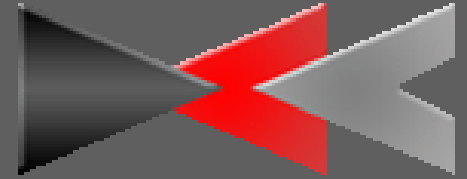




**DCC107**



# **Laboratório de Programação II**

**Programas com vários arquivos em C  
(criação de biblioteca)**

# Programas com vários arquivos fontes em C



2

- ❑ Arquivos cabeçalhos são aqueles que mandamos o compilador incluir no início de um programa e que sempre terminam em `.h` (*header*, em inglês);
- ❑ Exemplos: `stdio.h`, `conio.h`, `string.h`;
- ❑ Estes arquivos não possuem códigos fontes. Eles contêm protótipos de funções e/ou tipos de dados;
- ❑ O compilador lê estes protótipos e gera o código correto;
- ❑ O corpo das funções cujos protótipos estão no arquivo cabeçalho já estão compiladas.

# Programas com vários arquivos fontes em C



3

- Quando se deseja aproveitar algumas funções em vários programas, ou em módulos de programas, você pode escrever arquivos cabeçalhos e incluí-los também;
- Suponha que as funções sejam importantes em vários programas:
  - `int EPar(int a):` retorna 1 se a é par e 0 se é ímpar;
  - `EDivisivel(int a, int b):` retorna 1 se o resto da divisão de a por b for zero;
  - `int Square(int a):` retorna o quadrado de a.

# Programas com vários arquivos fontes em C



4

- No arquivo `funcao.h` fazemos a seguinte declaração:

```
//retorna 1 se a eh par e 0 se eh impar
int EPar(int a);
//retorna 1 se o resto da divisão de a por b for 0
int EDivisivel(int a, int b);
//retorna o quadrado de a
int Square(int a);
```

# Programas com vários arquivos fontes em C



5

- O código das funções será escrito num arquivo a parte chamado `funcao.c`:

```
#include "funcao.h"
```

```
int Square(int a)
{
```

```
    return (a * a); //retorna o quadrado de a
```

```
}
```

```
int EDivisivel(int a, int b)
```

```
{
```

```
    return (a % b == 0 ? 1 : 0);
```

```
//se o resto da divisao de a por b for 0 retorna 1,  
caso contrario 0
```

```
} //continua...
```

Este **include** é necessário, pois:

- 1) Podem existir definições que são necessárias na **funcao.c**
- 2) Garante que as funções implementadas correspondem às funções do cabeçalho.

Expressão condicional:  
retorna 1 se `a % b` for verdadeiro e 0 caso contrário

# Programas com vários arquivos fontes em C



6

## □ ... continuação arquivo funcao.c:

```
int EPar(int a) //retorna 1 se a eh par e 0 se eh impar
{ //verifica se a eh divisivel por dois
    return EDivisivel(a, 2);
}
```

## □ Por fim, no arquivo do programa principal main.c, colocamos o programa principal:

```
#include <stdio.h>
#include "funcao.h"
void main()
{
    int num, num2;
    //continua...
```

Usa-se aspas

# Programas com vários arquivos fontes em C



7

## □ ... continuação arquivo main.c:

```
printf("Numero 1: "); scanf("%d", &num); //le num
printf("Numero 2: "); scanf("%d", &num2); //le num2
//quadrado de num
printf("\nO quadrado de %d eh %d", num, Square(num));
//verifica se num eh par
if (EPar(num))
    printf("\nO numero %d eh par.\n", num);
else
    printf("\nO numero %d eh impar.\n", num);
//verifica se num eh divisivel por num2
if (EDivisivel(num, num2))
    printf("\n%d eh divisivel por %d.", num, num2);
else
    printf("\n%d nao eh divisivel por %d", num, num2);
}
```

