

PRC

Dados, Atribuição, Entrada e

Operações Aritméticas

Chessman Kennedy Faria Corrêa

ckennyfc@gmail.com

Programa Exemplo

- Crie um novo projeto, escreva e compile o programa a seguir.

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int numero1;
6      numero1 = 10;
7      printf("Numero: %d\n", numero1);
8
9      return 0;
10 }
```

Dados

- Os **dados** são os **valores** processados pelo programa.
- Os dados possuem **tipo**. Por exemplo, existem dados inteiros, dados reais, etc.
- Os dados são representados por **variáveis**.
- Uma variável é mantida em uma determinada parte da **memória** do computador.

Variável

- É usada para representar um dado.
- Resulta na reserva de um espaço da memória para armazenar um valor correspondente ao dado.
- É uma forma de acesso e manipulação de uma determinada posição de memória.

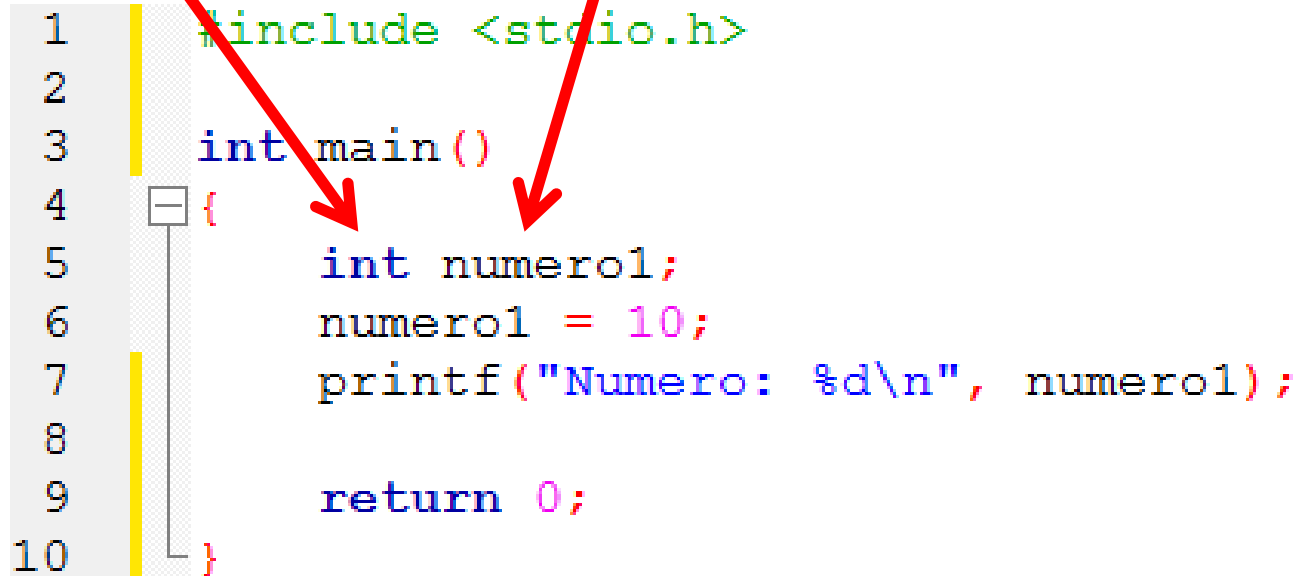
```
1  #include <stdio.h>
2
3  int main()
4  {
5      int numero1;
6      numero1 = 10;
7      printf("Numero: %d\n", numero1);
8
9      return 0;
10 }
```

Declaração de
uma variável
em C

Variável

Tipo da
variável

Nome da
variável



```
1  #include <stdio.h>
2
3  int main()
4  {
5      int numero1;
6      numero1 = 10;
7      printf("Numero: %d\n", numero1);
8
9      return 0;
10 }
```

Variável

- O nome de uma variável deve começar sempre com uma letra.
- Um programa pode ter qualquer quantidade de variáveis.
- **Dica 1:** declare as variáveis no início do programa.
- **Dica 2:** use nomes significativos.
- **Dica 3:** comece os nomes de variáveis com letras minúsculas (é uma convenção usada frequentemente).
- **Dica 4:** use letra maiúscula no início de cada palavra do nome.
 - quantidadeFuncionarios, totalPagamento, notaMinimaAprovacao
- **Dica 5:** Separar as declarações das variáveis e os comandos com uma linha em branco.

Atenção



- Letras maiúsculas e minúsculas em C++ são diferentes.
- Sendo assim, no exemplo:

