1) contenu de ce CD-ROM

Ce CD-ROM contient une notice d'utilisation de CodeBlocks :

- `CodeBlocks_V10.05.pdf` : c'est le fichier que vous lisez en ce moment. Il s'agit de la documentation d'installation et d'utilisation en français de CodeBlocks.

Le CD-ROM contient aussi divers logiciels :

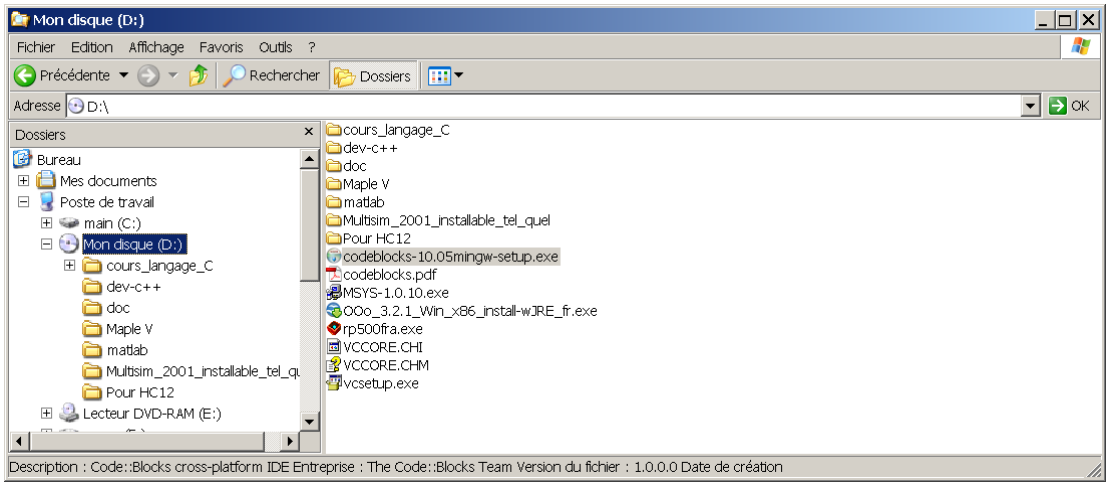
- `rp500fra.exe` : c'est Acrobat Reader, logiciel indispensable pour lire les documents au format pdf.
- **`codeblocks-10.05mingw-setup.exe`** : c'est le fichier exécutable contenant l'environnement de développement CodeBlocks version 10.05 (www.codeblocks.org).
- Dans le répertoire `Matlab` : Octave un logiciel clone de Matlab qui est très utilisé pour faire des calculs mathématiques ainsi que du traitement du signal.
- Dans le répertoire `Maple V` : Maple un logiciel de calcul symbolique mathématique version freeware.
- Dans le répertoire `Multisim_2001_installable_tel_quel` : logiciel de simulation de circuit analogique et numérique, à décompresser dans le répertoire `c:\multisim`. Le programme se lance en exécutant le fichier `multisim.exe`.
- Dans le répertoire `Pour HC12` : simulateur freeware pour microcontrôleur 68HC12.
- Dans le répertoire `doc` : c'est une copie du site CCM encyclopédie informatique http://www.commentcamarche.net. Il y a de plus le document `structure_des_ordinateurs.pdf` qui explique la structure interne d'un PC.
- Dans le répertoire `dev-c++` : c'est l'environnement de développement en C `Dev-C++ V5` avec sa documentation.

Sur ce CD-ROM, vous trouverez enfin dans le répertoire `cours_langage_C` :

- `introduction_ANSI_C.pdf` (230 pages) : c'est le cours C de Bernard CASSAGNE. Un must pour le programmeur expérimenté.
- `Langage_C.pdf` (47 pages) : un bon résumé bien compact.
- `COURS-C-HTML` : ce répertoire contient un cours C assez complet au format HTML (donc difficilement imprimable). Il faut ouvrir le fichier `home.htm` avec un navigateur Internet pour démarrer. C'est la copie du site http://www.ltam.lu/Tutoriel_Ansi_C/.
- `cours_IPA` : ce répertoire contient le polycopié de votre cours de programmation en langage C plus une documentation d'utilisation de visual C++ 6.0 en français.

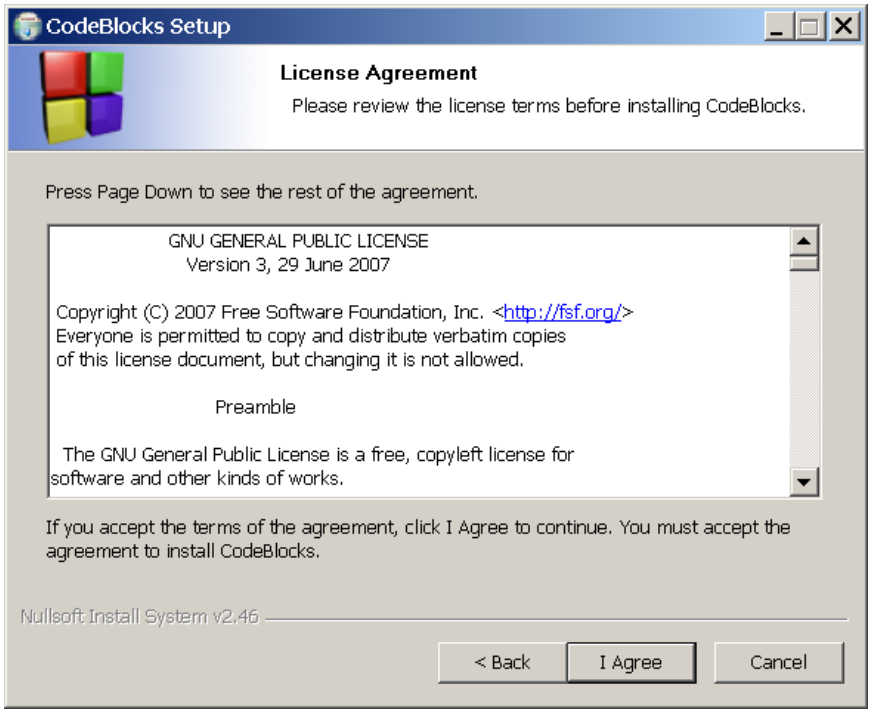
2) Installation de CodeBlocks *Utilisez un compte administrateur de Windows*

Cette procédure est valide pour Windows XP ou 2000. Pour Vista et Seven, elle doit marcher à l'identique mais avec au moins trois demandes de confirmation (du genre êtes-vous vraiment sûr que vous faites cela de votre plein gré) ; répondez oui à toutes ces demandes débiles et poursuivez l'installation. Mettez le CD-ROM dans le lecteur et double-cliquez sur le fichier **`codeblocks-10.05mingw-setup.exe`** qui se trouve dans le répertoire racine :

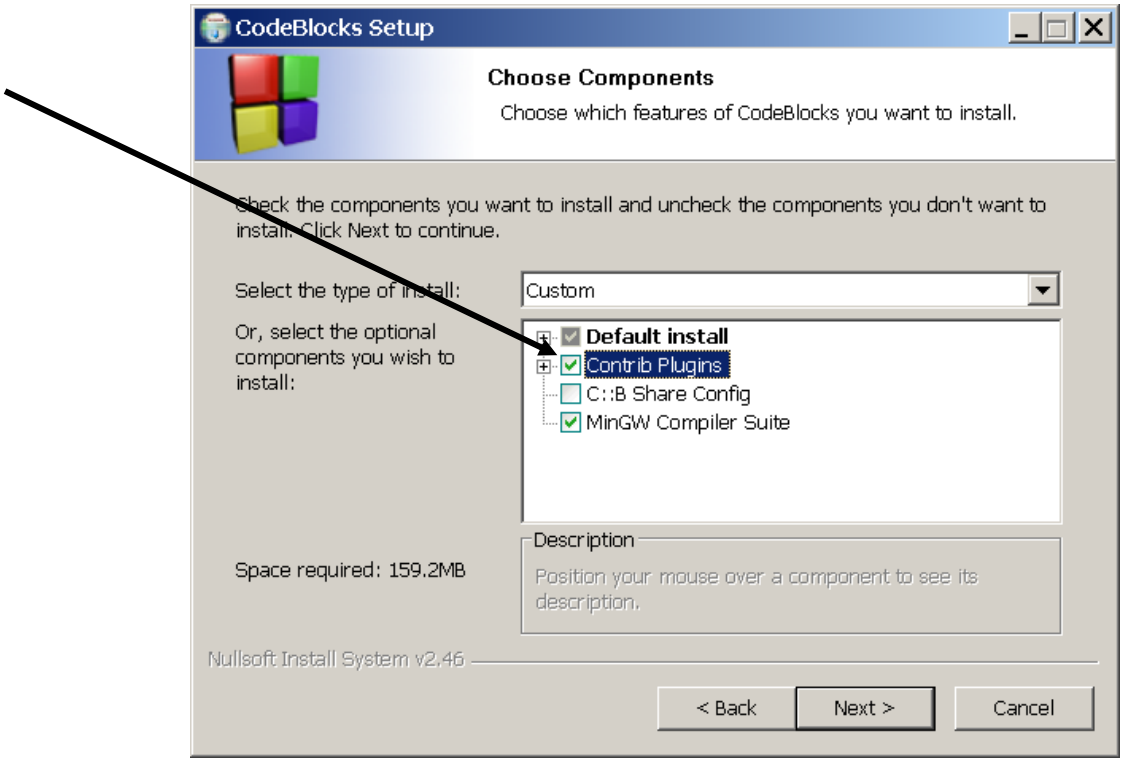


La procédure d'installation démarre. Dans la fenêtre suivante, cliquez sur le bouton « Next » :

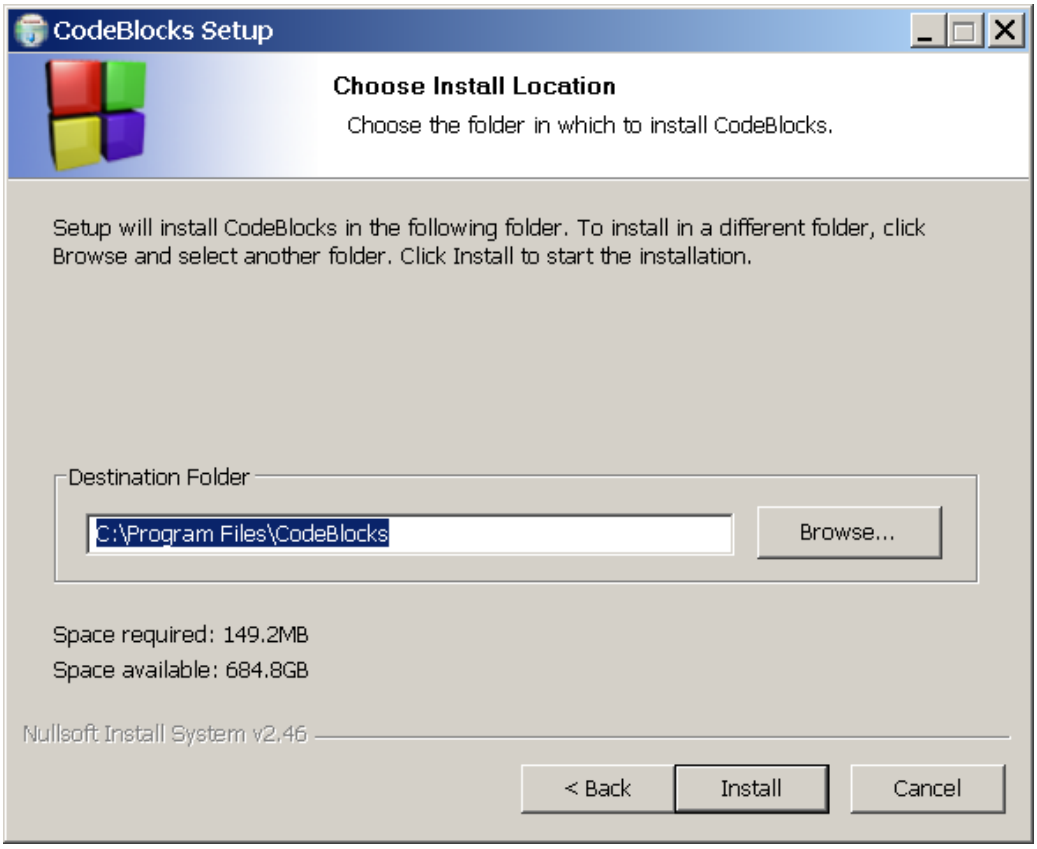
Dans la fenêtre suivante, cliquez sur le bouton « I Agree » :



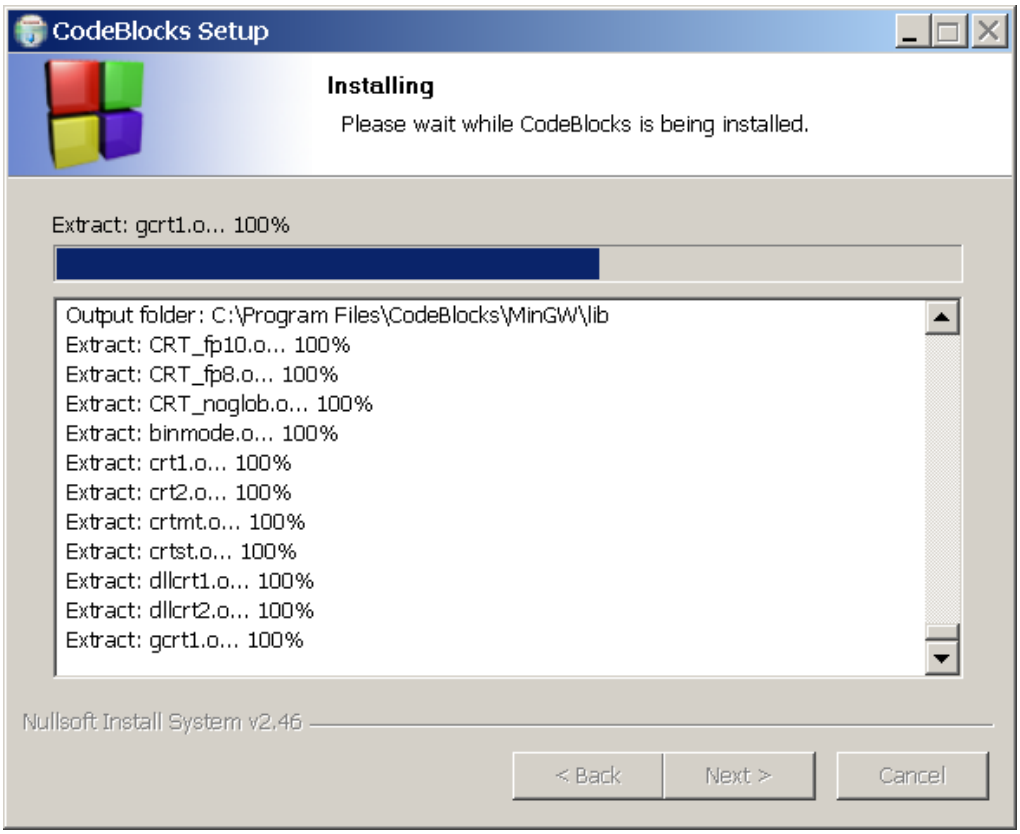
Dans la fenêtre suivante, cliquez sur la case « Contrib Plugins » pour la sélectionner (elle doit apparaître en vert et pas en gris) puis cliquez sur le bouton « Next » :



Le répertoire d'installation par défaut est `c:\Program Files\CodeBlocks`. Ne le changez pas sauf nécessité absolue. Cliquez sur le bouton « Install » :

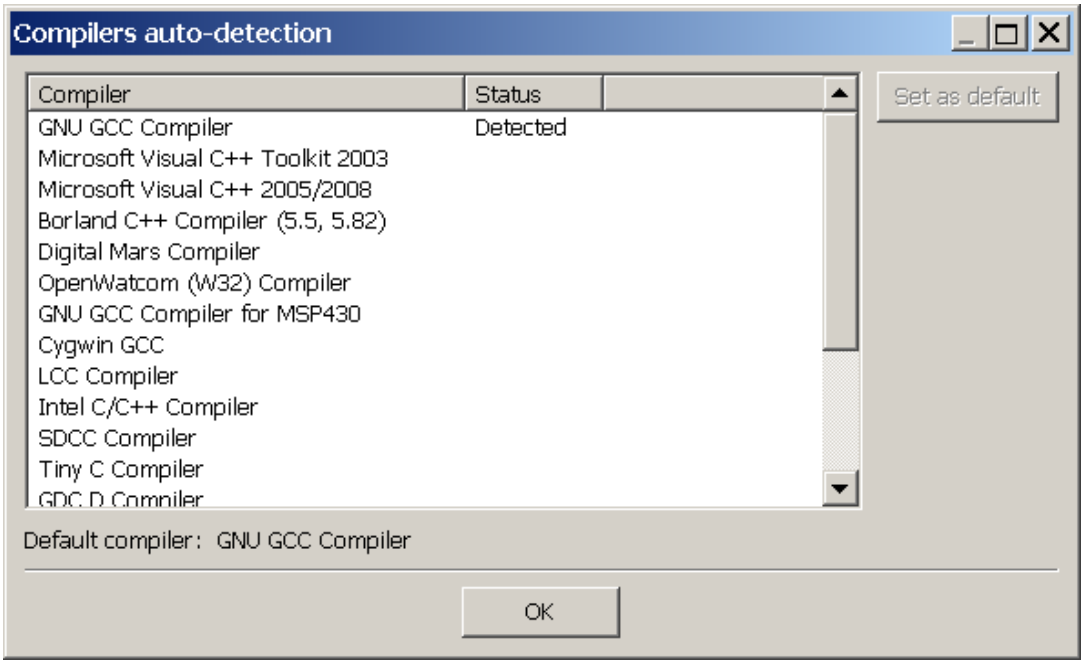


L'installation démarre :

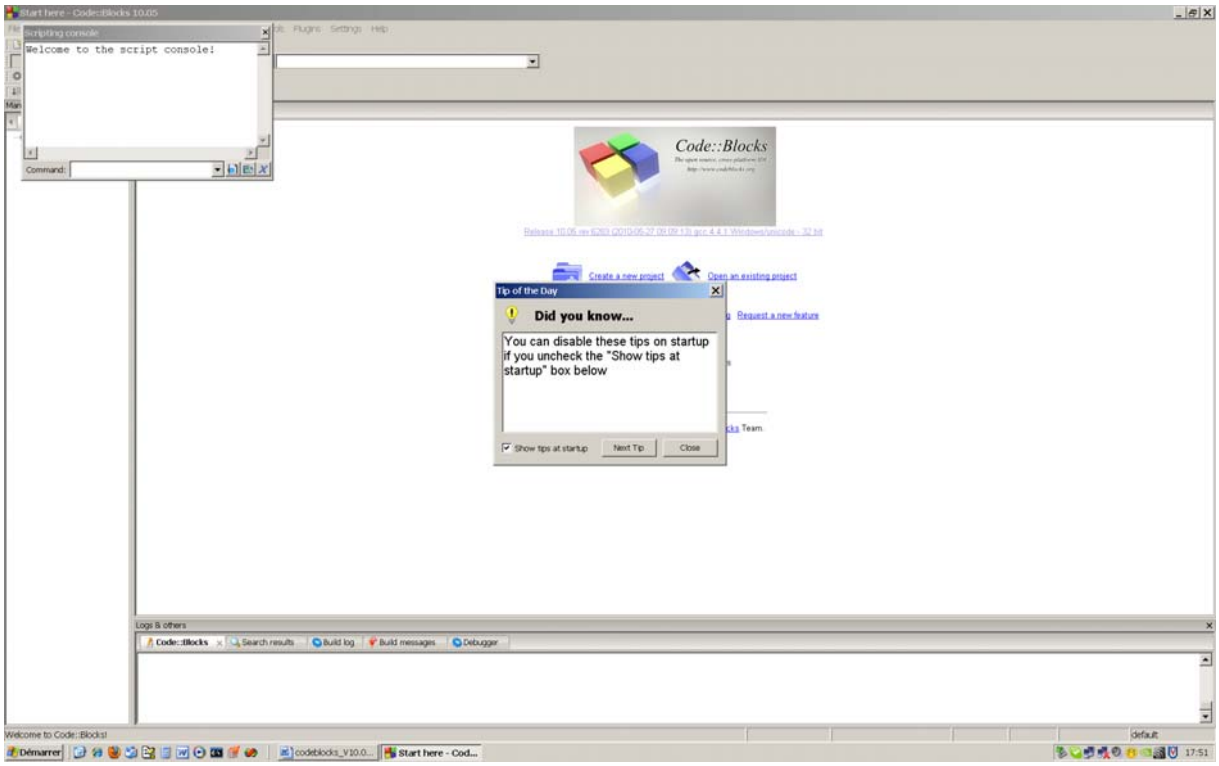


L'installation est terminée. Quand vous cliquerez sur le bouton « Oui » dans la fenêtre suivante, CodeBlocks sera automatiquement lancé.

