

Aula Prática 08

- **Instruções:**
 - Os exercícios deverão ser feitos em aula de laboratório durante o tempo da aula;
 - O professor/monitor irá esclarecer dúvidas em aula;
 - Crie uma pasta com seu nome e vá gravando seus programas implementados.
- **Para entrega:** submissão via moodle, conforme já explicado (veja instruções na própria página Web de submissão).

Exercício 1 a 4

Estude os 4 programas, veja as suas diferenças, faça as alterações pedidas. Para cada alteração, gere um programa diferente, conforme pedido.

Aprenda alterando um código já feito!

Exemplo 1	Exemplo 2
<pre>#include <stdio.h> #include <string.h> #define MAX 5 struct produto { char descricao[40]; float valor; }; int main() { struct produto prods[MAX]; /* definindo um vetor de 5 posicoes para o struc produto */ int i; for (i=0; i<MAX; i++) { puts("Nome do Produto?"); gets(prods[i].descricao); puts("Valor do Produto?"); scanf("%f", &prods[i].valor); getchar(); // Limpa o buffer de teclado } printf("\n\n"); for (i=0; i<MAX; i++) { printf("Descricao: %s - Valor: R\$ %.2f\n", prods[i].descricao, prods[i].valor); } return 0; }</pre>	<pre>#define LEN 50 struct endereco { char rua[LEN]; char cidade_estado_cep[LEN]; }; struct endereco obtem_endereco(void) { struct endereco ender; printf("\t Entre rua: "); gets(ender.rua); printf("\t Entre cidade/estado/cep: "); gets(ender.cidade_estado_cep); return ender; } void imprime_endereco(struct endereco ender) { printf("\t %s\n", ender.rua); printf("\t %s\n", ender.cidade_estado_cep); } void main(void) { struct endereco residencia; printf("Entre seu endereco residencial:\n"); residencia = obtem_endereco(); printf("\nSeu endereco eh:\n"); imprime_endereco(residencia); }</pre>
<p>(1) Altere o programa (exemplo 1) para que os dados digitados sejam exibidos na ordem inversa em que foram digitados. (código ap08-ex01-1.c)</p> <p>(2) Altere o programa (exemplo 1) para que seja impresso somente os produtos com valor acima de 120,00 reais. (código ap08-ex01-2.c)</p>	<p>(1) Altere o programa (exemplo2) para que os dados cidade, estado e cep sejam recebidos como : Cidade - char [15] Estado - char [3] CEP - char [10] (lembrar de colocar um caracter a mais para ter o final da string \0). (código ap08-ex02.c)</p>

Exemplo 3	Exemplo 4
<pre>#include <stdio.h> #include <string.h> struct tipo_endereco { char rua [50]; int numero; char bairro [20]; char cidade [30]; char sigla_estado [3]; long int CEP; }; struct ficha_pessoal { char nome [50]; long int telefone; struct tipo_endereco endereco; }; main (void) { struct ficha_pessoal ficha; strcpy (ficha.nome,"Luiz Osvaldo Silva"); ficha.telefone=4921234; strcpy (ficha.endereco.rua,"Rua das Flores"); ficha.endereco.numero=10; strcpy (ficha.endereco.bairro,"Cidade Velha"); strcpy (ficha.endereco.cidade,"Belo Horizonte"); strcpy (ficha.endereco.sigla_estado,"MG"); ficha.endereco.CEP=31340230; /* INSIRA AQUI AS LINHAS PARA IMPRIMIR O QUE FOI COPIADO PARA A STRUCT FICHA PESSOAL */ /* FIM DA AREA DE IMPRESSAO */ return 0; }</pre>	<pre>#include <stdio.h> #include <stdlib.h> #define TAM_VET 4 // declaracao do modelo data struct data{ int dia, mes, ano; }; // declaracao do modelo funcionario struct aluno{ char nome[50]; double mediaFinal; struct data nascimento; }; int main(){ int i; // declaracao e inicialização do vetor de estruturas vet struct aluno faculdade[TAM_VET]={{"Adriana Martins", 7.5, {10,5,1982} }, {"Luciano Araujo de Lima", 8, {23,8,1983} }, {"Marcos Paulo da Silva", 9.2, {14,1,1981} }, {"Jonathas Martins dos Santos", 10, {2,10,1983} } }; puts("Imprimindo os dados.\n"); for(i=0; i<TAM_VET; i++){ puts(faculdade[i].nome); printf("Media Final: %.2f", faculdade[i].mediaFinal); printf("\n%d/%d/%d\n\n", faculdade[i].nascimento.dia, faculdade[i].nascimento.mes, faculdade[i].nascimento.ano); } return (0); }</pre>
<p>(1) Altere o programa exemplo 3 para que se imprima o que foi copiado para dentro da struct ficha pessoal.</p> <p>(código ap08-ex03.c)</p>	<p>(1) Altere o programa exemplo 4 para que a entrada de dados seja feita por meio de digitação e não como feito dentro do programa.</p> <p>(código ap08-ex04.c)</p>

Observações importantes:

- gets já é comando ultrapassado, assim recomendo utilizar fgets, que funciona assim:
 - fgets (campo string, tamanho a ser lido, stdin);
 - Antes do comando fgets colocar um comando para limpar o *buffer* de entrada de dados:
 - fflush (stdin); // no Windows
 - fpurge (stdin); // no Mac
 - __fpurge (stdin); // no Linux

Exercício 5

Vamos elaborar um programa em C para praticar os seguintes conceitos:

- Estrutura ou registro - *struct*

O que deverá ser feito é descrito a seguir:

Um hospital da rede pública de saúde deseja fazer o cadastro dos seguintes dados sobre os pacientes que atende: **nome**, **idade**, **peso** e **altura**. Defina uma estrutura de dados conveniente para armazenar estes dados (*struct*).

```
typedef struct pessoa
{
    ...
    ...
} individuo;
```

Considere que o cadastro deverá ser armazenado em um **vetor** (denominado **vetorCadastro**), sendo que o vetor deverá ter um tamanho máximo permitido (`#define MAX 10`).

No início do código principal, pergunte ao usuário quantas pessoas serão cadastradas (n). Para cada pessoa a ser cadastrada, você receberá os dados e colocará no vetor do tipo da estrutura que você criou (individuo).

Em seu código **main** você deverá ter:

```
int main (int argc, char *argv[])
{
    individuo vetorCadastro[MAX];

    faça um estrutura de repetição para receber os cadastros das pessoas.

    faça um FOR (para cada pessoa), imprimindo seus campos:
        - vetorCadastro[i].nome;
        - vetorCadastro[i].altura;
        - etc.
}
```

Salve o código como **ap08-ex05.c**.

Exercício 6

Faça uma **função** chamada **nova_pessoa()** para receber o cadastro de cada pessoa e retornar um tipo **individuo** (do *struct* que você criou).

```
individuo nova_pessoa()
{
    individuo p;
    ...
    printf("Nome da pessoa:");
    gets (p.nome);
    ...
    printf("Idade da pessoa:");
    scanf("%d", &(p.idade);
    ...
}
```

E altere seu código **main** implementado anteriormente para que faça a chamada à função que vai gerando o registro de cada novo indivíduo.

Salve o código como **ap08-ex06.c**.