

## C - Algoritmos e Linguagem de Programação - LAB

1-Elaborar uma função para efetuar o produto escalar de dois vetores reais de dimensão máxima LIM inicializados pelo teclado, e retornar o produto. Protótipo: **double produto\_escalar(double a[], double b[], int n)**. Defina LIM como constante, via #define.

2-Elaborar uma função para efetuar a soma de duas matrizes reais de dimensão NxM inicializados pelo teclado, e retornar a matriz soma ao programa principal. Protótipo: **void soma\_matrizes(double a[][M], double b[][M], double c[][M], int n, int m)**. Defina N e M como constantes, via #define. *Passe parâmetros n = N e m = M*. Chamada da função: *soma(a, b, N, M)*; Para teste use #define N 3 e #define M 3. **Obs.:** Não há ; terminando comando #define

3-Escriva programa que leia as notas de 10 alunos via teclado, armazena as notas em um vetor e forneça relatório com uma lista das notas ordenadas, a maior nota, a menor nota e a média. Use o método de ordenação Bolha (Bubble Sort). VER LIVRO 1 pág. 210, 211).

4-Escriver uma função que receba um vetor e sua dimensão e reverta, nele mesmo, a ordem dos componentes de um vetor qualquer, de dimensão conhecida. Sua função deve minimizar tempo de execução e uso de memória (não pode ser utilizado vetor auxiliar para trocar elementos de posição). **Sugestão:** Para testar sua função, imprima o vetor antes da chamada da função e o mesmo vetor após a chamada.

5-Dada a matriz M1 (4x4): { { 1, 2, 3, 4 },  
                              { 5, 6, 7, 8 },  
                              { 0, 0, 0, 0 },  
                              {-1, -2, -3, -4 } }

Pede-se: a). Elaborar em C um programa para trocar de posição a linha 1 com a 2.  
b). Elaborar programa para calcular a soma dos elementos da diagonal principal.

6-Implemente em C o algoritmo de Busca Binária. Realize testes que demonstrem o funcionamento do algoritmo. Crie um vetor de inteiros (1 a 1.000), ordenados e verifique o número de comparações realizadas para encontrar um elemento na posição 1/3 do final.

```
long int binary(long int v[], long int n, long int x)
{
    long int low=0, high=n-1, mid;
    while (low <= high) {
        mid=(high+low)/2;
        if (x<v[mid]) high=mid-1;
        else if (x>v[mid]) low=mid+1;
        else return(mid);
    }
    return(-1);
}
```

Escreva programa para testar as seguintes funções e descreva o que elas fazem. Faça seu programa principal imprimir o que retorna de cada função. Execute passo a passo, se necessário.

```
7- void swapv (int a[], int b[], int n)
{
    int i, temp;
    for (i=0; i<n; i++) {
        temp = a[i];
        a[i] = b[i];
        b[i] = temp;
    }
}

8- int mystrlen( char s[])
{
    int i;
    for (i=0; s[i] != '\0'; i++);
    return (i);
}
```

```
9- void mystrcpy (char s[], char t[])
{
    in i=0;
    while ( (s[i] = t[i]) != '\0')
        i++;
}

10- int mystrcmp (char s1[], char s2[])
{
    int i = 0;
    while ( s1[i] == s2[i])
        if (s1[i++] == '\0') return (0);
    return (s1[i] - s2[i]);
}
```