```
float Salario, salario, SALARIO;
```

tem-se três variáveis diferentes.

- **Cuidado para não criar a mesma variável com diferentes combinações de letras maiúsculas e minúsculas.**

Tipos de Dados

- Existem vários **tipos de dados**.
- No C, cada tipo de dado é representado por uma palavra, chamada de **palavra reservada (ou palavra chave)**.
- Veja os tipos de dados simples do C (você vai aprender outros depois).

Tipos de Dados Simples do C/C++

Tipo	C/C++	Tamanho (bytes)	Faixa de Valores
Inteiro	int	4	Com sinal: -2.147.483.648 a 2.147.483.647 Sem sinal: 0 a 4.294.967.295
Inteiro curto	short	2	Com sinal: -32.768 ta 32.767 Sem sinal: 0 to 65.535
Ponto flutuante	float	4	+/- 3.4e +/- 38 (~7 dígitos)
Ponto flutuante de dupla precisão	double	8	+/- 1.7e +/- 308 (~15 dígitos)
Caractere	char	1	Tabela ASCII.

Declaração de Números sem Sinal

- Usar a palavra reservada **unsigned** antes do tipo.
- Exemplos:

```
unsigned int idade;
```

```
unsigned float salario;
```

```
unsigned double totalVendas;
```

```
unsigned short quantidadeAlunosTurma;
```

Atribuição

- Serve para colocar o valor de um dado em uma variável.
- Corresponde a colocar o valor na posição de memória representada pela variável.
- O valor ser atribuído a uma variável deve ser compatível com o seu tipo. Por exemplo, não se pode atribuir um valor real a uma variável do tipo inteiro!

```
3  int main()  
4  {  
5  
6      int numero1;  
7      numero1 = 10;  
8      printf("Numero: %d\n", numero1);  
9  
10     return 0;  
11 }
```

Atribuição em C (lê-se
numero1 recebe dez.

Atribuição

- Utiliza-se o "=" ;
- Exemplos:
totalAlunos = 0;
sexo = 'M';
quantidade = quantidade + 1;
salario = 350.00;

Saída de Valores em Variáveis

- Para a saída de um números é realizada usando a formatação **%d** na string de saída.
- O segundo parâmetro do printf é o valor ou a variável do tipo inteiro.

The diagram illustrates the use of the `printf` function to print an integer value. It features a code editor window on the left with line numbers 3 to 10. The code is as follows:

```
3 int main()  
4 {  
5     int numero1;  
6     numero1 = 10;  
7     printf("Numero: %d\n", numero1);  
8  
9     return 0;  
10 }
```

Two callout boxes provide additional context:

- A box labeled "Local onde valor inteiro deve ser exibido." (Local where integer value must be displayed) has a red arrow pointing to the `numero1` variable in the `printf` statement.
- A box labeled "Variável que contem o valor e ser exibido." (Variable that contains the value and will be displayed) has a red arrow pointing to the `numero1` variable in the assignment statement `numero1 = 10;`.

Saída de Valores Armazenados em Variáveis

Tipo de Dado	Formatação	Exemplo
Caractere	%c	Print ("Sexo: %c", sexo);
Inteiro	%d	printf("Idade: %d\n", idade);
Real	%f	printf("Salario: %f\n", salario);
Real definindo a quantidade de casas decimais a serem mostradas na tela.	%.nf (n = quantidade de dígitos decimais)	Printf("Salario: %.2f\n", salario);
String (cadeia de caracteres)	%s	printf("Nome: %s", nome);

- **Atenção:** Quando %.nf é usado:
 - O valor mostrado na tela não corresponde necessariamente ao valor que está armazenado na variável.
 - O valor mostrado na tela é arredondado.
 - Experimente os comandos a seguir:
 - printf("%.2f\n", 5.446); /*imprime 5.45 */
 - printf("%.1f\n", 5.446); /*imprime 5.4 */

Dicas de Saída

- Para mostrar o símbolo %, digite-o 2 vezes em seguida:

```
printf("%% de Aumento e: %f", percentual)
```

O que é preciso programar?

- Dados. Você acabou de aprender!
- Alocação da memória
- Entrada
- Saída. Você aprendeu um pouco mais!
- Processamento

Entrada de Dados

- É a leitura de dados de algum dispositivo de entrada.
- Você vai usar o teclado como dispositivo de entrada.
- A entrada corresponde a ler algum valor informado pelo teclado e colocá-lo em uma variável.

Entrada de Dados

- Crie um novo projeto e escreva o programa a seguir.

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int idade;
6      printf("Digite a idade: ");
7      scanf("%d", &idade);
8      printf("A idade informada foi: %d\n", idade);
9
10     return 0;
11 }
```

Inclusão da Biblioteca `stdio.h`

- A função para a entrada de dados encontra-se na biblioteca **`stdio.h`**.
- Para usar a função de entrada, é necessário incluir a biblioteca no código, usando a diretiva **`#include`**.

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int idade;
6      printf("Digite a idade: ");
7      scanf("%d", &idade);
8      printf("A idade informada foi: %d\n", idade);
9
10     return 0;
11 }
```

Inclusão da biblioteca `stdio.h`, que possui a função de entrada `scanf`.

Função de Entrada scanf

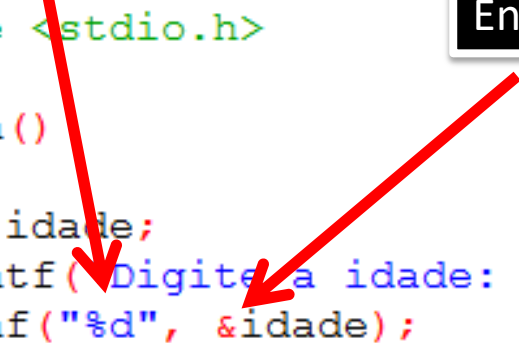
- A função **scanf** recebe dois parâmetros de entrada: a **formatação** do tipo de dado a ser lido e o **endereço da variável** que vai receber o valor.

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int idade;
6      printf("Digite a idade: ");
7      scanf("%d", &idade);
8      printf("A idade informada foi: %d\n", idade);
9
10     return 0;
11 }
```

Função de Entrada scanf

Leitura de um valor inteiro

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int idade;
6      printf("Digite a idade: ");
7      scanf("%d", &idade);
8      printf("A idade informada foi: %d\n", idade);
9
10     return 0;
11 }
```



Endereço (&) da variável

Leitura de Valores

Tipo de Dado	Formatação	Exemplo
Caractere	%c	scanf(" %c", sexo);
Inteiro	%d	scanf("%d", idade);
Real	%f	scanf("%f", salario);
String (cadeia de caracteres sem espaço)	%s	scanf("%s", nome);
String (com espaços)	%[^\n]	scanf("%[^\n]", nome);

Leitura de um caractere com scanf

- Para ler caracteres, é necessário limpar o buffer antes usando o comando `fflush(stdin)`:

```
1  /*
2  Informa o mês por extenso em Português ou Inglês.
3  */
4
5  #include <stdio.h>
6
7  int main(){
8
9      int mes;
10     char idioma;
11
12     printf("Digite o mes: ");
13     scanf("%d", &mes);
14
15     fflush(stdin); //Para limpar o buffer. Evita que o caractere não seja lido
16
17     printf("Digite o idioma (P = Portugues - I = Ingles): ");
18     scanf("%c", &idioma);
```

Outro Exemplo

```
1  #include <stdio.h>
2
3  int main()
4  {
5      float salario;
6      printf("Digite o salario: ");
7      scanf ("%f",&salario);
8      printf ("O salario informado foi: %.2f", salario);
9      return 0;
10 }
```


O que é preciso programar?

- **Dados.**
- Alocação da memória
- **Entrada. Você acabou de aprender.**
- **Saída.**
- Processamento

Operações Aritméticas

Operação	Símbolo
Soma	+
Subtração	-
Multiplicação	*
Divisão	/
Módulo (resto)	%

- / (divisão) retorna inteiro quando o dividendo e o divisor são inteiros.
- **Precedência das operações aritméticas:**
 1. ()
 2. * / %
 3. + -

Precedência das Operações Aritméticas

$$5 + 2 * 3 = 11$$

$$(5 + 2) * 3 = 21$$

$$2 * 2 + 4 = 8$$

$$2 * (2 + 4) = 12$$

Operações Aritméticas Combinadas com Atribuição

Operação	Símbolo	Exemplo	Correspondência
Soma	<code>+=</code>	<code>a += 100</code>	<code>a = a + 100</code>
Subtração	<code>-=</code>	<code>a -= b</code>	<code>a = a - b</code>
Multiplicação	<code>*=</code>	<code>a *= 50</code>	<code>a = a * 50</code>
Divisão	<code>/=</code>	<code>a /= 100</code>	<code>a = a / 100</code>
Módulo (resto da divisão de dois números inteiros)	<code>%=</code>	<code>a %= c</code>	<code>a = a % c</code>

Exemplo

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int valor1;
6      int valor2;
7      int soma;
8
9      printf("Digite o primeiro valor: ");
10     scanf("%d", &valor1);
11
12     printf("Digite o segundo valor: ");
13     scanf("%d", &valor2);
14
15     soma = valor1 + valor2;
16
17     printf("Resultado da soma: %d", soma);
18
19     return 0;
20 }
```