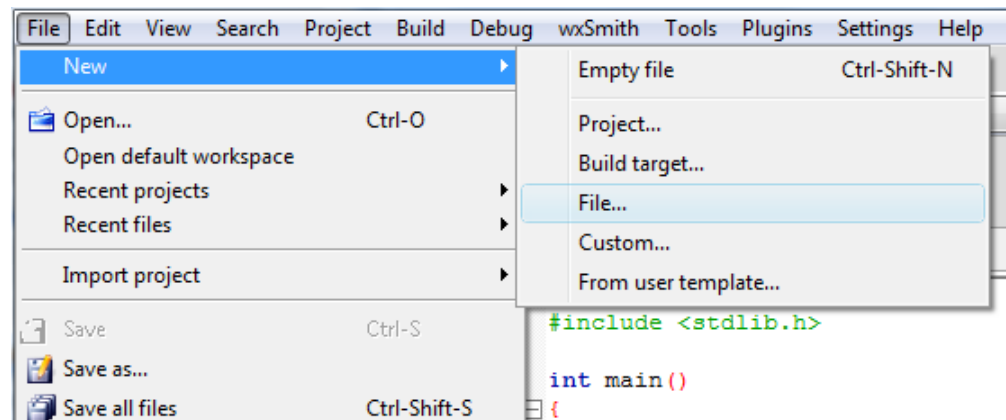
# Programas com vários arquivos fontes em C



8

## □ No **Code::Blocks**:

- Criar um novo projeto: File > New > Project... > Console application...
- Criar um novo arquivo (cabeçalho): File > New > File...





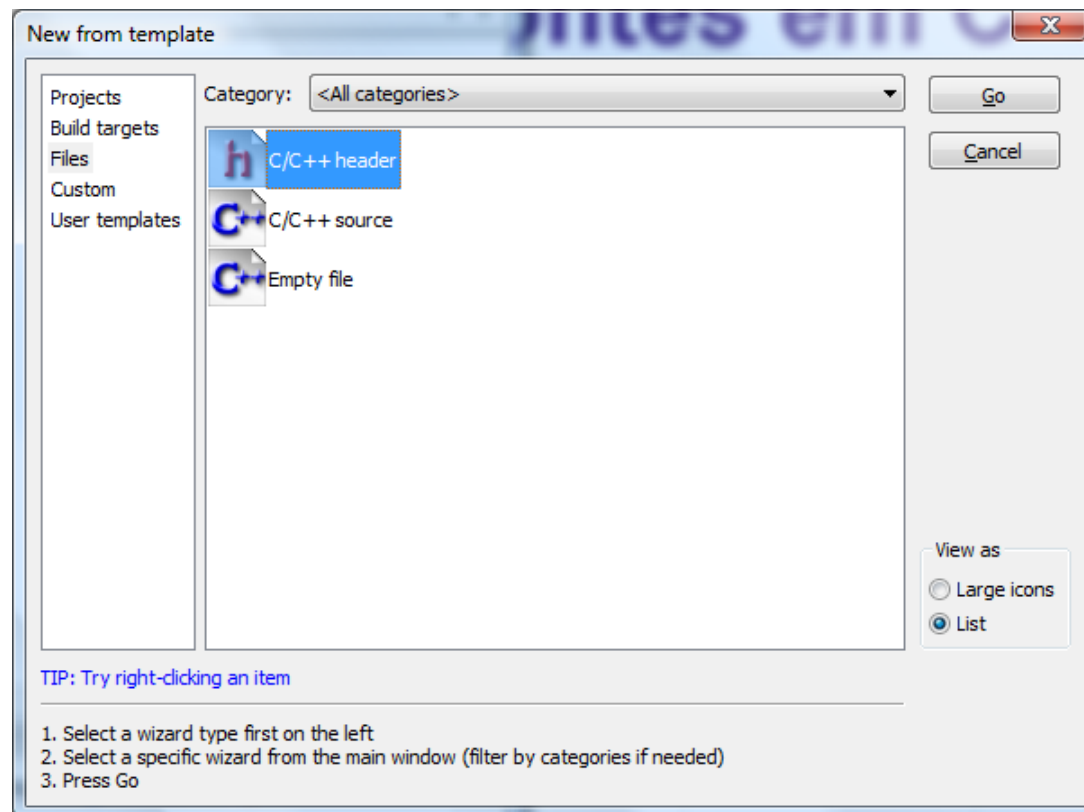
# Programas com vários arquivos fontes em C



9

## □ No **Code::Blocks**:

■ Escolha a opção *C/C++ header* e clique em **Go**:



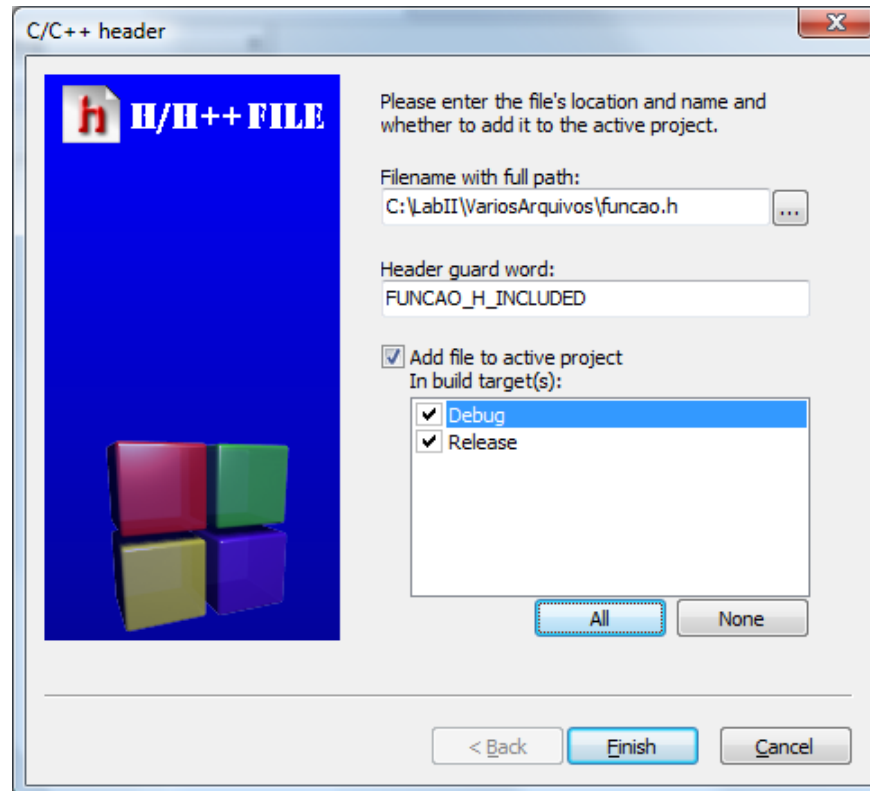
# Programas com vários arquivos fontes em C



10

## □ No **Code::Blocks**:

- ▣ Selecione a pasta, digite *funcao.h* para o nome do arquivo, clique em **All** e depois em **Finish**:

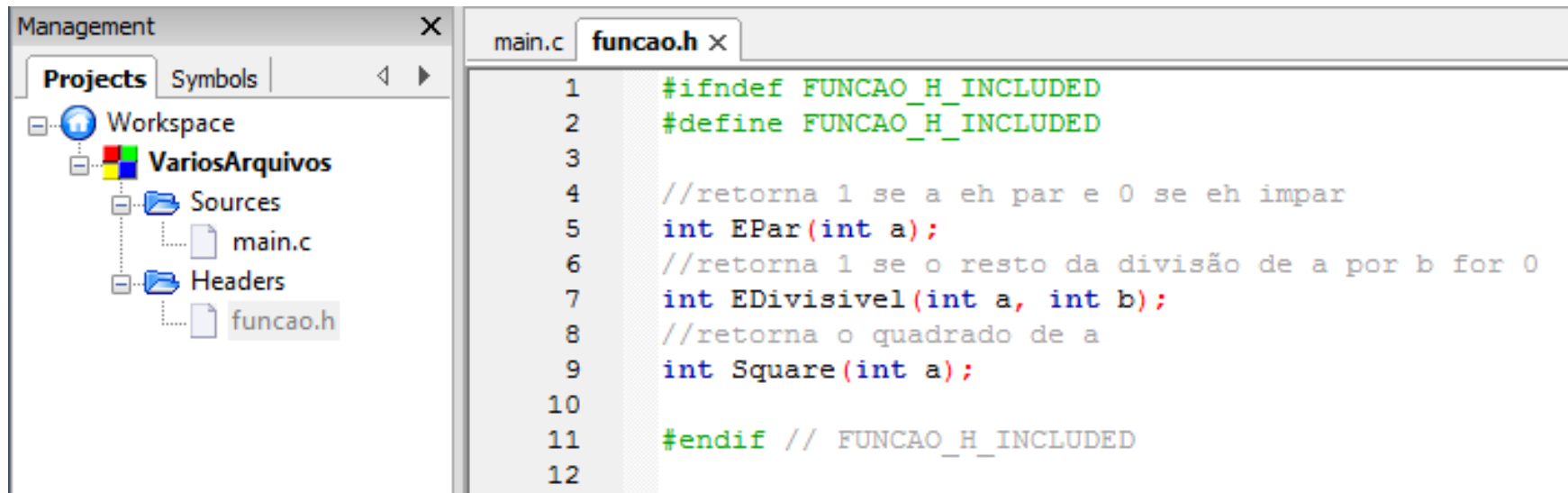


# Programas com vários arquivos fontes em C



11

- No **Code::Blocks**:
  - ▣ Digite o código de `funcao.h` e salve-o:



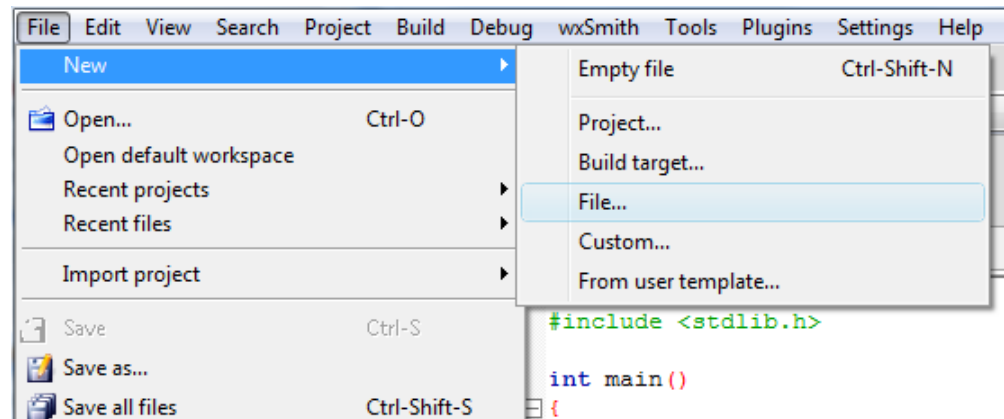
# Programas com vários arquivos fontes em C



12

## □ No **Code::Blocks**:

- ▣ Criar um novo arquivo (fonte): File > New > File...



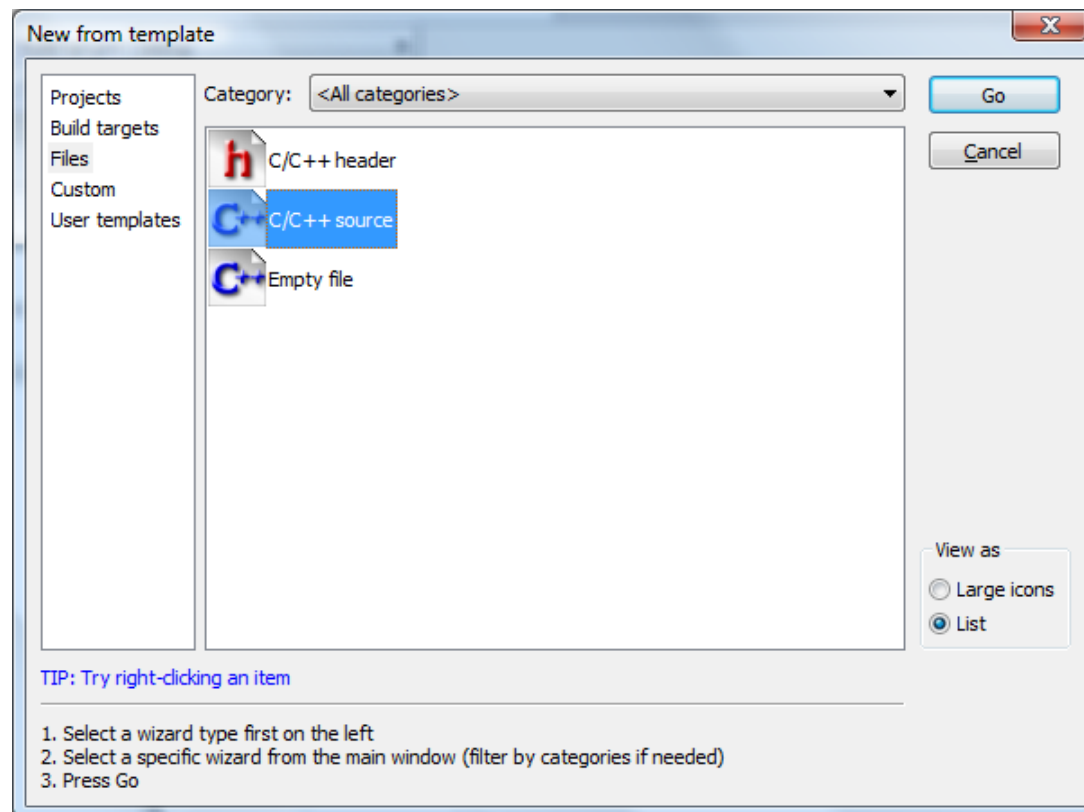
# Programas com vários arquivos fontes em C



