



ATIVIDADE DE PONTEIROS, ALOCAÇÃO DINÂMICA E STRINGS

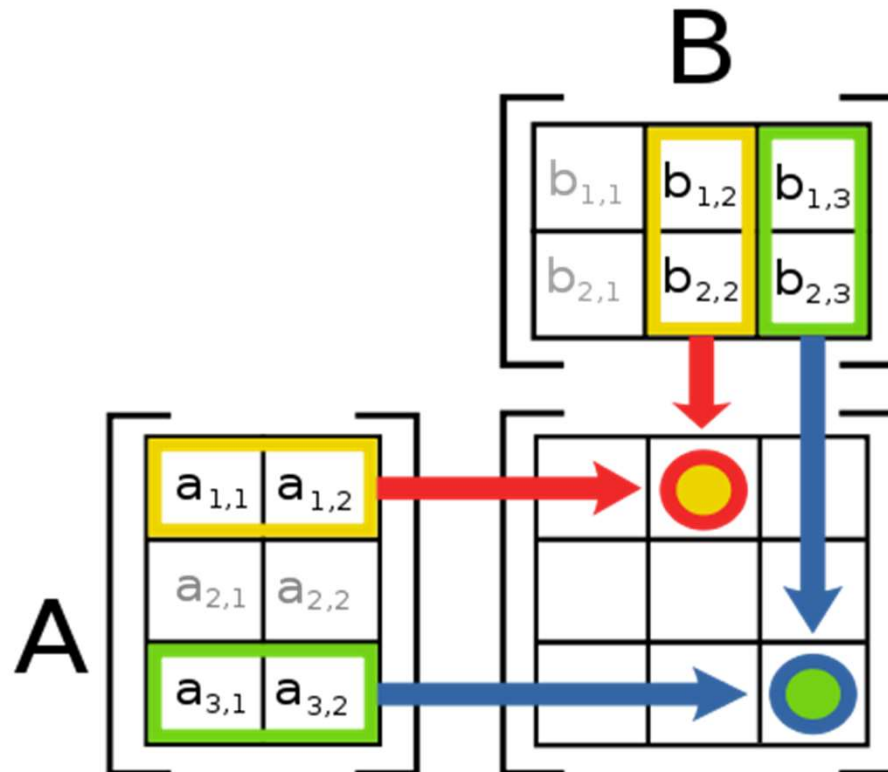
Prof. Humberto Razente

Sala 1B144

1

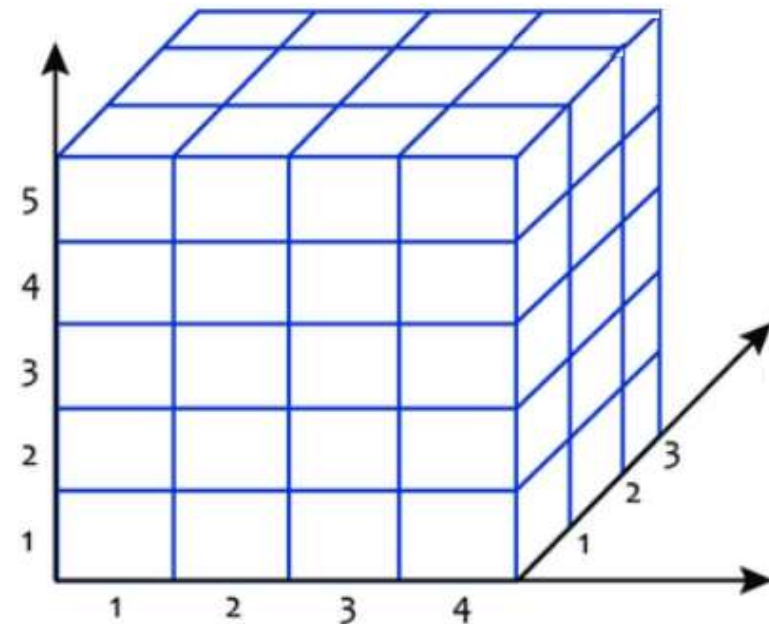
EXERCÍCIO 1

- Escreva um programa para calcular a multiplicação de uma matriz $M \times N$ por uma matriz $N \times P$ alocadas dinamicamente. A matriz resultado também deve ser alocada dinamicamente.
- Imprima as matrizes de entrada e a matriz resultante. Ao final, desaloque as matrizes.



EXERCÍCIO 2

- Escreva um programa que leia um cubo tridimensional com coordenadas X, Y e Z de números de ponto flutuante alocado dinamicamente com N posições para cada coordenada. O objetivo é fazer a agregação dos valores que estão em uma dimensão para valores específicos das outras duas. Por exemplo, se o usuário quiser a agregação quando $Y = 3$ e $Z = 4$, devem ser somados os valores das células com X de 1 a N para $Y = 3$ e $Z = 4$.
- Assim, após a leitura do cubo, em uma repetição devem ser oferecidas as opções: XY, XZ e YZ, seguido da leitura dos valores para o par escolhido.



EXERCÍCIO 3

- Escreva um programa que leia uma matriz de caracteres (um vetor de strings) com nomes e apresente quem seria o primeiro da lista (pela ordem alfabética).
 - Não se deve ordenar a lista
 - Equivalente a encontrar o menor valor em um vetor numérico
- Utilize a função strcmp para comparar strings.

```
int main() {  
    char M[20][30];  
  
    // leia cada string M[i]  
  
    // localize o elemento que seria primeiro  
    // de uma lista ordenada  
}
```



EXERCÍCIO 4

- Escreva laços para obter o mesmo resultado das seguintes funções:
 - strlen
 - strcpy
 - strcat

```
int main() {  
    char M[100] = "The quick brown fox jumps over the lazy dog";  
    char N[100] = "A ligeira raposa marrom saltou sobre o cachorro preguiçoso";  
    char R[100];  
  
    // 1) imprimir o tamanho das strings M e N  
    // 2) copiar M para R  
    // 3) concatenar N em R  
}
```



EXERCÍCIO 5

- Escreva um programa que ordene a seguinte lista de presença
- Utilize a função strcmp para comparar strings.

```
#include <stdio.h>
#include <stdlib.h>
#define QUANTIDADE 21
int main() {
    char temp[12];
    char lista[QUANTIDADE][12] = { "Vinicius", "Guilherme", "Vitor", "Lucas", "Igor", "Joao",
                                     "Pedro", "Abel", "Luiz", "Wemerson", "Rafael", "Pablo",
                                     "Saint", "Thais", "Matheus", "Douglas", "Gabriel",
                                     "Viviane", "Reginaldo", "Jose", "Leonardo" };

    printf("Antes de ordenar:\n=====\\n");
    for (int i = 0; i < QUANTIDADE; i++)
        printf("%s\\n", lista[i]);

    ...

    printf("\\n\\nApos ordenar:\\n=====\\n");
    for (int i = 0; i < QUANTIDADE; i++)
        printf("%s\\n", lista[i]);
    return 0;
}
```