Exemplo

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int valor1;
6      int valor2;
7      int soma;
8
9      printf("Digite o primeiro valor: ");
10     scanf("%d", &valor1);
11
12     printf("Digite o segundo valor: ");
13     scanf("%d", &valor2);
14
15     soma = valor1 + valor2;
16
17     printf("Resultado da soma: %d", soma);
18
19     return 0;
20 }
```

Declaração de variáveis

Entrada

Processamento

Saída

Funções Matemáticas

- Existe uma biblioteca que fornece diversas funções que realizam cálculos matemáticos básicos.
- Para usar estas funções, inclua o arquivo **math.h** em seu código:

```
#include <math.h>
```

Principais Funções Matemáticas

Função	Descrição	Exemplo
<code>double sqrt (double)</code>	Calcula a raiz quadrada de um número.	<code>x = sqrt(100)</code>
<code>double pow (double, double)</code>	Eleva um valor a um determinado expoente	<code>a = pow (x, 2)</code>
<code>double log (double)</code>	Calcula o logaritmo natural de um número, (base e)	<code>x = log(y)</code>
<code>double log10 (double)</code>	Calcula o logaritmo com base 10 de um número.	<code>x = log10(y)</code>
<code>double exp (double)</code>	Retorna o valor de e elevado à uma determinada potência.	<code>x = exp(100)</code>
<code>double ceil (double)</code>	Retorna o teto de um número (o maior valor inteiro mais próximo)	<code>c = ceil (7,15)</code> //Retorna 8
<code>double floor (double)</code>	Retorna o piso de um número (o menor valor inteiro mais próximo)	<code>f = floor(7,15)</code> //Retorna 7

Principais Funções Matemáticas

Função	Descrição	Exemplo
double fabs (double)	Retorna o valor absoluto de um número.	a = fabs (-100) //Retorna 100
double cos (double)	Retorna o cosseno de um ângulo em radianos.	c = cos(angulo)
double sin (double)	Retorna o seno de um ângulo em radianos.	s = sin(angulo)

Exemplo

```
1  /*Exemplo de uso de funções matemáticas*/
2
3  #include <stdio.h>
4  #include <math.h>
5
6  int main()
7  {
8      float valor;
9      printf("Digite um número: ");
10     scanf("%f", &valor);
11
12     printf("Raiz quadrada do numero: %.3f\n", sqrt(valor));
13     printf("Número elevado a 2: %.3f\n", pow(valor,2));
14     printf("Número elevado a 3: %.3f\n", pow(valor,3));
15     printf("Teto do numero: %f\n", ceil(valor));
16     printf("Piso do numero: %f\n", floor(valor));
17     printf("Valor absoluto do numero: %f\n", fabs(valor));
18     printf("Parte fracionaria do numero/2: %f\n", fmod(valor,2));
19 }
```

O que é preciso programar?

- **Dados.**
- Alocação da memória
- **Entrada.**
- **Saída.**
- **Processamento:** Operações aritméticas. Você acabou de aprender!

Comentários

- Serve para incluir anotações no código
- Não são compilados
- Normalmente são usados para:
 - Incluir dados de identificação, como autor do código, data de criação, alterações realizadas ao longo do tempo (estes dados são colocados no início do arquivo).
 - Explicar partes mais complexas do código.
 - Isolar partes do código que não devem ser compilados.

Comentários

- Comentário de uma linha:

//Este é um comentário de uma linha

- Comentário com várias linhas

/*

Este é um comentário
com várias linhas

*/

Exemplo

