



### **Lista de Exercícios 8 – Ponteiros – Parte2**

Nesta lista vocês usarão os conceitos de leitura de arquivo, alocação dinâmica de memória e ponteiros.

Para arquivos: lembre-se sempre do `fclose()`!

Para alocação dinâmica: (i) Ponteiro sem atribuição aponta para NULL. (ii) Não vai mais usar o array ou vai finalizar seu programa? Lembre-se do `free()`!

1. Crie um programa que aloque um array de float contendo o tamanho que o usuário informar como entrada para o programa. Atribua valores para cada posição desse array (use um loop e atribua para cada posição o valor de `i`). Imprima o endereço de cada posição desse array.
2. Crie um programa que contenha uma matriz de float contendo o número de linhas e colunas que o usuário informar como entrada para o programa. Atribua valores para cada posição desse array (use um loop e atribua para cada posição o valor de `i*j`). Imprima o endereço de cada posição dessa matriz.
3. Crie um programa que aloque um array de inteiros contendo o tamanho que o usuário informar como entrada para o programa (O TAMANHO MÁXIMO DO VETOR DEVE SER 1000, TRATE CASO O VALOR DIGITADO FOR MAIOR). Leia os valores desse array de um arquivo de texto `inputs_int_lista8_p2.txt` e, utilizando **apenas aritmética de ponteiros**, imprima o dobro de cada valor lido.
4. Crie um programa que aloque um array de inteiros contendo o tamanho que o usuário informar como entrada para o programa. Leia os valores desse array do teclado e imprima o endereço das posições contendo valores pares.
5. Considere a seguinte declaração de variáveis: `int A, *B, **C, ***D;`

Escreva um programa que leia a variável "A" e calcule e exiba o dobro, o triplo e o quádruplo desse valor utilizando apenas os ponteiros B, C e D. O ponteiro B deve ser usada para calcular o dobro, C o triplo e D o quádruplo.

6. Crie um programa que aloque um array de inteiros contendo o tamanho que o usuário informar como entrada para o programa (O TAMANHO MÁXIMO DO VETOR DEVE SER 1000, TRATE CASO O VALOR DIGITADO FOR MAIOR). Leia os valores desse array de um arquivo de texto `inputs_int_lista8_p2.txt`.  
Determina o maior elemento do array e o número de vezes que este elemento ocorreu no array. Por exemplo, para um array com os seguintes elementos: 5, 2, 15, 3, 7, 15, 8, 6, 15, seu programa deverá indicar que o número 15 é o maior e apareceu 3 vezes.
7. Crie um programa que aloque um array de inteiros contendo o tamanho que o usuário informar como entrada para o programa (O TAMANHO MÁXIMO DO VETOR DEVE SER 1000, TRATE CASO O VALOR DIGITADO FOR MAIOR). Leia os valores desse array de um arquivo de texto `inputs_int_lista8_p2.txt`. Declare um ponteiro que aponte para o array e, utilizando o ponteiro para percorrer o array, some mais um (+1) a cada posição do array.
8. Crie um programa que aloque três matrizes de float, cada uma com 5 linhas e 50 colunas. Preencha essas duas matrizes com os valores lidos do arquivo de texto `inputs_float_lista8_p2.txt`. Leia da linha 1 até a linha 250 para a primeira matriz e da linha 251



Universidade Federal  
do Espírito Santo

**CENTRO UNIVERSITÁRIO NORTE DO ESPÍRITO SANTO**  
**DEPARTAMENTO COMPUTAÇÃO E ELETRÔNICA**

Disciplina: Programação Estruturada / Programação II

até a linha 500. Calcule a soma dos elementos da primeira e segunda matriz e atribua à terceira matriz. Imprima a terceira matriz.

9. Escreva um programa em C para gerenciar uma lista dinâmica de números inteiros positivos inseridos pelo usuário. O programa deve:
  1. Alocar dinamicamente um vetor de capacidade 5.
  2. Exibir um menu com as opções (ficar em loop até o usuário digitar uma opção menor que zero):
    1. Inserir novo número
    2. Exibir todos os números
    3. Sair

Sempre que a capacidade do vetor se esgotar, o programa deve dobrar a capacidade do vetor utilizando `realloc()`, sem perder os números já inseridos.

10. Escreva um programa em C que:
  1. Crie uma matriz de inteiros, alocada dinamicamente com 3 linhas e 3 colunas.
  2. Preencha a matriz com valores lidos do teclado.
  3. Pergunte ao usuário se ele quer adicionar mais colunas à matriz ('S' ou 'N'). Caso 'S', seu programa deverá entrar em um loop e sempre ao final do loop você deve perguntar se o usuário quer adicionar mais colunas. Esse loop faz:
    1. Para cada coluna adicionada, use `realloc` para aumentar o tamanho de todas as linhas.
    2. O programa deve pedir ao usuário o valor da nova coluna para cada linha.
    3. Deve ser possível adicionar quantas colunas o usuário desejar (Perguntar se ele quer adicionar mais uma coluna).
  4. Quando o usuário não quiser mais adicionar, o programa deve exibir a matriz final com todas as linhas e colunas.
  5. Liberar toda a memória alocada ao final!