

Depuração e Teste de programas C

André Takeshi Endo

Roteiro

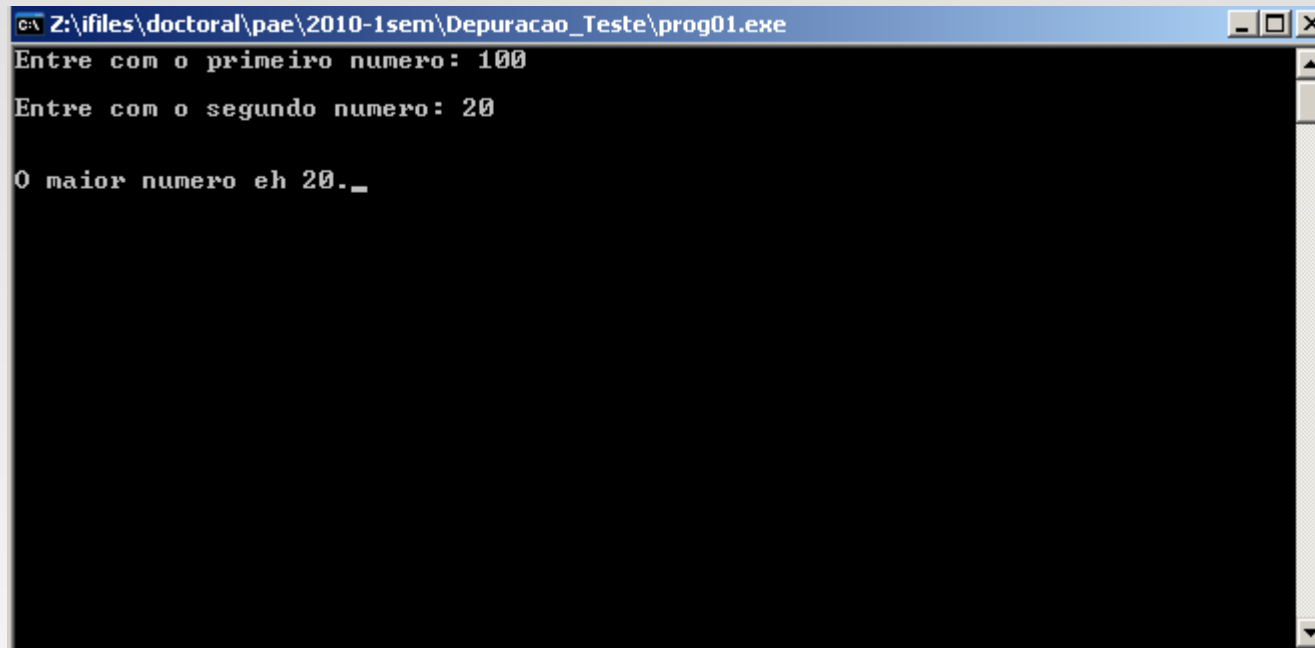
- Teste de Software
 - Definições
 - Caso de teste
- Depuração de programas
 - Dev-C++
 - Gdb + gcc (Linux)
- Cobertura dos testes
 - GCov

Teste de Software: definições

- Executar um programa com a intenção de encontrar defeitos.
- Defeito / Bug / Erro
 - Instrução ou comando incorreto
- Falha
 - Produção de uma saída incorreta

Teste de Software: definições

- Exemplo: Leia dois inteiros e imprima o maior.

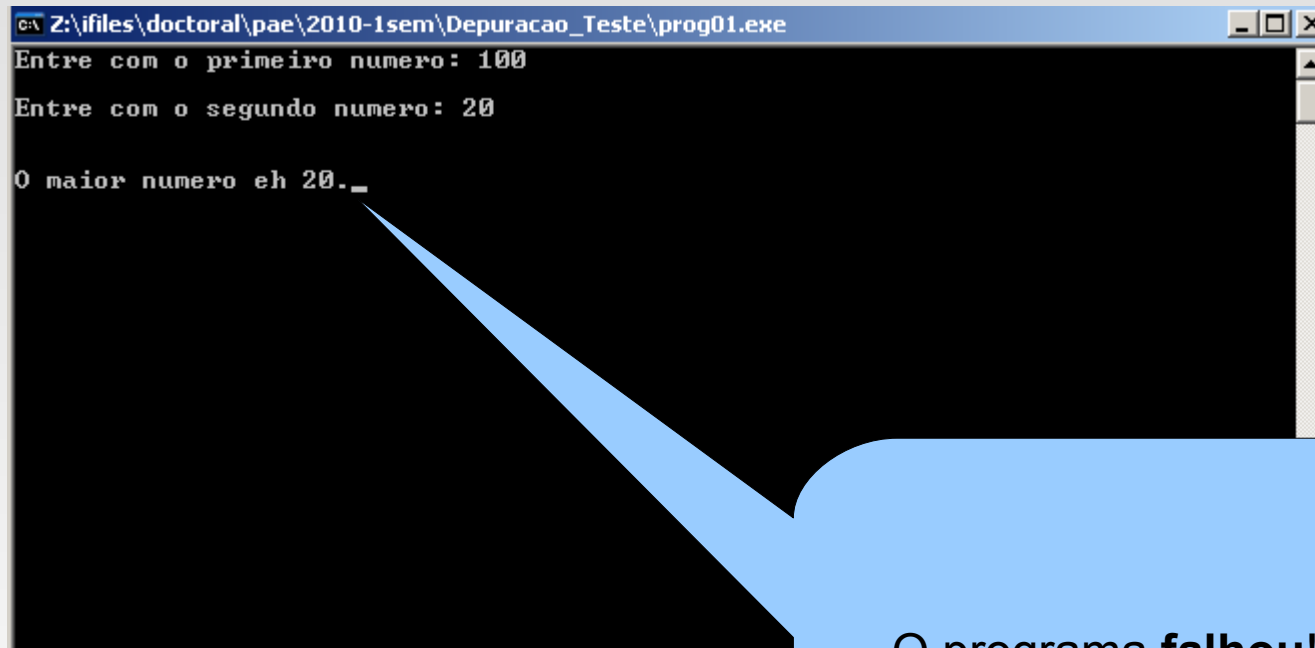


A screenshot of a Windows command prompt window. The title bar at the top reads "C:\ Z:\files\doctoral\pae\2010-1sem\Depuracao_Teste\prog01.exe". The window has a black background with white text. The text inside the window shows the following sequence of input and output: "Entre com o primeiro numero: 100", "Entre com o segundo numero: 20", and "O maior numero eh 20._". The cursor is positioned at the end of the last line.

```
C:\ Z:\files\doctoral\pae\2010-1sem\Depuracao_Teste\prog01.exe
Entre com o primeiro numero: 100
Entre com o segundo numero: 20
O maior numero eh 20._
```

Teste de Software: definições

- Exemplo: Leia dois inteiros e imprima o maior.



```
C:\Z:\files\doctoral\pae\2010-1sem\Depuracao_Teste\prog01.exe
Entre com o primeiro numero: 100
Entre com o segundo numero: 20
O maior numero eh 20._
```

O programa **falhou!**
Encontramos uma **falha**.