```
1  /*
2  Este programa calcula a soma de dois valores.
3  */
4
5  #include <stdio.h>
6
7  int main()
8  {
9      int valor1;
10     int valor2;
11     int soma;
12
13     //Entrada do primeiro valor
14     printf("Digite o primeiro valor: ");
15     scanf("%d", &valor1);
16
17     //Entrada do segundo valor
18     printf("Digite o segundo valor: ");
19     scanf("%d", &valor2);
20
21     //Soma dos valores
22     soma = valor1 + valor2;
23
24     //Exibição do resultado na tela
25     printf("Resultado da soma: %d", soma);
26
27     return 0;
28 }
```

Dicas sobre Comentários



- O programa deve ser autoexplicativo.
- Evite usar comentários sem necessidade.

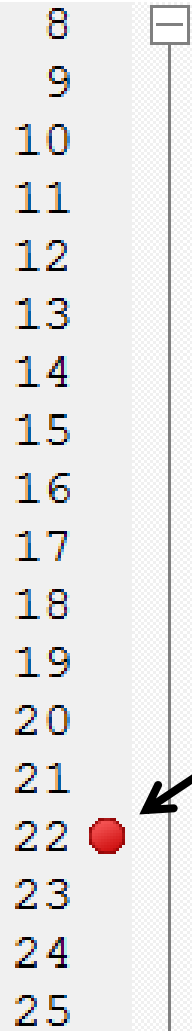
Depuração do Programa

- Depurar é a tarefa de encontrar defeitos no código fonte.
- Os ambientes de programação atuais possuem vários recursos para ajudar a realização desta tarefa.
- **Principais tarefas de depuração:**
 - Marcar a linha de código onde a execução do programa deve ser interrompida para você iniciar a depuração (ponto de parada ou *breakpoint*).
 - Acompanhar a execução do programa linha a linha.
 - Entrar dentro de uma função (será visto depois).
 - Observar os valores das variáveis que estão na memória no momento da depuração.

