



Funções (passagem por referência) e Ponteiros

ATENÇÃO: Todos os exercícios deverão ser resolvidos e serão enviados via PVANet, a partir de agora apenas o código.c de cada exercício.

Objetivos

Praticar os conceitos de **Ponteiros** e **Funções com Passagem de Parâmetro por Referência** na linguagem de programação C.

Exercícios

Os exercícios estão organizados de uma outra forma para representar uma **entrada** e **saída** de informação para cada programa. Na coluna **entrada** representa os valores que o usuário digitaria ao executar o programa, e na coluna **saída** a representação da **tela esperada** ao executar o programa.

Professor

1.1) Escrever e executar um programa em linguagem C que:

- i. Crie uma variável do tipo int.
- ii. Armazene um valor (número) do tipo inteiro na variável criada.
- iii. Crie um ponteiro que apontará para uma variável do tipo int.
- iv. Armazene no ponteiro o endereço de memória da variável do tipo int
- v. Utilizando o ponteiro (e não a variável do tipo int), mostre na tela o conteúdo da variável do tipo int (Para a qual o ponteiro se encontra apontando).

Entrada

A entrada de seu programa deverá ser um valor do tipo inteiro que deverá ter seu endereço de memória armazenado em um ponteiro.

Saída

A saída deverá ser o valor de entrada, utilizando o ponteiro para poder imprimir na tela.

Como mostrado abaixo:

ENTRADA	SAÍDA
10	10

1.2) Usando a passagem por referência, crie uma função que incremente duas unidades em uma variável qualquer do tipo int.

Em seguida, crie uma função main onde o usuário digita um valor N e o programa deverá mostrar todos os valores a partir do número 1 e incrementados segundo a função que você criou, menores ou iguais a N.

Entrada

A entrada deverá ser um número inteiro e positivo.

Saída

A saída do programa deverá ser os números de 1 até N, incrementados.

Como mostrado abaixo:

ENTRADA	SAÍDA
10	1 3 5 7 9
15	1 3 5 7 9 11 13 15

1.3) Defina uma struct para representar um **Funcionario** de um supermercado contendo os campos de **nome**, **idade** e **salário**. Crie na função main, dois funcionários e um ponteiro para o tipo Funcionario, digite as informações de ambos e depois faça a impressão dos campos de cada um usando o ponteiro.

Transforme o seu código, para que as operações de entrar com as informações e a impressão, sejam feitas através de funções, e teste novamente seu programa.

1.4) Usando a struct e funções do exercício anterior, defina um vetor para 5 funcionarios, entre com as informações e imprima cada uma delas. Por fim, crie duas funções que mostre, a partir do vetor, uma para dizer quem é o funcionário mais novo, e outra para dizer qual o funcionário com o maior salário. Teste o seu código.

Roteiro

2.1) Escrever e executar um programa em linguagem C que:

- Crie uma variável do tipo int.
- Armazene um valor (número) do tipo inteiro na variável criada.
- Crie um ponteiro "p1" do tipo ponteiro para int.
- Crie um ponteiro "p2" também do tipo ponteiro para int.
- Armazene no ponteiro "p1" o endereço de memória da variável do tipo int
- Armazene o ponteiro "p1" no ponteiro "p2". Em outras palavras, p2 recebe p1, só isso (p2 = p1).
- Mostre na tela o conteúdo dos endereços de memória apontados pelos ponteiros p1 e p2.

2.2) Escrever e executar um programa em linguagem C que:

- Declare uma estrutura (struct) para representar uma pessoa contendo as informações (variáveis):
 - nome
 - idade
- Crie uma variável do tipo da struct declarada;
- Preencha as variáveis (os campos) da estrutura com valores do mesmo tipo delas.
Exemplo: "João da Silva", 29;
- Crie um ponteiro que apontará para uma variável do tipo da estrutura (struct) criada.
- Armazene no ponteiro o endereço de memória da estrutura.
- Utilizando o ponteiro (e não a variável do tipo da struct), mostre na tela o conteúdo dos campos "nome" e "idade" da variável tipo da struct (Para a qual o ponteiro se encontra apontando).

Entrada

A entrada de seu programa deverá ser o nome, de no máximo 20 caracteres, e a idade; armazenados em uma estrutura.

Saída

A saída deverá ser o nome e a idade de entrada utilizando o ponteiro que aponta para a estrutura, com espaçamento entre as informações e sem pulo de linha.

Como mostrado abaixo:

Entrada	Saída
Hugh Laurie 56	Hugh Laurie 56

2.3) Usando a passagem por referência, crie as seguintes funções:

- void preencheVetor(int *vet, int n)
- void imprimeVetor(int *vet, int n)
- int retornaMaiorElemento(int *vet, int n)

Implemente o corpo das 3 funções e crie um vetor na Main com 1000 posições chamando em seguida cada uma das funções criadas, cujo número de elementos será digitado pelo usuário e mostrando, quando tiver, seu resultado.

Entrada

A entrada de seu programa deverá ser um número n inteiro positivo que será utilizado para definir o tamanho do vetor a ser preenchido e os números de acordo com a limitação do número n.

Saída

A saída deverá ser o vetor com os números preenchidos na entrada e o maior elemento do vetor, separados por um pulo de linha.

Como mostrado abaixo:

ENTRADA	SAÍDA
5 1 2 3 4 5	1 2 3 4 5 5

2.4) Crie uma função que recebe como parâmetro, um ponteiro para um tipo estruturado Aluno (que contém Nome, Matricula e Nota) e usando esse ponteiro, preencha os dados da variável aluno que será passada como parâmetro. Feito isso, crie um programa main com um vetor de 10 alunos onde cada posição do vetor será preenchida usando a função criada.

Crie outra função que recebe como parâmetro o vetor criado e mostre na tela as informações (Nome, Matricula e Nota) do aluno que possui a maior nota dentre os alunos cadastrados.

Entrada

A entrada de seu programa deverá ser as informações dos 10 alunos (Nome, Matricula e Nota).

Saída

A saída deverá ser as informações do alunos com a maior nota.

Como mostrado abaixo:

ENTRADA	SAÍDA
Joaquim Fernandes 1234 23.5 ... Adriana Santos 4321 34.3	Aluno(a) com maior nota: Adriana Santos 4321 34.3