Teste de Software: definições

- Exemplo: Leia dois inteiros e imprima o maior.

```
#include<stdio.h>
#include<conio.h>

int main(void)
{
    int num1, num2;

    printf("Entre com o primeiro numero: ");
    scanf("%d", &num1);
    printf("\nEntre com o segundo numero: ");
    scanf("%d", &num2);

    if(num1 < num2)
        printf("\n\nO maior numero eh %d.", num1);
    else
        printf("\n\nO maior numero eh %d.", num2);

    getch();
}
```

Teste de Software: definições

- Exemplo: Leia dois inteiros e imprima o maior.

```
#include<stdio.h>
#include<conio.h>

int main(void)
{
    int num1, num2;

    printf("Entre com o primeiro numero: ");
    scanf("%d", &num1);
    printf("\nEntre com o segundo numero: ");
    scanf("%d", &num2);

    if(num1 < num2)
        printf("\n\nO maior numero eh %d.", num1);
    else
        printf("\n\nO maior numero eh %d.", num2);

    getch();
}
```

O **defeito** deste programa é
A troca do sinal.
O correto seria:
if(num1 > num2)

Teste de Software: definições

- Caso de teste
 - <entradas; saídas esperadas>
- Exemplo: Faça um programa que receba como entrada um inteiro positivo e responda se o número é primo ou não.
 - < 5; primo >
 - < 49; nao eh primo >
 - < 37; primo >
 - < 10; nao eh primo >
 - < -15; entrada invalida >
 - < icc; entrada invalida >

Depuração: Conceitos

- *Debugging*
- Processo de encontrar e corrigir defeitos
- Como encontrar o defeito?
 - Mensagens do compilador
 - Revisão de código
 - Ferramentas de depuração

Depuração: Revisão do código

- 3 defeitos

```
#include<stdio.h>
#include<conio.h>

int main(void)
{
    int num1, num2;

    printf("Entre com o primeiro numero: ");
    scanf("%f", &num1);
    printf("\nEntre com o segundo numero: ");
    scanf("%d", num2);

    if(num1 < num2)
        printf("\n\nO maior numero eh %d.", num1);
    else
        printf("\n\nO maior numero eh %d.", num2);

    getch();
}
```

Depuração: Revisão do código

- 3 defeitos

```
#include<stdio.h>
#include<conio.h>

int main(void)
{
    int num1, num2;

    printf("Entre com o primeiro numero: ");
    scanf("%f", &num1);
    printf("\nEntre com o segundo numero: ");
    scanf("%d", num2);

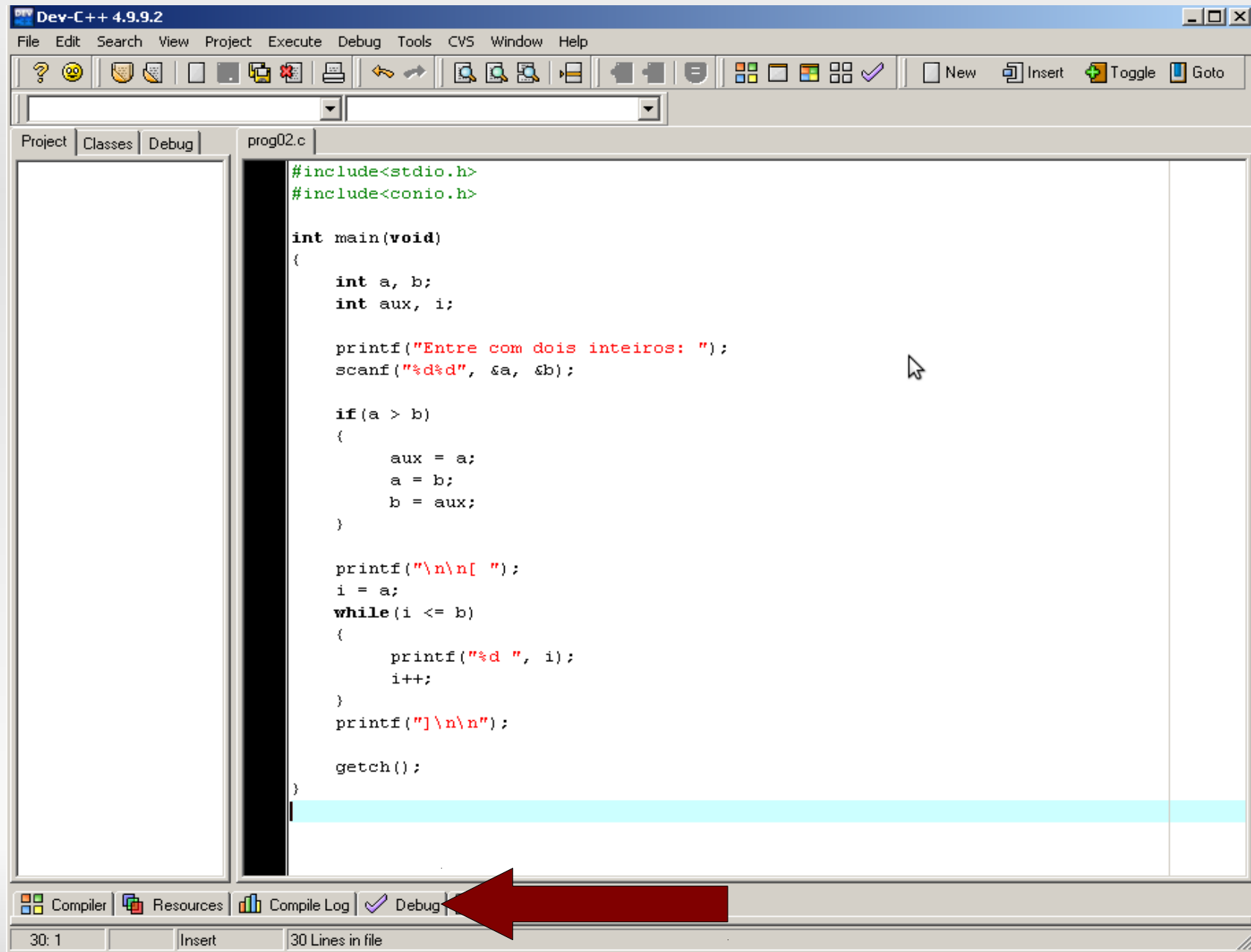
    if(num1 < num2)
        printf("\n\nO maior numero eh %d.", num1);
    else
        printf("\n\nO maior numero eh %d.", num2);

    getch();
}
```