Criando um Ponto de Parada

- Um ponto de parada marca a linha de código em que a execução do programa deve ser interrompida para você iniciar a tarefa de depuração.
- Para inserir um ponto de para no Codeblocks, posicione o cursor botão do mouse ao lado do número da linha e clique o botão esquerdo.
- Irá aparecer um círculo vermelho ao lado da linha.

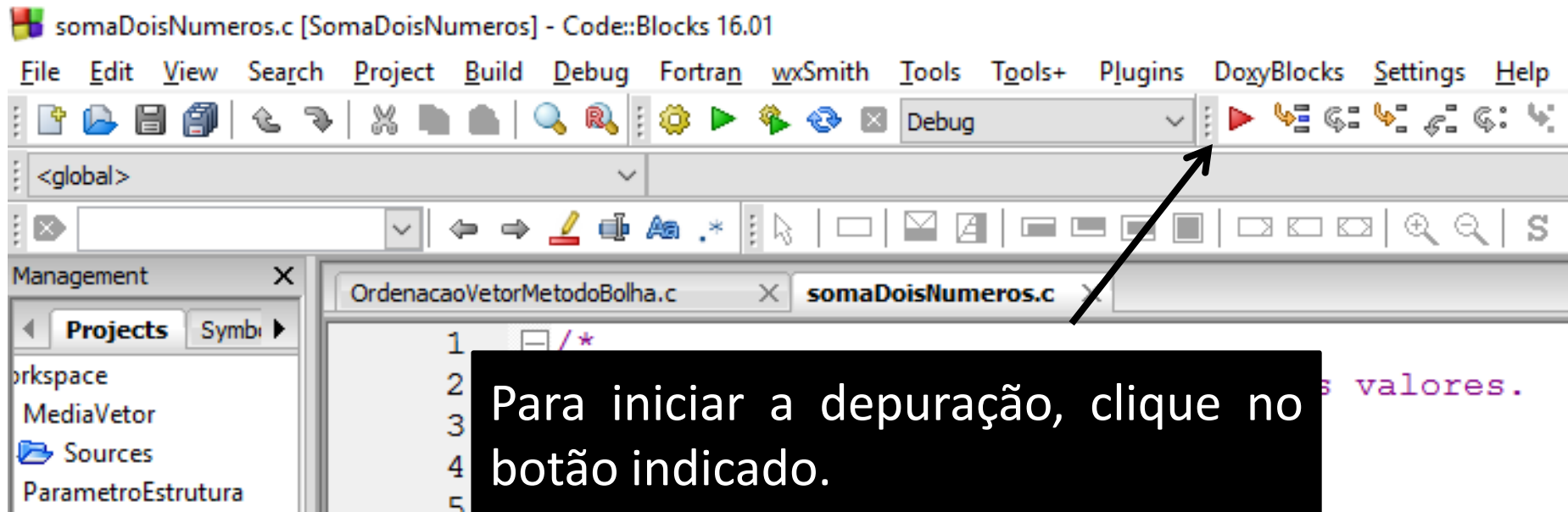
Criando um Ponto de Parada



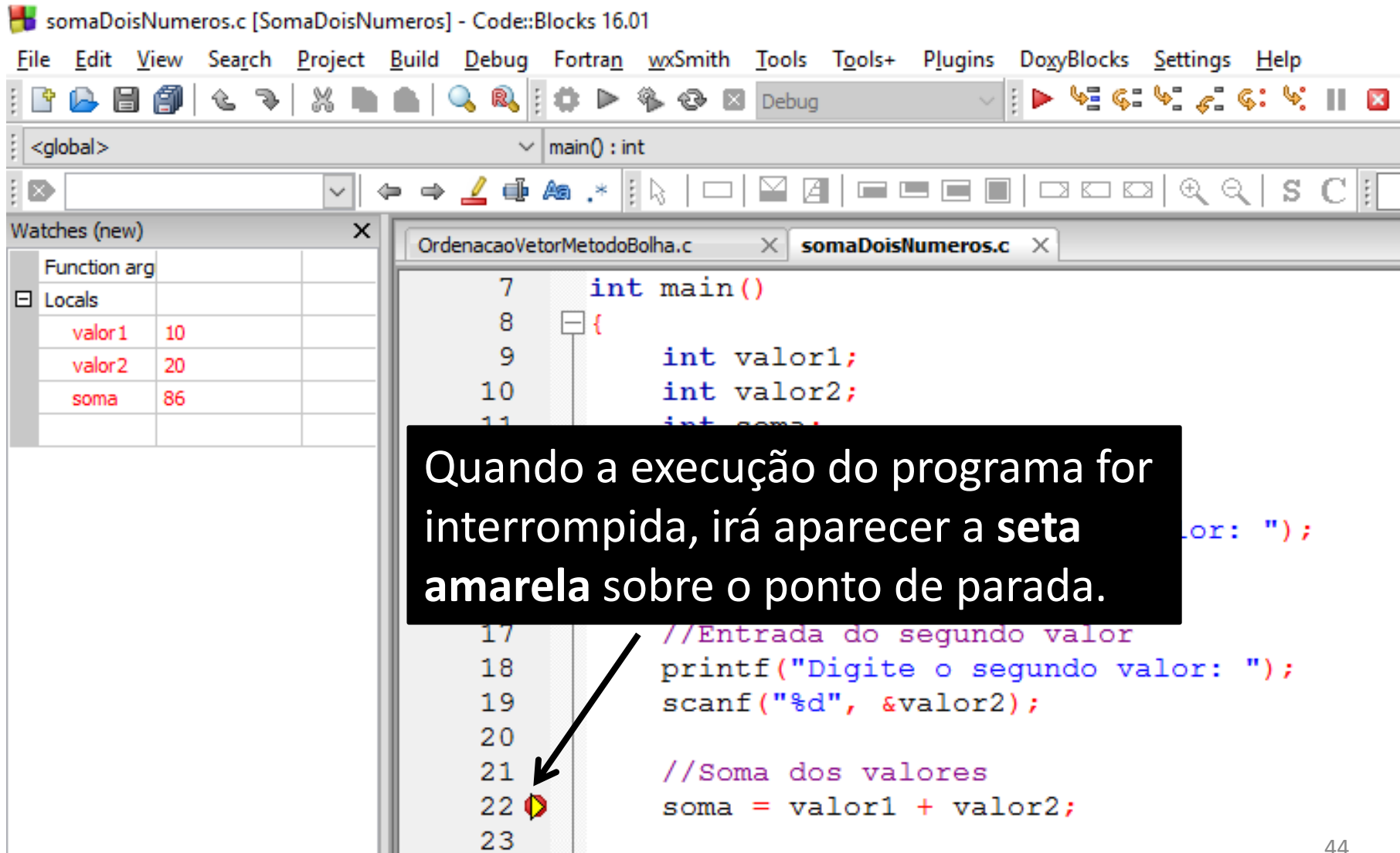
```
8  {
9      int valor1;
10     int valor2;
11     int soma;
12
13     //Entrada do primeiro valor
14     printf("Digite o primeiro valor: ");
15     scanf("%d", &valor1);
16
17     //Entrada do segundo valor
18     p
19     s
20
21     /
22     s
23
24     //Exibição do resultado na tela
25     printf("Resultado da soma: %d", soma);
```

Posicione o cursor do mouse ao lado do número da linha onde você quer que o programa pare e clique com o botão esquerdo. Irá aparecer o círculo vermelho.

Iniciando a Depuração



Parada da Execução do Programa



somaDoisNumeros.c [SomaDoisNumeros] - Code::Blocks 16.01

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global> main() : int

Watches (new)

Function arg	
valor1	10
valor2	20
soma	86

```
7 int main()
8 {
9     int valor1;
10    int valor2;
11    int soma;
12
13    //Entrada do primeiro valor
14    printf("Digite o primeiro valor: ");
15    scanf("%d", &valor1);
16
17    //Entrada do segundo valor
18    printf("Digite o segundo valor: ");
19    scanf("%d", &valor2);
20
21    //Soma dos valores
22    soma = valor1 + valor2;
23
```

Quando a execução do programa for interrompida, irá aparecer a seta amarela sobre o ponto de parada.

Observando os Valores das Variáveis

somaDoisNumeros.c [SomaDoisNumeros] - Code::Blocks 16.01

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global> main() : int

Debug

Watches (new)

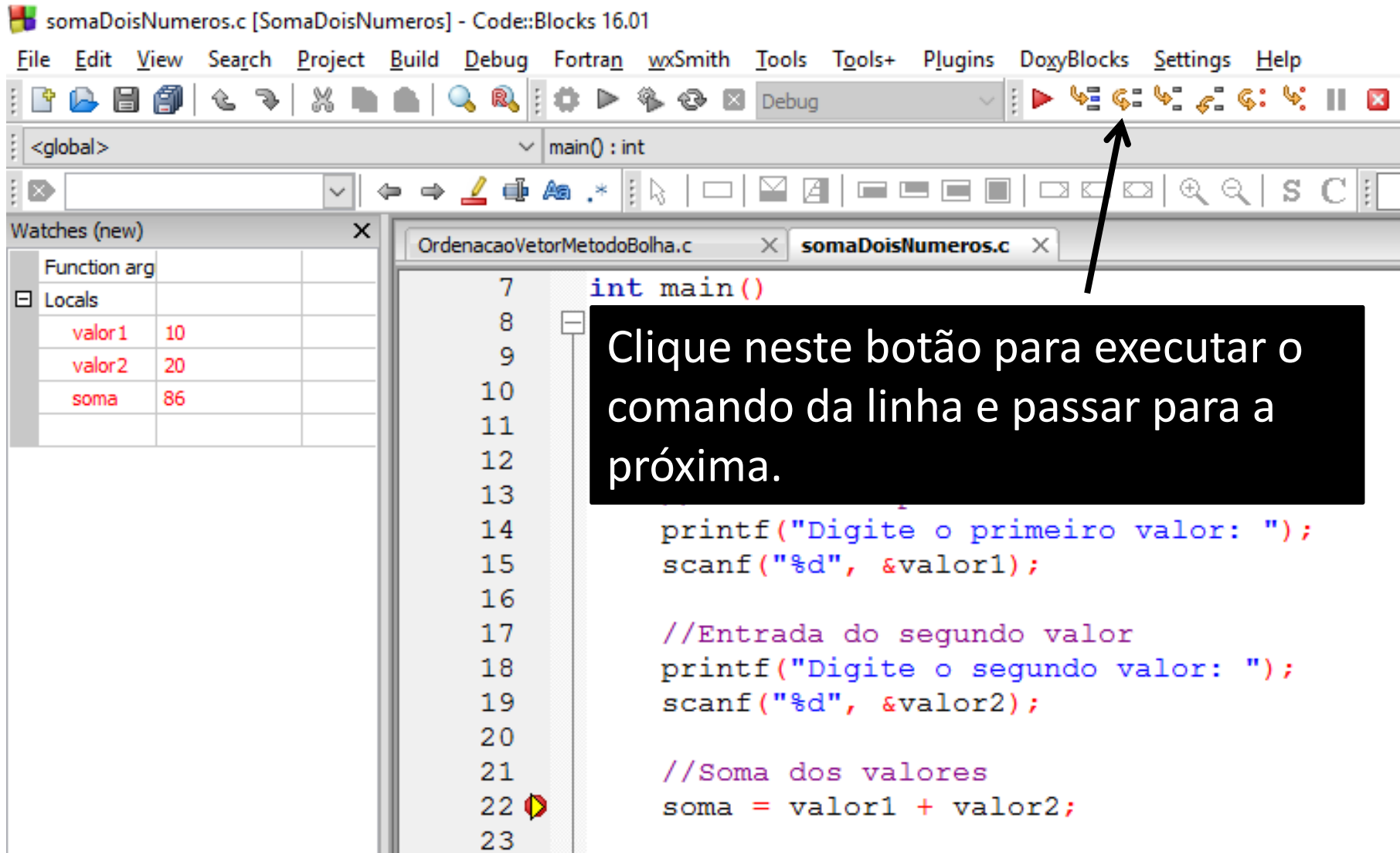
Function arg	
Locals	
valor1	10
valor2	20
soma	86

OrdenacaoVetorMetodoBolha.c x somaDoisNumeros.c x

