

Funções

- Protótipos de funções

Protótipos são declarações de funções. Isto é, você declara uma função que irá usar. O compilador toma então conhecimento do formato daquela função antes de compilá-la. Possibilitando assim, antes da compilação da função, a validação da utilização da mesma. Um protótipo tem o seguinte formato:

TipoDeRetorno NomeDaFunção (DeclaraçãoDeParâmetros);

onde o *TipoDeRetorno*, o *NomeDaFunção* e a *DeclaraçãoDeParâmetros* são os mesmos que você pretende usar quando realmente escrever a função.

```
#include <stdio.h>
float Square (float a);
int main ()
{
    float num;
    printf ("Entre com um numero: ");
    scanf ("%f",&num);
    num=Square(num);
    printf ("\n\nO seu quadrado vale: %f\n",num);
    return 0;
}
float Square (float a)
{
    return (a*a);
}
```

Funções

- Arquivos headers

Arquivos headers, ou arquivos cabeçalhos, são aqueles que temos mandado o compilador incluir no início de nossos exemplos e que sempre terminam em **.h**. Estes arquivos não contêm os códigos das funções. Eles só contêm *protótipos* de funções.

Funções

O corpo das funções cujos protótipos estão no arquivo-cabeçalho, no caso das funções do próprio C, já estão compilados e normalmente são incluídos no programa no instante da "linkagem". Este é o instante em que todas as referências a funções cujos códigos não estão nos arquivos fontes são resolvidas, buscando este código nos arquivos de bibliotecas.

Funções

- Arquivos headers (continuação)

Se você criar algumas funções que queira aproveitar em vários programas futuros, ou módulos de programas, você pode escrever arquivos-cabeçalhos e incluí-los também.

Vamos supor que a função 'int EPar(int a)', vista anteriormente, seja importante em vários programas, e desejemos declará-la num módulo separado.

Funções

- Arquivos headers (continuação)

No arquivo de cabeçalho chamado por exemplo de “funcao.h” teremos a seguinte declaração:

```
int EPar(int a);
```

Funções

- Arquivos headers (continuação)

O código da função será escrito num arquivo a parte. Vamos chamá-lo de “funcao.c”. Neste arquivo teremos a definição da função:

```
int EPar (int a)
{
    return (a%2?0:1);
}
```

Funções

- Arquivos headers (continuação)

Por fim, no arquivo do programa principal teremos a função main(). Vamos chamar este arquivo aqui de “princip.c”.

```
#include <stdio.h>
#include "funcao.h"
main ()
{
    int num;
    printf ("Entre com numero: ");
    scanf ("%d",&num);
    if (EPar(num))
        printf ("\nO numero e par.\n");
    else
        printf ("\nO numero e impar.\n");
}
```


Funções

- Arquivos headers (continuação)

Este programa poderia ser compilado usando a seguinte linha de comando para o gcc:

```
gcc princip.c funcao.c -o saida
```

onde “saida” seria o nome do arquivo executável gerado.

Funções

Exercício:

Escreva um programa que faça uso da função `EDivisivel(int a, int b)`, criada anteriormente. Além desta função o programa deverá usar: `Max(int a, int b)`, que retorna o maior dos parâmetros; `VMedio(int a, int b)`, que retorna o valor médio (inteiro) entre `a` e `b`, você deverá escrever estas funções. Organize o seu programa em três arquivos: o arquivo `exercicio.c`, conterá o programa principal que deve se utilizar das funções descritas; o arquivo `func.c` conterá o corpo das funções; o arquivo `func.h` conterá os protótipos das funções. Compile os arquivos e gere o executável.