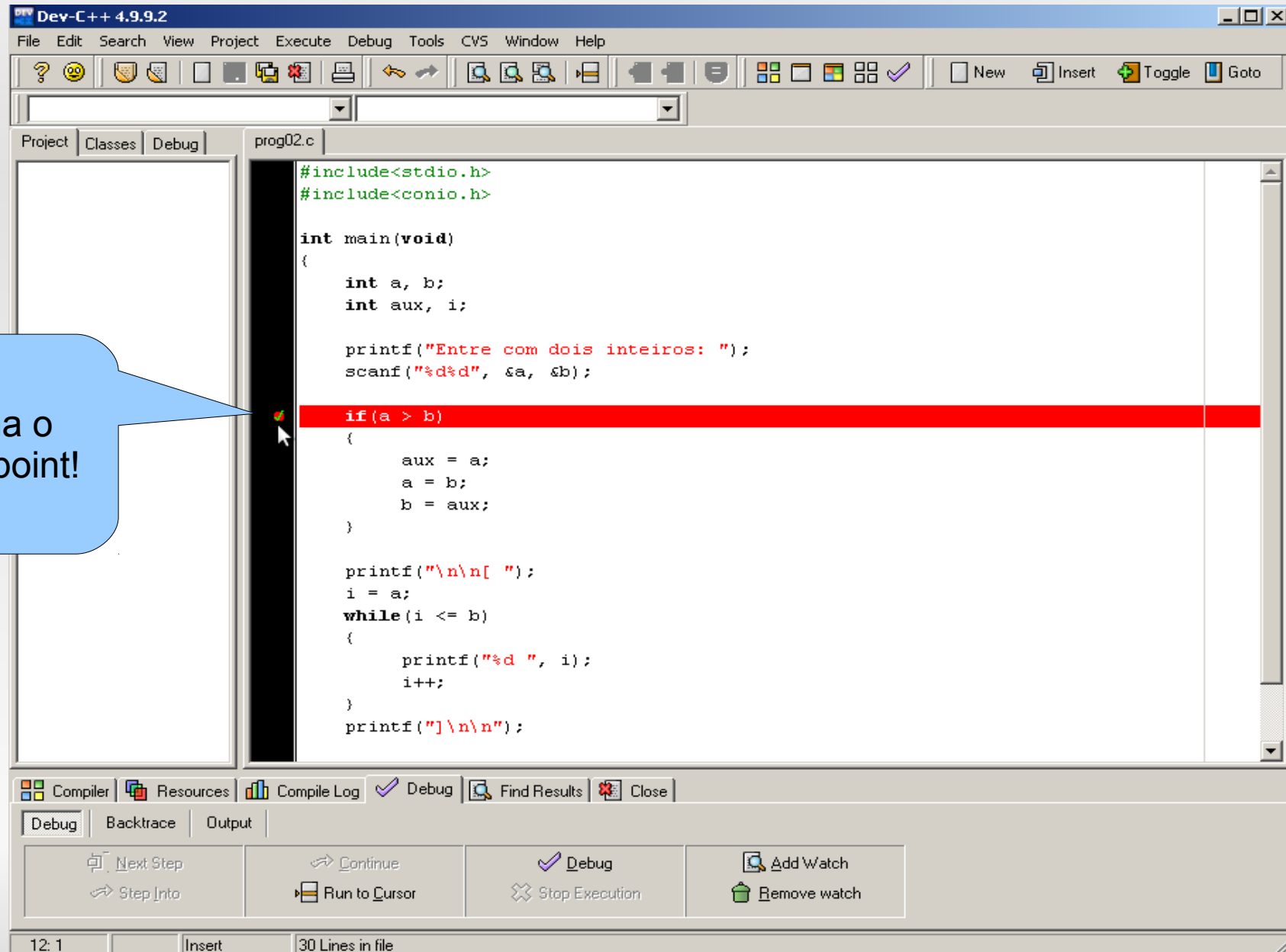
Depuração: Dev-C++

- gdb - GNU Project Debugger <[link](#)>
- Recursos gráficos
- Breakpoints
 - Até que ponto o programa deve executar normalmente
- Watch
 - Quais variáveis devem ser acompanhadas?

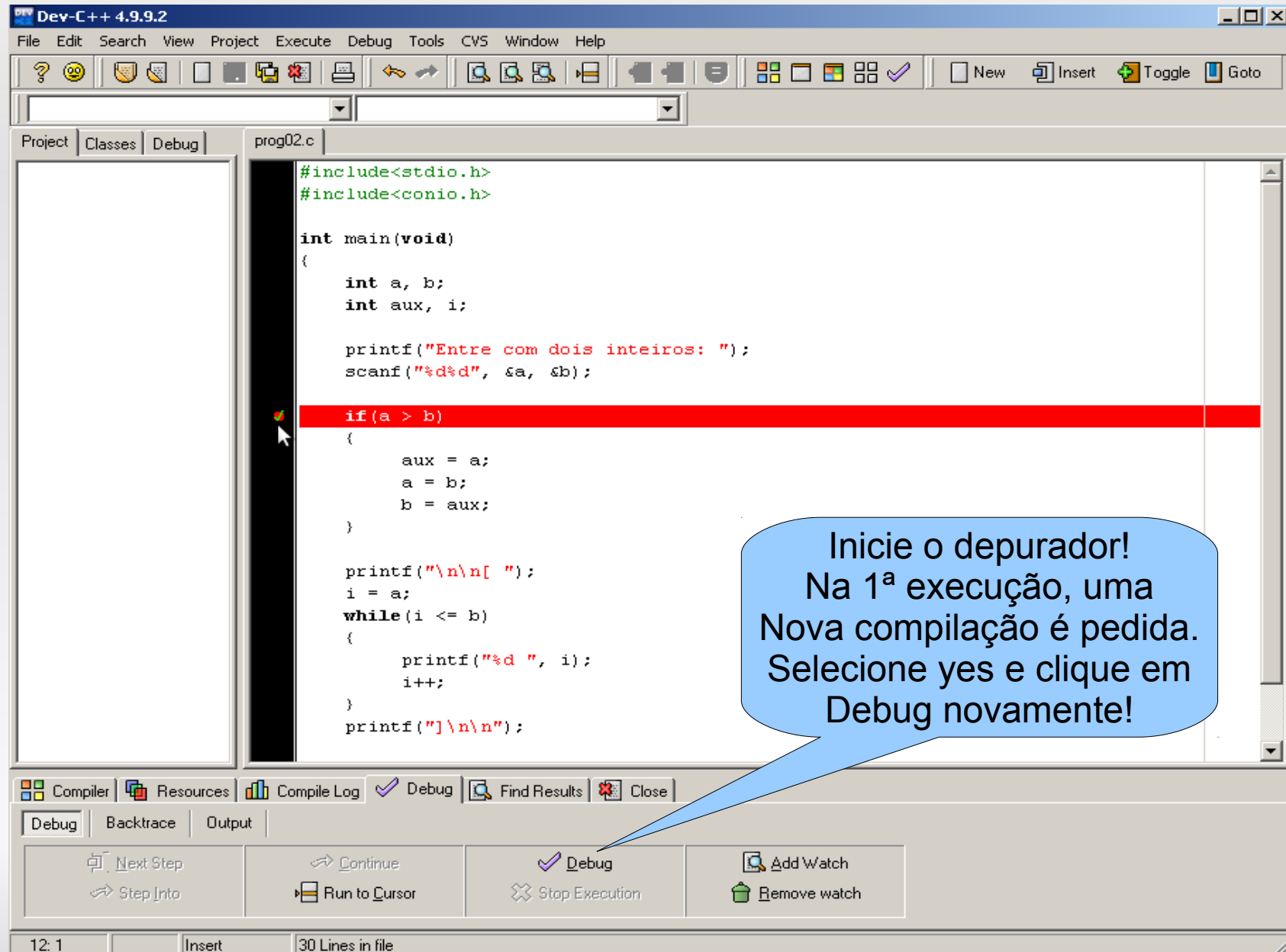
Depuração: Dev-C++ (passo 1)