13

## □ No **Code::Blocks**:

■ Escolha a opção *C/C++ source* e clique em **Go**:

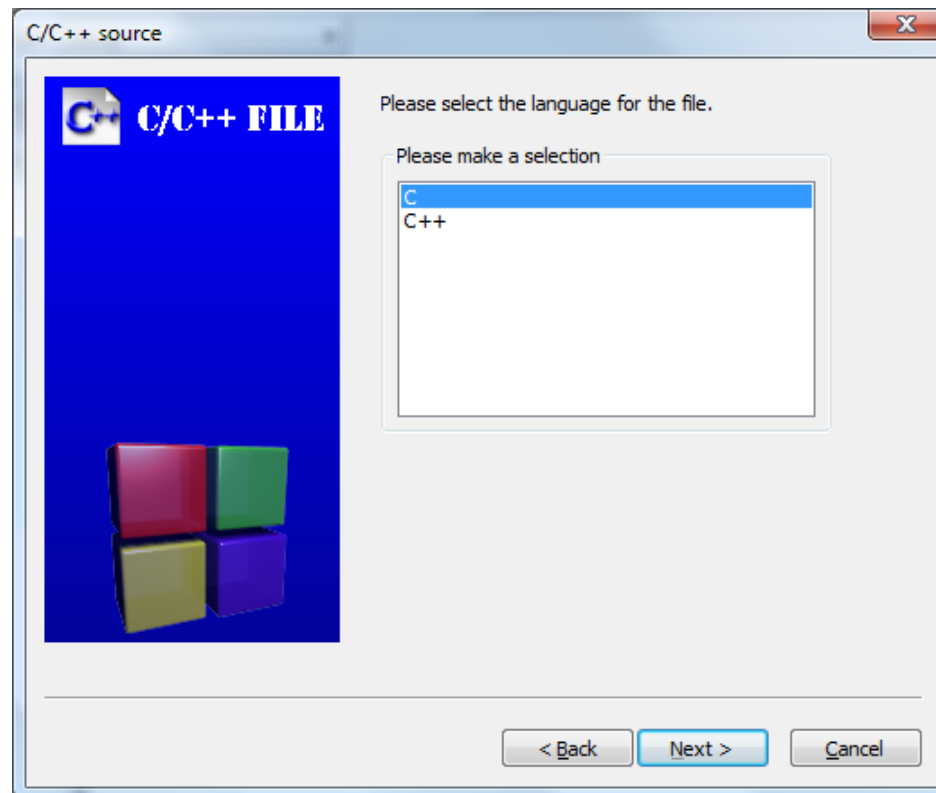


# Programas com vários arquivos fontes em C



14

- No **Code::Blocks**:
  - ▣ Escolha a opção *C* e clique em **Next**:



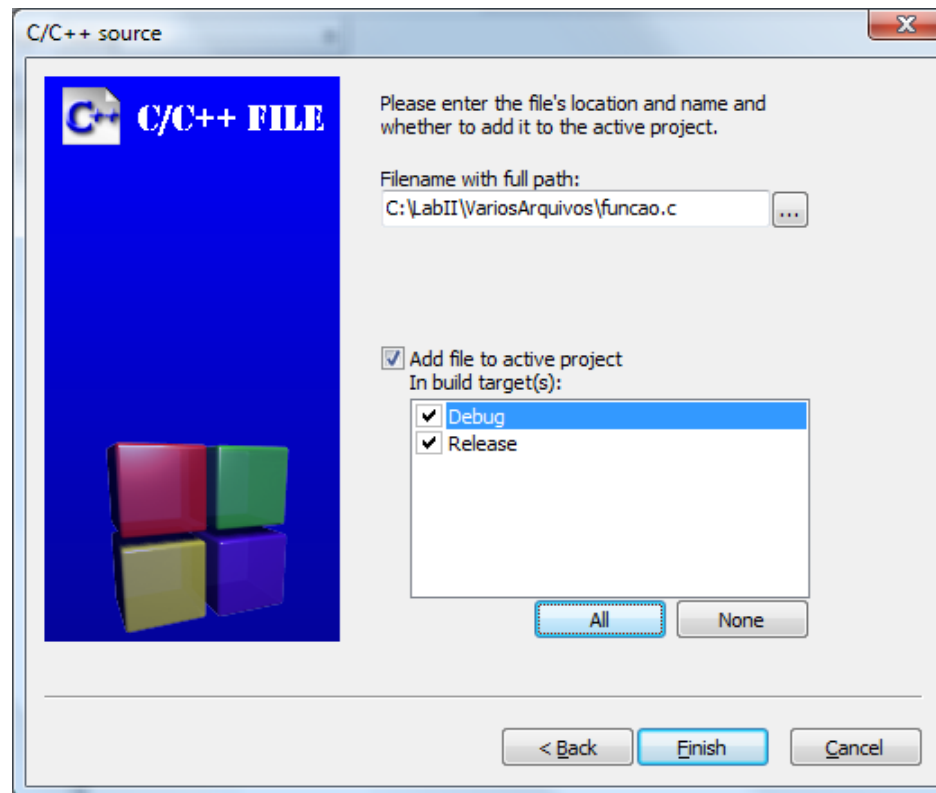
# Programas com vários arquivos fontes em C