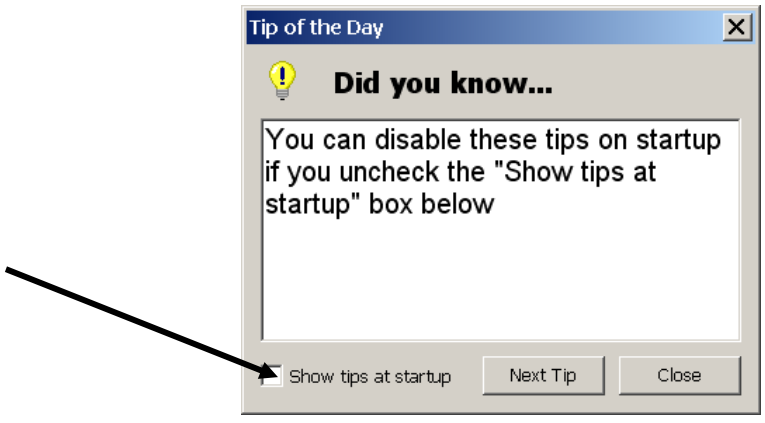
Au cours de ce premier lancement, certaines questions concernant la configuration du logiciel vont vous être posées. Elles n'apparaîtront plus lors des lancements ultérieurs. CodeBlocks sait travailler avec de nombreux compilateurs. Il doit au moins détecter le compilateur fourni avec ce CD, `GNU GCC`. Cliquez sur « OK » dans la fenêtre ci-dessous :



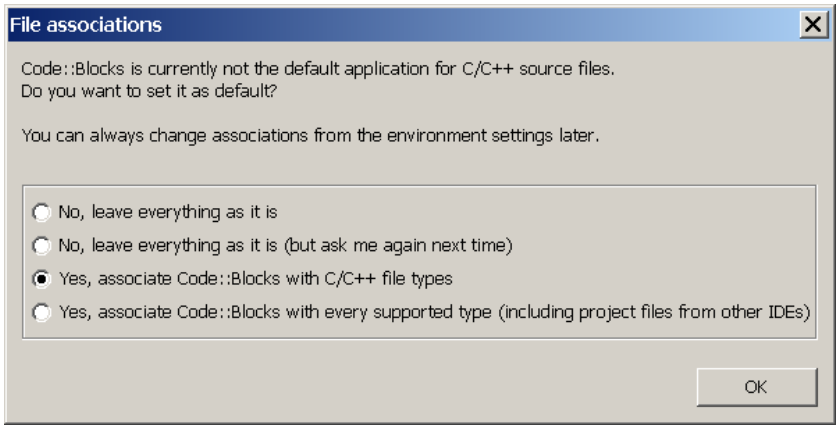
CodeBlocks apparaît à l'écran.



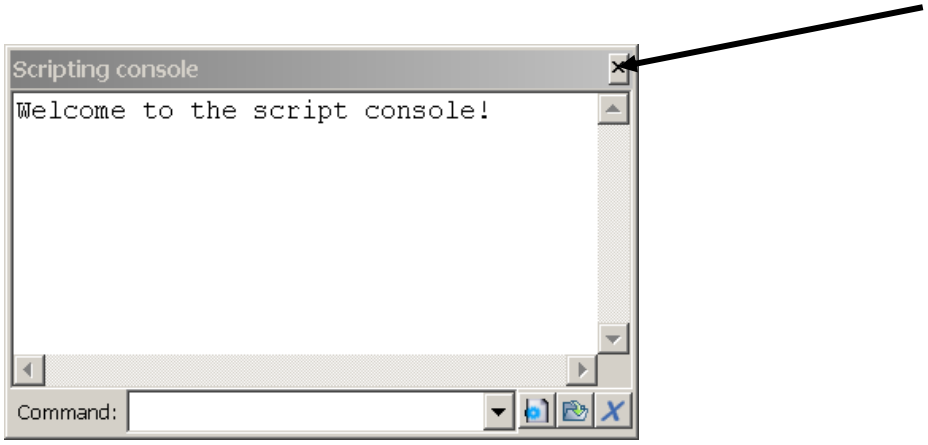
La fenêtre « Tip of the Day » apparaît à chaque lancement du logiciel. Décochez la case du bas (Show tips at startup) pour supprimer cette option puis cliquez sur le bouton « Close ».



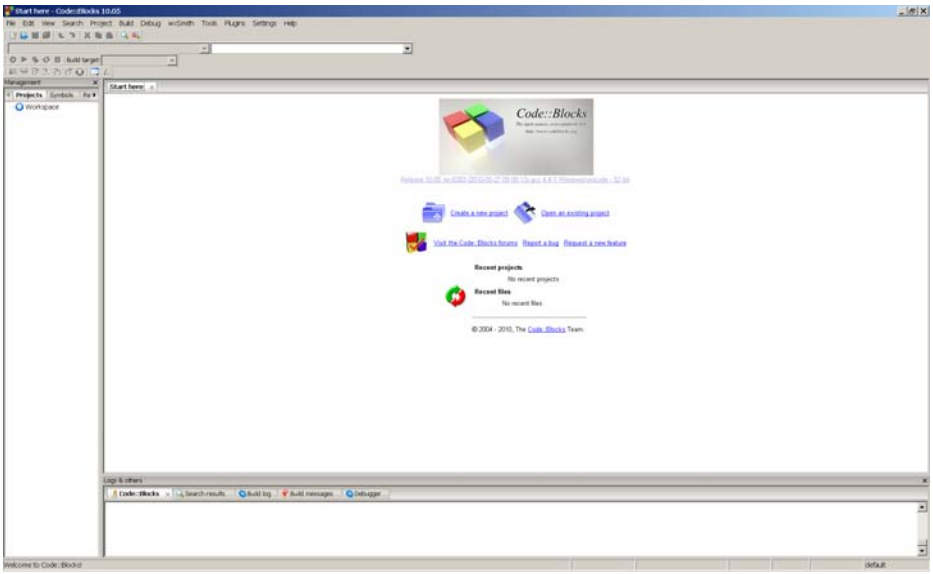
CodeBlocks demande maintenant si vous souhaitez qu'il soit l'outil de développement par défaut pour les fichiers C/C++. Cliquez sur le bouton « OK » » dans la fenêtre ci-dessous pour que cela soit le cas :



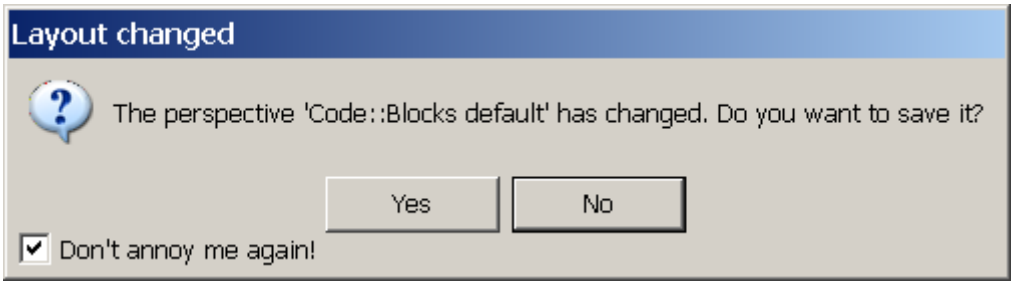
La console suivante peut être fermée en cliquant sur la croix, vous n'en aurez pas besoin dans l'immédiat :



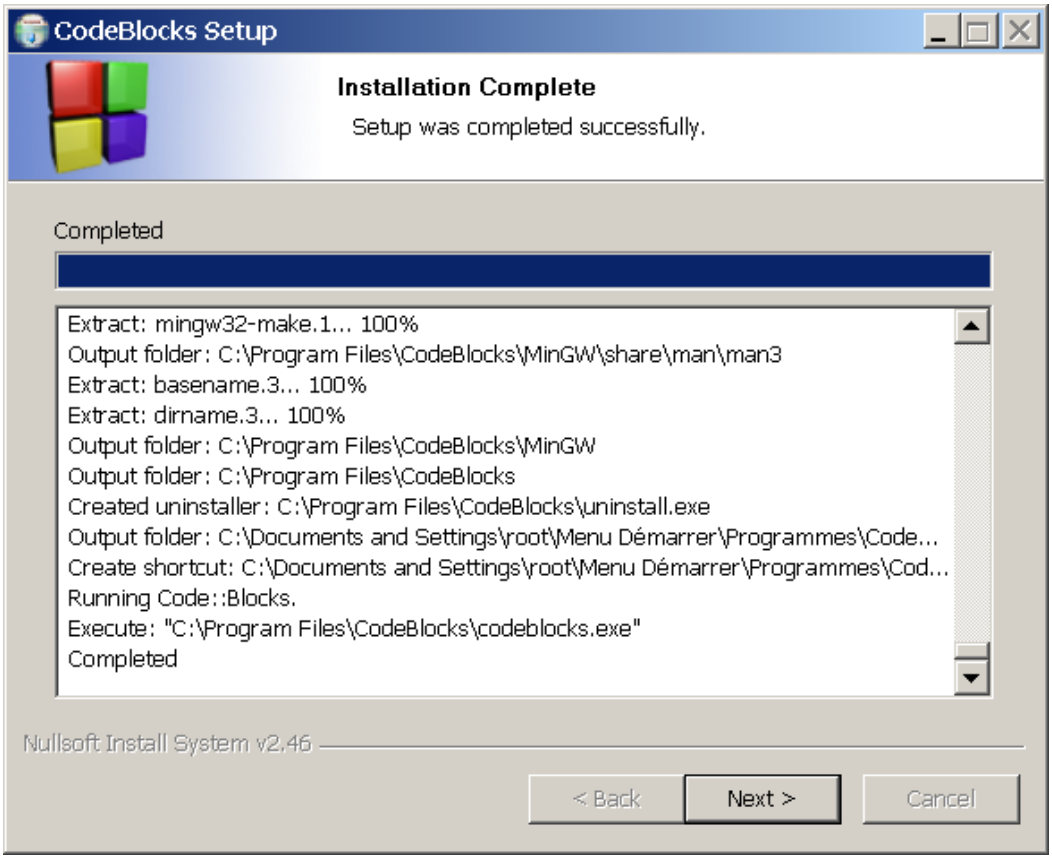
CodeBlocks peut maintenant être utilisé. Pour quitter l'application, cliquez sur le menu File puis sur Quit :



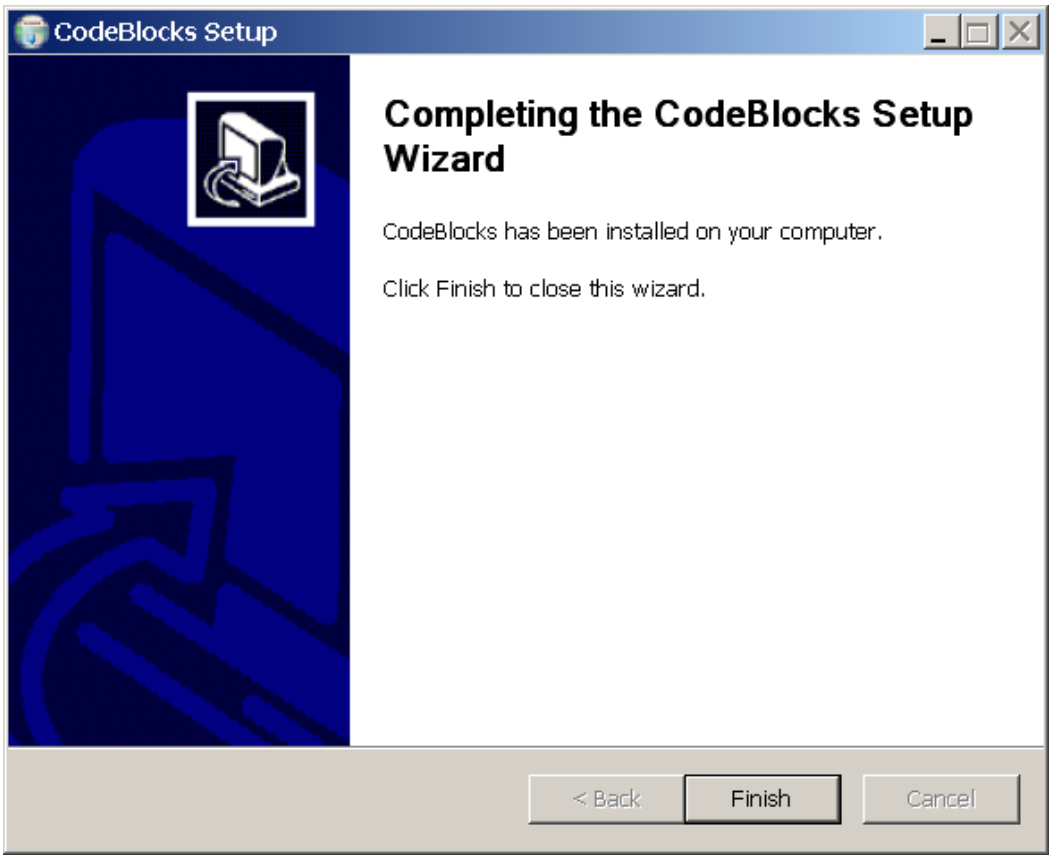
Dans la fenêtre qui s'ouvre, cochez la case « Don't annoy me again ! » puis cliquez sur le bouton « Yes ».



La fenêtre d'installation doit maintenant être fermée. Pour cela, cliquez sur le bouton « Next » dans la fenêtre suivante :



Puis sur le bouton « Finish » :

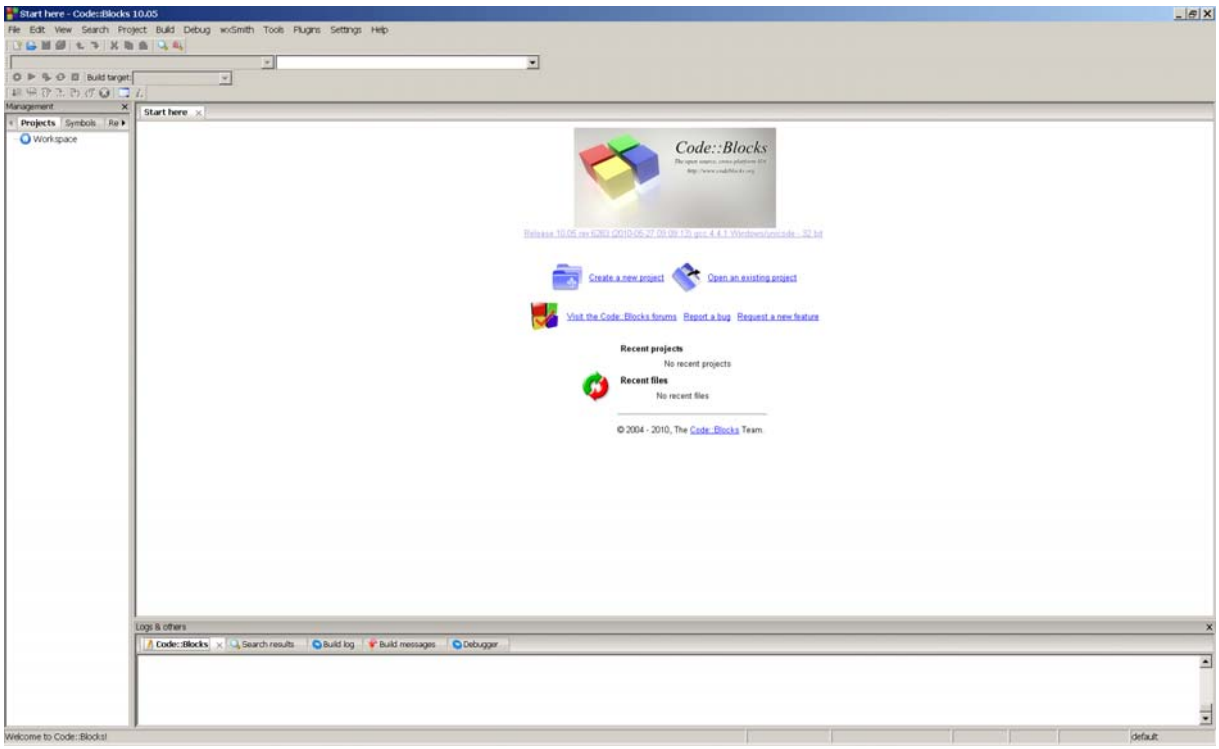


La version stable de CodeBlocks est en anglais. Mais il existe une version traduite en français : http://grandzebu.net/informatique/cpp/codeblocks.htm. Cependant, cette version n'est pas à jour et l'installation est beaucoup plus compliquée.

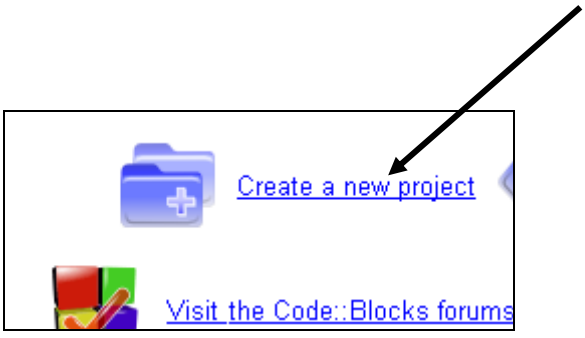
3) Premier programme

Vous pouvez maintenant relancer CodeBlocks en allant dans le menu « démarrer », « Tous les Programmes », « Codeblocks » 

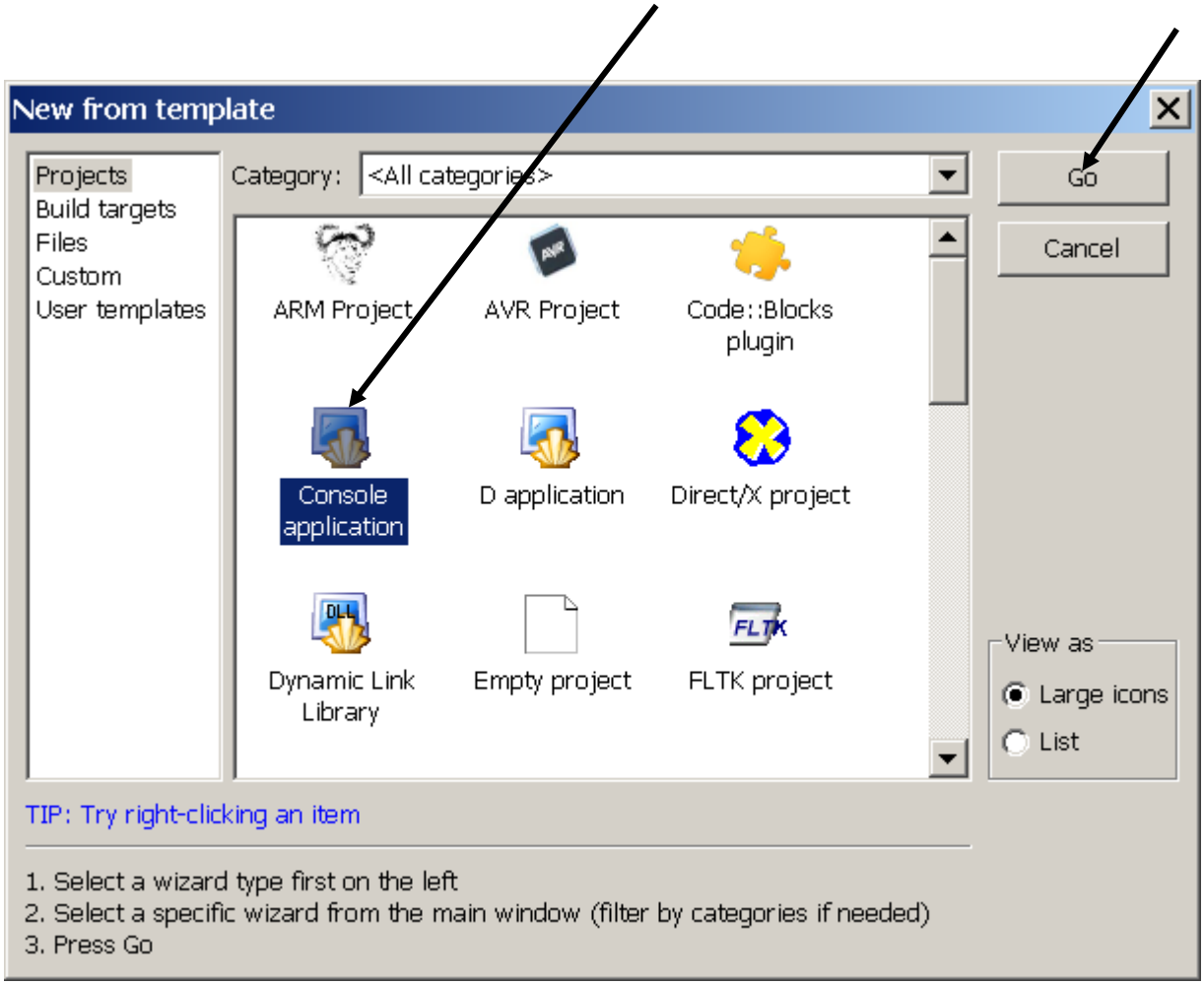
:



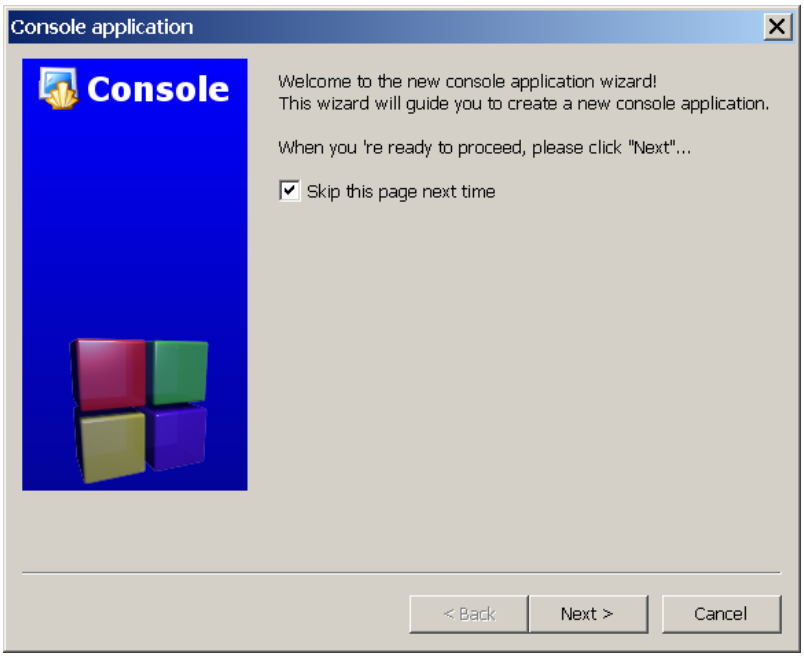
Dans la page principale, cliquez sur le lien « Create a new project » :



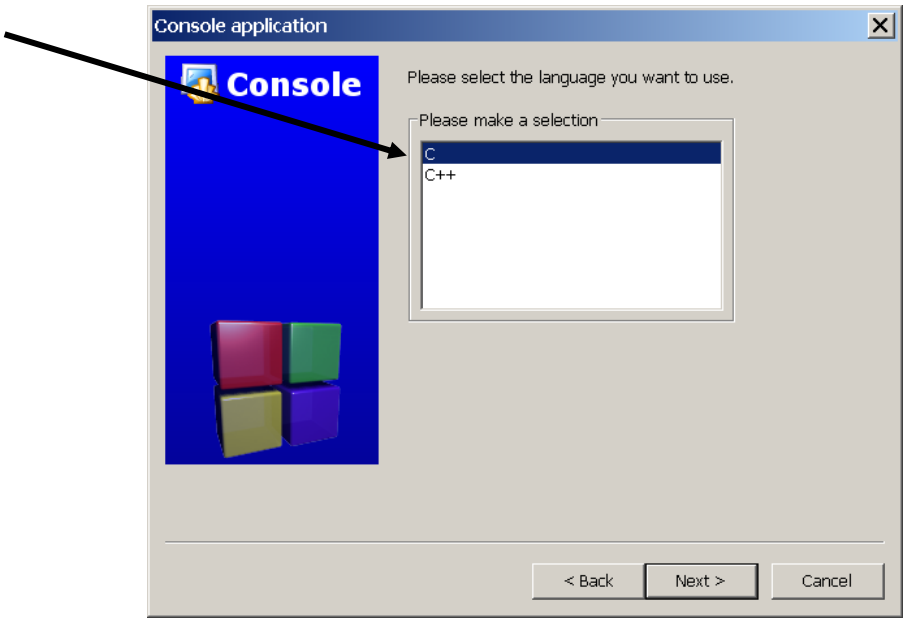
La fenêtre de création de projet apparaît à l'écran. Cliquez sur l'icône « Console Application » pour créer une application non-graphique. Vous **devez** obtenir la fenêtre suivante avant de cliquer sur le bouton « Go » :



Dans la fenêtre qui s'ouvre alors, cochez la case « Skip this page next time » puis cliquez sur le bouton « Next > » :

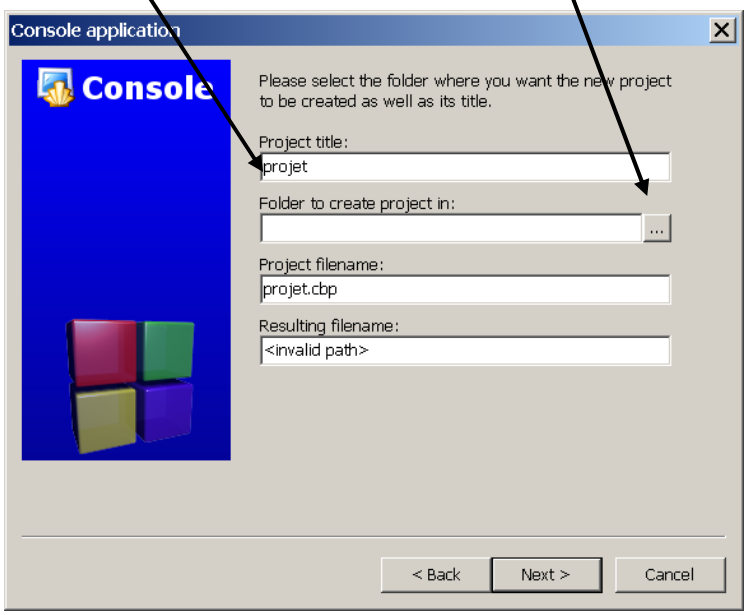


Sélectionnez le langage C dans la fenêtre suivante, puis cliquez sur le bouton « Next » :

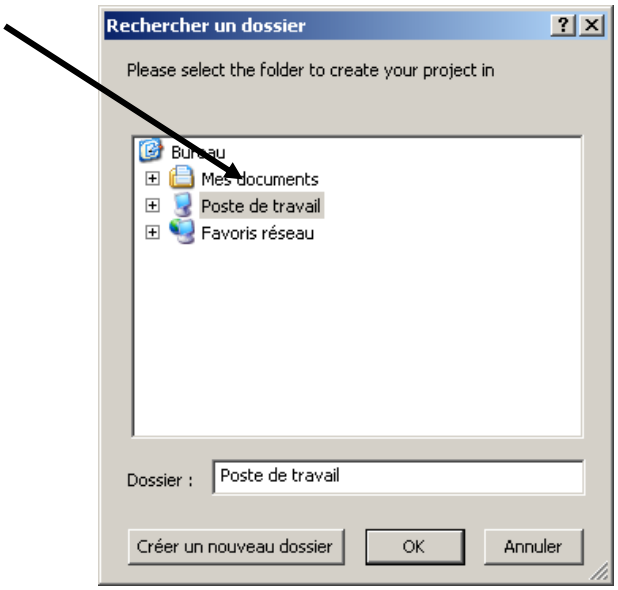


ATTENTION : il est très important de sélectionner langage C (fichier avec extension .c) et pas C++ (fichier avec extension .cpp). Si vous vous trompez, le compilateur ne va pas compiler correctement vos programmes.

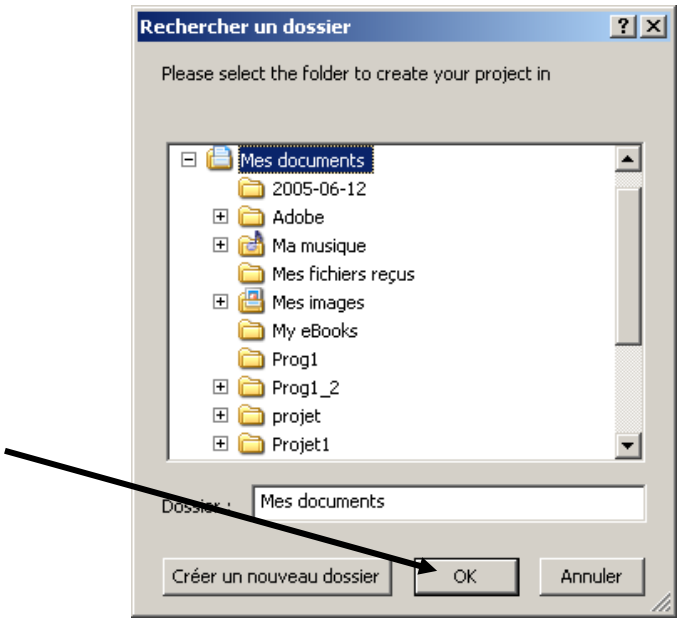
Le programme vous demande maintenant le nom de votre projet. Vous allez tapez `projet` dans le champ « Project title » puis cliquer sur le bouton ... pour montrer le répertoire dans lequel vous voulez enregistrer votre projet :



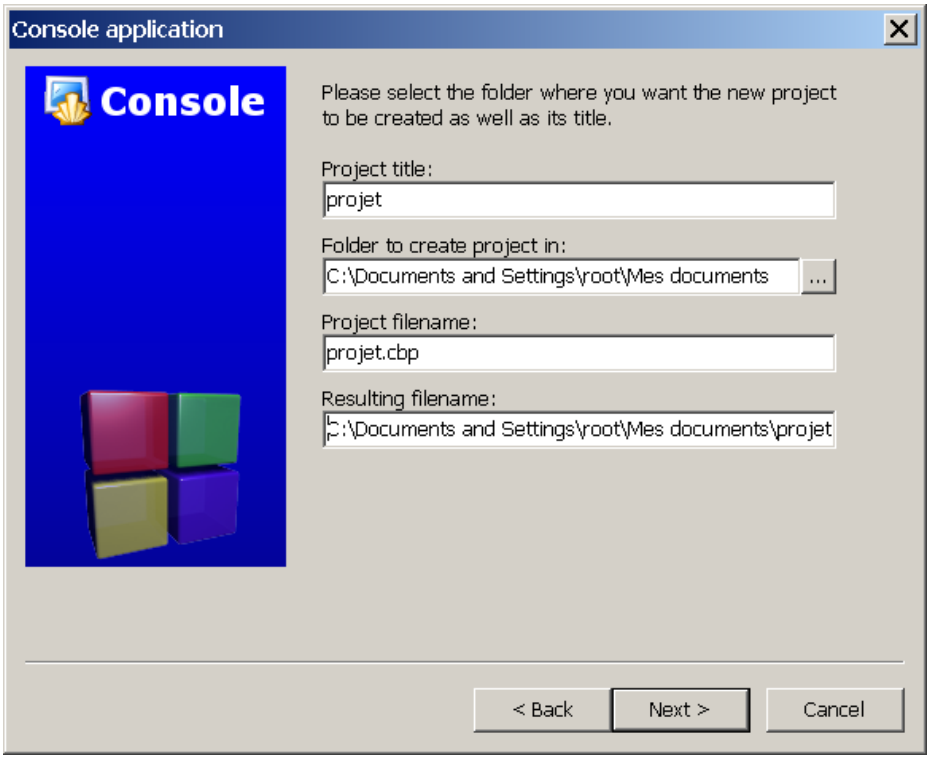
La fenêtre suivante s'ouvre. Dans notre exemple, nous allons enregistrer `projet` dans `Mes documents`. Double-cliquez sur le dossier « Mes documents » pour l'ouvrir :



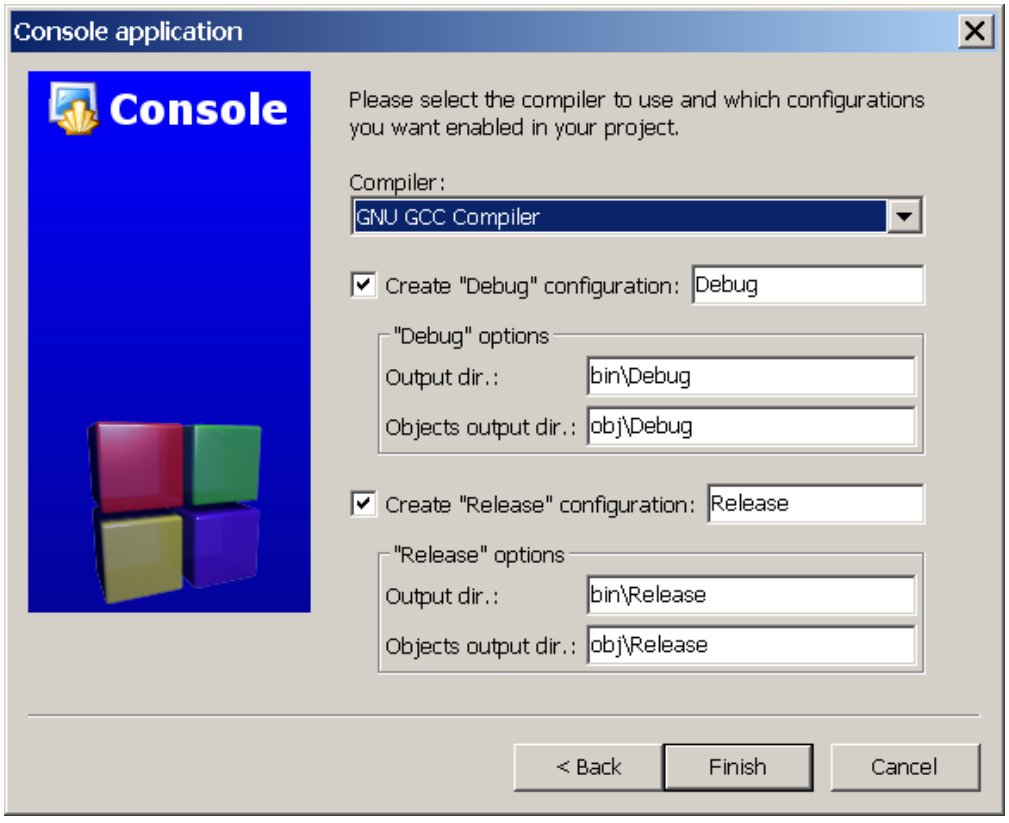
Puis cliquez sur le bouton « OK » :



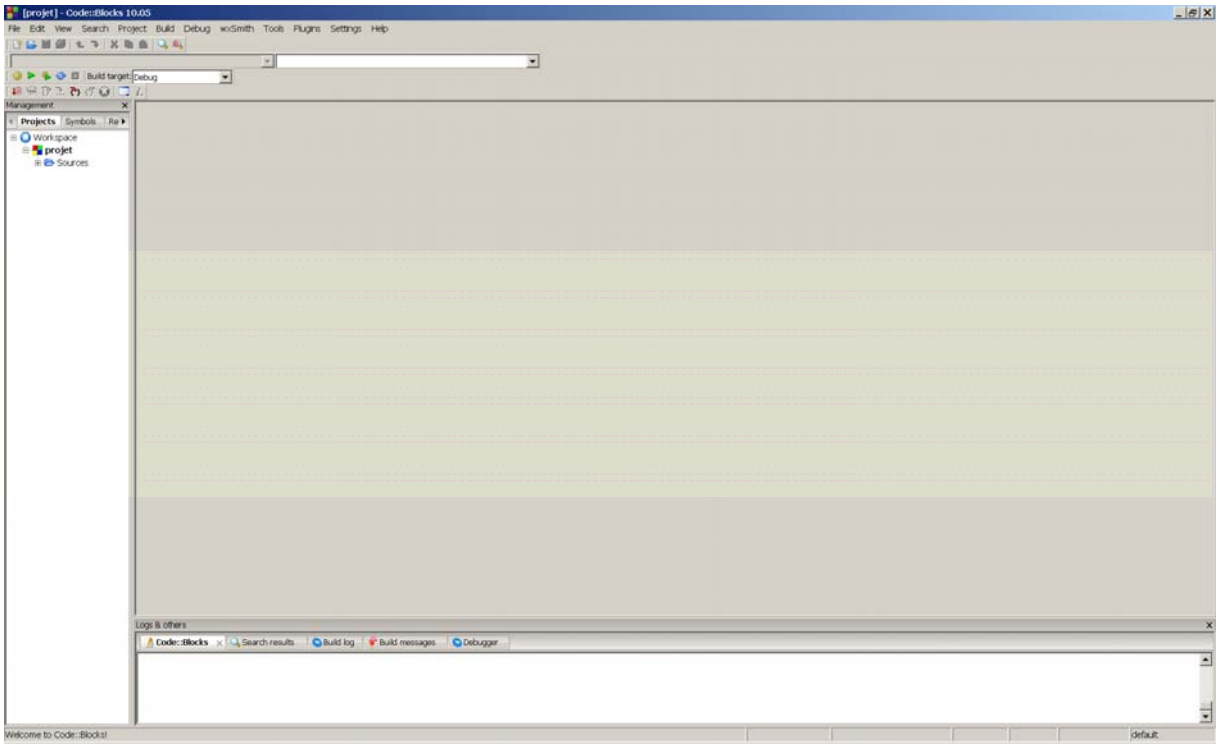
CodeBlocks va créer automatiquement un répertoire `projet` dans `Mes documents` et créer le fichier de gestion du projet `projet.cbp` à l'intérieur. Cliquez sur le bouton « Next > » après avoir obtenu la fenêtre suivante :



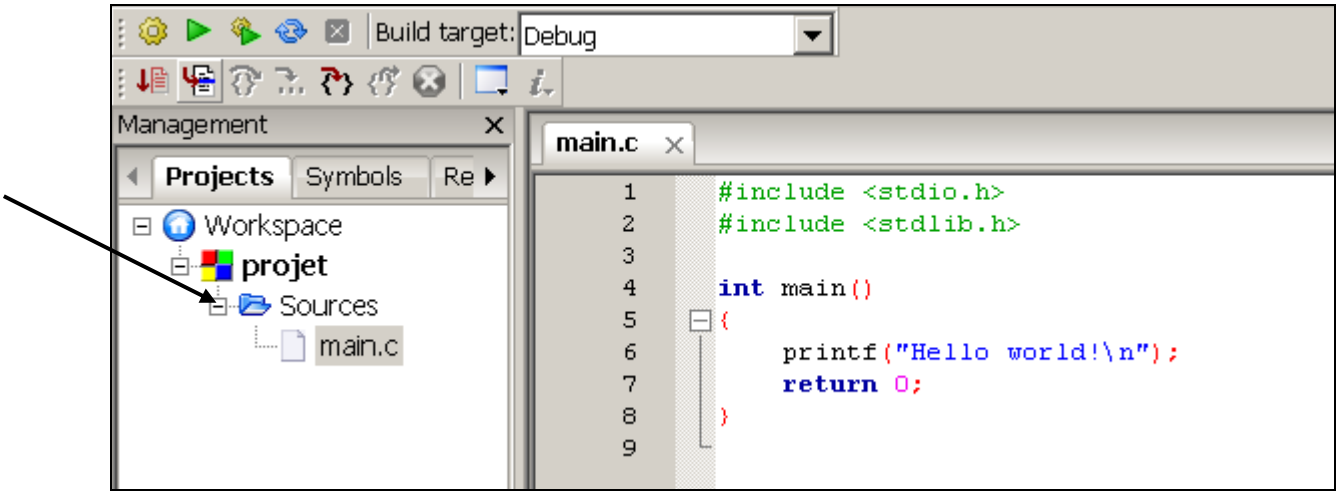
Cliquez sur le bouton « Finish » dans la fenêtre suivante :



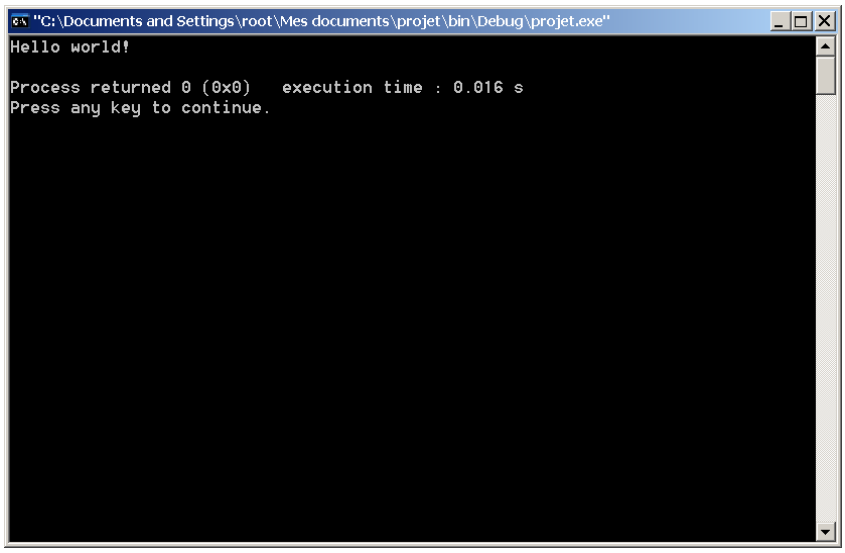
La fenêtre de CodeBlocks a changé.



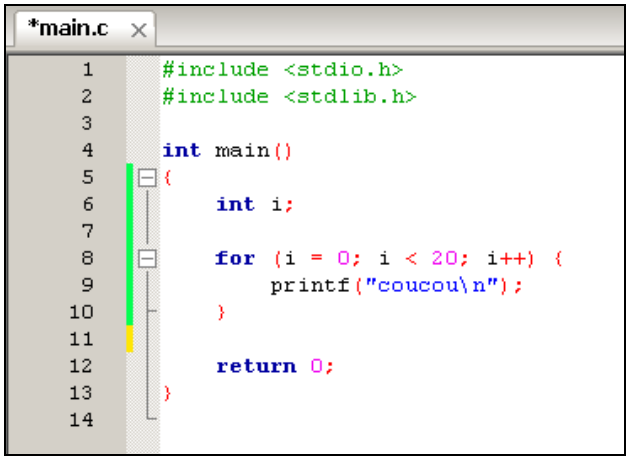
Le projet a été créé ainsi qu'un fichier source par défaut `main.c` que nous allons éditer. Pour cela, faites apparaître le fichier dans la fenêtre de gestion du projet en cliquant sur le signe + à gauche de « Sources », puis double-cliquez sur `main.c`.



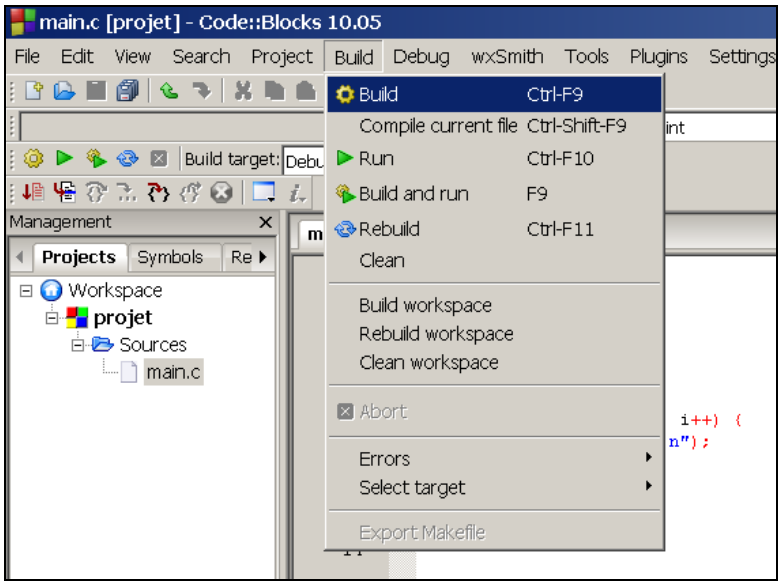
Vous pouvez exécuter cette amorce de programme en cliquant sur le bouton  qui va réaliser une compilation suivie d'une exécution. La console suivante apparaît à l'écran :



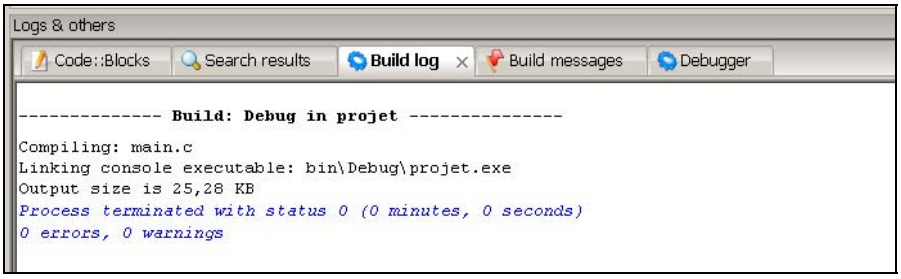
Elle vous indique le temps d'exécution du programme et la valeur retournée par la fonction `main` (ici, 0). Tapez sur une touche du clavier pour fermer la console. Dans la fenêtre d'édition appelée « `main.c` », saisissez le texte suivant :



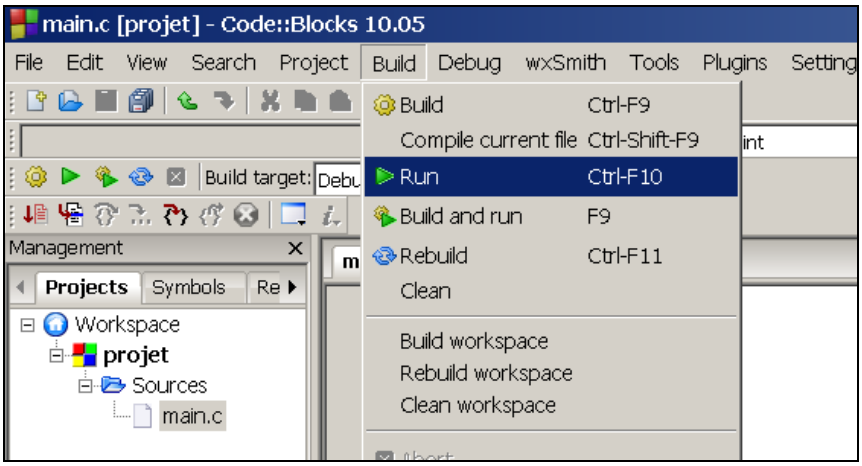
Cliquez sur l'icône  (Save) pour sauver votre programme. Pour compiler ce programme, cliquez sur le menu « Build » puis sur le sous-menu « Build » ou encore sur l'icône  (en réalité, la sauvegarde est faite automatiquement au moment du build, mais il ne faut pas oublier de sauver régulièrement votre travail).



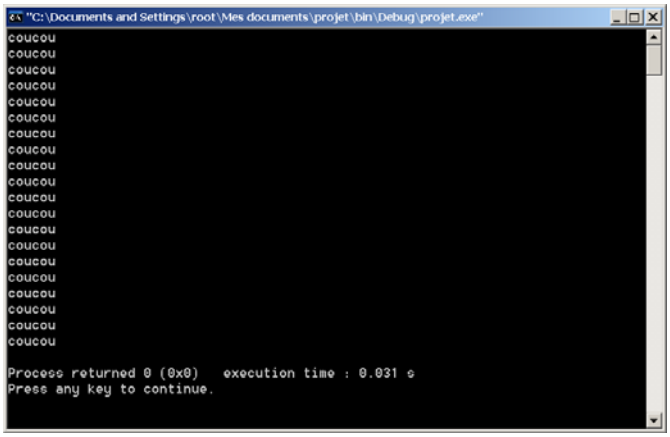
La fenêtre inférieure de CodeBlocks (la fenêtre Build Log) vous informe sur déroulement de la compilation. A l'issue de la compilation, vous devez obtenir la fenêtre suivante (avec 0 erreurs et 0 warnings). Nous verrons plus loin comment corriger les erreurs de compilation.



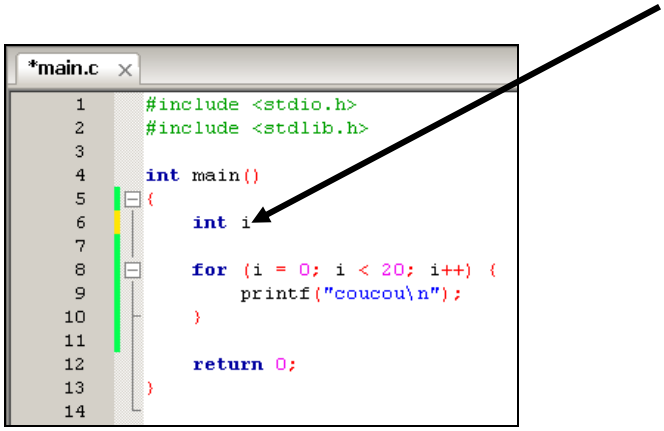
Pour lancer le programme, cliquez sur le menu « Build » puis sur le sous-menu « Run » ou cliquez sur l'icône  :



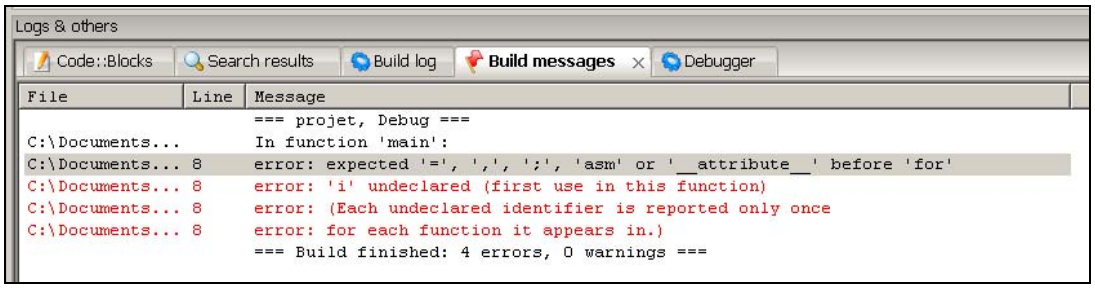
La fenêtre de commande (ou console) suivante apparaît à l'écran :



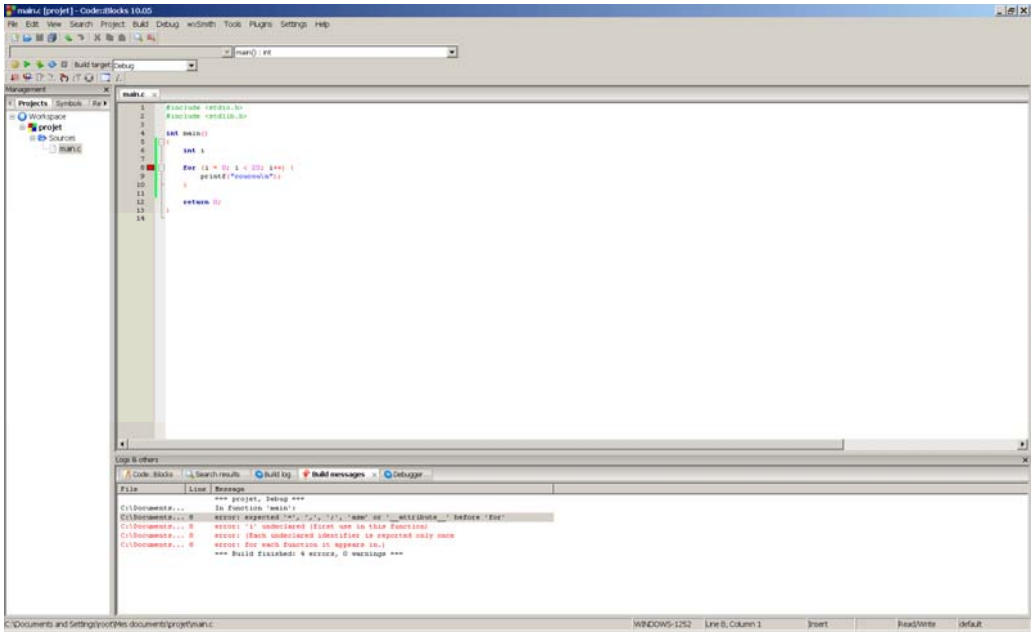
Appuyez sur une touche du clavier pour fermer cette fenêtre. Voyons maintenant ce qui se passe en cas d'erreur de compilation. Pour cela, nous allons créer une erreur de syntaxe en supprimant le point virgule à la fin de la ligne de déclaration de la variable i :



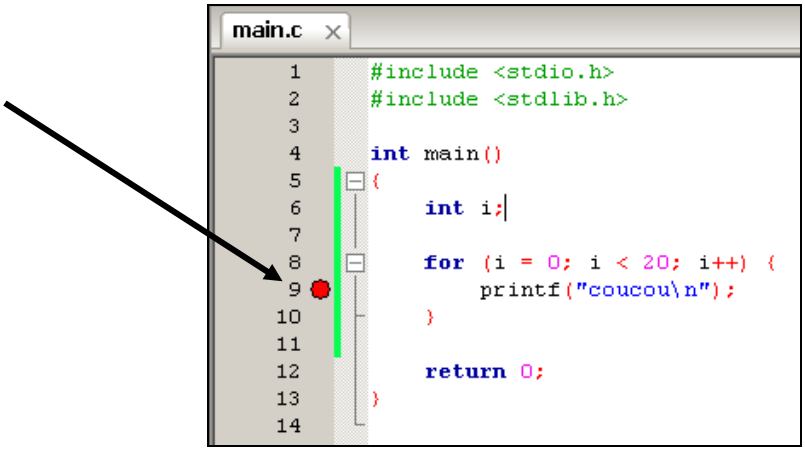
ATTENTION : si vous n'avez pas fermé la console, la compilation est impossible. Les boutons correspondants sont gris  excepté le bouton « Abort » qui met fin au programme. Si tel est le cas, pensez à vérifier grâce à la barre de taches que la console n'est pas en cours d'exécution. Relancez la compilation. Le compilateur voit maintenant plusieurs erreurs. La fenêtre inférieure de CodeBlocks vous indique la liste des erreurs.



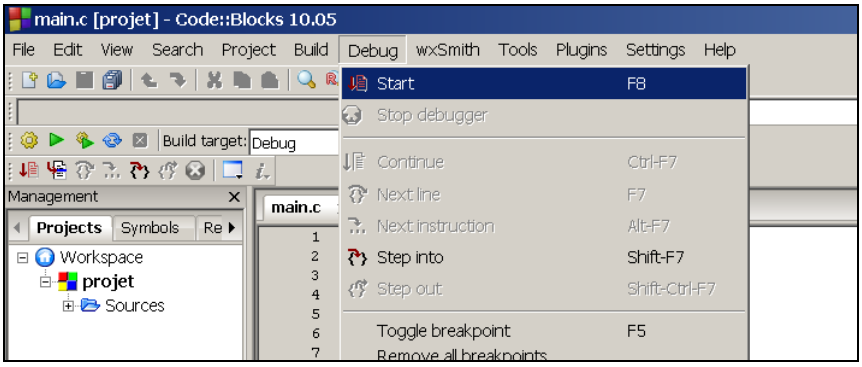
La ligne correspondant à la première erreur est marquée avec un carré rouge dans l'éditeur. D'une manière générale, pour identifier la ligne la plus proche de l'erreur, il suffit de double-cliquer sur le message dans la fenêtre inférieure et la ligne apparaît marquée avec un carré rouge dans l'éditeur.



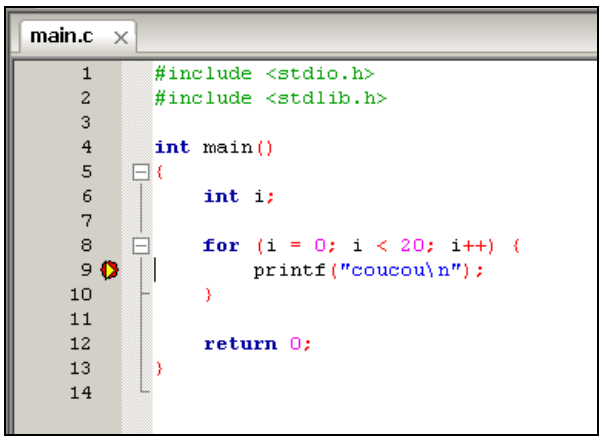
Dans notre exemple, le message vous indique qu'il y a une erreur de syntaxe avant le for. Rajoutez le point-virgule manquant sur la ligne précédente, enregistrez le fichier puis recompilez le programme. **Après avoir corrigé vos erreurs de compilation**, il se peut que votre programme ne fonctionne pas comme vous le souhaitez. Pour corriger ces erreurs de fonctionnement, vous pouvez utiliser le debugger. Pour commencer, il faut placer un point d'arrêt dans le programme. Pour cela, cliquez dans la fenêtre d'édition à droite du numéro de ligne 9. Un point d'arrêt (breakpoint en anglais) apparaît sur la ligne 9 :



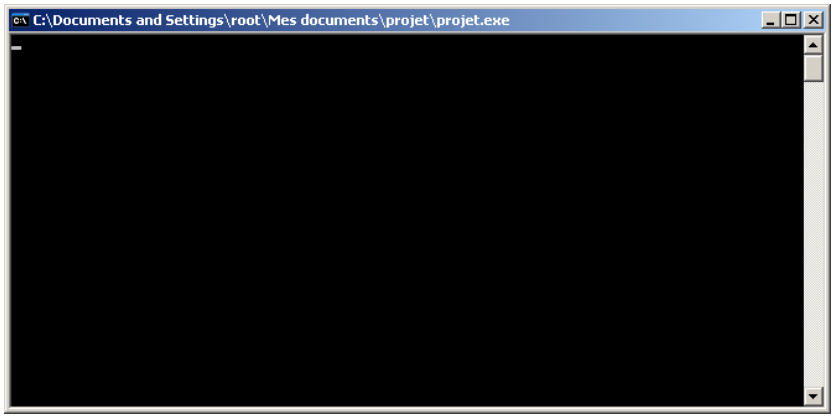
Cliquez ensuite sur le menu « Debug » puis sur le sous-menu « Start » (raccourci clavier, touche F8 ou bouton  « Debug / Continue ») pour lancer le debugger :



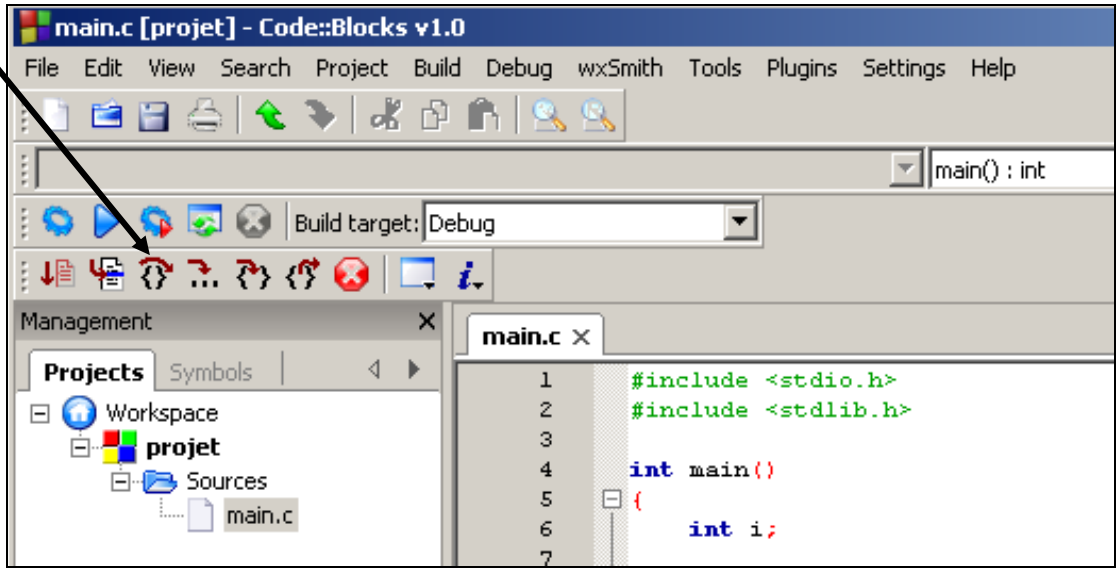
Le programme est arrêté sur la ligne 9. La ligne qui va être exécutée est maintenant marquée par un triangle jaune :



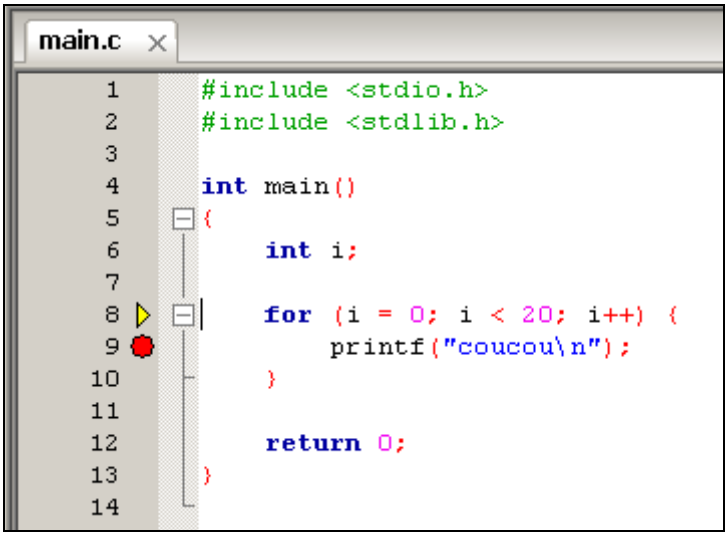
et la fenêtre de commande nommée `projet` où va s'afficher la sortie du programme est ouverte à l'écran :



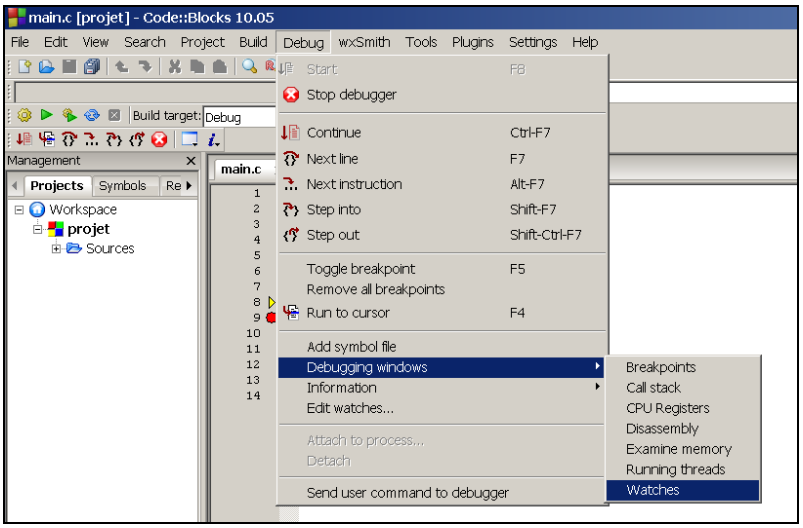
Vous pouvez maintenant avancer dans le programme ligne par ligne. On appelle cela le mode pas à pas. Il suffit pour cela de cliquer sur l'icône « Next line »  dans la barre d'outils :



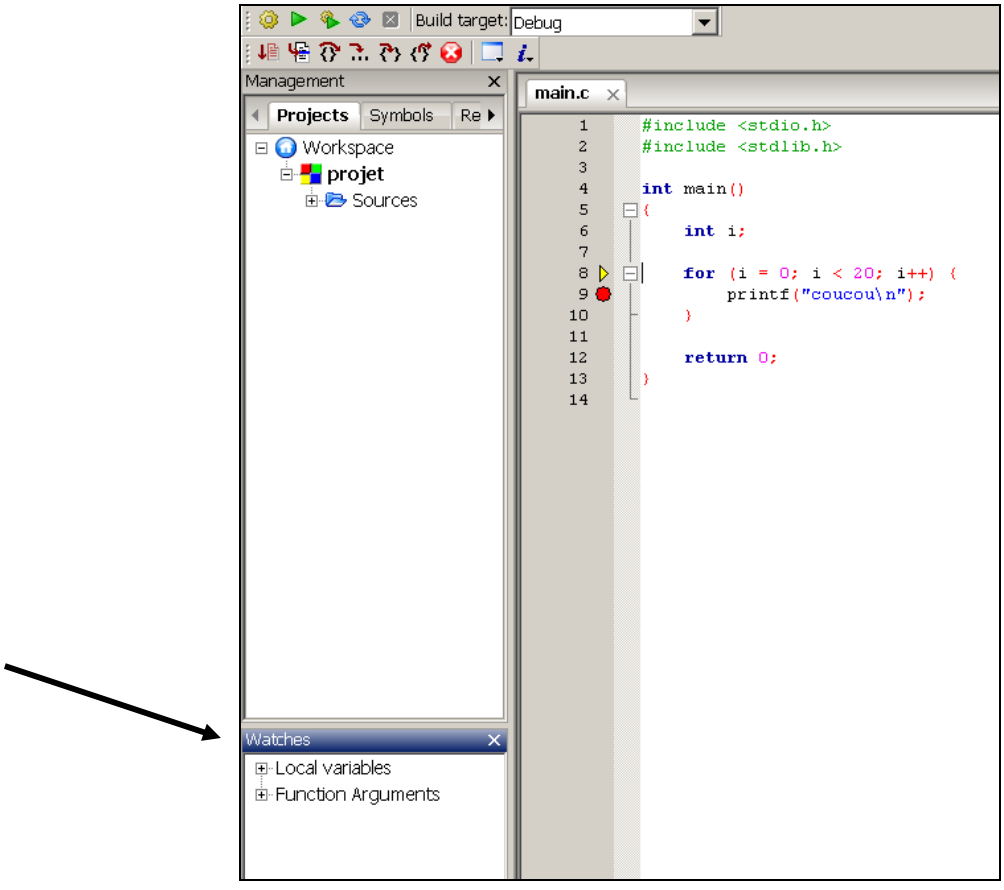
A chaque fois que vous cliquez sur « Next line », le programme exécute une ligne supplémentaire, ce qui fait avancer le triangle jaune.



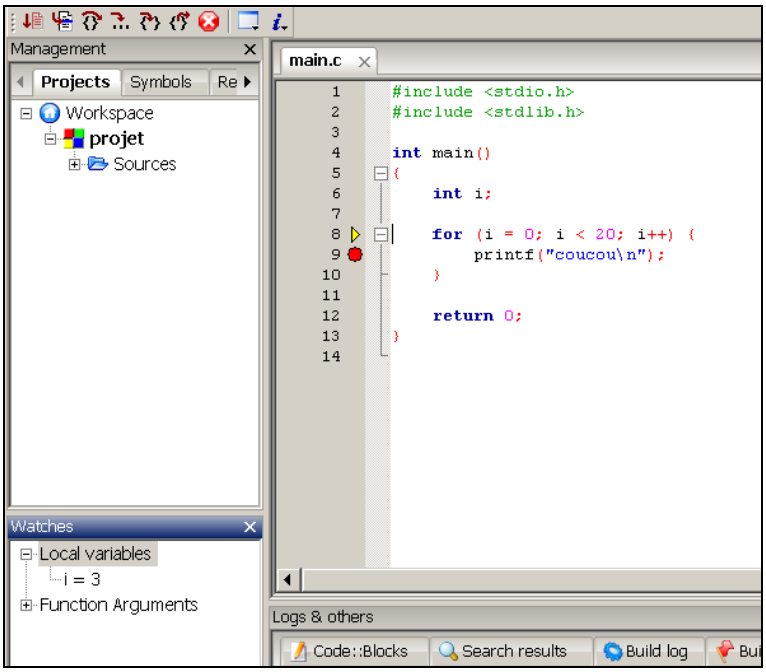
Vous pouvez inspecter (to watch en anglais) simplement le contenu d'une variable en faisant apparaître la fenêtre « Watches » dans CodeBlocks de la manière suivante :



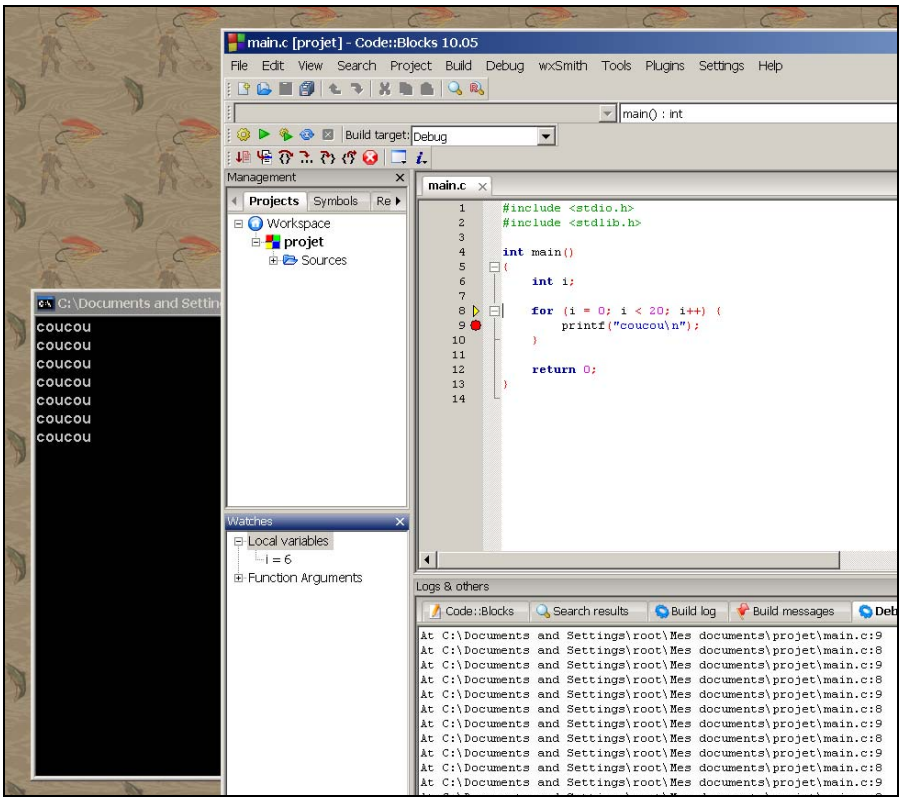
Quand la fenêtre apparaît, faites la glisser sous la fenêtre « Projects » jusqu'à obtenir la disposition suivante :



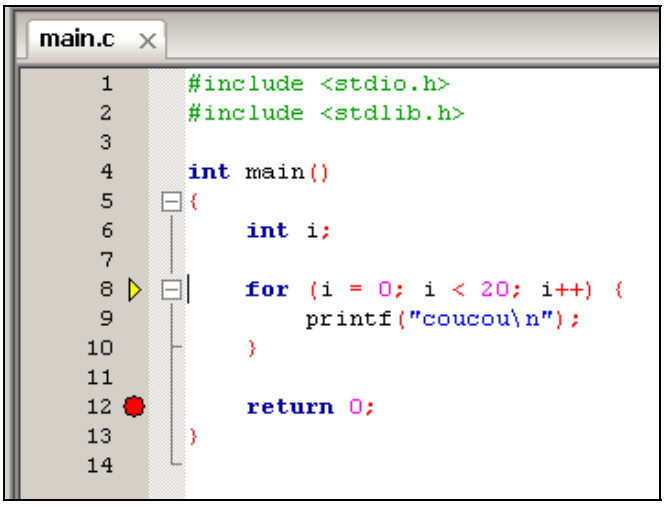
Vous pouvez voir toutes les variables de la fonction `main` dans cette fenêtre. Dans notre exemple, il s'agit de la variable `i` (cliquez sur le + si elle n'apparait pas).



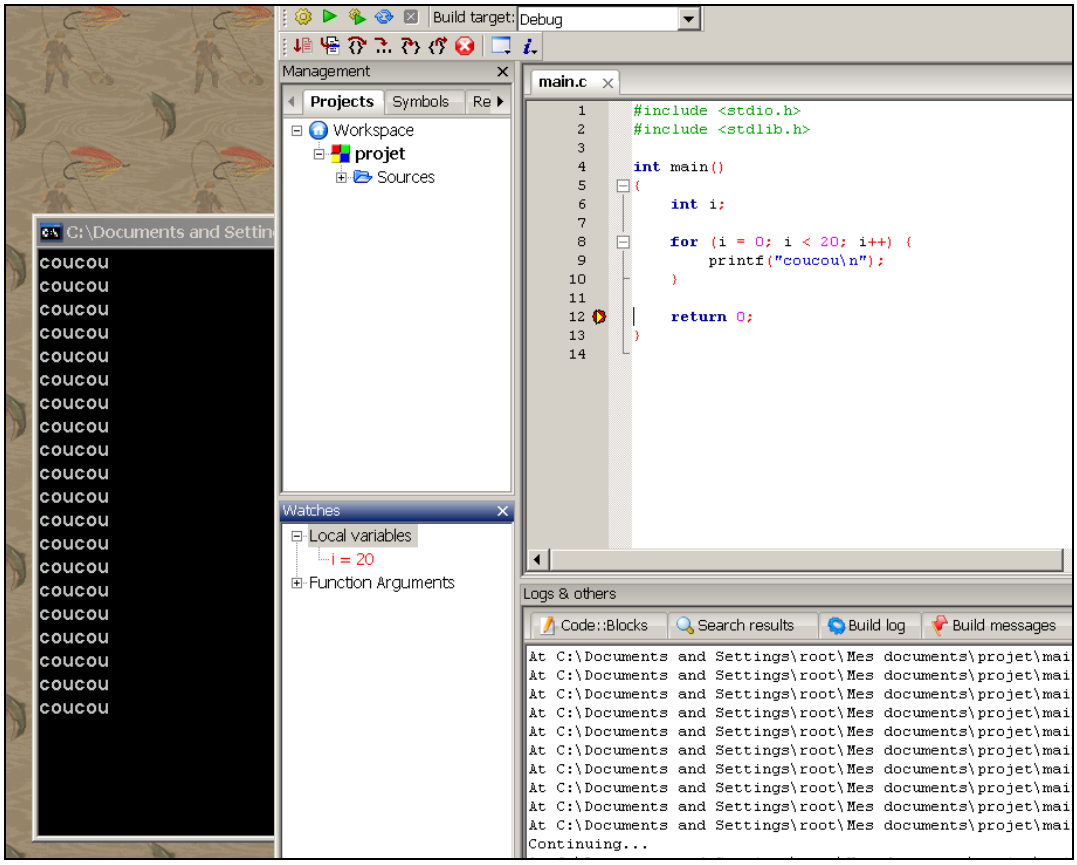
En cliquant plusieurs fois sur le bouton « Next line », vous voyez la valeur de `i` évoluer et les `printf("coucou\n");` apparaître dans la fenêtre de commande.



Vous pouvez supprimer le point d’arrêt en cours de debug en cliquant dessus et en placer un autre sur la ligne 12 :



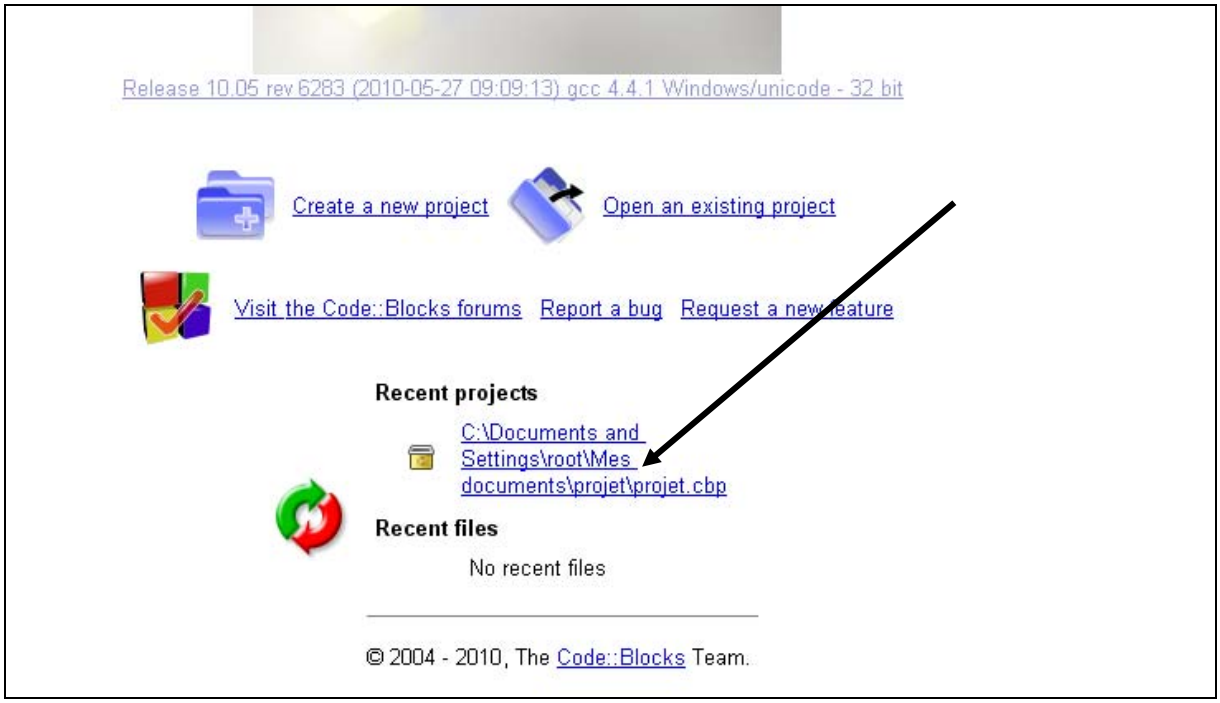
Vous pouvez ensuite lancer le programme jusqu'au nouveau point d'arrêt en cliquant sur le bouton « Continue »  . Le programme a fini la boucle `for`. `i` est égal à 20 et 20 lignes ont été affichées dans la fenêtre de commande.



Pour arrêter le debugger, cliquez sur le bouton « Stop debugger »  . La fenêtre de commande se ferme. Fermez CodeBlocks.

4) Deuxième programme

Ouvrez CodeBlocks. Pour rouvrir le projet précédent, cliquez sur le lien suivant :

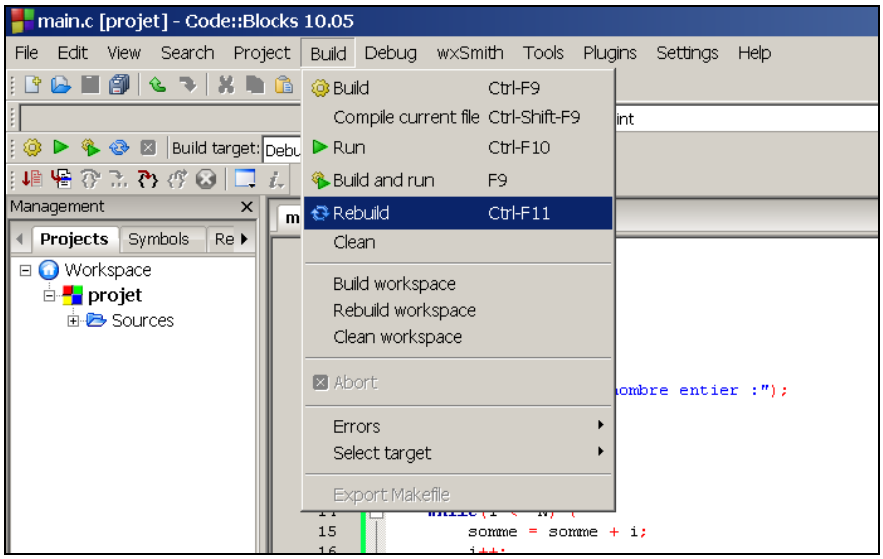


Modifiez `main.c` pour obtenir le programme suivant :

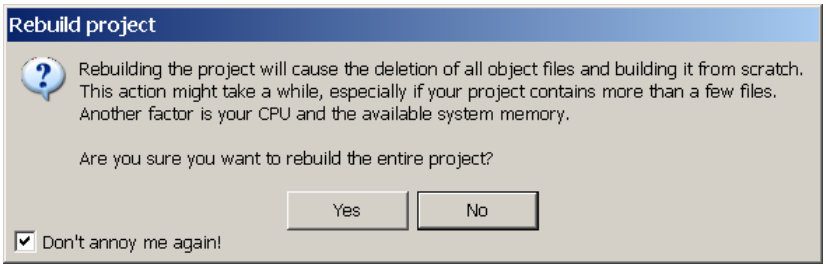
main.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int i, somme, N;
7
8      printf("Entrez un nombre entier :");
9      scanf("%d", &N);
10
11     i = 1;
12     somme = 0;
13
14     while(i <= N) {
15         somme = somme + i;
16         i++;
17     }
18
19     printf("La somme des %d premiers entiers vaut : %d\n", N, somme);
20
21     return 0;
22 }
23
```

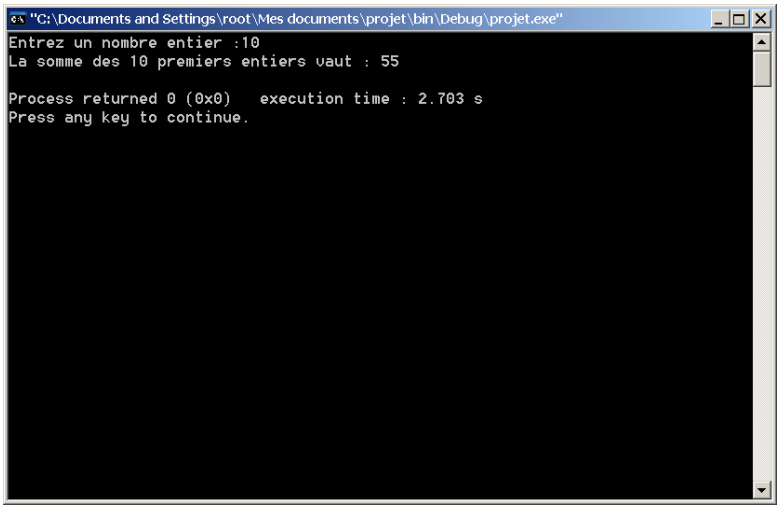
Recompilez tout le projet :



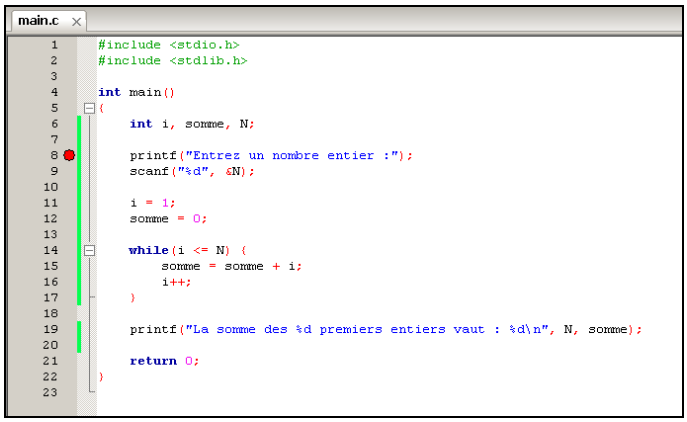
Dans la fenêtre qui s'ouvre, cochez la case « Don't annoy me again ! » puis cliquez sur le bouton « Yes » pour recompiler le projet en partant de zéro :



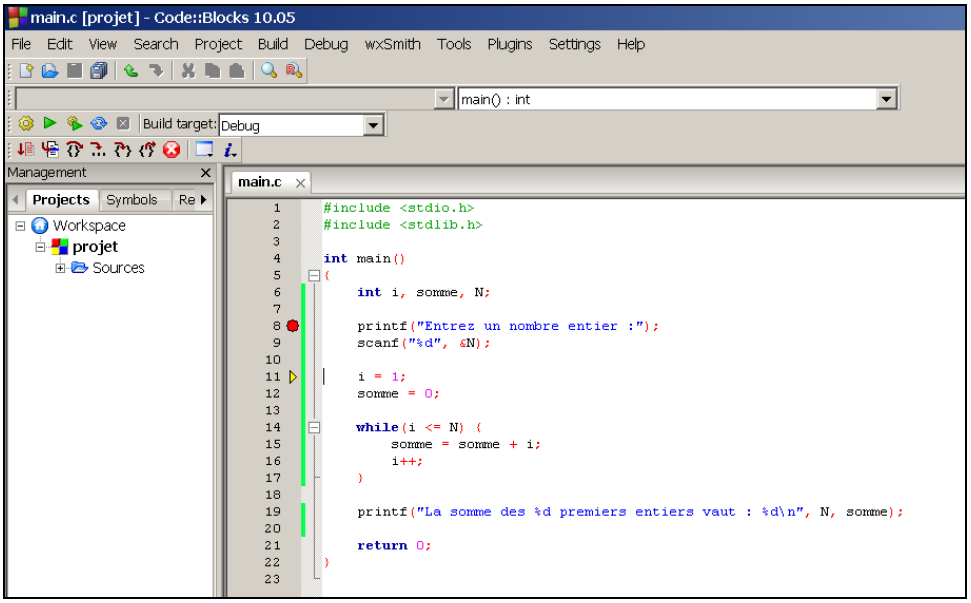
puis exécutez-le ( ) :



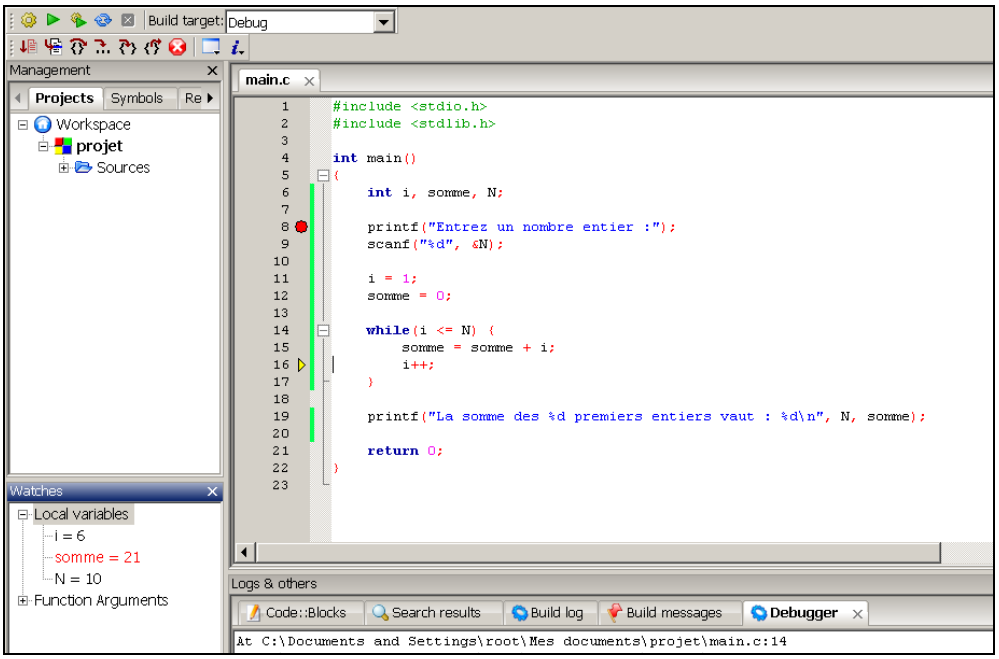
Tapez sur une touche du clavier pour fermer la fenêtre. Placez un point d'arrêt sur la ligne 8 :



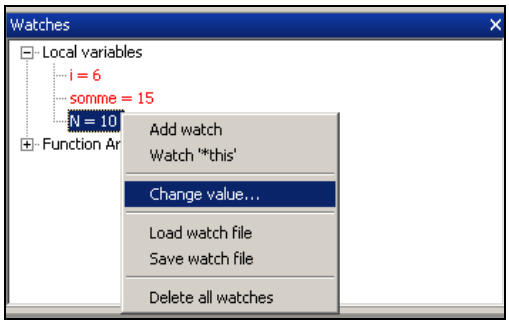
puis lancez le debugger (F8). Le debugger s'arrête sur le premier printf du programme. Avancez en mode pas à pas (Next line ). Quand vous essayez de dépasser la ligne du `scanf`, le debugger cesse d'avancer et les boutons de debug deviennent inactifs (ils deviennent gris)  . Le `scanf` attend que vous saisissiez une valeur dans la fenêtre de commande. Cette fenêtre doit être active pour que vous puissiez taper une valeur à l'intérieur. Pour rendre une fenêtre active, il faut cliquer dessus de façon à ce que la bande supérieure (le bandeau) de la fenêtre devienne bleue. Cliquez sur la fenêtre de commande, tapez une valeur puis appuyez sur la touche Entrée. Au moment où vous tapez sur Entrée, le debugger se débloque et avance d'une ligne. Les boutons de debug sont à nouveau actifs.



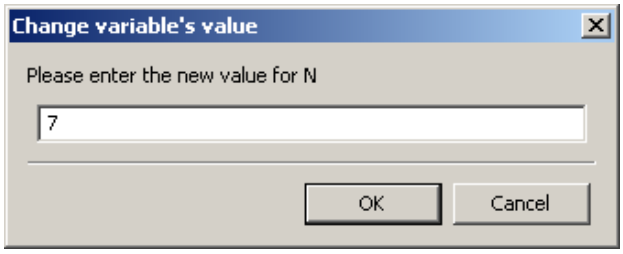
Vous pouvez maintenant examiner les variables `i`, `somme` et `N` en mode pas à pas.



Vous avez la possibilité de modifier la valeur d'une variable pendant l'exécution du programme. Il faut pour cela sélectionner la variable dans la fenêtre « Watches », puis cliquer avec le bouton droit de la souris. Dans le menu qui apparaît, cliquez sur « Change value… » :



Dans la fenêtre qui apparaît, tapez la nouvelle valeur de la variable puis cliquez sur « OK »



Vous pouvez continuer le mode pas à pas avec la nouvelle valeur de la variable. Fermez le debugger et CodeBlocks. Vous avez tous les éléments nécessaires pour commencer à programmer.

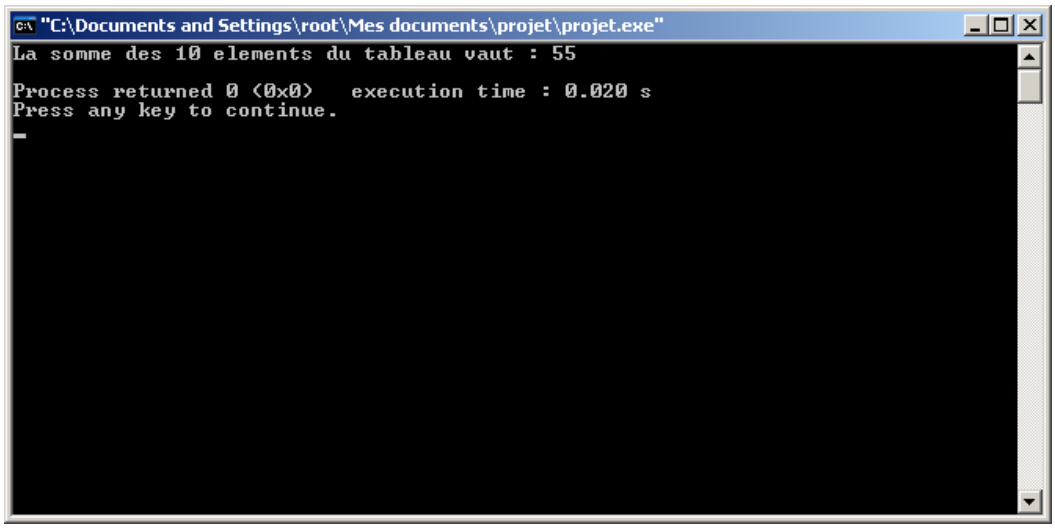
5) troisième programme : contenu des tableaux et inspection de la mémoire

Ouvrez CodeBlocks et ré-ouvrez le projet précédent. Modifiez `main.c` pour obtenir :

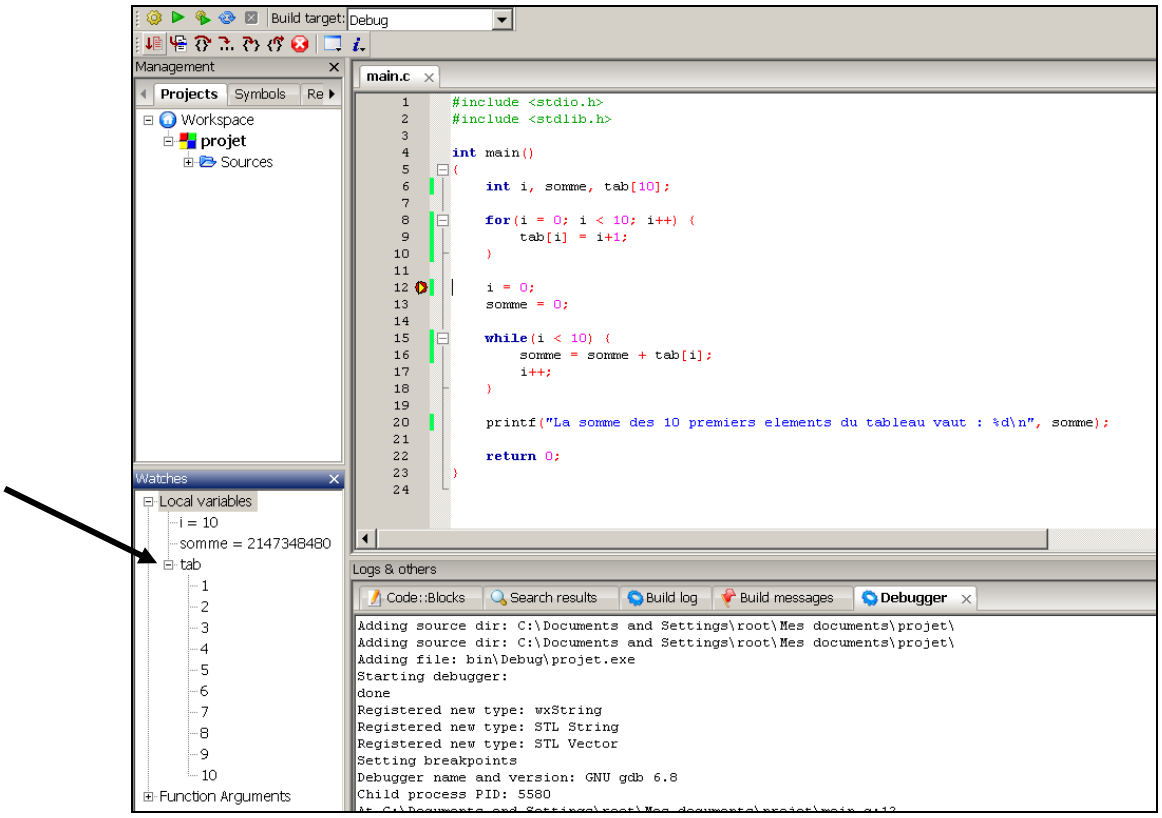
main.c

```
1   #include <stdio.h>
2   #include <stdlib.h>
3
4   int main()
5   {
6       int i, somme, tab[10];
7
8       for(i = 0; i < 10; i++) {
9           tab[i] = i+1;
10      }
11
12      i = 0;
13      somme = 0;
14
15      while(i < 10) {
16          somme = somme + tab[i];
17          i++;
18      }
19
20      printf("La somme des 10 elements du tableau vaut : %d\n", somme);
21
22      return 0;
23  }
24
```

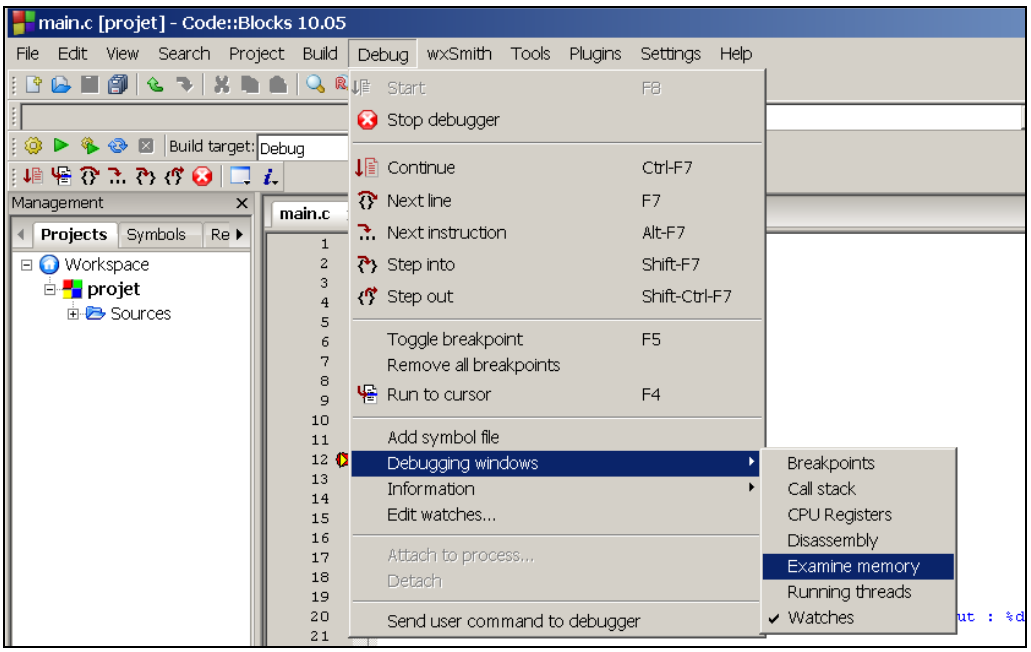
Re-compilez le projet et exécutez-le :



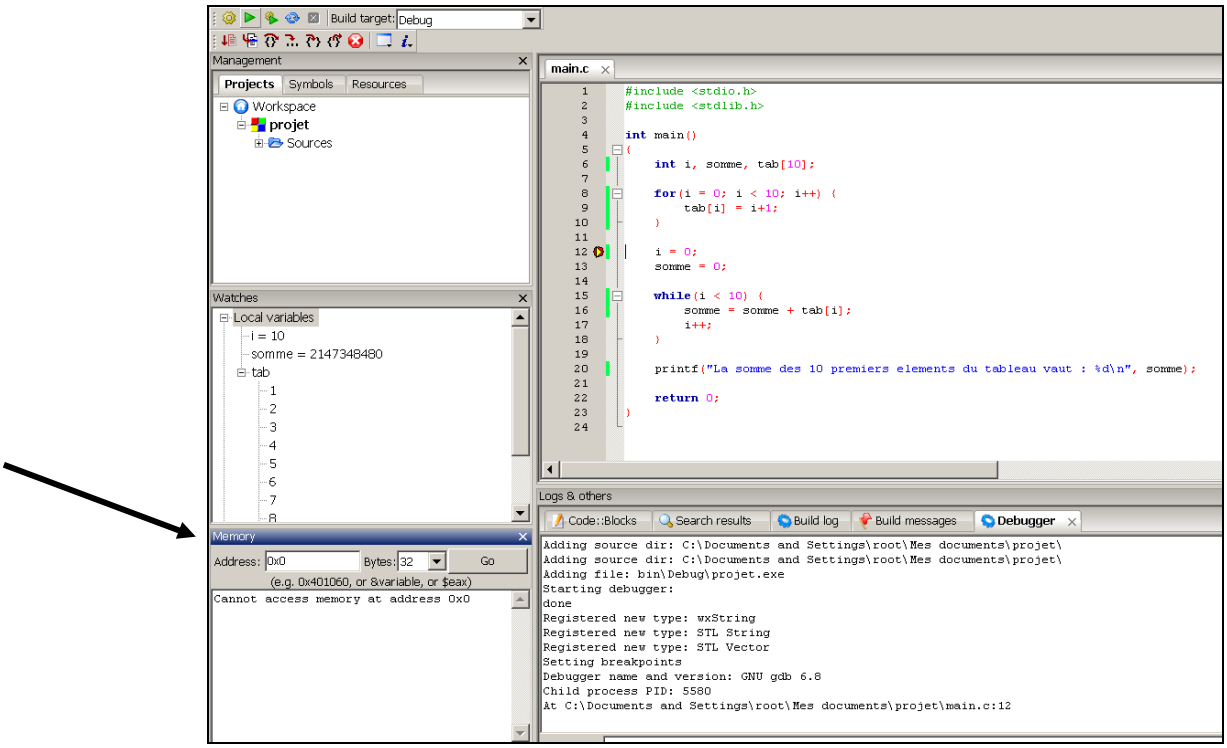
Tapez sur une touche du clavier pour fermer la fenêtre. Placez un point d'arrêt sur la ligne 12 puis lancez le debugger (F8). Vous trouverez dans la fenêtre « Watches » les variables `i`, `somme` et `tab`. Comme `tab` est un tableau, vous pouvez cliquez sur le signe + à gauche de son nom et visualiser le contenu des 10 cases :



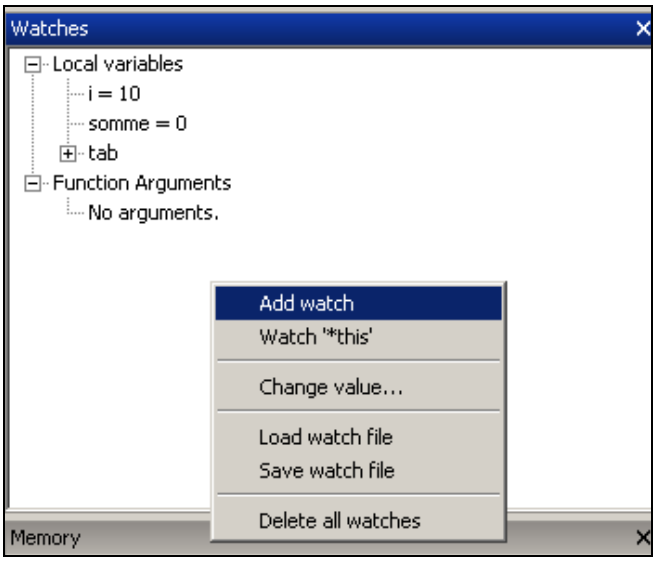
Arrêtez le debugger. Il existe une autre méthode pour visualiser le contenu d'une variable ou d'un tableau : la fenêtre d'inspection de la mémoire. Pour la faire apparaître, cliquez sur :



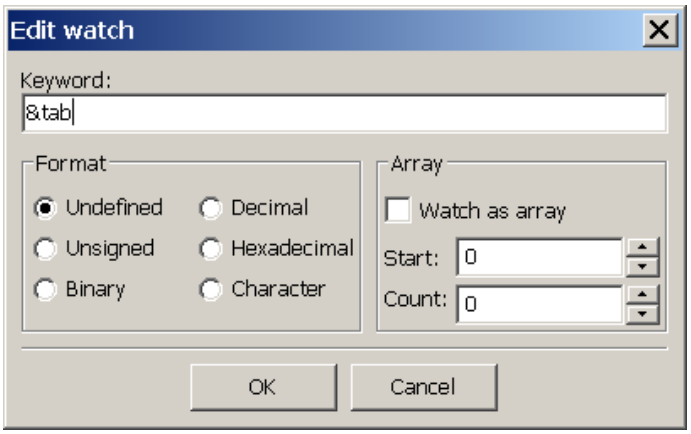
Faites glisser la fenêtre « Memory » pour la placer sous la fenêtre « Watches » afin d'obtenir la disposition suivante :



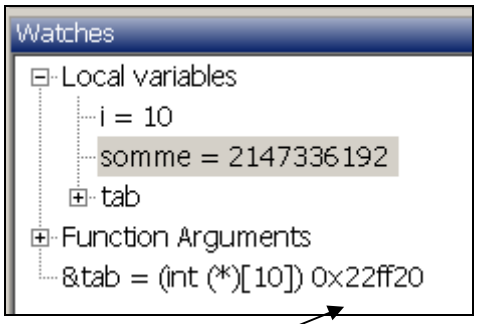
Lancez le debugger (F8). Nous avons besoin de connaître l'adresse du tableau. Pour cela, cliquez avec le bouton droit de la souris dans la partie inférieure de la fenêtre « Watches ». Un menu contextuel apparaît. Cliquez sur « Add watch » :



Dans le champ « Keyword » de la fenêtre qui apparaît, taper &tab puis cliquez sur le bouton « OK » :

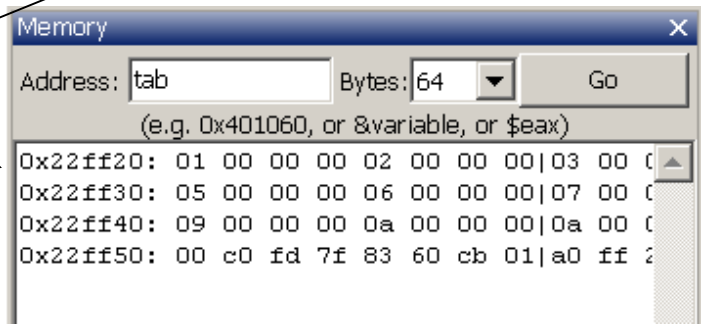


L'adresse du tableau apparaît maintenant dans la fenêtre « Watches » :



Dans le champ « Address » de la fenêtre « Memory », tapez le nom du tableau tab :

&tab[0]



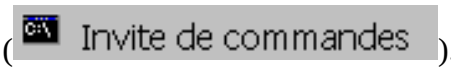
Vous pouvez vérifier que l'adresse du premier octet (0x0022ff30) correspond bien à l'adresse du tableau vu précédemment. Les 4 premiers octets valent 01 00 00 00, ce qui est égal à la valeur 1 en entier (avec l'octet de poids faible sur l'adresse de poids faible, ce qui correspond au codage little-indian d'Intel). Les 36 octets suivants correspondent aux 9 valeurs entières du tableau (2, 3, 4, 5, 6, 7, 8, 9 et 10) avec le même codage. Vous pouvez inspecter n'importe quelle adresse dans la mémoire virtuelle du programme grâce à la fenêtre « Memory ». Fermez CodeBlocks.

6) quatrième programme : passage de paramètres à la fonction main

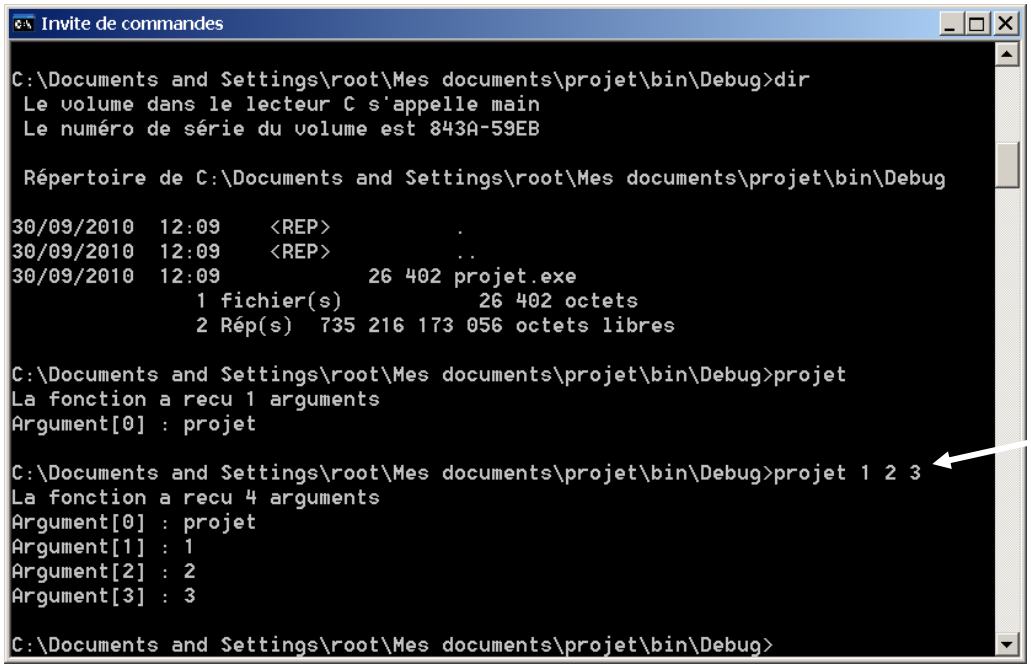
Pour passer des paramètres à la fonction main, il existe deux méthodes. Ouvrez CodeBlocks et ré-ouvrez le projet précédent. Modifiez `main.c` pour obtenir :

main.c

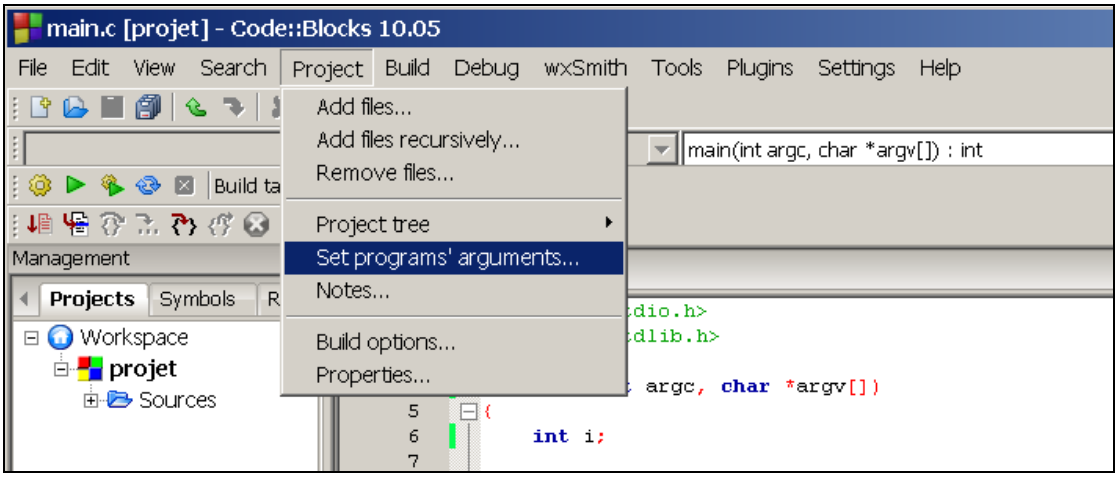
```
1   #include <stdio.h>
2   #include <stdlib.h>
3
4   int main(int argc, char *argv[])
5   {
6       int i;
7
8       printf("La fonction main a recu %d arguments\n", argc);
9
10
11      for(i = 0; i < argc; i++) {
12          printf("argument[%d] : %s \n", i, argv[i]);
13      }
14
15      return 0;
16  }
17
```

Ce programme accepte des paramètres sur la ligne de commande (voir §7 du cours). Mais comment les passer à la fonction main avec CodeBlocks ? La première méthode utilise la console de Windows (

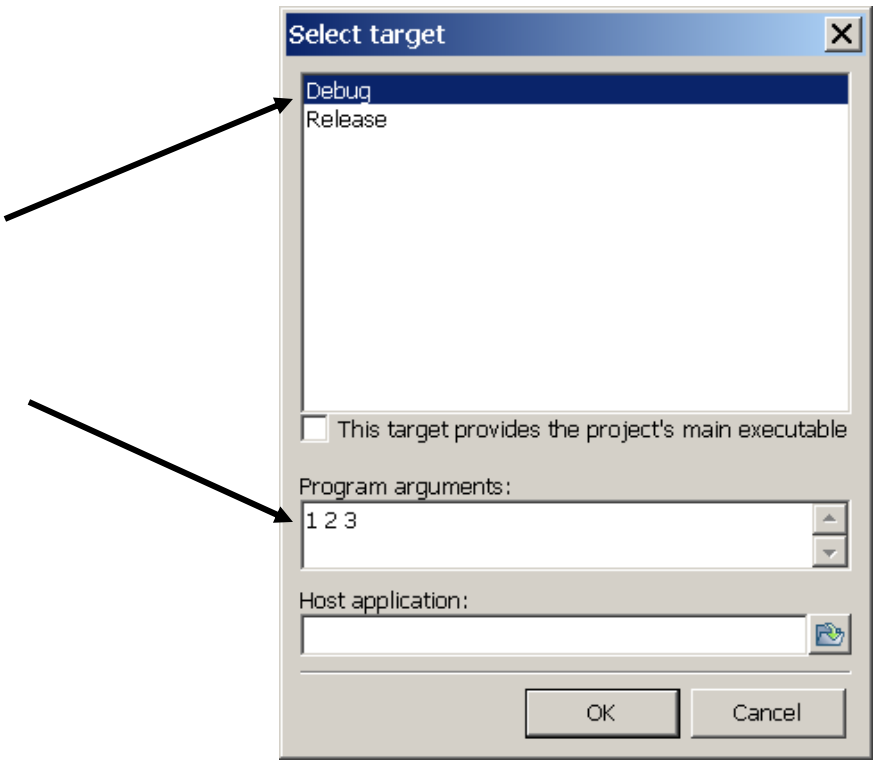
). Il suffit de l'ouvrir, de se placer dans le répertoire du projet grâce à la commande `cd "Mes documents"\projet\bin\Debug`, puis de tapez le nom du programme suivi des paramètres (exemple : `projet 1 2 3`):



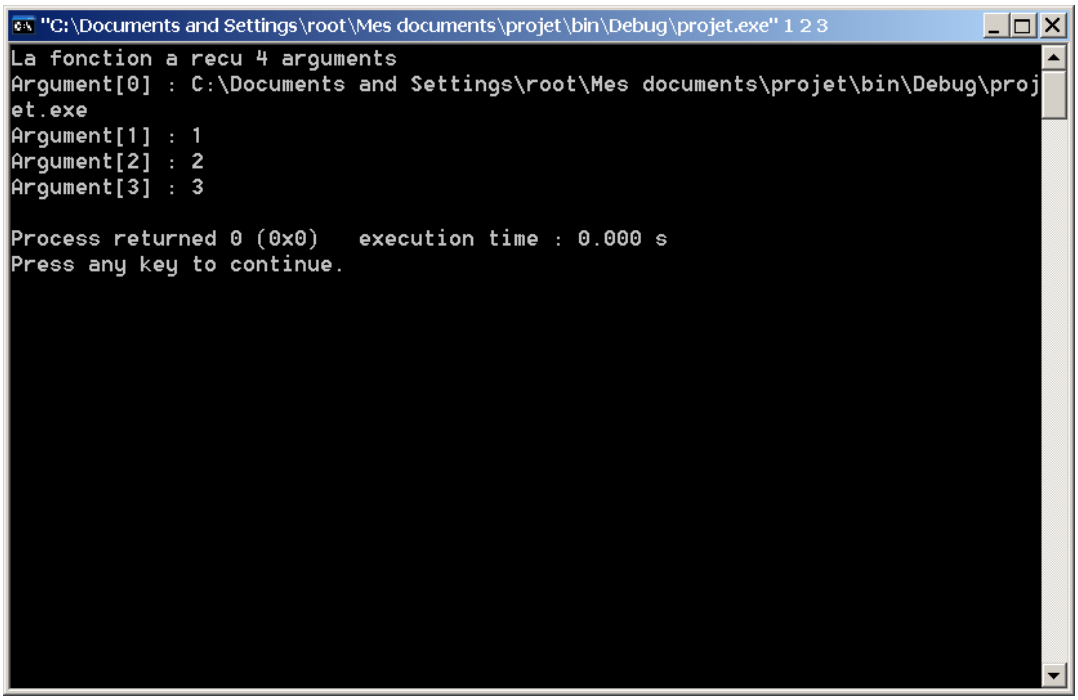
C’est la méthode traditionnelle que l’on utilise sous Linux. Si vous souhaitez passer les paramètres sans utiliser la console en restant sous CodeBlocks, il faut aller dans le menu « Project », cliquer sur « Set programs’ arguments… » :



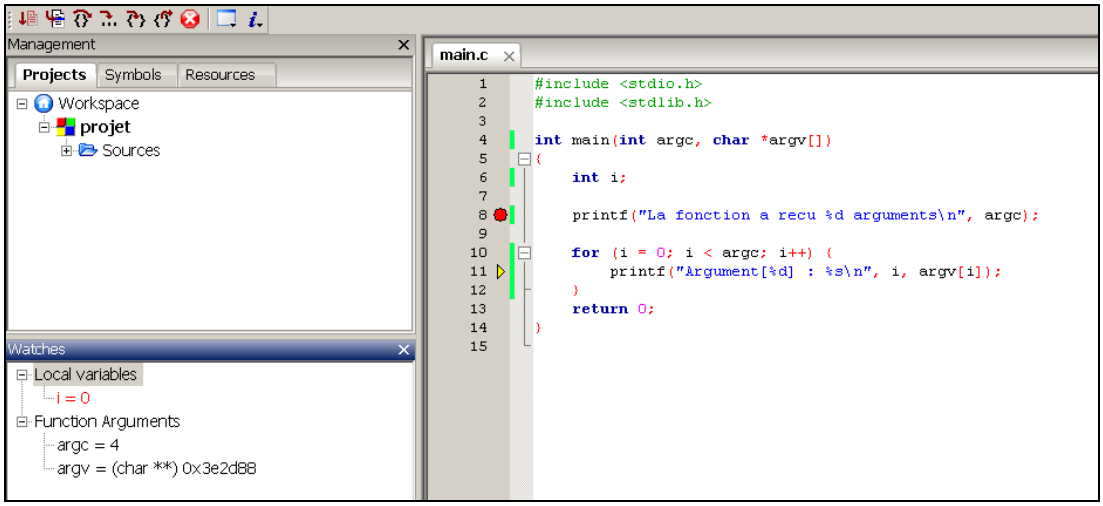
Sélectionnez le mode Debug, tapez les paramètres de la fonction main dans le champ « Program arguments » puis cliquez sur « Ok » :



Et enfin lancez le programme ( ):



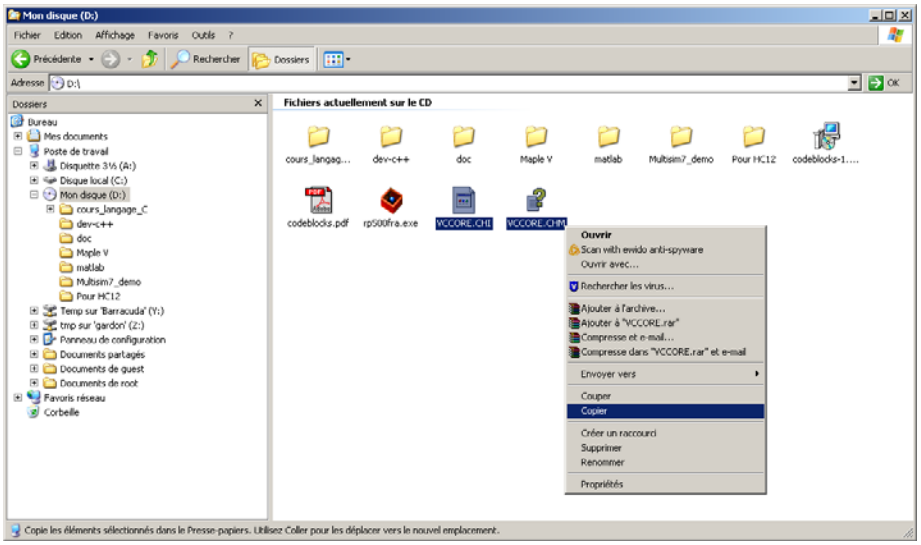
Cette méthode fonctionne aussi avec le debugger :



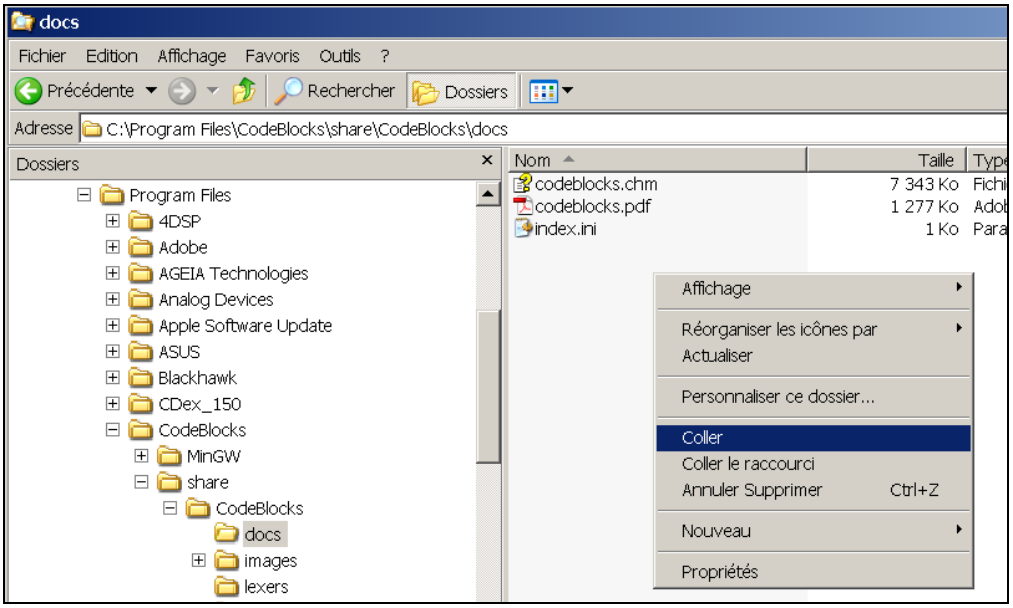
Fermez CodeBlocks.

7) installation de la documentation

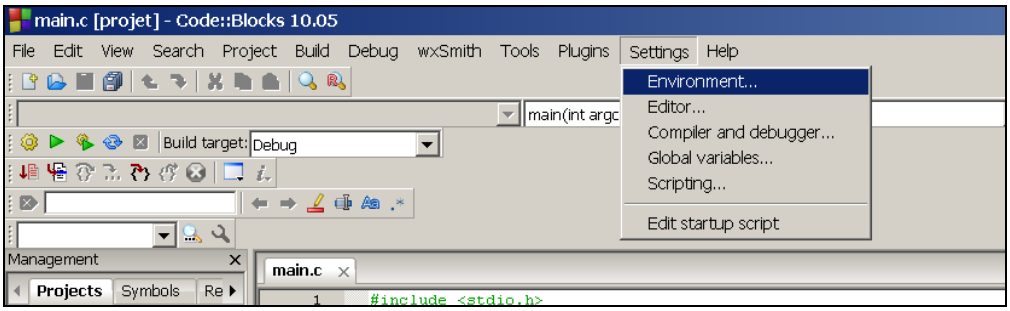
Codeblocks ne propose pas d'aide sur le langage mais prévoit l'utilisation de fichiers d'aide externes. Vous pouvez ainsi installer les fichiers d'aide de Visual C++ qui sont présents sur le CDROM. Il est en effet très pratique d'avoir une bonne documentation facilement accessible sur le C ANSI : voici comment l'installer sur votre ordinateur. Il faut pour cela copier les deux fichiers VCCORE.CHI et VCCORE.CHM qui se trouve sur la racine du CD dans le répertoire d'installation de CodeBlocks. Dans l'explorateur Windows, cliquez sur le CD, sélectionnez ces deux fichiers, faites un clic droit puis cliquez sur « Copier » :



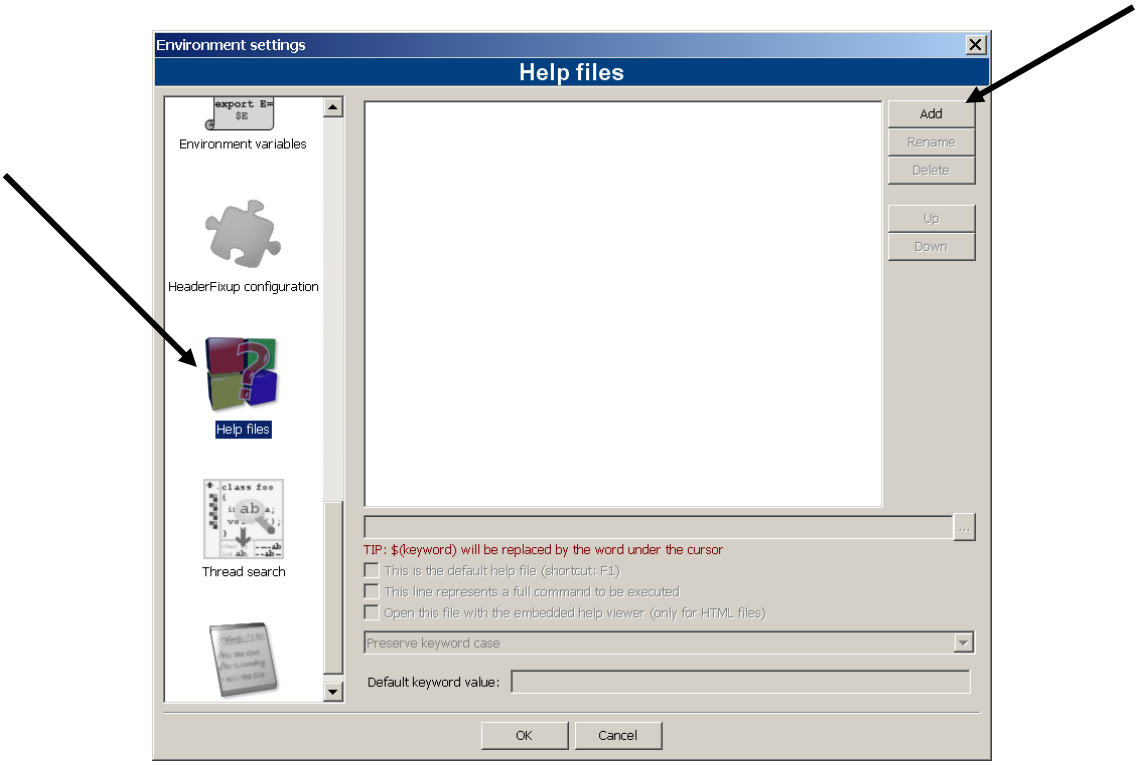
Dans le répertoire d'installation de la doc de CodeBlocks (`C:\Program Files\CodeBlocks\share\CodeBlocks\docs`), copiez ces deux fichiers :



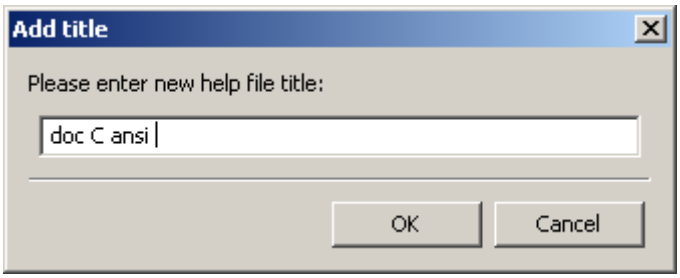
Démarrez CodeBlocks puis cliquez sur :



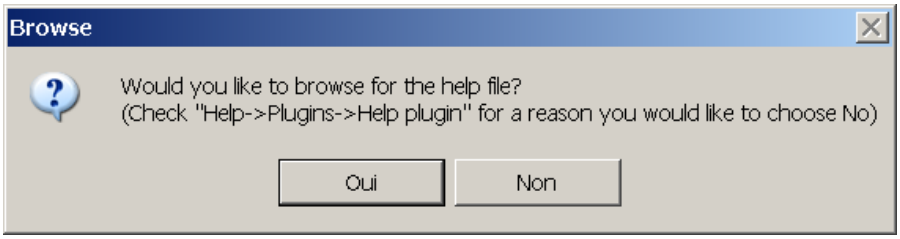
Dans la fenêtre qui s'ouvre, sélectionnez l'icône « Help files » (tout en bas, en descendant avec l'ascenseur). Cliquez ensuite sur le bouton « Add » :



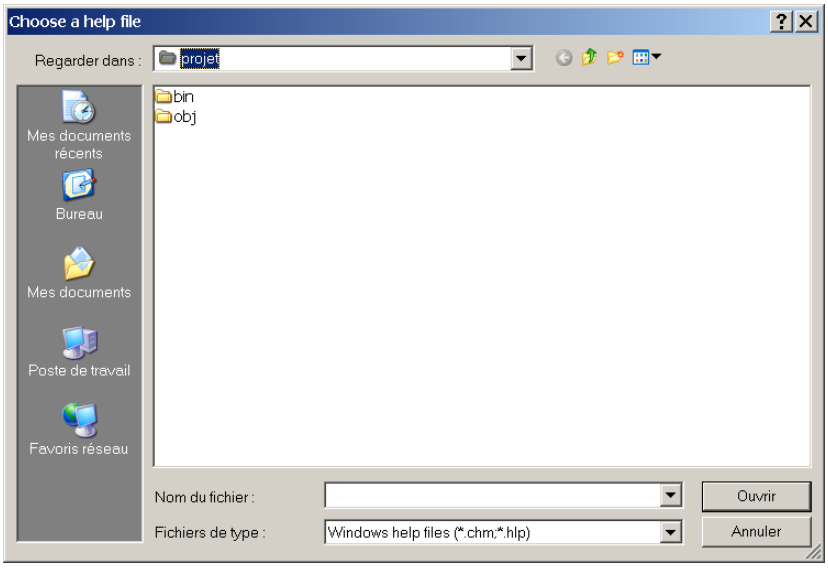
Dans la fenêtre suivante, tapez « `doc C ansi` » puis cliquez sur « OK » :



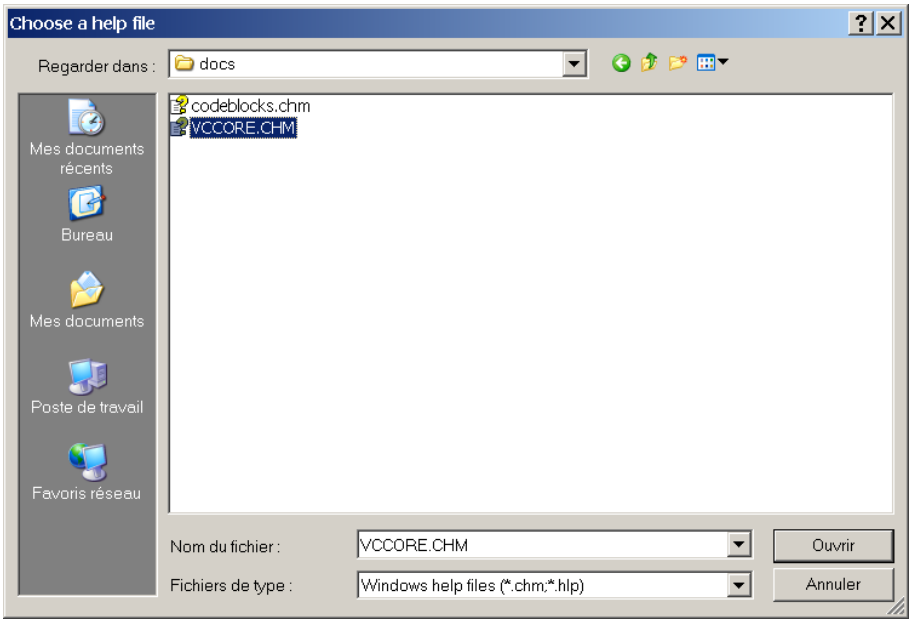
Dans la fenêtre qui s'ouvre, cliquez sur « Oui » :



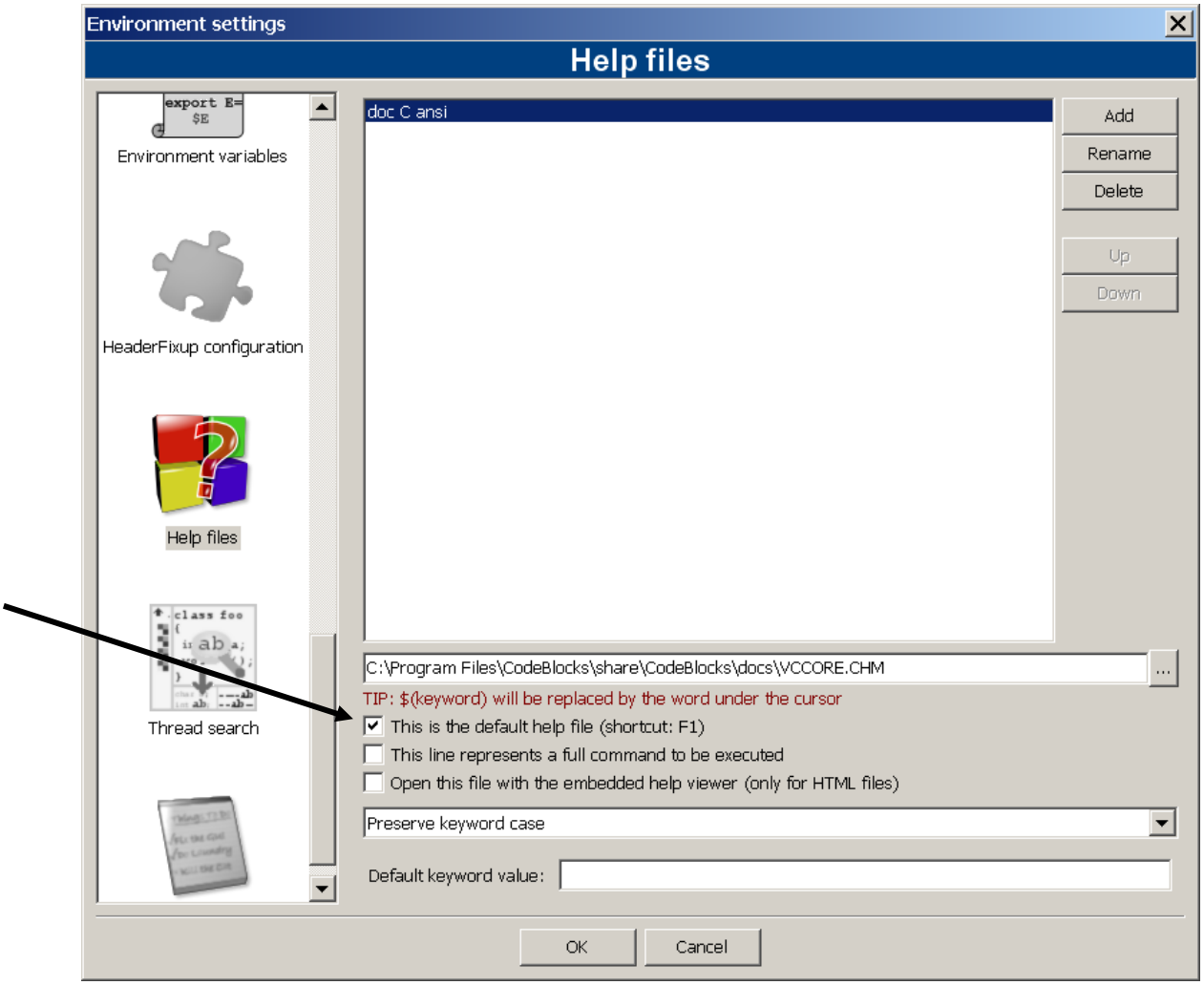
Une fenêtre de sélection du fichier d'aide s'ouvre alors :



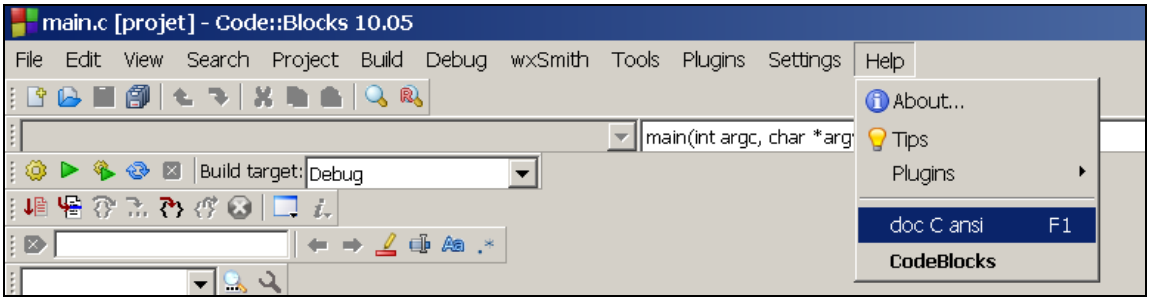
Allez dans le répertoire `docs` de CodeBlocks (`C:\Program Files\CodeBlocks\doc`), sélectionnez le fichier VCCORE.CHM puis cliquez sur « Ouvrir » :



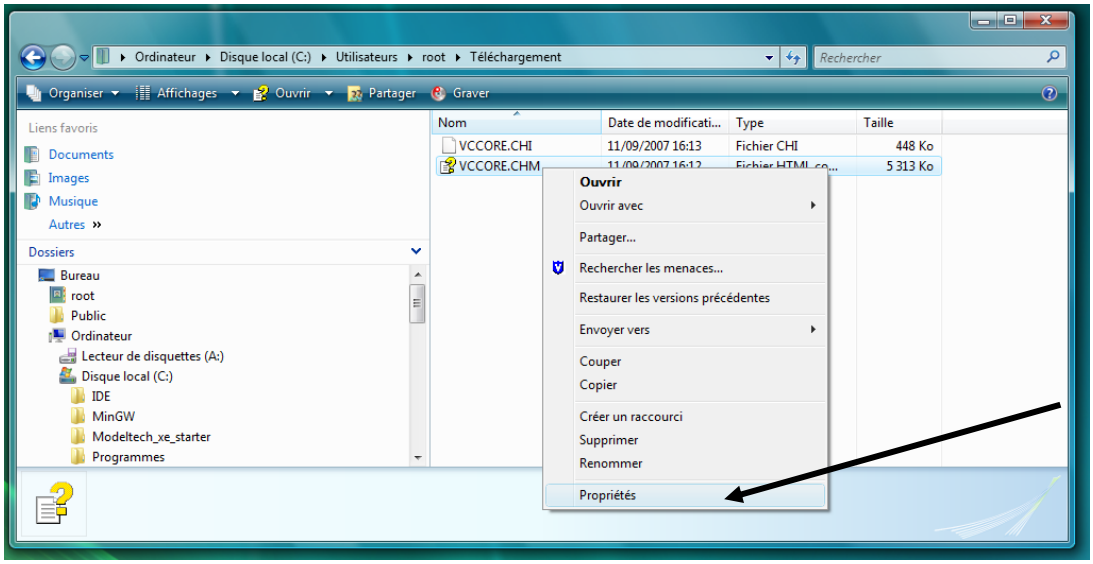
Dans la fenêtre de configuration suivante, cochez la case « This is the default help file », puis cliquez sur « OK » :



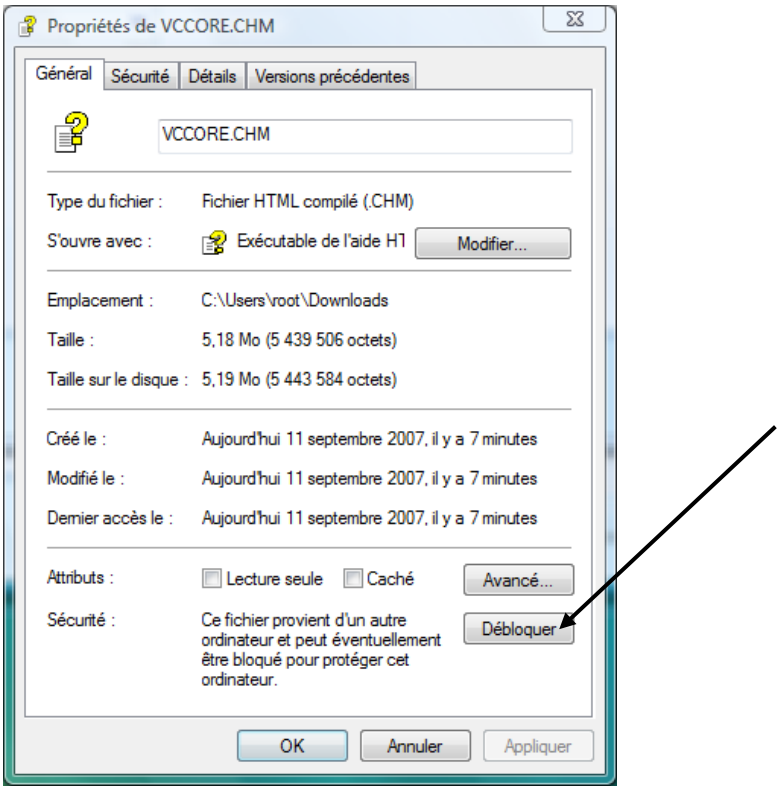
La nouvelle aide apparaît maintenant en bas du menu « