

Funções

- Variáveis globais

Variáveis globais são declaradas fora de todas as funções do programa. Elas são conhecidas e podem ser alteradas por todas as funções do programa. Quando uma função tem uma variável local com o mesmo nome de uma variável global a função dará preferência à variável local. Vamos ver um exemplo:

```
int z,k;
func1 (...)
{
    int x,y;
    ...
}
func2 (...)
{
    int x,y,z;
    ...
    z=10;
    ...
}
main ()
{
    int count;
    z=7;
    func2(...);
    ...
}
```

Exercício:

Analise o seguinte programa e indique o que será impresso na saída padrão.

```
#include <stdio.h>
int num;
int func(int a, int b)
{
    a = (a+b)/2;
    num -= a+1;
    return a;
}
main()
{
    int first = 0, sec = 50;
    num = 10;
    printf("\nnum antes = %d\tfirst antes = %d\tsec antes = %d", num,
first, sec);
    num += func(first, sec);
    printf("\nnum depois = %d\tfirst depois = %d\tsec depois = %d",
num, first, sec);
}
```

Funções

Exercício:

Escreva o código fonte de um programa, na linguagem C, que manipule um vetor de inteiros com dez elementos. O programa deve possuir uma função responsável pela inicialização do vetor e outra função que efetue a impressão do vetor na saída padrão. Por meio das funções mencionadas, o programa deve inicializar o vetor e em seguida retorná-lo no monitor.

Funções

- Passagem de parâmetros por valor e passagem por referência

Vimos que quando chamamos uma função os parâmetros formais da função recebem uma copia dos valores dos parâmetros que são passados para a função. Isto quer dizer que não são alterados os valores que os parâmetros têm fora da função.

Funções

- Passagem de parâmetros por valor e passagem por referência (continuação)

Este tipo de chamada de função é denominado chamada por valor. Isto ocorre porque são passados para a função apenas os valores dos parâmetros e não os próprios parâmetros. Veja o exemplo a seguir:

```
#include <stdio.h>
float sqr (float num)
{
    num=num*num;
    return num;
}
main ()
{
    float num,sq;
    printf ("Entre com um numero: ");
    scanf ("%f",&num);
    sq=sqr(num);
    printf ("\n\nO numero original e: %f\n",num);
    printf ("O seu quadrado vale: %f\n",sq);
}
```

Funções

- Passagem de parâmetros por valor e passagem por referência (continuação)

Outro tipo de passagem de parâmetros para uma função ocorre quando alterações nos parâmetros formais, dentro da função, alteram os valores dos parâmetros que foram passados para a função. Este tipo de chamada de função tem o nome de "chamada por referência".

Funções

- Passagem de parâmetros por valor e passagem por referência (continuação)

O C só faz chamadas por valor. Isto é bom quando queremos usar os parâmetros formais à vontade dentro da função, sem termos que nos preocupar em estar alterando os valores dos parâmetros que foram passados para a função. Mas, isto também pode ser ruim, porque, às vezes, podemos querer que mudanças nos valores dos parâmetros formais reflitam nos parâmetros que foram passados para a função.

Funções

- Passagem de parâmetros por valor e passagem por referência (continuação)

Quando queremos alterar as variáveis que são passadas para uma função, nós podemos declarar seus parâmetros formais como sendo *ponteiros*. Os ponteiros são a "referência" que precisamos para poder alterar a variável fora da função. O único inconveniente é que, quando usarmos a função, teremos de lembrar de colocar um **&** na frente das variáveis que estivermos passando para a função. Veja um exemplo:

```
#include <stdio.h>
void incrementa (int *a,int b)
{
    *a+=b;
}
main ()
{
    int num;
    num=100;
    printf ("\nValor de num antes da chamada %s%d",
"da funcao: ", num);
    incrementa (&num,50);
    printf ("\nValor de num depois da chamada %s%d",
"da funcao: ", num);
}
```

Funções

Exercício:

Construa um programa em 'C' que possua uma função responsável por fazer a troca de valores entre duas variáveis. O programa deve receber dois valores inteiros, através da entrada padrão, armazená-los em variáveis, e com a utilização da função mencionada trocar o conteúdo das variáveis e depois retorná-las na saída padrão.

Funções

Exercício:

Escreva o código fonte de um programa, na linguagem C, que manipule um vetor de inteiros com dez elementos. O programa deve possuir uma função responsável pela inicialização do vetor e outra função que efetue a impressão do vetor na saída padrão. Por meio das funções mencionadas, o programa deve inicializar o vetor e em seguida retorná-lo no monitor. **Obs. o programa não pode possuir variáveis globais.**