15

## □ No **Code::Blocks**:

- ▣ Selecione a pasta, digite *funcao.c* para o nome do arquivo, clique em **All** e depois em **Finish**:



# Programas com vários arquivos fontes em C



16

## □ No Code::Blocks:

▣ Digite o código fonte da *funcao.c* e salve-o:

```
1  #include "funcao.h"
2
3  int Square (int a)
4  {
5      return (a * a); //retorna o quadrado de a
6  }
7
8  int EDivisivel(int a, int b)
9  {
10     return (a % b == 0 ? 1 : 0);
11     /*se o resto da divisao de a por b for 0 retorna 1,
12     caso contrario 0*/
13 }
14
15 int EPar (int a) //retorna 1 se a eh par e 0 se eh impar
16 { //verifica se a eh divisivel por dois
17     return EDivisivel(a, 2);
18 }
19
```



# Programas com vários arquivos fontes em C



17

## □ No **Code::Blocks**:

- ▣ Coloque o código do programa principal no arquivo *main.c* e salve-o :

```
1  #include <stdio.h>
2  #include "funcao.h"
3  void main()
4  {
5      int num, num2;
6      printf("Numero 1: "); scanf("%d", &num); //le num
7      printf("Numero 2: "); scanf("%d", &num2); //le num2
8
9      //quadrado de num
10     printf("\nO quadrado de %d eh %d", num, Square(num));
11
12     //verifica se num eh par
13     if (EPar(num))
14         printf("\nO numero %d eh par.\n", num);
15     else
16         printf("\nO numero %d eh impar.\n", num);
17
18     //verifica se num eh divisivel por num2
19     if (EDivisivel(num, num2))
20         printf("\n%d eh divisivel por %d.", num, num2);
21     else
22         printf("\n%d nao eh divisivel por %d", num, num2);
23 }
```

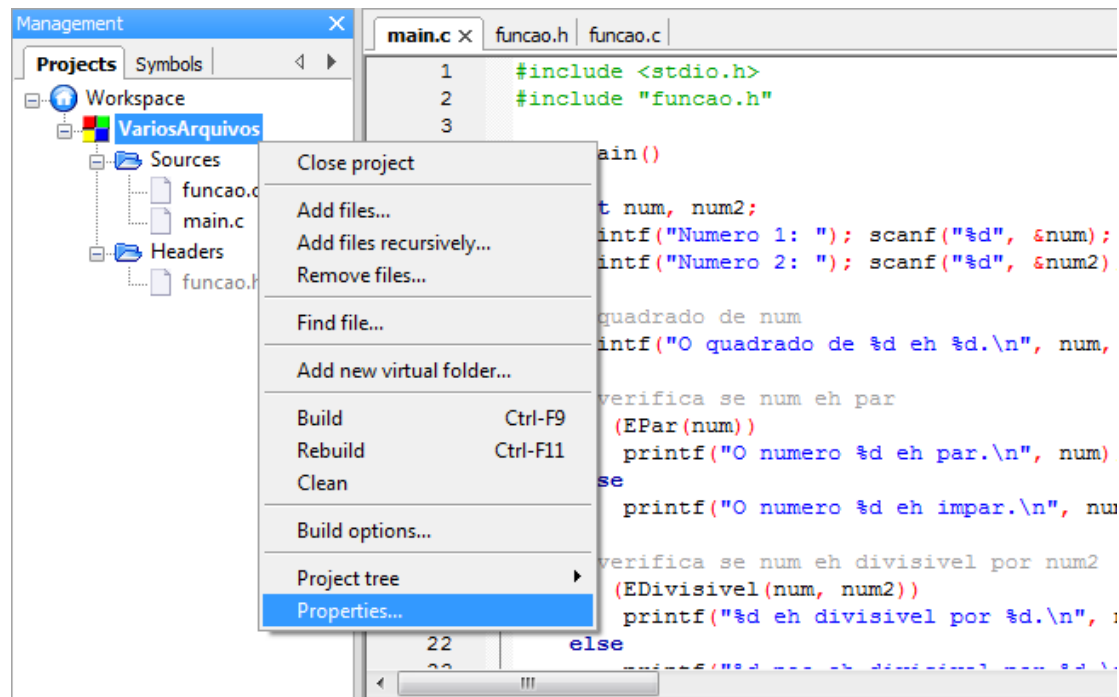
# Programas com vários arquivos fontes em C



18

## □ Forma Alternativa 1:

- Com o botão direito do mouse, clique sobre o nome do projeto (no caso “VariosArquivos”) e selecione *Properties...*



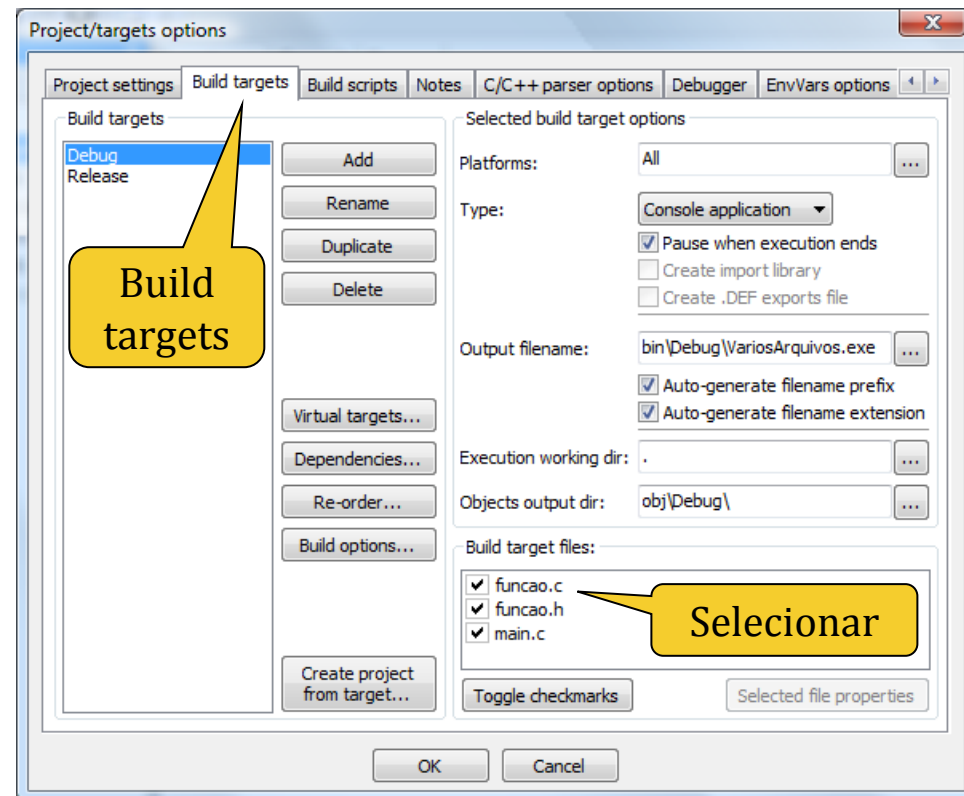
# Programas com vários arquivos fontes em C



19

## □ Forma Alternativa 1:

- Na aba *Build targets* opção *Build targets files* marcar a opção **funcao.c** e **funcao.h** e clique em OK;
- Isso é necessário para que o compilador compile estes arquivos fonte;
- Deve-se fazer para todo arquivo fonte do projeto.



