

Manual do Code::Blocks para o Windows XP, Vista e 7

Instalação:

- 1) Utilize o link abaixo para iniciar automaticamente o processo para baixar o arquivo de instalação do Code::Blocks:

<http://prdownload.berlios.de/codeblocks/codeblocks-12.11mingw-setup.exe>

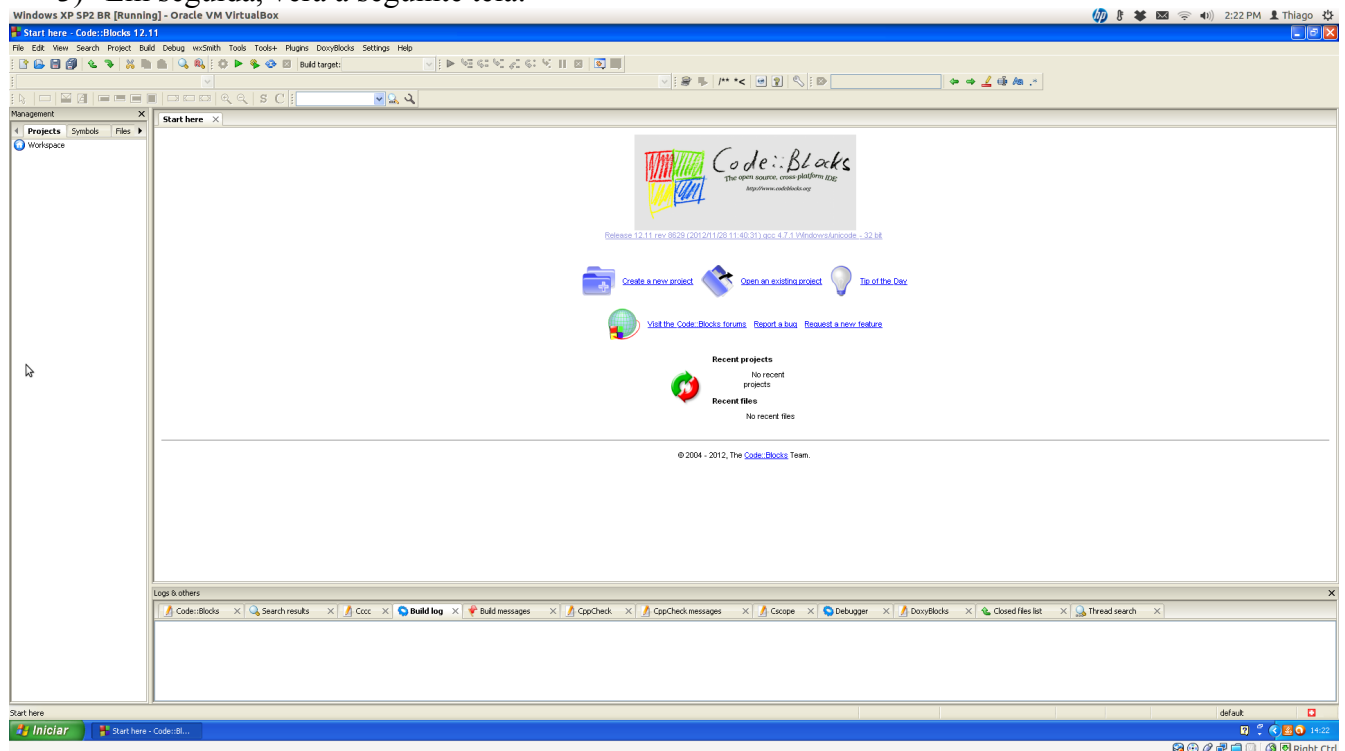
- 2) Abra a pasta na qual o arquivo codeblocks-12.11mingw-setup.exe foi baixado e execute o instalador.

- 3) Aceite todas as opções padrões e prossiga com a instalação clicando em Próximo (Next).

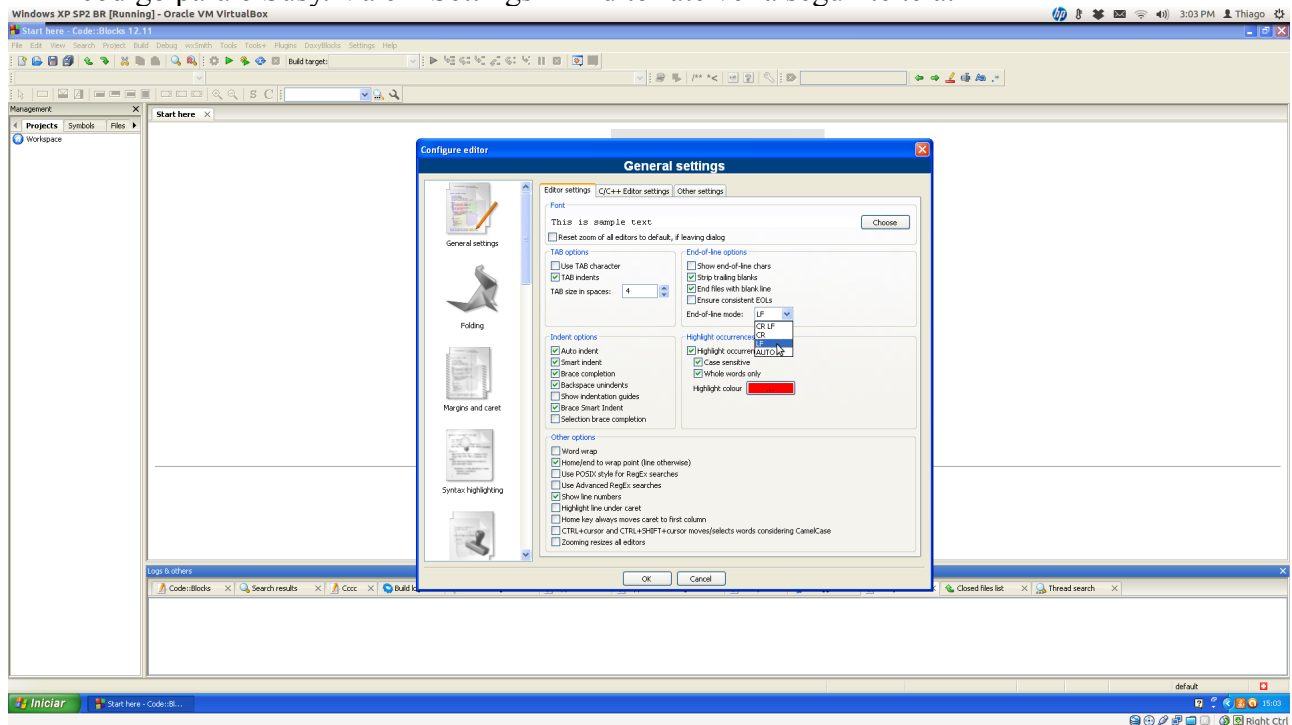
IMPORTANTE: O Code::Blocks também funciona no Linux e deve estar disponível no laboratório de informática do PB.

Primeiro uso do Code::Blocks

- 1) Abra o Code::Blocks pelo link criado na Área de Trabalho, ou em Iniciar → Todos os Programas → Codeblocks → Codeblocks
- 2) Na primeira vez que o programa for iniciado, ele buscará o compilador GCC no sistema (o mesmo utilizado no Linux). O programa deve ser encontrado já que está incluso no arquivo instalador do link anterior, portanto apenas dê Ok ou Continuar.
- 3) Em seguida, verá a seguinte tela:

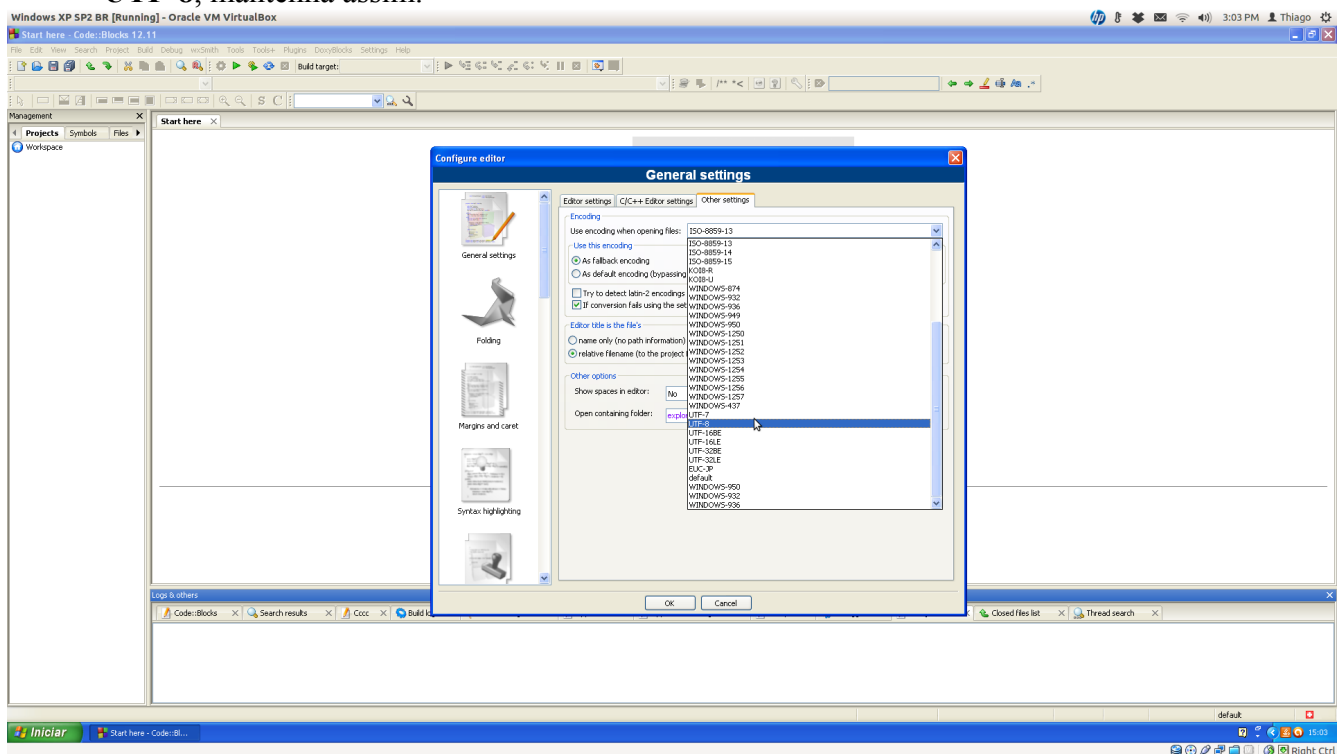


- 4) Precisamos agora configurar algumas questões técnicas para evitar problemas na hora de enviar código para o Susy. Vá em Settings → Editor até ver a seguinte tela:



- 5) Altere o End-of-line mode (visto na figura acima) de CR LF para **LF** na caixa de seleção correspondente. Caso já esteja como **LF**, mantenha assim.

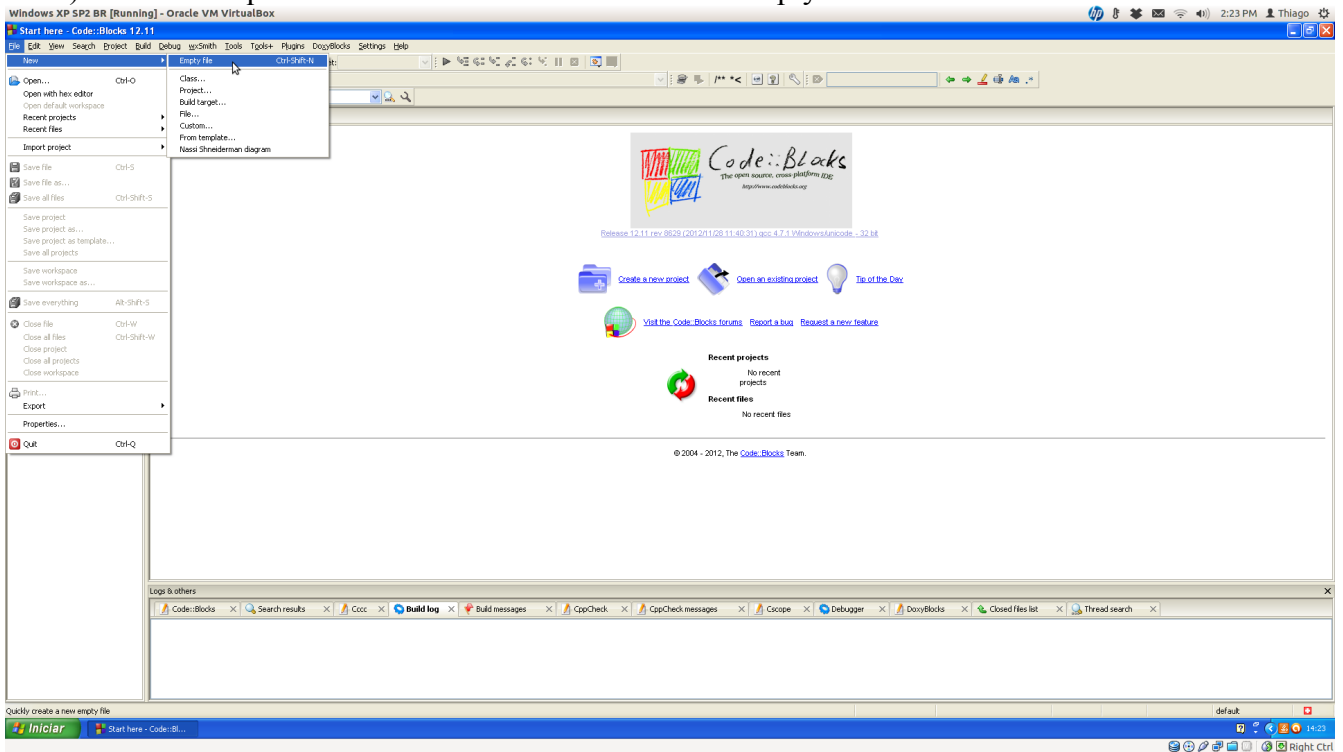
- 6) Vá para a aba “Other settings” e altere o encoding de Windows 1252 (ou qualquer um que esteja sendo utilizado) para **UTF-8** na caixa de seleção correspondente. Caso já esteja como **UTF-8**, mantenha assim.



7) Clique em Ok para salvar as alterações acima.

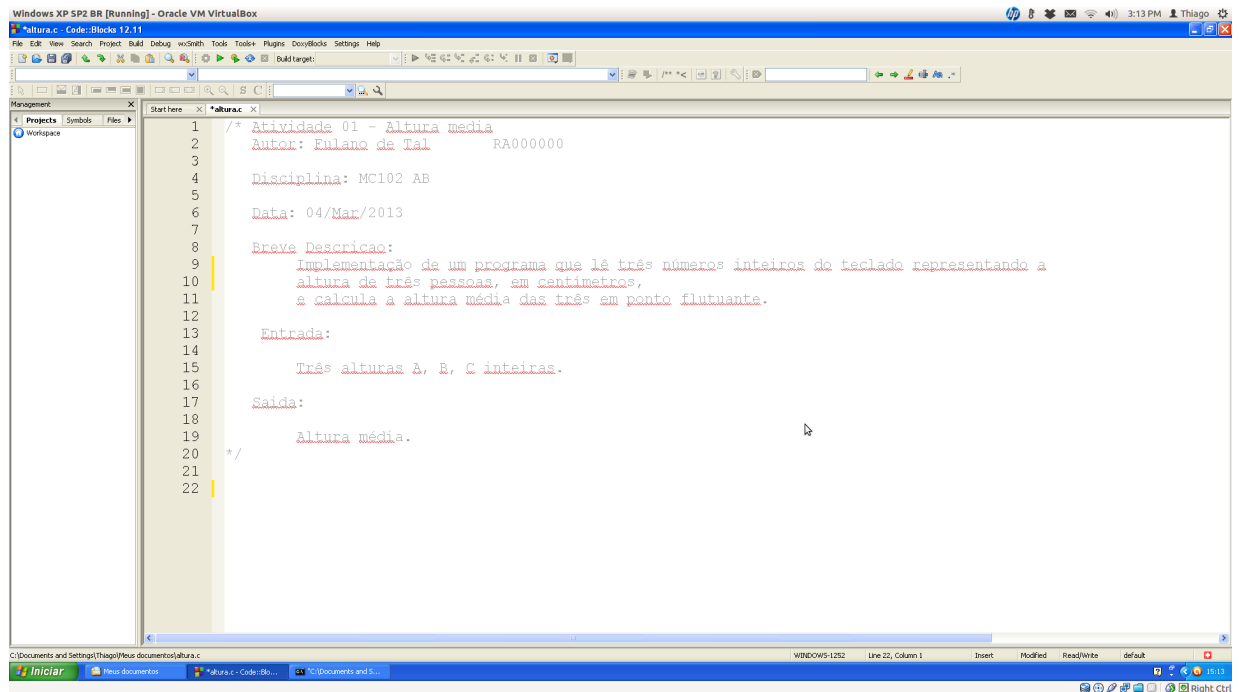
Criação de um arquivo .c e uso do Code::Blocks

8) Crie um arquivo novo utilizando File → New → Empty file.



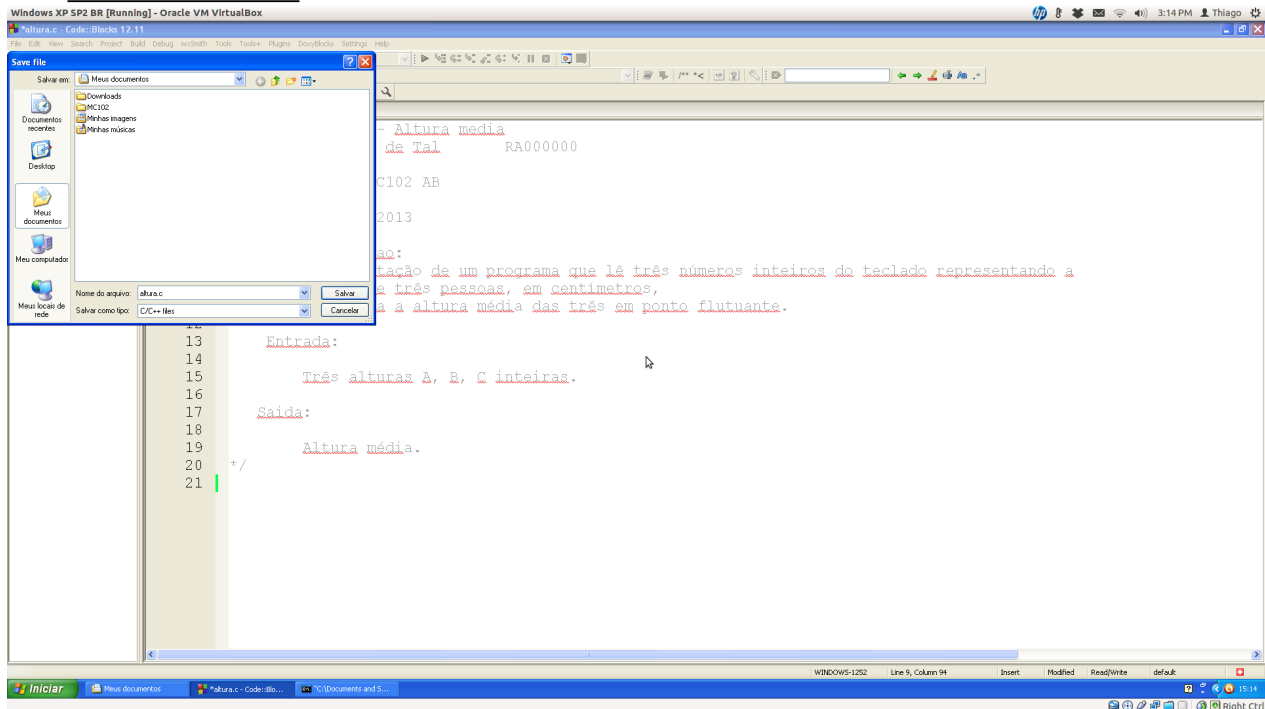
9) O arquivo criado está pronto para ser transformado em código fonte para programas.

10) Sempre comece o arquivo com um cabeçalho com todas as informações relevantes para correção e descrição do programa criado, isto é, Nome, RA, Entrada, Saída, Descrição.

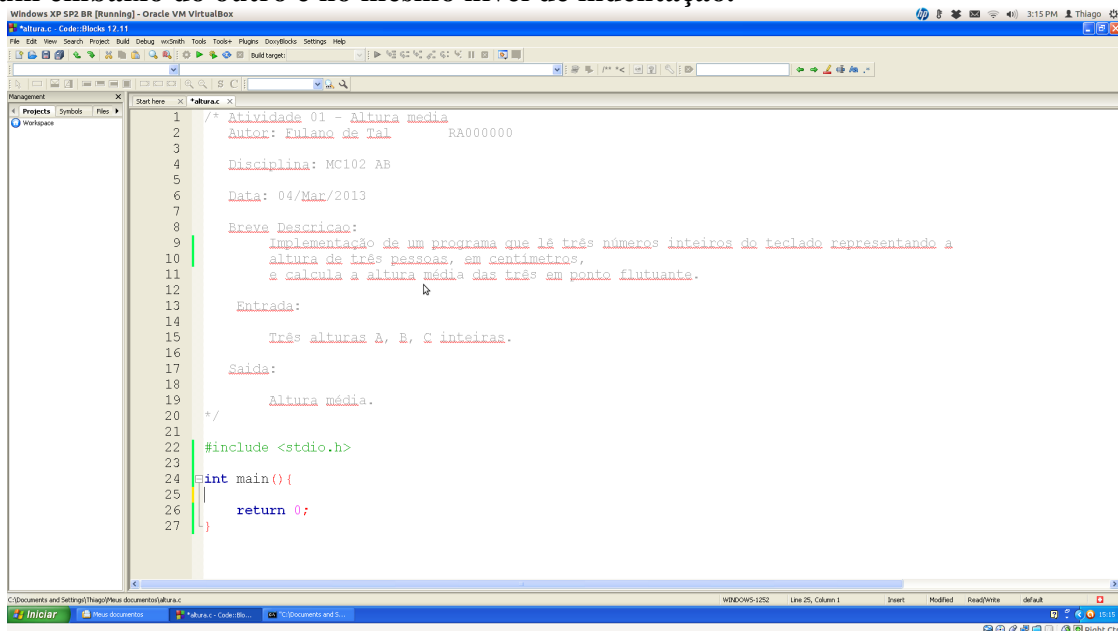


- 11) Salve o arquivo após digitar o cabeçalho, utilizando um nome apropriado terminado com a extensão .c . Para tanto, vá em File → Save File (CTRL + S) e escolha um nome. No caso de exemplo utilizamos altura.c .

Organize seus programas em pastas do melhor modo que preferir, mas seja ORGANIZADO.

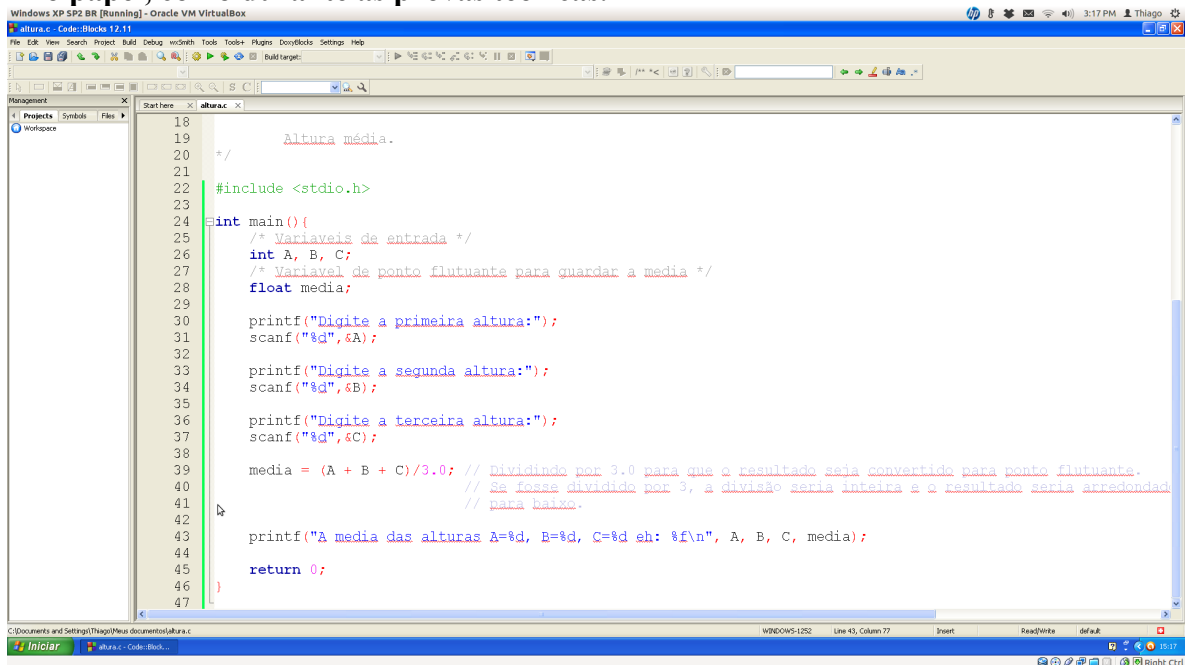


- 12) Crie agora a função main, logo após inserir a biblioteca stdio.h. Para indentar o programa (inserir espaço horizontal quando adicionar código entre { }) utilize a tecla TAB para mover a linha para a direita. Note que cada comando dentro de { } deverá aparecer sequencialmente um embaixo do outro e no mesmo nível de indentação.



- 13) Insira o restante do código lembrando sempre de **indentá-lo** com a tecla TAB e **comentá-lo**

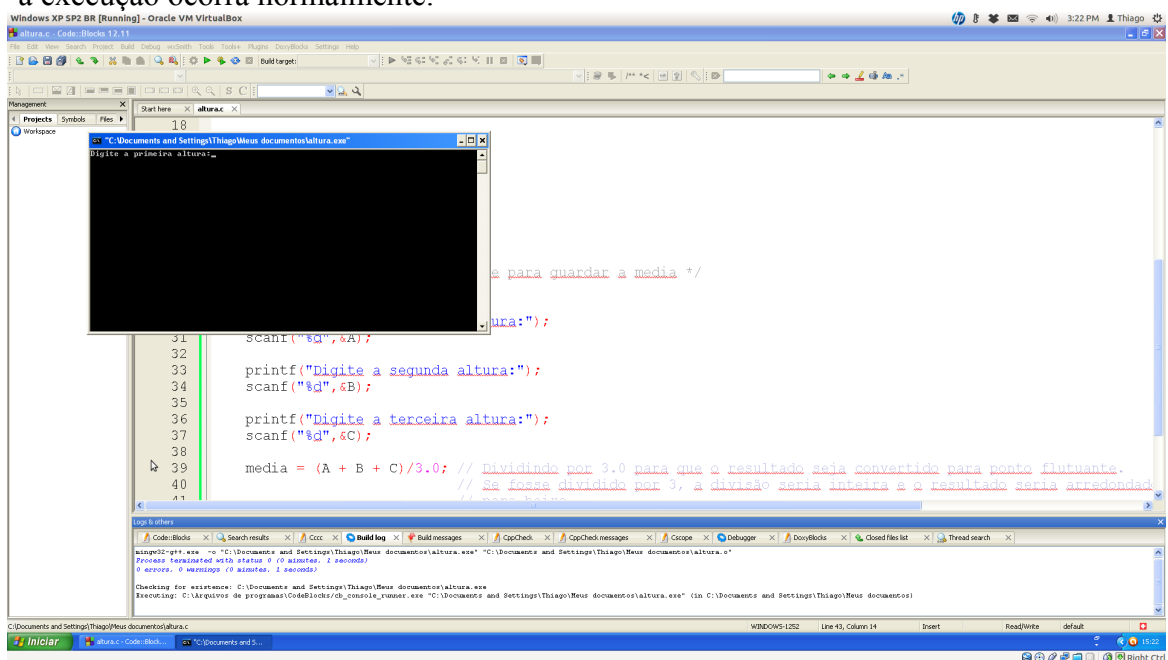
apropriadamente. Note que o Code::Blocks tende a indentar o código automaticamente, mas haverá vários momentos em que a indentação deverá ser corrigida manualmente ou feita no papel, como durante as provas teóricas.



```
18      Altura média.
19
20  */
21
22  #include <stdio.h>
23
24  int main(){
25      /* Variáveis de entrada */
26      int A, B, C;
27      /* Variável de ponto flutuante para guardar a media */
28      float media;
29
30      printf("Digite a primeira altura:");
31      scanf("%d",&A);
32
33      printf("Digite a segunda altura:");
34      scanf("%d",&B);
35
36      printf("Digite a terceira altura:");
37      scanf("%d",&C);
38
39      media = (A + B + C)/3.0; // Dividindo por 3.0 para que o resultado seja convertido para ponto flutuante.
40                             // Se fosse dividido por 3, a divisão seria inteira e o resultado seria arredondado
41                             // para baixo.
42
43      printf("A media das alturas A=%d, B=%d, C=%d eh: %f\n", A, B, C, media);
44
45      return 0;
46  }
```

Compilação e execução do programa

- 1) Após alterar o arquivo .c, salve-o e aperte a tecla F9 (Build → Build and Run) para compilá-lo e executá-lo. Pressione também F2 para aparecer a um retângulo abaixo do código com a saída da compilação, caso já não esteja aparecendo.
- 2) Se a compilação for efetuada sem erros o programa será imediatamente executado como abaixo. Se o seu antivírus ou o Windows pedir permissão para executar o programa, conceda-a para que a execução ocorra normalmente.



```
18      Altura média.
19
20  */
21
22  #include <stdio.h>
23
24  int main(){
25      /* Variáveis de entrada */
26      int A, B, C;
27      /* Variável de ponto flutuante para guardar a media */
28      float media;
29
30      printf("Digite a primeira altura:");
31      scanf("%d",&A);
32
33      printf("Digite a segunda altura:");
34      scanf("%d",&B);
35
36      printf("Digite a terceira altura:");
37      scanf("%d",&C);
38
39      media = (A + B + C)/3.0; // Dividindo por 3.0 para que o resultado seja convertido para ponto flutuante.
40                             // Se fosse dividido por 3, a divisão seria inteira e o resultado seria arredondado
41                             // para baixo.
42
43      printf("A media das alturas A=%d, B=%d, C=%d eh: %f\n", A, B, C, media);
44
45      return 0;
46  }
```

Log & others

- Code::Blocks
- Search results
- Cccc
- Build log
- Build messages
- CopCheck
- CopCheck messages
- Scope
- Debugger
- DocyBlocks
- Closed files list
- Thread search

```
mingw2-gcc.exe -o "C:\Documents and Settings\Thiago\Meu documento\altura.exe" "C:\Documents and Settings\Thiago\Meu documento\altura.c"
Process terminated with status 0 (0 minutes, 1 seconds)
0 erros, 0 warnings (0 minutos, 1 segundos)

Checking for existence: C:\Documents and Settings\Thiago\Meu documento\altura.exe
Warning: C:\Program de programar\Code\blocks\cb_console_runner.exe "C:\Documents and Settings\Thiago\Meu documento\altura.exe" (in C:\Documents and Settings\Thiago\Meu documento)
```

- 3) Em nosso programa, cada altura será perguntada individualmente e cada resposta deverá ser dada também individualmente já que utilizamos 3 chamadas a scanf. Após digitar cada altura, pressione ENTER para que scanf efetue a leitura da altura e guarde na variável apropriada.

```
#include <stdio.h>

int main() {
    /* Variáveis de entrada */
    int A, B, C;
    /* Variável de ponto flutuante para guardar a média */
    float media;

    printf("Digite a primeira altura:");
    scanf("%d", &A);

    printf("Digite a segunda altura:");
    scanf("%d", &B);

    printf("Digite a terceira altura:");
    scanf("%d", &C);

    media = (A + B + C)/3.0; // Dividindo por 3.0 para que o resultado seja convertido para ponto flutuante.
                           // Se fosse dividido por 3, a divisão seria inteira e o resultado seria arredondado para baixo.

    printf("A média das alturas A=%d, B=%d, C=%d eh: %f\n", A, B, C, media);

    return 0;
}
```

Erros de compilação

- 1) Do mesmo modo que o programa GCC apresenta erros no terminal caso algo tenha sido feito de modo errado no código C, para compilar o código em Linux, o Code::Blocks também o faz no painel retangular abaixo do código (aberto anteriormente com F2).

```
#include <stdio.h>

int main() {
    /* Variáveis de entrada */
    int A, B, C;
    /* Variável de ponto flutuante para guardar a média */
    float media;

    printf("Digite a primeira altura:");
    scanf("%d", &A);

    printf("Digite a segunda altura:");
    scanf("%d", &B);

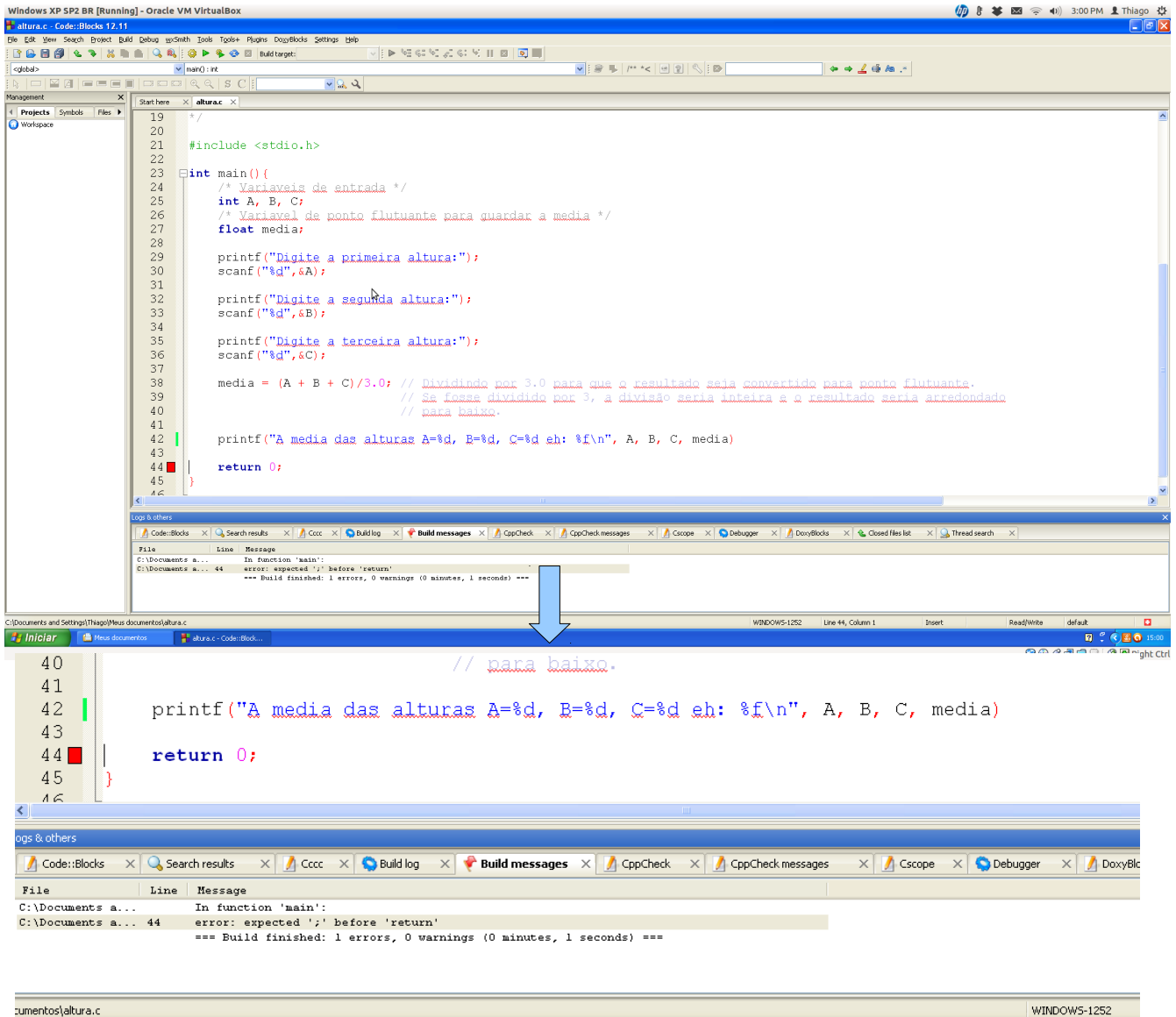
    printf("Digite a terceira altura:");
    scanf("%d", &C);

    media = (A + B + C)/3.0; // Dividindo por 3.0 para que o resultado seja convertido para ponto flutuante.
                           // Se fosse dividido por 3, a divisão seria inteira e o resultado seria arredondado para baixo.

    printf("A média das alturas A=%d, B=%d, C=%d eh: %f\n", A, B, C, media);

    return 0;
}
```

- 2) Caso um ponto-e-vírgula tenha sido esquecido após o último printf, por exemplo, a seguinte saída de erro aparecerá no painel:



```
19  /*
20
21  #include <stdio.h>
22
23  int main(){
24      /* Variaveis de entrada */
25      int A, B, C;
26      /* Variavel de ponto flutuante para guardar a media */
27      float media;
28
29      printf("Digite a primeira altura:");
30      scanf("%d",&A);
31
32      printf("Digite a segunda altura:");
33      scanf("%d",&B);
34
35      printf("Digite a terceira altura:");
36      scanf("%d",&C);
37
38      media = (A + B + C)/3.0; // Dividindo por 3.0 para que o resultado seja convertido para ponto flutuante.
39                             // Se fosse dividido por 3, a divisão seria inteira e o resultado seria arredondado
40                             // para baixo.
41
42      printf("A media das alturas A=%d, B=%d, C=%d eh: %f\n", A, B, C, media)
43
44      return 0;
45  }
```

Log & others

File	Line	Message
C:\Documents a...	44	In function 'main': error: expected ';' before 'return'
=== Build finished: 1 errors, 0 warnings (0 minutes, 1 seconds) ===		

40 // para baixo.
41
42 printf("A media das alturas A=%d, B=%d, C=%d eh: %f\n", A, B, C, media)
43
44 return 0;
45 }

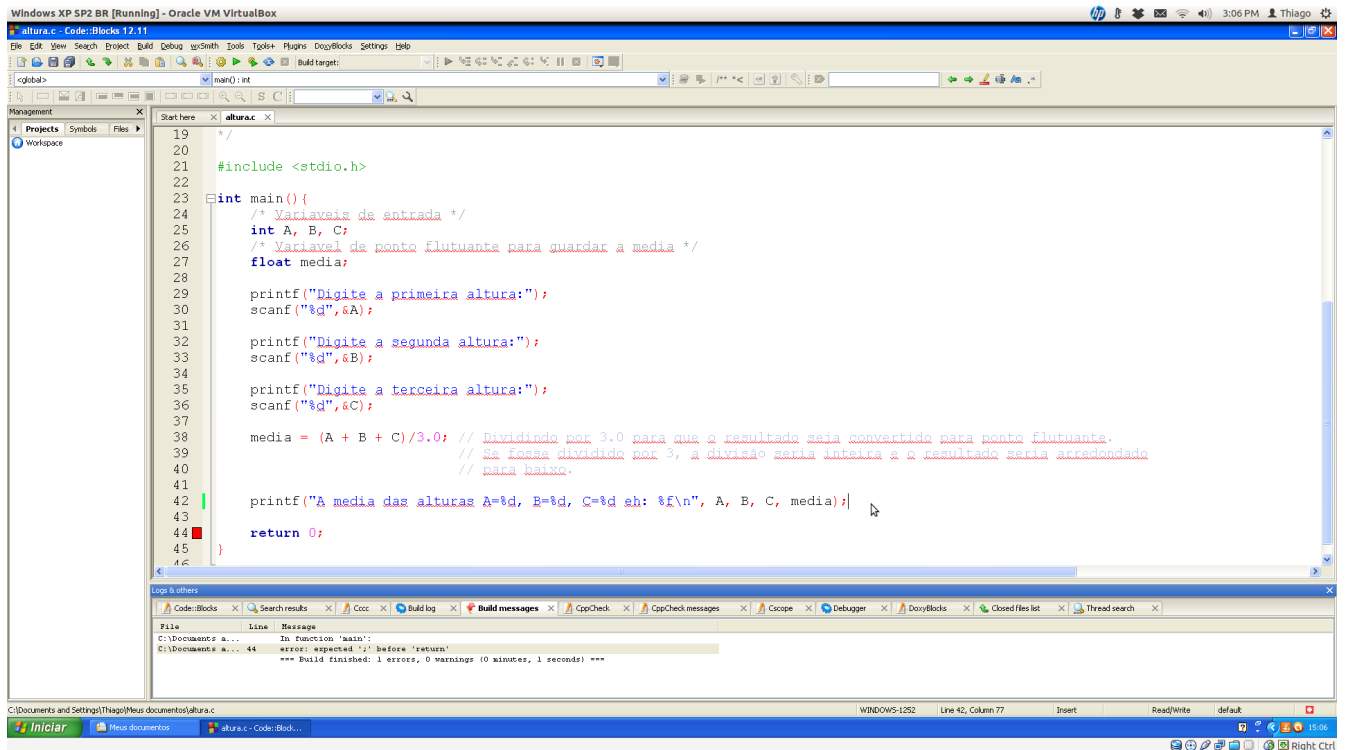
Log & others

File	Line	Message
C:\Documents a...	44	In function 'main': error: expected ';' before 'return'
=== Build finished: 1 errors, 0 warnings (0 minutes, 1 seconds) ===		

tumentos\altura.c WINDOWS-1252

- 3) Note que o erro foi apontado na linha 44, onde o compilador menciona que faltou um “;” antes do comando return (Line: 44 error: expected ';' before 'return'). Logo, você deve buscar na linha anterior (e na anterior da anterior sucessivamente), até achar o erro. Isto é, **neste caso** você deve procurar onde faltou “;”. Note também que um quadrado vermelho aparece ao lado da linha de código que apresenta erro.

4) Após o conserto do erro:

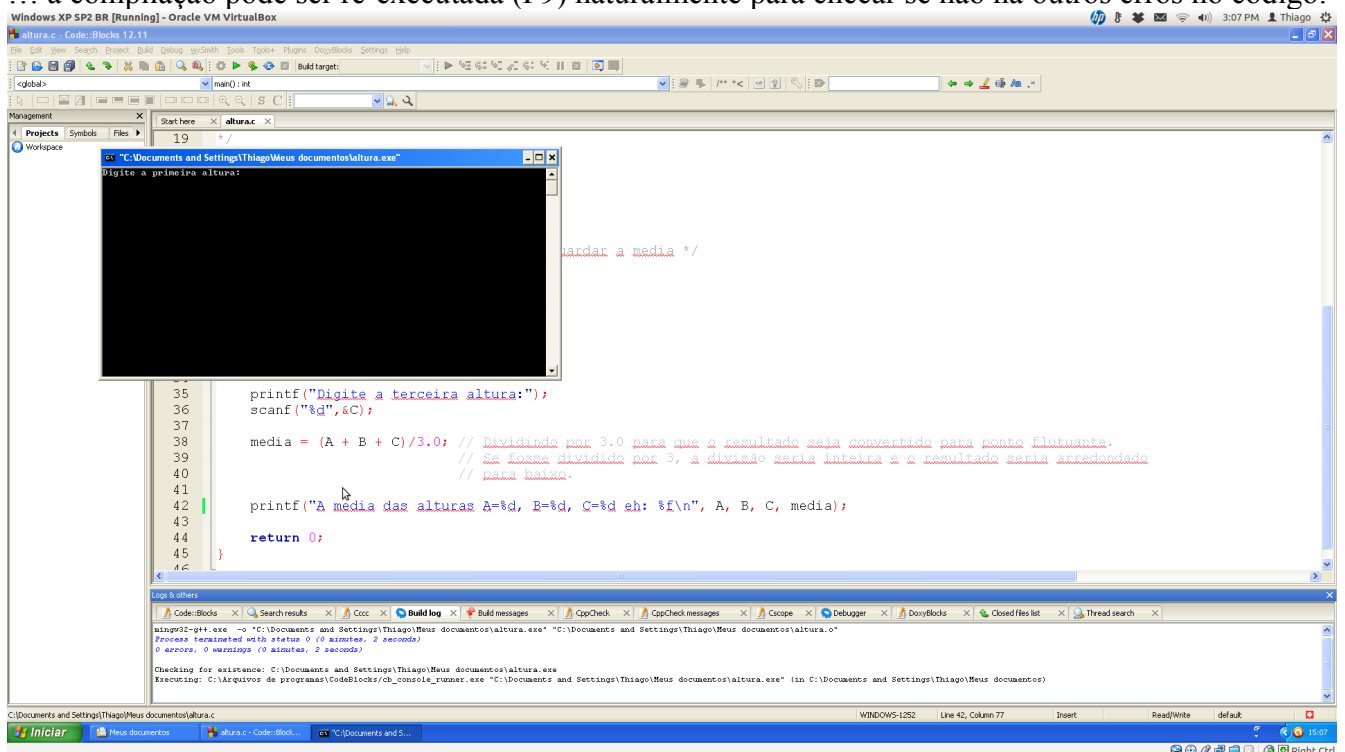


```
19  */
20
21  #include <stdio.h>
22
23  int main() {
24      /* Variáveis de entrada */
25      int A, B, C;
26      /* Variável de ponto flutuante para guardar a média */
27      float media;
28
29      printf("Digite a primeira altura:");
30      scanf("%d",&A);
31
32      printf("Digite a segunda altura:");
33      scanf("%d",&B);
34
35      printf("Digite a terceira altura:");
36      scanf("%d",&C);
37
38      media = (A + B + C)/3.0; // Dividindo por 3.0 para que o resultado seja convertido para ponto flutuante.
39                             // Se fosse dividido por 3, a divisão seria inteira e o resultado seria arredondado
40                             // para baixo.
41
42      printf("A média das alturas A=%d, B=%d, C=%d eh: %f\n", A, B, C, media);
43
44      return 0;
45  }
```

Build messages

File Line Message
C:\Documents and Settings\Thiago\Meu documentos\altura.c In function 'main':
C:\Documents and Settings\Thiago\Meu documentos\altura.c 44 error: expected ';' before 'return'
*** Build finished: 1 errors, 0 warnings (0 minutes, 1 seconds) ***

... a compilação pode ser re-executada (F9) naturalmente para checar se não há outros erros no código:



```
19  */
20
21  #include <stdio.h>
22
23  int main() {
24      /* Variáveis de entrada */
25      int A, B, C;
26      /* Variável de ponto flutuante para guardar a média */
27      float media;
28
29      printf("Digite a primeira altura:");
30      scanf("%d",&A);
31
32      printf("Digite a segunda altura:");
33      scanf("%d",&B);
34
35      printf("Digite a terceira altura:");
36      scanf("%d",&C);
37
38      media = (A + B + C)/3.0; // Dividindo por 3.0 para que o resultado seja convertido para ponto flutuante.
39                             // Se fosse dividido por 3, a divisão seria inteira e o resultado seria arredondado
40                             // para baixo.
41
42      printf("A média das alturas A=%d, B=%d, C=%d eh: %f\n", A, B, C, media);
43
44      return 0;
45  }
```

Build messages

File Line Message
C:\Documents and Settings\Thiago\Meu documentos\altura.exe Checking for existence: C:\Documents and Settings\Thiago\Meu documentos\altura.exe
Executing: C:\Arquivos de programas\CodeBlocks\cb_console_runner.exe "C:\Documents and Settings\Thiago\Meu documentos\altura.exe" (in C:\Documents and Settings\Thiago\Meu documentos)
Process terminated with status 0 (0 minutes, 2 seconds)
0 errors, 0 warnings (0 minutes, 2 seconds)

- 5) O processo de compilação, conserto de erros e execução de testes é iterativo, sendo necessária a re-execução de todos os passos até que todos os erros de programação e/ou lógicos sejam sanados.

Apêndice

Código do programa acima

/* Atividade 01 - Altura media
Autor: Fulano de Tal RA000000

Disciplina: MC102 AB

Data: 04/Mar/2013

Breve Descricao:

Implementação de um programa que lê três números inteiros do teclado representando a altura de três pessoas, em centímetros, e calcula a altura média das três em ponto flutuante.

Entrada:

Três alturas A, B, C inteiras.

Saida:

Altura média.

*/

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

```
int main(){
```

```
    /* Variaveis de entrada */
```

```
    int A, B, C;
```

```
    /* Variavel de ponto flutuante para guardar a media */
```

```
    float media;
```

```
    printf("Digite a primeira altura:");
```

```
    scanf("%d",&A);
```

```
    printf("Digite a segunda altura:");
```

```
    scanf("%d",&B);
```

```
    printf("Digite a terceira altura:");
```

```
    scanf("%d",&C);
```

```
    media = (A + B + C)/3.0; // Dividindo por 3.0 para que o resultado seja convertido para ponto flutuante.
```

```
                          // Se fosse dividido por 3, a divisão seria inteira e o resultado seria arredondado
```

```
                          // para baixo.
```

```
    printf("A media das alturas A=%d, B=%d, C=%d eh: %f\n", A, B, C, media);
```

```
    return 0;
```

```
}
```