

**Paradigmas de Linguagens de Programação**

**Programação Imperativa - Linguagem C**

**Professor:** Daniel Abella

**Aluna:** Fabrícia Martins

**Período:** 2016.1

# **Lista de Exercícios 2**

As questões relacionadas abaixo devem ser resolvidas e disponibilizadas no git no repositório chamado facisa-plp-162

**1.** Observe o trecho de programa abaixo:

**int \*p, x=1;**

**p = &x;**

**\*p = \*p + 1;**

Supondo que as variáveis x e p sejam inicialmente alocadas, respectivamente, nos endereços de memória FF01 e FF07, e considerando que o tipo int ocupe 2 bytes de memória, qual o resultado final de, respectivamente:

a) X

**2**

b) &x

**FF01**

c) p

**FF01**

d) \*p

**2**

e) &p

**FF07**

**2.** Faça um programa que:

**a)** declare uma variável n do tipo int, uma variável x do tipo float, uma variável pn do tipo ponteiro para int e uma variável px do tipo ponteiro para float;

**b)** atribua para n o valor 5, para x o valor 3.8, para pn o endereço de n e para px o endereço de x;

**c)** exiba o conteúdo e o endereço de cada variável.

**R.:**

**#include <stdio.h>**

**int main(){**

**int n, \*pn;**

**float x, \*px;**

**n = 5;**

**x = 3.8;**

**pn = &n;**

**px = &x;**

**printf("Endereço de n: %p\n",&n);**

**printf("Conteudo de n: %d\n",n);**

**printf("Endereço de x: %p\n",&x);**

**printf("Conteudo de x: %f\n",x);**

**printf("Endereço de pn: %p\n",pn);**

**printf("Conteudo de pn: %d\n",\*pn);**

**printf("Endereço de px: %p\n",px);**

**printf("Conteudo de px: %f\n",\*px);**

**return 0;**

**}**

**3.** Escreva um procedimento em C chamado ***inverte*** que inverta o sinal de uma variável do tipo float recebida como parâmetro. Exemplo:

**float x = 3.7;**

**inverte(&x); // x irá assumir o valor -3.7**

**R.:**

**void inverte(float\* valor){**

**if(\*valor > 0){**

**printf("%f\n",(\*valor)\*(-1));**

**}**

**else if(\*valor < 0){**

**printf("%f\n",(\*valor)-(\*valor)\*2);**

**}**

**}**

**int main(){**

**float x = 3.7;**

**inverte(&x); // x irá assumir o valor -3.7**

**return 0;**

**}**

**4.** Uma determinada empresa resolver conceder um abono salarial aos seus empregados. O valor desse abono será igual para todos os empregados da empresa. Então implemente:

a) uma função (procedimento) chamada ***add*** que receba dois parâmetros X e Y (do tipo float) e adicione a X o valor de Y.

b) um programa que leia o valor do abono salarial e, para cada empregado, faça o seguinte:

* leia a matrícula do empregado (a leitura da matrícula 0 encerra os dados de entrada);
* leia o seu salário atual (em R$);
* modifique o salário do funcionário, adicionando ao seu salário atual o valor do abono
  + Use a sua função *add*
* exiba o nome do empregado e o seu novo salário.

**R.:**

**#include <stdio.h>**

**#include <stdlib.h>**

**void add(float\* x, float\* y){**

**(\*x) += (\*y);**

**}**

**int main(){**

**float abono, salario;**

**long matriculaEmpregado;**

**printf("Digite o valor do abono salarial:\n");**

**scanf("%f",&abono);**

**printf("Digite a matricula do empregado:\n");**

**scanf("%ld",&matriculaEmpregado);**

**if(matriculaEmpregado == 0){**

**exit(0);**

**}**

**printf("Digite o salario atual do empregado:\n");**

**scanf("%f",&salario);**

**add(&salario, &abono);**

**printf("Matricula empregado: %ld - Salario Atual: %f",matriculaEmpregado, salario);**

**return 0;**

**}**

**5.** Seja a seguinte struct que é utilizada para descrever os produtos que estão no estoque de uma loja:

**struct Produto {**

**char nome[30]; /\* Nome do produto \*/**

**int codigo; /\* Codigo do produto \*/**

**double preco; /\* Preco do produto \*/**

**};**

Escreva um programa em C que declare um vetor de Produto com 10 itens de produtos e atribua os valores "Pé de Moleque", 13205 e R$0,20 aos membros da posição 0 e os valores "Cocada Baiana", 15202 e R$0,50 aos membros da posição 1 do vetor. Após isso, exiba todos os itens da loja.

**R.:**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <string.h>**

**void add(float\* x, float\* y){**

**(\*x) += (\*y);**

**}**

**int main(){**

**int i;**

**struct Produto {**

**char nome[30]; /\* Nome do produto \*/**

**int codigo; /\* Codigo do produto \*/**

**double preco; /\* Preco do produto \*/**

**};**

**Produto produtos[10];**

**struct Produto produto1;**

**strcpy(produto1.nome,"Pé de Moleque");**

**produto1.codigo = 13205;**

**produto1.preco = 0.20;**

**struct Produto produto2;**

**strcpy(produto2.nome,"Cocada Baiana");**

**produto2.codigo = 15202;**

**produto2.preco = 0.50;**

**produtos[0] = produto1;**

**produtos[1] = produto2;**

**for(i=0; i<10; i++){**

**printf("Produto%d: \nNome:%s\nCodigo:%d\nPreco:%f\n",i,produtos[i].nome,produtos[i].codigo, produtos[i].preco);**

**}**

**return 0;**

**}**

**6.** Faça as mudanças que forem necessárias no item anterior para usar um ponteiro para Produto ao invés de um vetor de Produtos. Faça a alocação de memória de forma que se possa armazenar N produtos (N será lido) na área de memória apontada por este ponteiro e refaça as atribuições feitas no item anterior;

**R.:**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <string.h>**

**void add(float\* x, float\* y){**

**(\*x) += (\*y);**

**}**

**int main(){**

**int i;**

**struct Produto {**

**char nome[30]; /\* Nome do produto \*/**

**int codigo; /\* Codigo do produto \*/**

**double preco; /\* Preco do produto \*/**

**};**

**//Produto produtos[10];**

**struct Produto \*produtos;**

**produtos = (struct Produto\*)malloc(sizeof(struct Produto)\*2);**

**struct Produto produto1;**

**strcpy(produto1.nome,"Pé de Moleque");**

**produto1.codigo = 13205;**

**produto1.preco = 0.20;**

**struct Produto produto2;**

**strcpy(produto2.nome,"Cocada Baiana");**

**produto2.codigo = 15202;**

**produto2.preco = 0.50;**

**produtos[0] = produto1;**

**produtos[1] = produto2;**

**for(i=0; i<2; i++){**

**printf("Produto%d: \nNome:%s\nCodigo:%d\nPreco:%f\n",i,produtos[i].nome,produtos[i].codigo, produtos[i].preco);**

**printf("\n");**

**}**

**free(produtos);**

**return 0;**

**}**

**7.** Uma indústria faz a folha mensal de pagamentos de seus 80 empregados baseada no seguinte: existe uma tabela com os dados de cada funcionário (matrícula, nome e salário bruto).

Escreva um programa em C que leia e processe a tabela e emita, para cada funcionário, seu contracheque, cujo formato é dado a seguir:

Matrícula: \_\_\_\_\_\_\_\_

Nome: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Salário Bruto: \_\_\_\_\_\_\_\_\_

Dedução INSS: \_\_\_\_\_\_\_\_\_

Salário Liquido: \_\_\_\_\_\_\_\_\_\_

O desconto do INSS é de 12% do salário bruto.

O salário líquido é a diferença entre o salário bruto e a dedução do INSS.

**8.** Escreva um programa que, utilizando um vetor de n números inteiros (n<=100), faça as seguintes operações:

a) leia o tamanho do vetor;

b) leia os elementos do vetor

c) multiplique todos os elementos do vetor por 2;

d) exiba o vetor.

Obs: utilize a notação de ponteiros ao invés da notação de vetor.