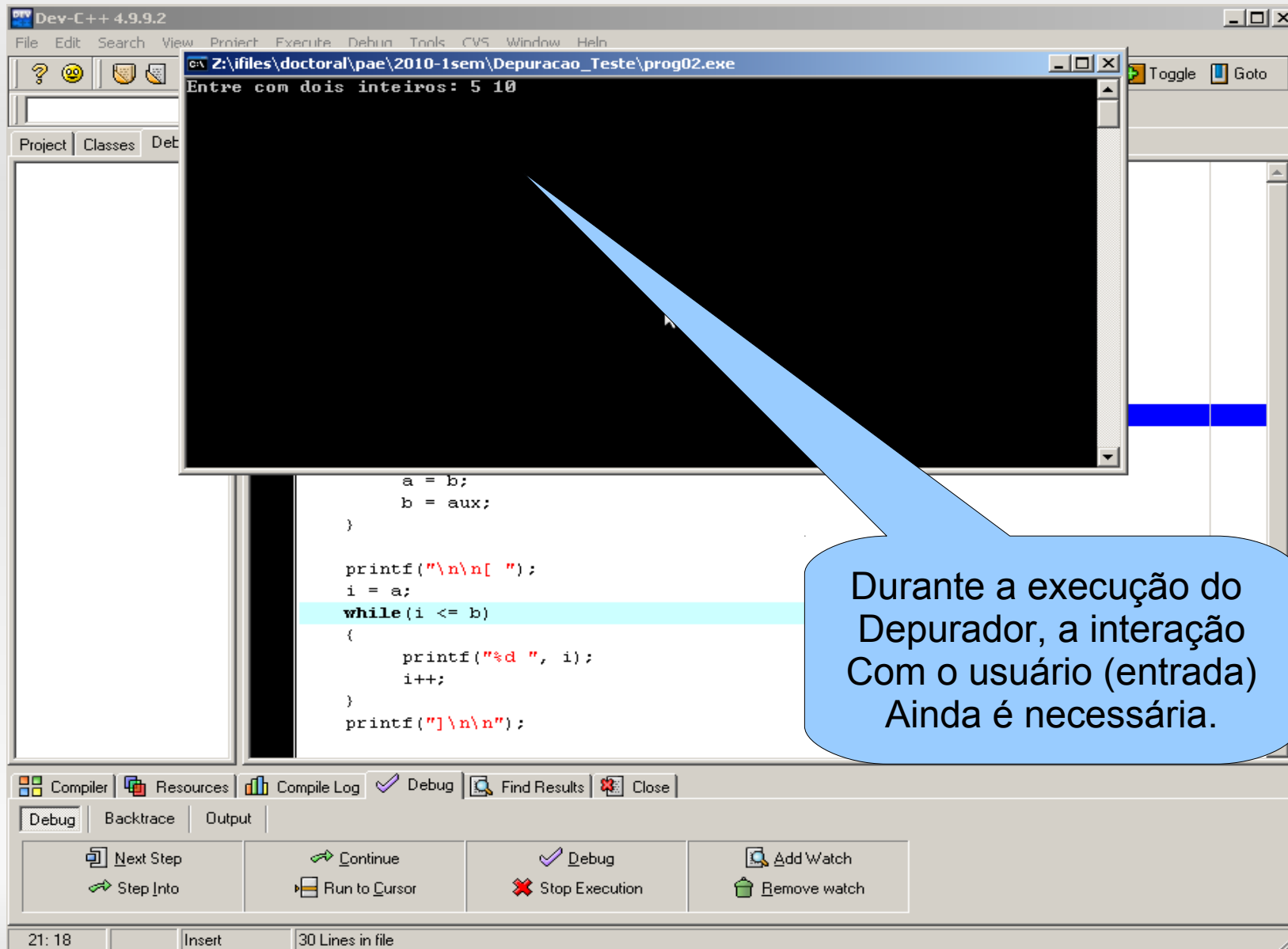
Depuração: Dev-C++ (passo 2)



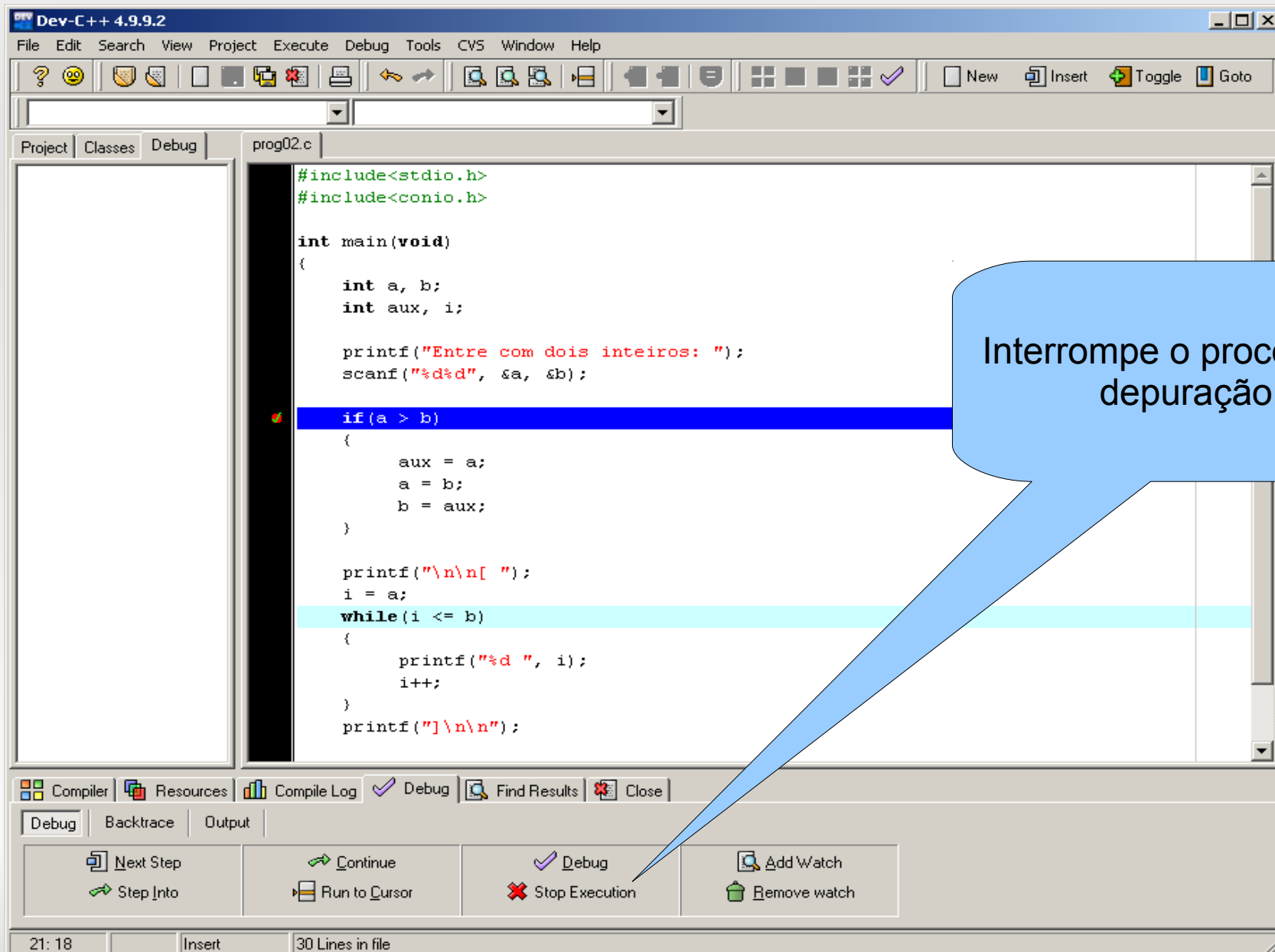
Depuração: Dev-C++ (passo 3)



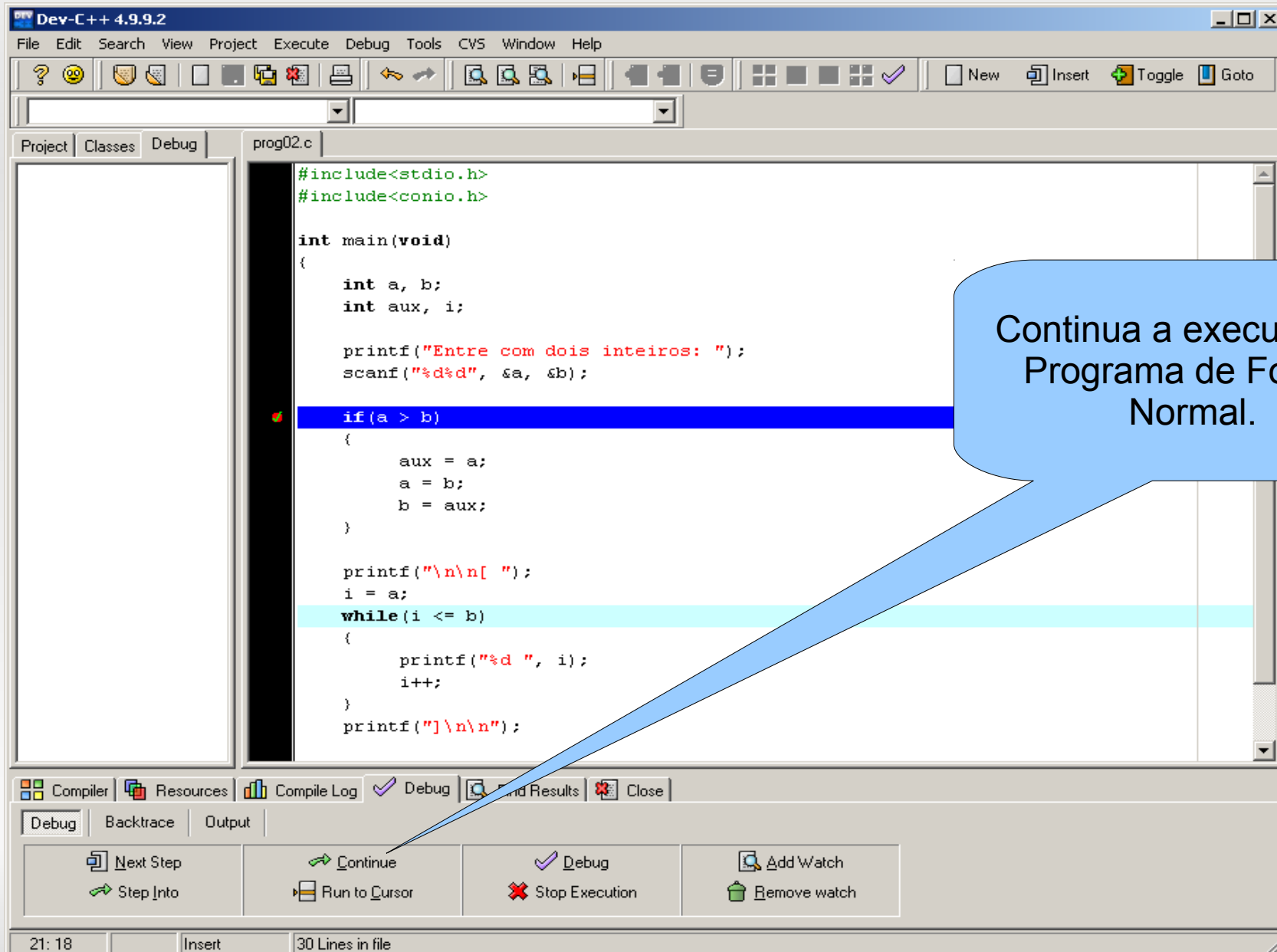
Depuração: Dev-C++ (passo 4)



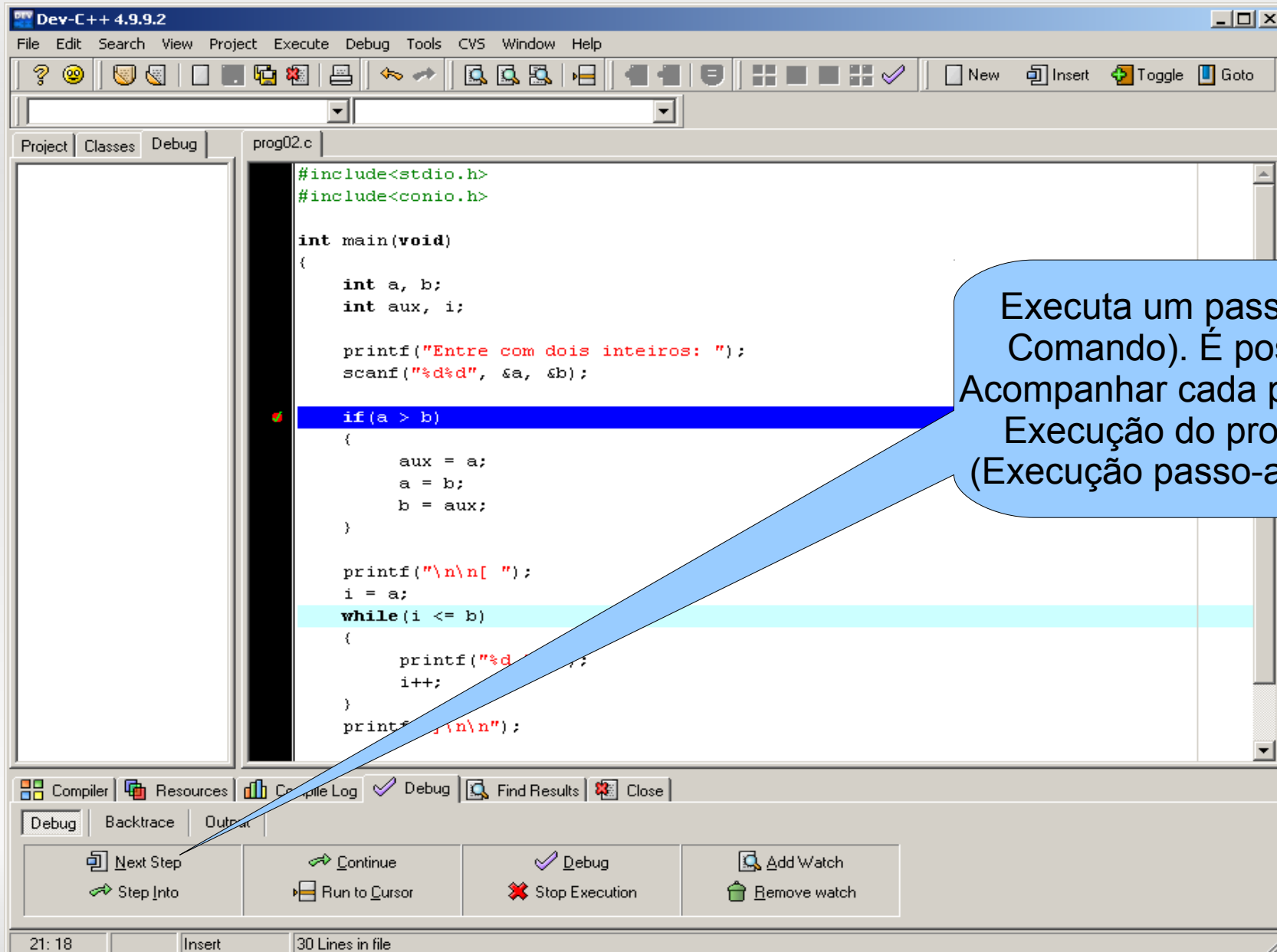
Depuração: Dev-C++ (Opções)



