da string informada a partir de uma posição inicial e quantidade de caracteres também informadas.
 - d. Maiusc: retorna uma string informada em maiúsculo.
 - e. Minusc: retorna uma string informada em minúsculo.
 - f. InverteStr: retorna a string informada invertida.
 - g. Del: remove de uma string a quantidade de caracteres informada a partir de uma posição informada. Não tem retorno, deverá manipular a string original.
 - h. Ins: insere uma sub-string em uma string informada, a partir da posição também informada. Não tem retorno, deverá manipular a string original.
 - i. LFill: preenche uma string informada, à esquerda, com um caractere informado. Não tem retorno, deverá manipular a string original.
 - j. RFill: preenche uma string informada, à direita, com um caractere informado. Não tem retorno, deverá manipular a string original.
 - k. Trim: remove espaços em branco de uma string informada. Não tem retorno, deverá manipular a string original.
 - l. Subst: substitui todas as ocorrências de um caractere informado, na string informada. Não tem retorno, deverá manipular a string original.
32. Faça uma função de nome "encrypt", que recebe uma string e realiza a criptografia desta string. Essa criptografia pode ser uma mera troca

de caracteres com base na tabela ASCII e/ou troca de posição de caracteres. Em seguida faça um programa que a partir de uma senha informada pelo usuário utilize a função criada para codificá-la e apresente na tela a senha criptografada.

Tabela:

33. Faça o programa do mapa de troco. Nele o usuário informa um valor monetário a pagar e o programa calcula qual a menor quantidade de cédulas/moedas serão necessárias para pagar exatamente este valor. Por exemplo: se o usuário informar R\$ 1.260,00, o programa deverá informar 6 notas de R\$ 200,00; 1 nota de R\$ 50,00; 1 nota de R\$ 10,00. Pense em utilizar uma matriz de 2 dimensões, onde a primeira coluna contenha o valor de cada cédula/moeda e a segunda terá a quantidade.
34. Fazer um tradutor de código Morse. O usuário poderá entrar com um caractere e obter o código correspondente ou entrar com o código e obter o caractere correspondente.
35. Ler três vetores A, B e C com 10 elementos cada. A partir deles, montar uma matriz M de 3 linhas por 10 colunas, onde cada linha deverá receber os valores referentes a um dos vetores. Apresentar a matriz resultante.
36. Criar uma matriz N de 10x3, onde cada elemento será a soma do índice de sua coluna com o índice de sua linha. Apresentar a matriz resultante.
37. A partir da matriz M e N criadas nos exercícios anteriores, construir a matriz R de mesmo tamanho, onde cada elemento de R será a soma dos elementos correspondentes de M e N. Apresentar a matriz resultante.
38. Ler uma matriz A de 3 linhas e 3 colunas. Calcular e exibir uma matriz B de mesmo tamanho, sendo que a primeira linha de B receba a primeira coluna de A, a segunda linha de B receba a segunda coluna de A e a terceira linha de B receba a terceira coluna de A. Apresentar a matriz resultante.
39. Dada uma matriz A de tamanho 3 x 3, preenchida sequencialmente a partir de 0 até 8. Calcule a diferença entre a diagonal principal e secundária da matriz.
40. Carregar uma matriz de 5x5 com caracteres aleatórios entre A e Z. Apresentar no final a string formada pelos caracteres da diagonal principal dessa matriz.

41.Considere a matriz MAT, com o conteúdo apresentado a seguir:

$$MAT = \begin{bmatrix} O & Q & * & I \\ E & * & E & S \\ R & E & U & T \\ A & * & * & S \end{bmatrix}$$

Através de teste de mesa, apresente o conteúdo da matriz MAT, após a execução do trecho de programa abaixo:

```
...  
Para i de 1 até 3 faça  
  Para j de i + 1 até 4 faça  
    Aux ← MAT[i,j];  
    Mat[i,j] ← Mat[j,i];  
    Mat[j,i] ← Aux;  
  Fim-Para  
Fim-Para  
Aux ← MAT[1,1];  
Mat[1,1] ← Mat[4,4];  
Mat[4,4] ← Aux;  
Aux ← Mat[2,2];  
Mat[2,2] ← Mat[3,3];  
Mat[3,3] ← Aux;  
...
```

42.Ler o nome e 3 notas de 5 alunos, calcular a média de cada aluno (entre 2 das 3 notas pois a menor deve ser descartada). Apresentar todas as notas e médias no final. Permitir ordenar por ordem alfabética de nome e ordem decrescente de media final.

43.Jogo da mágica. Nesta mágica as cartas de um baralho serão apresentadas na tela dispostas em 7 linhas e 3 colunas. O programa será capaz de adivinhar uma carta selecionada pelo usuário, que apenas indicará em qual coluna a carta se encontra. [Solicitar demonstração ao professor]. Para implementar esse interessante jogo, implemente os seguintes passos:

- a. Montar um vetor de 52 posições contendo todas as cartas de cada naipe do baralho. Dica: utilize uma string de 2 posições

para representar cada carta. Exemplo: o rei de espadas seria 'K'

+ #006 (caractere ASCII 6); o ás de copas seria 'A' + #003; o 5 de ouros seria '5' + #004; a dama de paus seria 'Q' + #005. b. Embaralhar as cartas dentro desse vetor.

c. Montar um vetor de 21 posições contendo as 21 primeiras cartas do baralho já embaralhado. Nesta mágica apenas utilizaremos 21 cartas.

d. Montar uma matriz de 7 linhas por 3 colunas onde a primeira célula receberá a última carta do vetor a segunda a penúltima e assim sucessivamente até que a última célula que se encontra na 7ª linha da 2ª coluna receba a primeira carta do vetor de 21 cartas.

e. Apresentar a matriz na tela e solicitar que o usuário escolha uma carta e indique em qual coluna ela se encontra.

f. Preencher o vetor de 21 posições a partir das cartas da matriz 7 x 3 sendo que as cartas da coluna informada pelo usuário deverão ocupar as posições de 8 a 14 do vetor e as demais colunas as posições de 1 a 7 e 15 a 21 respectivamente.

g. Repetir os passos de "d" a "f" por mais 2 vezes sendo que o usuário não escolhera uma nova carta, apenas indicará em que coluna está a carta escolhida inicialmente.

h. Exibir na tela a 11ª carta do vetor de 21 posições.