Prática 09 – Funções com Matrizes

Desenvolva os seguintes algoritmos em linguagem de programação C:

1. Escreva uma função para calcular a média dos elementos de um vetor. Retornar a média.

2. Escreva uma função que receba um vetor numérico unidimensional como parâmetro e verifique se há algum número negativo no vetor. Se houver, retornar 1; se não houver, retornar 0.

3. Escreva uma função que receba um *array* unidimensional de 100 posições como parâmetro e retorne o menor elemento do *array*. Escreva outra função para retornar o maior elemento do *array*.

4. Escreva uma função para corrigir provas de múltipla escolha de 10 questões. A função deve receber como parâmetro o gabarito da prova – um vetor de caracteres de 10 elementos – e as respostas do aluno – também um vetor de caracteres de 10 elementos. Retornar a quantidade de acertos do aluno.

5. A amplitude de uma seqüência é a diferença entre o maior e o menor valor da seqüência. Escreva um programa para ler uma seqüência de números reais, armazenar em um vetor e calcular a amplitude. Estruturar nas seguintes funções:
   - *le_array*: ler os números
   - *menor*: determina o menor valor e retorna esse valor
   - *maior*: determina o maior valor e retorna esse valor
   - *amplitude*: recebe o maior e o menor valores e retorna a diferença

6. Um dos recursos disponibilizados pelos editores de texto é a determinação do número de palavras de um texto. Escreva um programa que determine o número de palavras de uma frase. Funções:
   - *le_frase*: ler uma frase
   - *palavra*: determina o número de palavras e retorna esse valor

   Declare apenas os *arrays* como globais. TODAS as outras variáveis devem ser locais.

7. Acrescente ao programa do exercício anterior uma outra função para contar quantas vezes cada uma das vogais aparecem na frase. Esta última função deve retornar a vogal que mais aparece. Mostrar essa vogal no *main*.

8. Escreva uma função para calcular a soma de duas matrizes. Defina as matrizes como globais.

9. Dada uma matriz 4 x 4, fazer uma função para alterá-la multiplicando os elementos da diagonal principal por 3.

10. Escreva uma função que efetue o seguinte processamento com cada elemento da coluna 0 de uma matriz global de N linhas x 3 colunas:
    - calcule o quadrado e guarde na coluna 1
    - calcule o cubo e guarde na coluna 2

11. Escreva uma função que receba uma matriz como parâmetro e retorne a soma de todos os elementos da matriz.

12. Escreva uma função que retorne o menor elemento de uma matriz.

13. Escreva um programa que receba uma matriz A de dimensão 2x3 e crie uma matriz B de dimensão 3x2. A matriz B será a transposta da matriz A. A geração da matriz B deve ser feita em uma função.

14. Escreva um programa que leia uma matriz A e verifique se é simétrica. Propriedade da matriz simétrica: Aij = Aji. A leitura da matriz deve ser feita em uma função e a verificação de simetria em uma outra função, que deve retornar 1 caso seja simétrica ou 0 caso não seja.

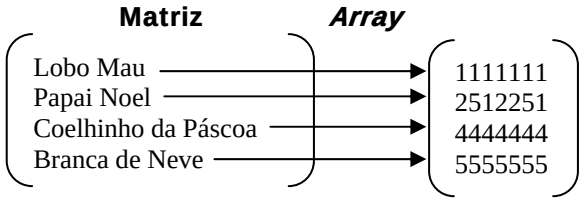
15. Observe o programa abaixo. A função *somalin* calcula a soma dos elementos da linha i da matriz e retorna esse resultado. Escreva essa função.

```
int A[6][6];
 int main () {
     int i, s;
     le_mat();
     for (i=0;i<6;i++) {
              s = somalin (i);
              printf ("%d ", s);
     }
}
```

16. O programa abaixo recebe uma matriz 6x6 e calcula, para uma linha escolhida pelo usuário, o menor elemento dessa linha. O cálculo do menor elemento é feito pela função *menor*. Escreva essa função, sabendo que ela recebe o número da linha que será percorrida e retorna o menor elemento da linha. Escreva também a função *le_linha*, sabendo-se que o usuário digita –1 caso queira encerrar o programa.

```
int A[6][6];
int main () {
     int i, s, lin, men;
     le_mat();
     lin = le_linha();
     while (lin != -1) {
              men = menor (lin);
              printf ("O menor elemento e': %d ", men);
     }
}
```

17. Escreva um programa para manter uma "agenda" telefônica. A agenda será composta de uma matriz de nomes e um *array* de telefones. O usuário irá digitar um nome e telefone e o programa preenche matriz e *array* na mesma posição i. Matriz e *array* devem ser locais, definidos no *main*.



18. Crie uma função "mostra" para mostrar a matriz e o *array* em formato tabelado – como acima (sem as setas).

19. Crie uma função consulta para verificar se o nome digitado já existe na matriz. Caso já exista, retornar 0 (zero); senão, retornar –1. Essa função será usada em duas situações no programa:
    - Ao digitar nome e telefone para incluir, verificar se o nome já existe. Se sim, mostrar mensagem e não incluir na matriz. Voltar a ler nome/telefone em seguida.
    - Após digitação de todos os dados, perguntar ao usuário se deseja visualizar o telefone de alguém em especial ou de todos. Se for a primeira opção, usar a função criada neste item (3). Se for a segunda opção use a função criada no item 2.