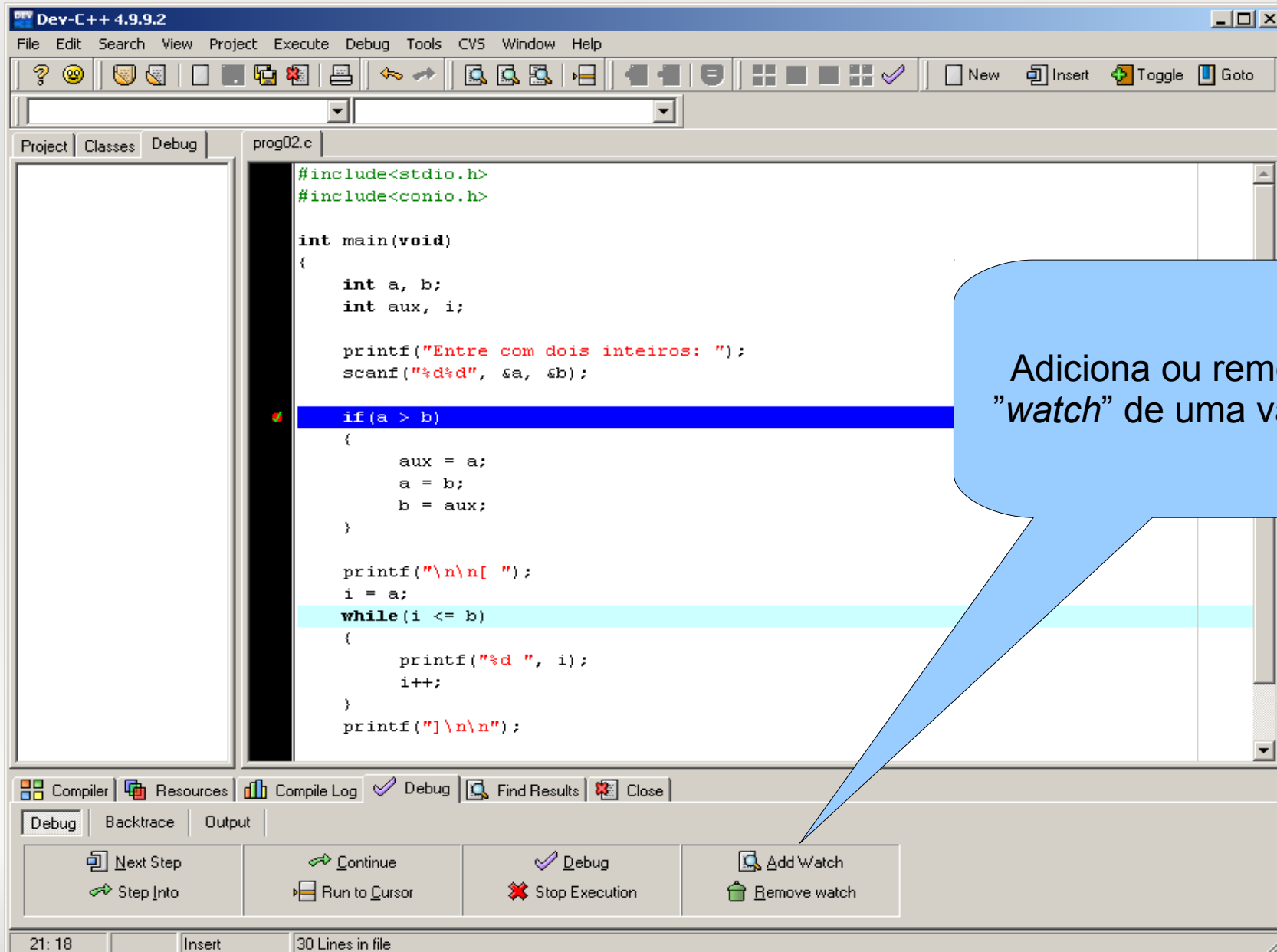
Depuração: Dev-C++ (Opções)



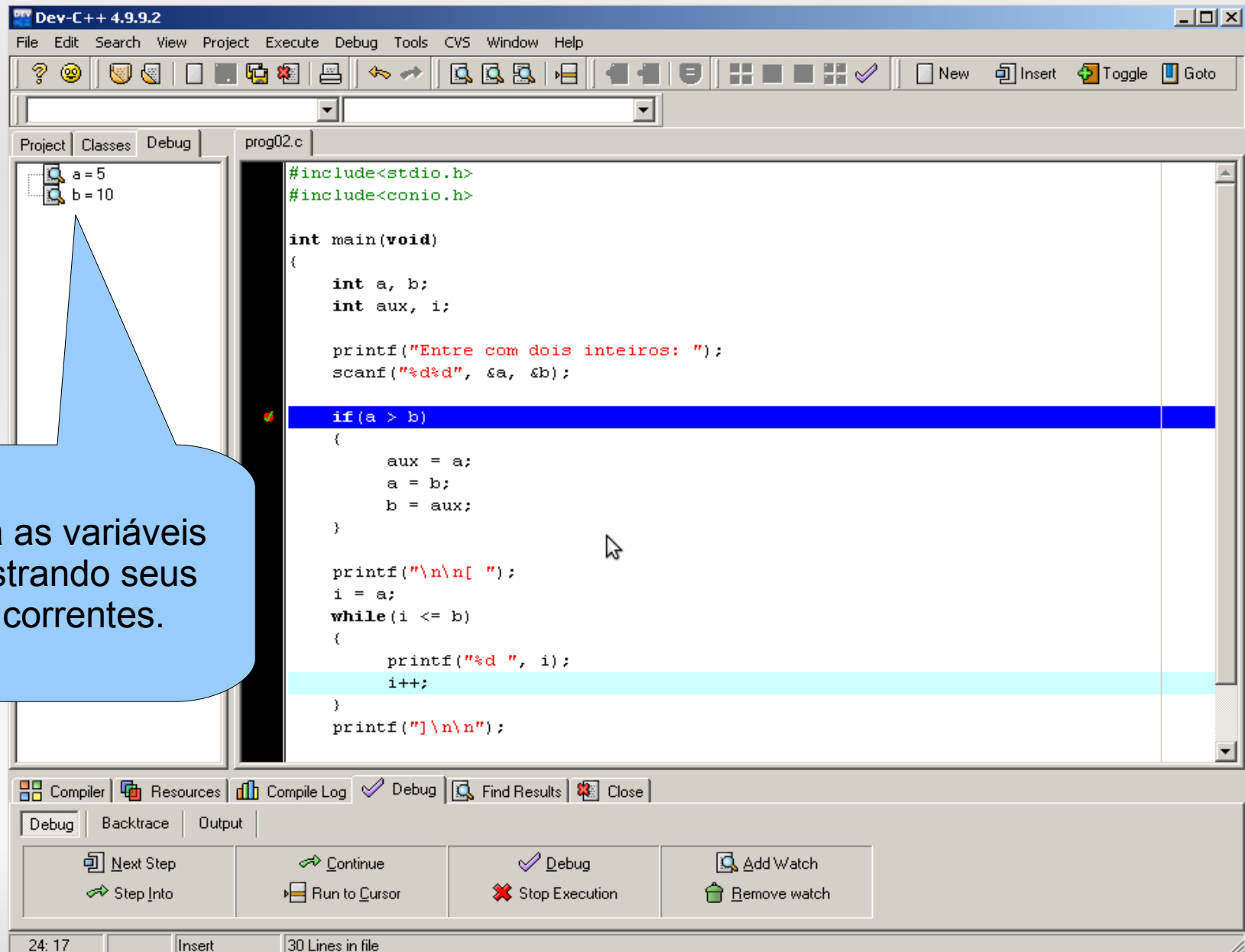
Depuração: Dev-C++ (Opções)



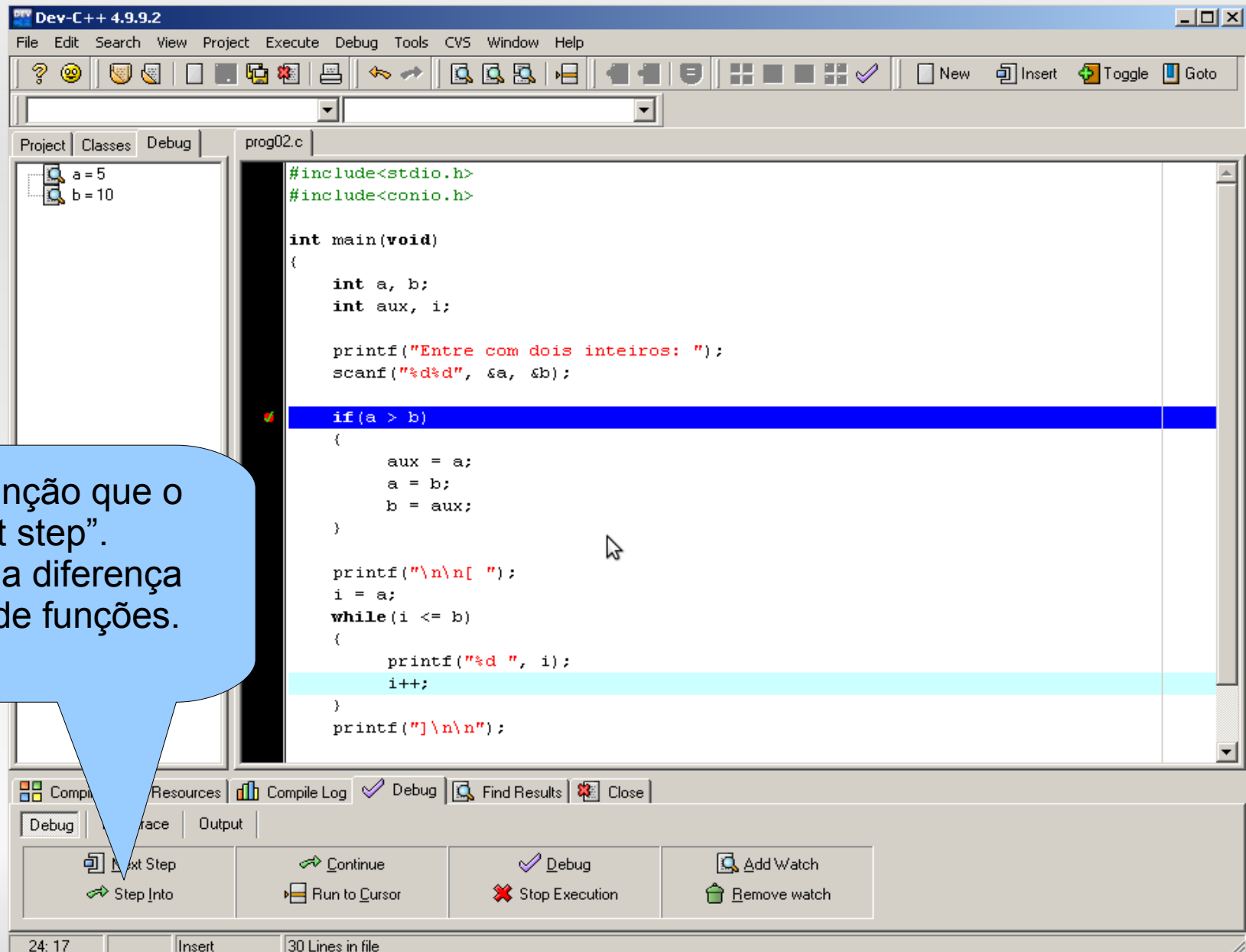
Depuração: Dev-C++ (Opções)



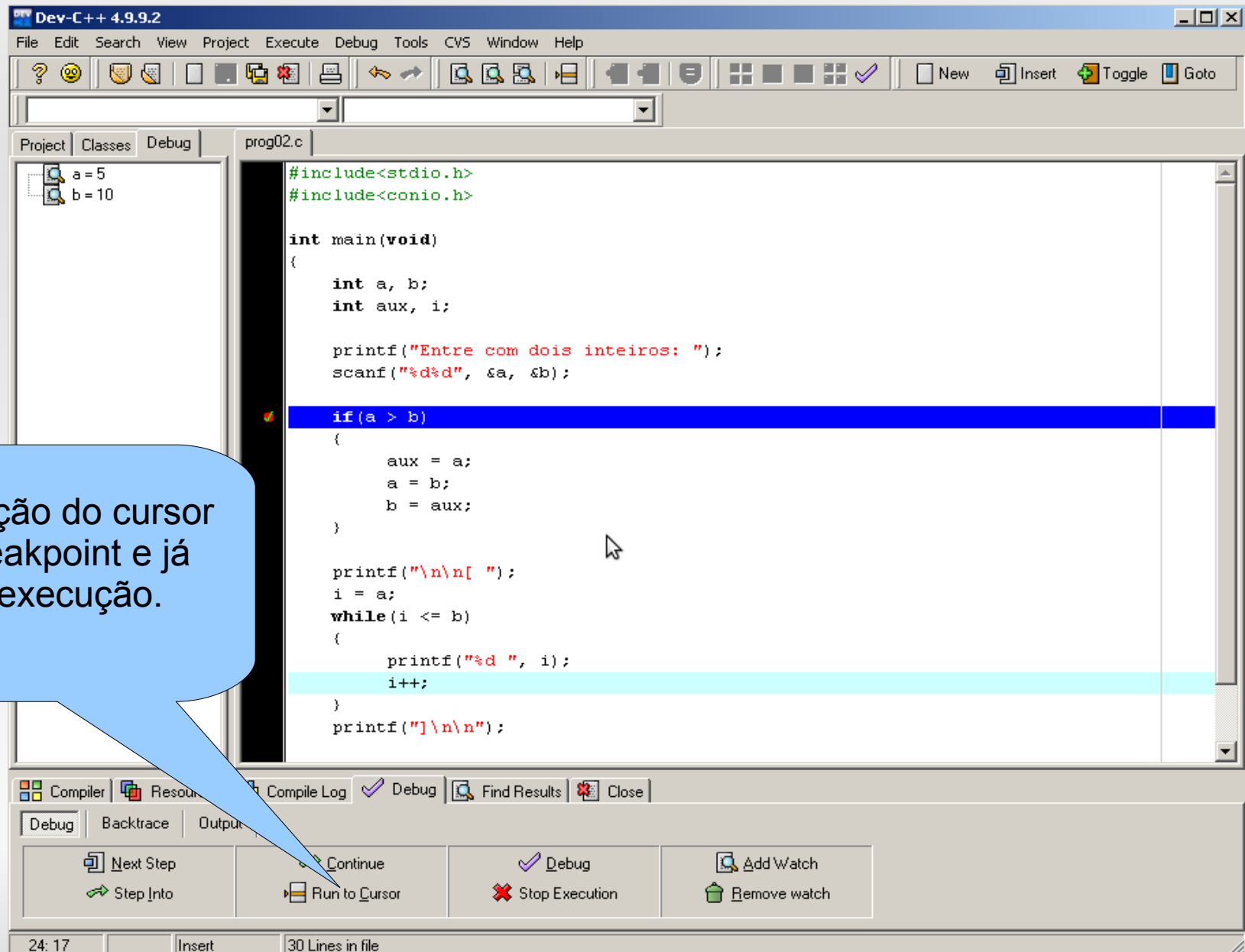
Depuração: Dev-C++ (Opções)



Depuração: Dev-C++ (Opções)



Depuração: Dev-C++ (Opções)



Usa a posição do cursor
Como breakpoint e já
Inicia a execução.

Depuração: Dev-C++

- Programa 02: faça um programa que receba como entrada dois inteiros a e b e imprima o intervalo:
 - $[a, b]$ se $a \leq b$
 - $[b, a]$, caso contrário ($b < a$).
- O código-fonte está disponível

Depuração: Dev-C++

- Exercício 01: utilizando o programa 02, faça:
 - A) Coloque um *breakpoint* na linha "int a, b;" e faça uma execução passo-a-passo para o caso de teste
 - <5, 10; [5 6 7 8 9 10] >
 - B) Coloque um *breakpoint* na linha "int a, b;" e faça uma execução passo-a-passo para o caso de teste
 - <11, 7; [7 8 9 10 11] >
 - Neste exercício, adicione o *watch* para as variáveis a, b, i, aux.

Depuração: Dev-C++

- Exercício 02: encontre os 3 defeitos no programa 3 (prog03.c) e corrija-os.
 - Leia atentamente a descrição do que o programa deve fazer.
 - Descreva, na forma de comentários no código, os casos de teste usados para revelar os defeitos.
 - Como o depurador ajudou/ajudaria a identificar esses defeitos?

Depuração: gcc + gdb (Linux)

- GCC (Dev-C++)
- gdb – linha de comando
- Utilize os terminais de comando do linux
 - Konsole, Gnome terminal
- Compilando o programa com informação de depuração (-g)

```
>> gcc -g myprogram.c -o myprogram <enter>
```

```
>> ./myprogram <enter>
```

```
>> gdb myprogram <enter>
```

Depuração: gcc + gdb (Linux)

- gdb – linha de comando

(gdb) digite comandos aqui <enter>

Comando	Descrição
q	Sai do gdb.
r	Executa o programa do início.
l	Lista porções do código.
b <i>numero</i>	Coloca um breakpoint na linha <i>numero</i>
n	Executa a próxima instrução.
p <i>var</i>	Imprime o valor atual da variável <i>var</i> .
c	Continua a execução do programa sem paradas.

Cobertura dos Testes: Gcov (Linux)

- Quais linhas do código foram executados com meus testes?
- Os desvios do programa foram cobertos?
 - Desvios condicionais (if, while, for).
- Ferramenta GCov

Cobertura dos Testes: Gcov (Linux)

- Compilar os arquivos usando opções específicas do gcc
- Em seguida, execute os testes

```
>> gcc -fprofile-arcs -ftest-coverage myprogram.c -o myprogram <enter>  
>> ./myprogram <enter>
```

- Verifique quantos % das linhas foram cobertas

```
>> gcov myprogram.c <enter>  
  
File 'myprogram.c'  
Lines executed:88.89% of 9  
test_01.c:creating 'myprogram.c.gcov'
```

Cobertura dos Testes: Gcov (Linux)

- Quais linhas não foram executadas?
 - Abra o arquivo myprogram.c.gcov

```
-: 0:Source:test_01.c
-: 0:Graph:test_01.gcno
-: 0:Data:test_01.gcda
-: 0:Runs:1
-: 0:Programs:1
-: 1:#include<stdio.h>
-: 2:
1: 3:int main(void)
-: 4:{
-: 5:   int num1, num2;
-: 6:
1: 7:   printf("Entre com o primeiro numero: ");
1: 8:   scanf("%d", &num1);
1: 9:   printf("\nEntre com o segundo numero: ");
1: 10:  scanf("%d", &num2);
-: 11:
1: 12:  if(num1 < num2)
1: 13:      printf("\n\nO maior numero eh %d.", num1);
-: 14:  else
#####: 15:      printf("\n\nO maior numero eh %d.", num2);
-: 16:
1: 17:  return 0;
-: 18:}
```

Exercícios (linux)

- Compile e execute os programas 2 e 3 no linux.
 - Remova o conio.h e a função getch()
- Utilize o gdb no modo linha de comando
 - Experimente os comandos apresentados
- Utilize o Gcov
 - Teste os programas e verifique se 100% das linhas foram cobertas. Caso não foram, faça mais testes para cobrir as linhas que faltam.

Mais Informações

- Gdb (Linux)
 - <http://www.gnu.org/software/gdb/>
 - Interface Gráfica: <http://www.kdbg.org/index.php>
 - DDD: <http://www.gnu.org/software/ddd/>
- Gcov
 - <http://gcc.gnu.org/onlinedocs/gcc/Gcov.html>
- Defeitos comuns da linguagem C
 - H. Schildt; C completo e total; 1997.
- Teste de Software / Depuração
 - G. J. Myers; The art of software testing; 2004.

Mais Informações

- Correções ou melhorias nos slides
 - andreendo@utfpr.edu.br