

Lista de exercícios para aula: Ponteiros, Structs e Alocação Dinâmica.

Parte I — Ponteiros

1. Trocar valores de variáveis por meio de ponteiros.
 - a) Crie um programa que declare duas variáveis inteiras e use um ponteiro para trocar seus valores.
 - b) Faça o mesmo para duas variáveis do tipo float e char.
2. Crie um programa que leia um número inteiro e imprima:
 - a) O valor digitado.
 - b) O endereço da memória.
 - c) O conteúdo da memória apontado pelo ponteiro.
3. Crie um programa que declare um vetor de inteiros com 5 posições e utilizando um ponteiro faça:
 - a) Preencha o vetor.
 - b) Mostre os dados armazenados.
 - c) Calcule a média dos dados armazenados.

Desafio parte I:

Escrever um programa para gerar uma matriz de 4 x 5 números reais, multiplicar a primeira coluna por qualquer outra da matriz e mostrar a soma dos produtos. O programa deve ser decomposto em subprogramas e utilizar ponteiros para acessar os elementos da matriz.

Parte II - Ponteiros e Registros

1. Utilizando um ponteiro para acessar os campos do registro.
 - a) Defina um registro Aluno com os campos nome e idade.
 - b) Defina uma variável do tipo registro.
 - c) Defina um ponteiro para este tipo de registro.
 - d) Por meio do ponteiro atribua valores a cada campo deste registro.
2. Modifique o exercício anterior para receber os dados de 3 alunos (Vetor de registros) e fazer toda a manipulação dos registros por meio de um ponteiro.
 - a) Preencher todos os registros
 - b) Mostrar o conteúdo.

Desafio parte II:

É necessário armazenar a seguinte informação de uma pessoa: nome, idade, altura e peso. Escrever uma função que leia os dados de uma pessoa, recebendo como parâmetro um ponteiro e outra função que os visualize.

Parte III — Alocação dinâmica de memória

1. Utilize o operador new() para alocar um bloco com 5 números inteiros.
 - a) Preencha este bloco com valores inteiros.
 - b) Mostre os dados preenchidos.
 - c) Após o uso libere a memória por meio do operador delete().
2. Desenvolva um programa que :
 - a) Peça ao usuário quantos números deseja armazenar.
 - b) Aloque dinamicamente por meio do operador new().

- c) Preencha toda a estrutura.
 - d) Apresente o maior valor armazenado.
 - e) Após o uso libere a memória por meio do operador delete().
3. Desenvolva um programa que crie um registro Pessoa com nome(string/vetor de caracteres) e cpf (inteiro).
- a) Use new() para alocar dinamicamente uma variável para cadastrar uma pessoa.
 - b) Leia e exiba os dados desta pessoa.
 - c) Após o uso libere a memória por meio do operador delete().

Desafio parte III:

Escrever um programa que leia um número determinado de inteiros (1.000 como máximo) do terminal e os visualize na mesma ordem e com a condição de que se escreva somente uma vez cada inteiro. Se o inteiro já foi impresso, não deve ser visualizado novamente. Por exemplo, se os números seguintes são lidos do terminal 55, 78,25, 55, 24,3 e 7 o programa deve visualizar o seguinte: 55,78,25,24,3 e 7. Deve-se trabalhar com ponteiros para tratar com o *array* no qual foram armazenados os números.