

**Universidade do Estado de Santa Catarina – UDESC**  
**Centro de Ciências Tecnológicas CCT**

**Prof. André Tavares da Silva – andre.silva@udesc.br**

**Exercícios Resolvidos – Linguagem de Programação (LPG0001)**

---

1) Faça um programa para ler uma frase e exibi-la na vertical (uma letra por linha). Refaça o programa anterior utilizando ponteiros.

```
#include <stdio.h>
#include <string.h>

int main() {
    char frase[100];
    int i = 0;

    printf("Entre com uma frase: ");
    fgets(frase, 100, stdin);
    while(frase[i])
        printf("%c\n", frase[i++]);

    return 0;
}
```

**OU**

```
#include <stdio.h>
#include <string.h>

int main() {
    char frase[100];
    char *p = frase;

    printf("Entre com uma frase: ");
    fgets(frase, 100, stdin);
    while(*p)
        printf("%c\n", *p++);

    return 0;
}
```

**Universidade do Estado de Santa Catarina – UDESC**  
**Centro de Ciências Tecnológicas CCT**

**Prof. André Tavares da Silva – andre.silva@udesc.br**

**Exercícios Resolvidos – Linguagem de Programação (LPG0001)**

---

2) Faça uma função para multiplicação de uma matriz 4x3 por outra 3x4, armazenando o resultado em uma terceira matriz.

```
/* sem ponteiros */
void multiplica_matrizes(int matA[][3], int matB[][4], int matC[][3])
{
    int i,j,k;
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            matC[i][j] = 0;
            for (k = 0; k < 4; k++) {
                matC[i][j] += matA[k][i] * matB[j][k];
            }
        }
    }
}
```

```
/* com ponteiros */
/* int (*matA)[3] equivale a int matA[][3] é preciso do [3] para saber como avançar nas linhas */
/* *(*matC + i) + j) equivale a matC[i][j] */
void multiplica_matrizes(int (*matA)[3], int (*matB)[4], int (*matC)[3])
{
    int i,j,k;
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            *(*matC + i) + j) = 0;
            for (k = 0; k < 4; k++) {
                *(*matC + i) + j) += *(*matA + k) + i) * *(*matB + j) + k);
            }
        }
    }
}
```

```
/* com ponteiros 2 */
void multiplica_matrizes(int (*matA)[3], int (*matB)[4], int (*matC)[3])
{
    int i,j,k;
    for (i = 0; i < 3; i++) {
        for (j = 0; j < 3; j++) {
            int *pB = *(matB + j); /* aponta para o inicio da linha em B */
            int *pC = *(matC + i) + j; /* aponta para C[i][j] */
            *pC = 0;
            for (k = 0; k < 4; k++) {
                *pC += *(*matA + k) + i) * *pB++; /* calcula e vai para o próximo elemento em B */
            }
        }
    }
}
```

**Universidade do Estado de Santa Catarina – UDESC**  
**Centro de Ciências Tecnológicas CCT**

**Prof. André Tavares da Silva – andre.silva@udesc.br**

**Exercícios Resolvidos – Linguagem de Programação (LPG0001)**

---

3) Faça uma função que some duas matrizes 4x3 e exiba o resultado (a matriz resultante). Crie um programa principal que preencha aleatoriamente (randomicamente) os valores das matrizes e chame a função para somá-las. Utilize ponteiros para passar as matrizes como parâmetro para a função (passagem por “referência”).

```
#include <stdio.h>
#include <stdlib.h> /* para rand() e srand() */
#include <time.h> /* para time() */

void soma_matrizes(int (*matA)[3], int (*matB)[3], int (*matC)[3])
{
    for (int i = 0; i < 4; i++)
        for (int j = 0; j < 3; j++)
            *(*matC + i) + j) = *(*matA + i) + j) + *(*matB + i) + j);
}

int main() {
    int A[4][3], B[4][3], C[4][3];
    int i, j;

    srand(time(NULL)); // gera nova semente de valores aleatórios
    for (i=0; i<4; i++) { // gera matrizes aleatórias
        for (j=0; j<3; j++) {
            A[i][j]=rand()%100; // valores entre 0 e 99
            B[i][j]=rand()%100; // valores entre 0 e 99
        }
    }

    soma_matrizes(A,B,C);

    for(i=0; i<4; i++) { // exibe matriz
        for(j=0; j<3; j++)
            printf("%3d ",C[i][j]);
        printf("\n");
    }

    return 0;
}
```

ou

```
void soma_matrizes(int (*matA)[3], int (*matB)[3], int (*matC)[3])
{
    int *pA = &matA[0][0];
    int *pB = &matB[0][0];
    int *pC = &matC[0][0];
    for (int i = 0; i < 4*3; i++, pA++, pB++, pC++) { // soma 12 valores consecutivos
        *pC = *pA + *pB;
    }
}
```

Pergunta: a matriz resultante será passada por parâmetro ou será retorno da função?

Resposta: em C não tem como retornar um vetor ou uma matriz. É possível retornar apenas um ponteiro para vetor ou matriz, mas não os conteúdos. Então a matriz resultante precisa ser **passada por parâmetro**.