```
7 int main()
8 {
9     int valor1;
10    int valor2;
11    int soma;
12
13    //Entrada do primeiro valor
14    scanf("%d", &valor1);
15
16    //Entrada do segundo valor
17    scanf("%d", &valor2);
18
19    //Soma dos valores
20    soma = valor1 + valor2;
21
22
23
```

Você pode observar os valores das variáveis aqui.

Executando o Programa Linha a Linha



somaDoisNumeros.c [SomaDoisNumeros] - Code::Blocks 16.01

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global> main() : int

Watches (new)

Function arg	
Locals	
valor1	10
valor2	20
soma	86

OrdenacaoVetorMetodoBolha.c x somaDoisNumeros.c x

```
7 int main()
8
9
10
11
12
13
14 printf("Digite o primeiro valor: ");
15 scanf("%d", &valor1);
16
17 //Entrada do segundo valor
18 printf("Digite o segundo valor: ");
19 scanf("%d", &valor2);
20
21 //Soma dos valores
22 soma = valor1 + valor2;
23
```

Clique neste botão para executar o comando da linha e passar para a próxima.

Executando o Programa Linha a Linha

*somaDoisNumeros.c [SomaDoisNumeros] - Code::Blocks 16.01

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global> main() : int

Watches (new)

Function arg	
Locals	
valor1	10
valor2	20
soma	30

Observe que o valor da variável soma mudou!

A seta amarela indica a próxima linha que será executada.

```
7
8
9
10
11 int soma;
12
13 //Entrada do primeiro valor
14
15
16
17
18
19
20
21 //Soma dos valores
22 soma = valor1 + valor2;
23
24 //Exibição do resultado na tela
25 printf("Resultado da soma: %d", soma);
```

Exercícios

1. Criar um programa para exibir a placa, o ano e a cor de um automóvel.
2. Criar um programa para exibir o nome, a categoria e a produtora de um filme.
3. Criar um programa para ler dois números, calcular a subtração e mostrar na tela o resultado.
4. Criar um programa para ler e somar três números inteiros, mostrando o resultado na tela.
5. Criar um programa para ler e calcular a média aritmética de quatro números reais e mostrar o resultado na tela.
6. Criar um programa para ler, calcular e exibir o resultado da seguinte equação:
 $y = ax + b$.
7. Criar um programa para ler, calcular e exibir o resultado da seguinte equação:
 $s = s_0 + vt$.
8. Criar um programa para ler, calcular e exibir o resultado da seguinte equação:
 $v = v_0 + at$.
9. Criar um programa para ler, calcular e exibir o resultado da seguinte equação:
 $s = s_0 + v_0t + at^2/2$.
10. Criar um programa para ler, calcular e exibir o resultado da seguinte equação:
 $x = (-b \pm \text{raiz}(b^2 - 4ac))/2a$ (não esqueça que existem duas saídas). Use a função `sqrt` para calcular a raiz quadrada.

Você Estudou

- Variável.
- Diferenças entre letras maiúsculas e minúsculas.
- Tipos de dados simples.
- Atribuição.
- Saída de valores armazenados em variáveis.
- Entrada de dados.
- Inclusão de uma biblioteca (stdio).
- Operações aritméticas.
- Comentários.
- Depuração de programas.