

Dicas para Melhorar o Código

Legibilidade do Código

Legibilidade do código

Todo bom programador preza pela legibilidade de seu código

- Auxilia na compreensão do mesmo
- Facilita o conserto de erros

A legibilidade pode ser aperfeiçoada com métodos simples

- Comentários das seções críticas do código
- Indentação do código

Comentando o código

Comentários das seções críticas do código

Facilita o entendimento do código por outras pessoas

Deve ser usado nas lógicas principais do programa (condicionais, laços)

Use comentários, iniciados por `// COMMENT`

Blocos de comentários com `/* COMMENT BLOCK */`

Exemplo:

- `if (N > MAIOR) { // Checa se o novo número deve ser guardado`
- `while (contador < N) // Executa o laço de 1 até N para calcular o fatorial`

Indentando o código

A indentação do código facilita o entendimento do código
Separa através do espaçamento os blocos de comandos
Mostra intuitivamente a hierarquia no código.

Dicas simples:

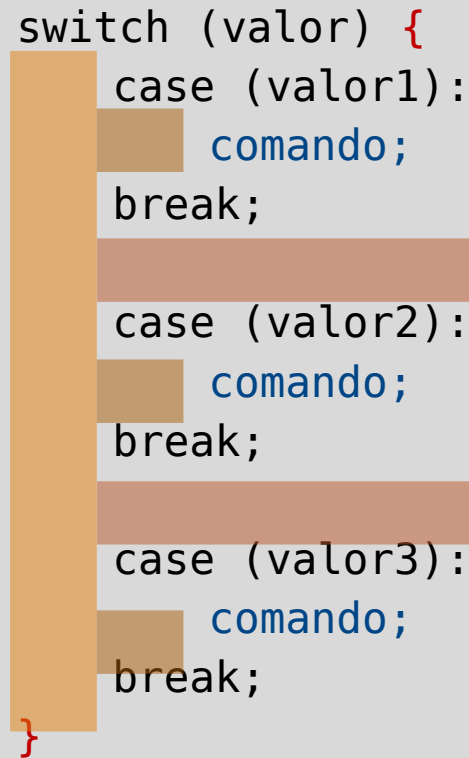
Podemos incluir um ENTER para separar a declaração de variáveis dos comandos do programa.

Podemos usar 1 TAB em todos comandos que estão dentro da condicional if. Ou para todos comando que estão dentro de um laço while().

Indentando o código

Exemplo 1

```
switch (valor) {  
case (valor1):  
comando;  
break;  
case (valor2):  
comando;  
break;  
case (valor3):  
comando;  
break;  
}
```



The diagram shows the same code as the left panel, but with visual aids for indentation. A long orange vertical bar is on the left, spanning the entire switch block. Three horizontal orange bars are positioned to the right of the code, each corresponding to a case block. The first horizontal bar is aligned with the 'case (valor1):' line, the second with 'case (valor2):', and the third with 'case (valor3):'. The code itself is color-coded: 'switch' and the closing brace '}' are red, 'case' is black, 'comando' is blue, and 'break;' is black.

```
switch (valor) {  
case (valor1):  
comando;  
break;  
case (valor2):  
comando;  
break;  
case (valor3):  
comando;  
break;  
}
```

Indentando o código

Exemplo 2

```
#include <stdio.h>
main (){int contador=0, numero;
scanf("%d",&numero); while
(contador < numero)
{ printf("%d\n", contador);
contador++;}}
```

```
#include <stdio.h>

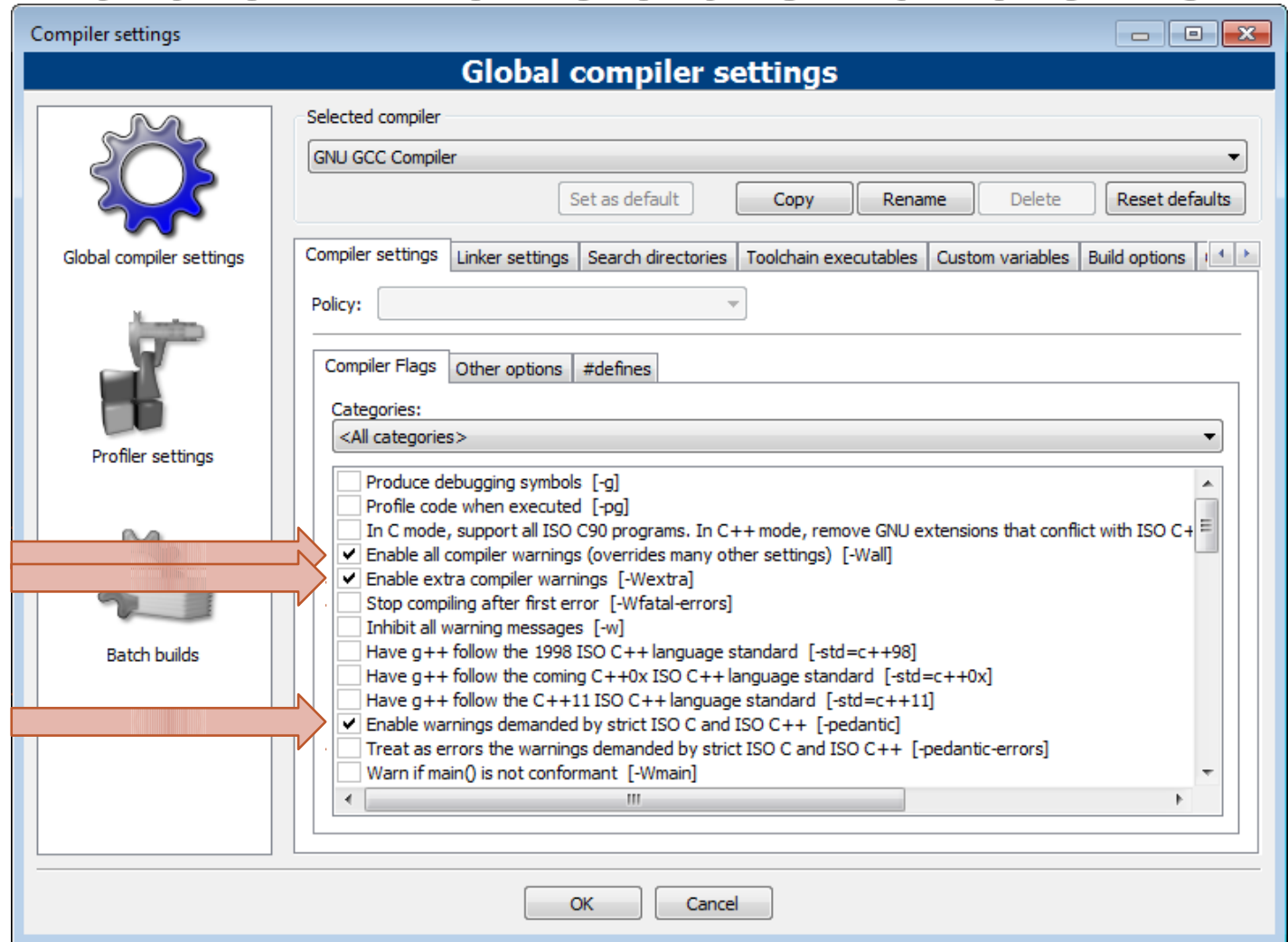
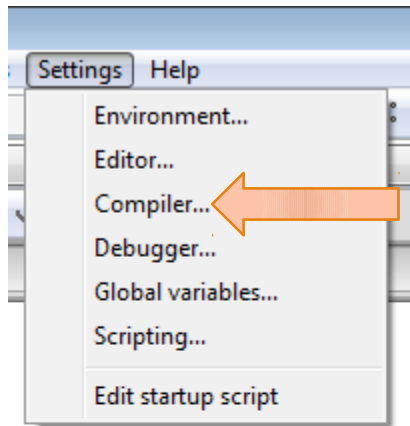
main () {
    int contador=0, numero;

    scanf ("%d",&numero);

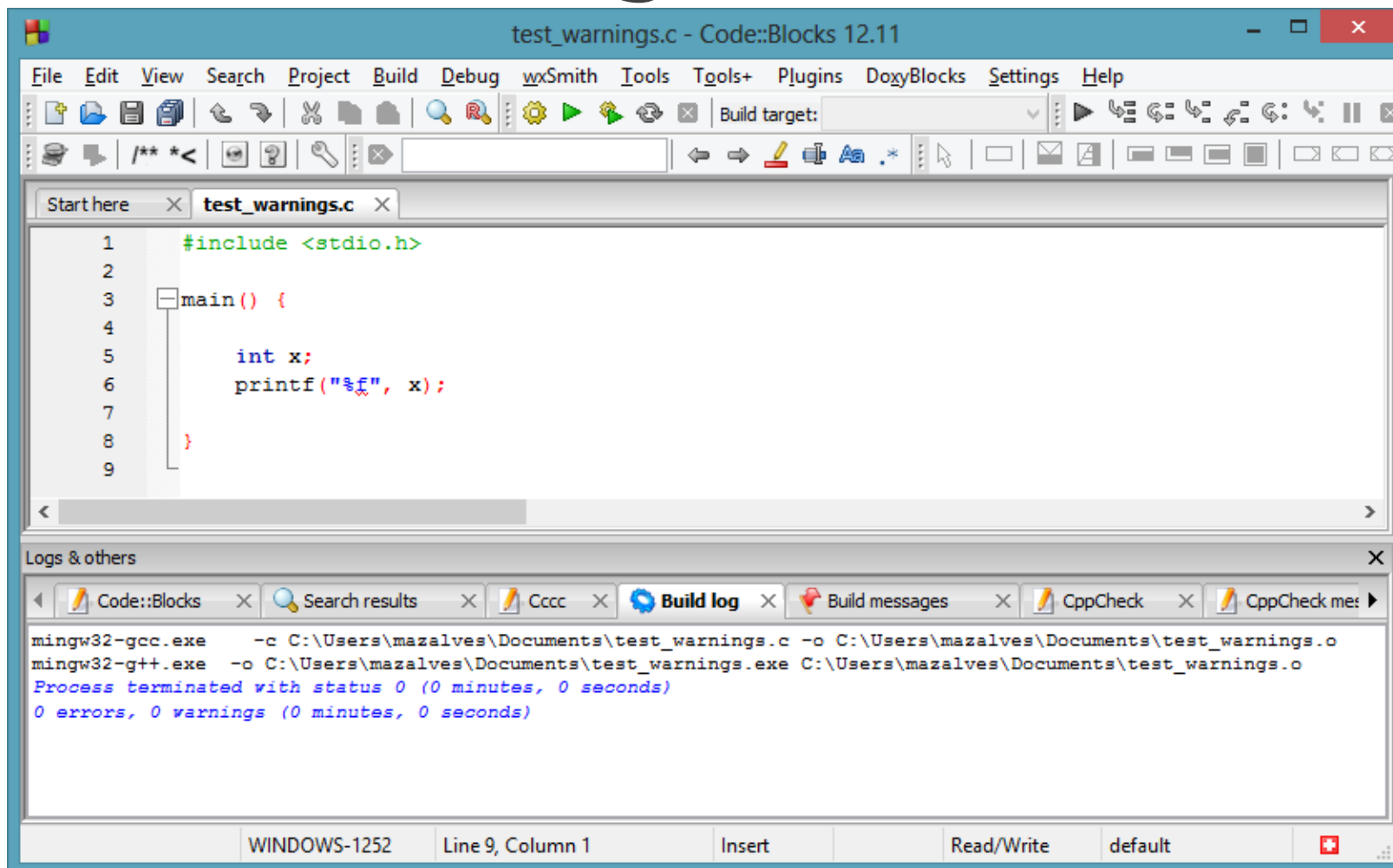
    // Conta 0 até Numero
    while (contador < numero) {
        printf("%d\n", contador);
        contador++;
    }
}
```

Obtendo mais mensagens do compilador

Ativando mais avisos do compilador no code-blocks



Sem Mensagens Extra



The screenshot shows the Code::Blocks 12.11 IDE. The main editor window displays a C program named `test_warnings.c` with the following code:

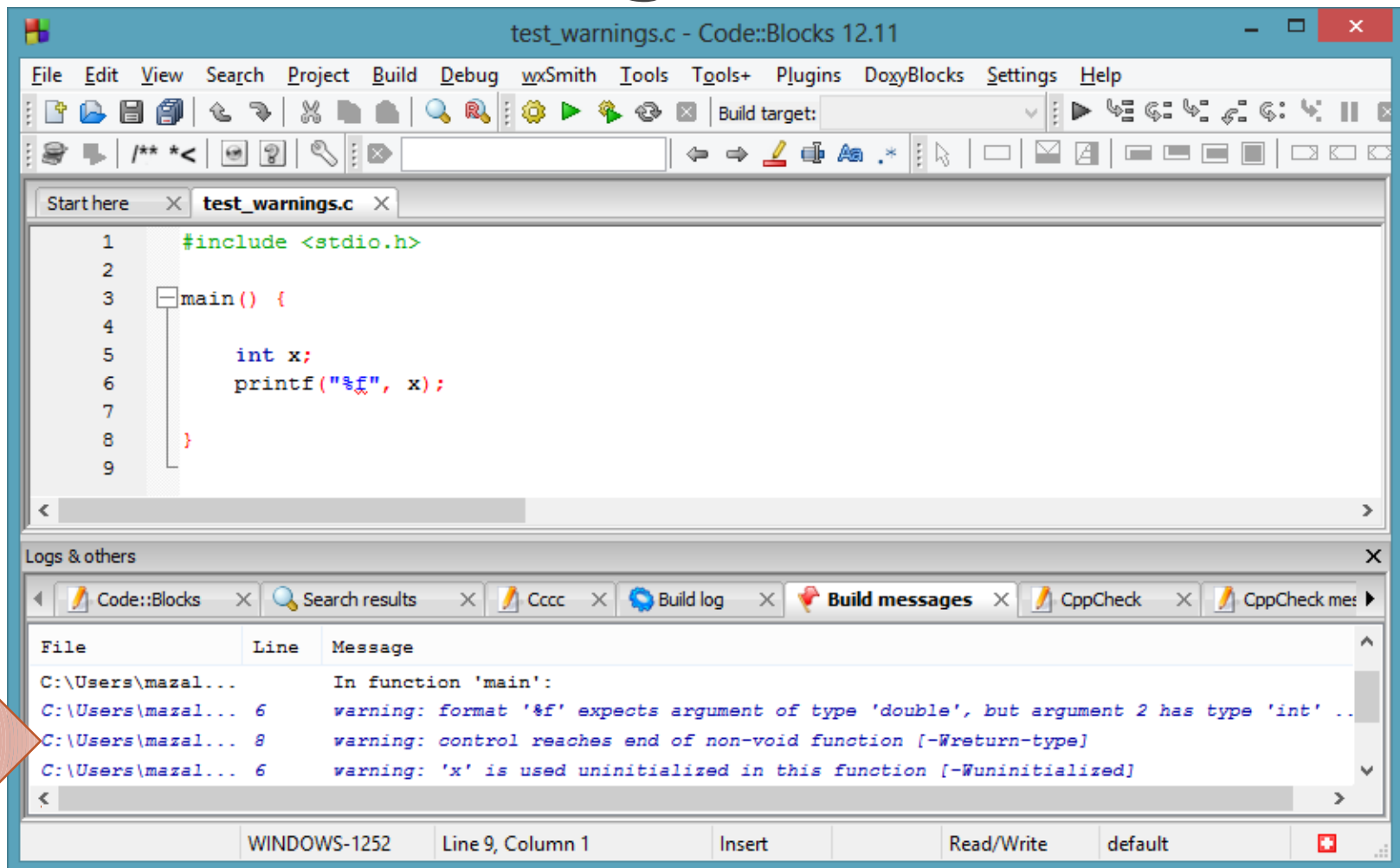
```
1  #include <stdio.h>
2
3  main() {
4
5      int x;
6      printf("%f", x);
7
8  }
```

The 'Logs & others' panel at the bottom shows the compilation output:

```
mingw32-gcc.exe -c C:\Users\mazalves\Documents\test_warnings.c -o C:\Users\mazalves\Documents\test_warnings.o
mingw32-g++.exe -o C:\Users\mazalves\Documents\test_warnings.exe C:\Users\mazalves\Documents\test_warnings.o
Process terminated with status 0 (0 minutes, 0 seconds)
0 errors, 0 warnings (0 minutes, 0 seconds)
```

The status bar at the bottom indicates the current position is Line 9, Column 1.

Com Mensagens Ativas



The screenshot shows the Code::Blocks 12.11 IDE with a C program named `test_warnings.c`. The code is as follows:

```
1  #include <stdio.h>
2
3  main() {
4
5      int x;
6      printf("%f", x);
7
8  }
```

The bottom panel, titled "Logs & others", displays the "Build messages" tab. It shows three warnings generated by the compiler:

- In function 'main':
- warning: format '%f' expects argument of type 'double', but argument 2 has type 'int' ..
- warning: control reaches end of non-void function [-Wreturn-type]
- warning: 'x' is used uninitialized in this function [-Wuninitialized]

A large orange arrow points from the left towards the "Build messages" tab in the logs panel.

At the bottom of the IDE, the status bar indicates "WINDOWS-1252", "Line 9, Column 1", and the "Insert" mode.

Com Mensagens Ativas

```
1  #include <stdio.h>
2
3  main() {
4
5      int x;
6      printf("%f", x);
7
8  }
9
```

Line	Message
za....	In function 'main':
za.... 6	warning: format '%f' expects argument of type 'double', but argument 2 has type 'int' .
za.... 8 ???	warning: control reaches end of non-void function [-Wreturn-type]
za.... 6	warning: 'x' is used uninitialized in this function [-Wuninitialized]

Entendendo as mensagens (tradutor)

Tradutor

Do: inglês ▾



Para o: português ▾

Traduzir

português inglês espanhol Detectar idioma

warning: format '%f' expects argument of type 'double', but argument 2 has type 'int' ✕
warning: 'x' is used uninitialized in this function



inglês português espanhol

aviso: 'f%' formato espera argumento de 'duplo' tipo, mas tem argumento 2 'int' tipo
aviso: 'x' é usado nesta função não inicializado

