

Exercícios sobre Passagem por Referência

1. Altere as chamadas de funções abaixo de forma que a função chamada não tenha retorno e, ao mesmo tempo, a variável que antes receberia o valor retornado possa ser alterado diretamente na função chamada.

ano = lerDado();	lerDado(&ano);
a = recebendoUmValor();	recebendoUmValor(&a);
idade = calcularIdade(ano);	calcularIdade(ano, &idade);
b = calcularAlgo(a);	calcularAlgo(a, &b);
c = somar(a,b);	somar(a, b, &c);
c = fazAlgo(a,b);	fazAlgo(a, b, &c);
d = getRA();	getRA(&d);
e = getValor();	getValor(&e);

2. Altere as funções abaixo, de forma que não retornem mais valores e, ao mesmo tempo, atribuem o valor definido diretamente no endereço de memória recebido.

<pre>int retornandoUmNumero(){ int a; a = 40; return a; }</pre>	<pre>int retornaValorAleatorio(){ int valor; srand(time(NULL)); valor = rand()%10; return valor; }</pre>
<pre>int calcularIdade(int ano){ int idade; idade = 2019-ano; return idade; }</pre>	<pre>int calcularAno(int idade){ int ano; ano = 2019-idade; return ano; }</pre>
<pre>int somar(int a, int b){ int res; res = a+b; return res; }</pre>	<pre>int subtrair(int a, int b){ int res; res = a-b; return res; }</pre>

<pre>void retornandoUmNumero(int *a){ *a = 40; }</pre>	<pre>void retornaValorAleatorio(int *valor){ srand(time(NULL)); *valor = rand()%10; }</pre>
--	---

<pre>void calcularIdade(int ano, int *idade){ *idade = 2019-ano; }</pre>	<pre>void calcularAno(int idade, int *ano){ *ano = 2019-idade; }</pre>
<pre>void somar(int a, int b, int *res){ *res = a+b; }</pre>	<pre>void subtrair(int a, int b, int *res){ *res = a-b; }</pre>

3. Altere os códigos abaixo de forma que eles não retornem mais valores. No lugar, use ponteiros.

<pre>int main(int argc, char** argv) { int valor; int quadrado; printf("\nDigite um valor:"); scanf("%d",&valor); calcularQuadrado(valor, &quadrado); printf("\n%d ao quadrado = %d", valor, quadrado) return 0; }</pre>	<pre>void calcularQuadrado(int valor, int *quadrado){ *quadrado = valor * valor; }</pre>
<pre>int main(int argc, char** argv) { float celsius; float kelvin; printf("\nDigite a temperatura em Celsius:"); scanf("%f",&celsius); converterCparaK(celsius, &kelvin); printf("\n%.2f Celsius = %.2f Kelvin",celsius,kelvin); return 0; }</pre>	<pre>void converterCparaK(float celsius, float *kelvin){ *kelvin = c + 273.15f; }</pre>

<pre> int main(int argc, char** argv) { float celsius; float kelvin; printf("\nDigite a temperatura em Celsius:"); scanf("%f",&celsius); converterCparaK(celsius, &kelvin); printf("\n%.2f Celsius = %.2f Kelvin",celsius,kelvin); return 0; } </pre>	<pre> void converterCparaK(float celsius, float *kelvin){ *kelvin = c + 273.15f; } </pre>
---	---

<pre> int main(int argc, char** argv){ int valor1, valor2; int resultado; printf("\nDigite um valor: "); scanf("%d",&valor1); printf("\nDigite outro valor:"); scanf("%d",&valor2); multiplicar(valor1, valor2, &resultado); printf("\n%d x %d = %d ",valor1, valor2, resultado); return 0; } </pre>	<pre> void multiplicar(int valor1, int valor2, int *resultado){ *resultado = a*b; } </pre>
---	--

4. Para cada código do exercício 3, preencha a tabela de endereços, supondo que a primeira variável seja sempre alocada no endereço 55FF20.

a.

Identificador	Endereço	Valor
valor	0055FF20	3
quadrado	0055FF1C	9
val	0055FF18	3
quad	0055FF14	9

Tabela algoritmo antigo

Identificador	Endereço	Valor
valor	0055FF20	3
quadrado	0055FF1C	9
valor	0055FF18	3
quadrado	0055FF14	0055FF1C

Tabela algoritmo corrigido

b.

Identificador	Endereço	Valor
celsius	0055FF20	0
kelvin	0055FF1C	273.15
ε	0055FF18	0
k	0055FF14	273.15

Tabela algoritmo antigo

Identificador	Endereço	Valor
celsius	0055FF20	0
kelvin	0055FF1C	273.15
celsius	0055FF18	0
kelvin	0055FF14	0055FF1C

Tabela algoritmo corrigido

c.

Identificador	Endereço	Valor
celsius	0055FF20	0
kelvin	0055FF1C	273.15
ε	0055FF18	0
k	0055FF14	273.15

Tabela algoritmo antigo

Identificador	Endereço	Valor
celsius	0055FF20	0
kelvin	0055FF1C	273.15
celsius	0055FF18	0
kelvin	0055FF14	0055FF1C

Tabela algoritmo corrigido

d.

Identificador	Endereço	Valor
valor1	0055FF20	2
valor2	0055FF1C	5
resultado	0055FF18	10
a	0055FF14	2
b	0055FF10	5
res	0055FF0C	10

Tabela algoritmo antigo

Identificador	Endereço	Valor
valor1	0055FF20	2
valor2	0055FF1C	5
resultado	0055FF18	10
valor1	0055FF14	2
valor2	0055FF10	5
resultado	0055FF0C	0055FF18

Tabela algoritmo corrigido