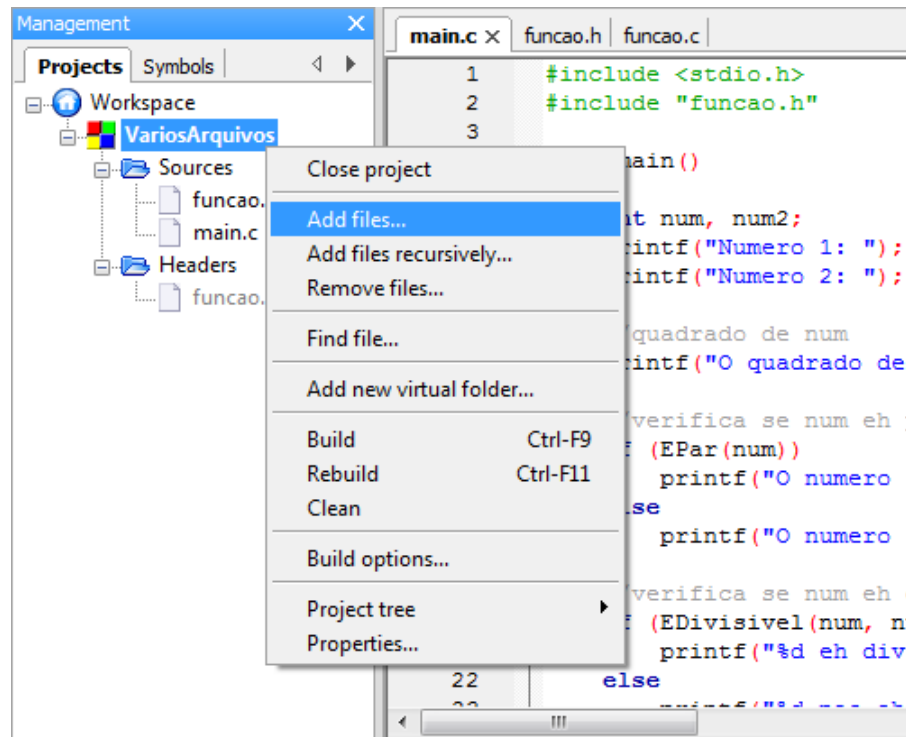
# Programas com vários arquivos fontes em C



20

## □ Forma Alternativa 2:

- ▣ Com o botão direito do mouse, clique sobre o nome do projeto e depois na opção *Add files...*



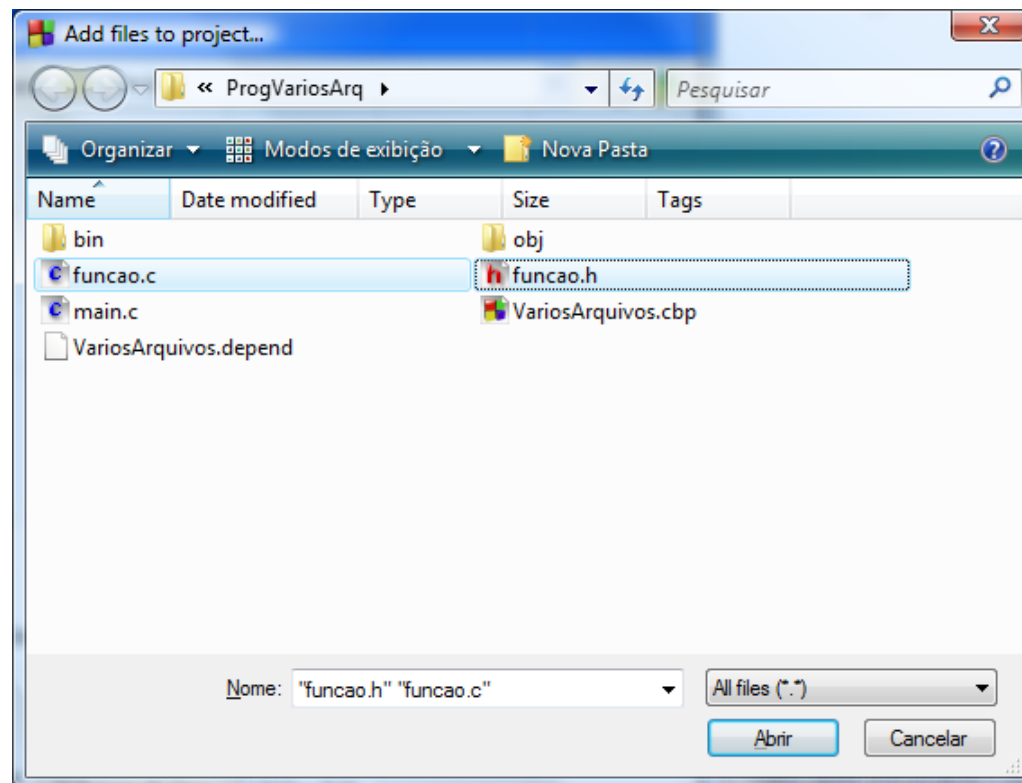
# Programas com vários arquivos fontes em C



21

## □ Forma Alternativa 2:

- Escolha os arquivos **funcao.c** e **funcao.h** e clique em **Abrir**:



# Programas com vários arquivos fontes em C



22

- **Forma Alternativa 2:**
  - ▣ Marque a opção **Select All** e clique em **OK**:

