

<b>Campus Cachoeiro de Itapemirim</b>	
<b>Curso Técnico em Informática</b>	
<b>Disciplina:</b> Programação 1	<b>Professor:</b> Rafael Vargas Mesquita
<b>Lista 20 de Exercícios – Ponteiros</b>	

## LISTA 20

1. Implemente um procedimento para calcular dois impostos relacionados ao salário de uma pessoa. Como uma função em C não pode retornar dois valores, vamos utilizar passagem por referência. Para tal lembre-se do conceito de ponteiros.

Os descontos a serem calculados são:

- INSS: 10% do salário;
- IR: 7,5% do salário;

Seu procedimento deve receber como parâmetros:

- Passados por valor: salário;
- Passados por referência: INSS e IR;

2. Já vimos que a utilização de ponteiros é importante quando precisamos que uma função retorne mais de um valor. Todavia podemos justificar a utilização de ponteiros também por outros motivos. Considere as duas situações a seguir, e comente a vantagem da solução com ponteiros em relação a solução sem ponteiros:

<b>Campus Cachoeiro de Itapemirim</b>	
<b>Curso Técnico em Informática</b>	
<b>Disciplina:</b> Programação 1	<b>Professor:</b> Rafael Vargas Mesquita
<b>Lista 20 de Exercícios – Ponteiros</b>	

a) Ponteiros ou Variáveis Locais

Solução com Ponteiros	Solução com Variáveis Locais
<pre>#include&lt;stdio.h&gt;  float soma(float *num1, float *num2){     return (*num1 + *num2); }  int main(){     float n1, n2, resultado;      printf("Digite o numero 1: ");     scanf("%f", &amp;n1);     printf("Digite o numero 2: ");     scanf("%f", &amp;n2);      resultado = soma(&amp;n1, &amp;n2);      printf("Soma = %f", resultado); }</pre>	<pre>#include&lt;stdio.h&gt;  float soma(float num1, float num2){     return (num1 + num2); }  int main(){     float n1, n2, resultado;      printf("Digite o numero 1: ");     scanf("%f", &amp;n1);     printf("Digite o numero 2: ");     scanf("%f", &amp;n2);      resultado = soma(n1, n2);      printf("Soma = %f", resultado); }</pre>

Obs.: Considere que variáveis do tipo float ocupam 8 bytes na memória do computador. E ponteiros 4 bytes.

<b>Campus Cachoeiro de Itapemirim</b>	
<b>Curso Técnico em Informática</b>	
<b>Disciplina:</b> Programação 1	<b>Professor:</b> Rafael Vargas Mesquita
<b>Lista 20 de Exercícios – Ponteiros</b>	

b) Ponteiros ou Variáveis Globais

Solução com Ponteiros	Solução com Variáveis Globais
<pre>#include&lt;stdio.h&gt;  void troca(float *pa, float *pb){     float temp;      temp = *pa;     *pa = *pb;     *pb = temp; }  float main(){     float a, b, c, d;     printf("Digite o valor de a: ");     scanf("%f", &amp;a);     printf("Digite o valor de b: ");     scanf("%f", &amp;b);     printf("Digite o valor de c: ");     scanf("%f", &amp;c);     printf("Digite o valor de d: ");     scanf("%f", &amp;d);      troca(&amp;a, &amp;b);     troca(&amp;c, &amp;d);      printf("\n\nImprimindo...\n");     printf("a: %f\n", a);     printf("b: %f\n\n", b);     printf("c: %f\n", c);     printf("d: %f\n\n", d); }</pre>	<pre>#include&lt;stdio.h&gt;  float a, b, c, d;//Globais  void troca_ab(){     float temp;      temp = a;     a = b;     b = temp; }  void troca_cd(){     float temp;      temp = c;     c = d;     d = temp; }  float main(){     printf("Digite o valor de a: ");     scanf("%f", &amp;a);     printf("Digite o valor de b: ");     scanf("%f", &amp;b);     printf("Digite o valor de c: ");     scanf("%f", &amp;c);     printf("Digite o valor de d: ");     scanf("%f", &amp;d);      troca_ab();     troca_cd();      printf("\n\nImprimindo...\n");     printf("a: %f\n", a);     printf("b: %f\n\n", b);     printf("c: %f\n", c);     printf("d: %f\n\n", d); }</pre>

<b>Campus Cachoeiro de Itapemirim</b>	
<b>Curso Técnico em Informática</b>	
<b>Disciplina:</b> Programação 1	<b>Professor:</b> Rafael Vargas Mesquita
<b>Lista 20 de Exercícios – Ponteiros</b>	

3. Responda verdadeiro (V) ou falso (F) para as seguintes afirmativas com relação à aritmética de ponteiros. Justifique as falsas:
- a) ( ) Após a atribuição `pa=&a[0]`; `pa` e `a` possuem valores idênticos;
  - b) ( ) A atribuição `pa=&a[0]`; pode ser escrita como `pa=a`;
  - c) ( ) Uma referência a `a[i]` pode ser escrita como `*(a+i)`
  - d) ( ) `&a[i]` e `a+i` são idênticos
  - e) ( ) `a+i` é o endereço do *i*-ésimo elemento após `a`
  - f) ( ) `pa[i]` é idêntico a `*(pa+i)`
  - g) ( ) `pa=a` é uma operação válida
  - h) ( ) `pa++` é uma operação válida
  - i) ( ) `a=pa` é uma operação válida
  - j) ( ) `a++` é uma operação válida
4. Escrever um programa para ler uma frase qualquer do teclado e imprimir, esta mesma frase, um caracter por vez. A frase deve ser armazenada inicialmente em um vetor de no máximo 255 posições. Para mostrar a frase, caracter por caracter, utilize um ponteiro.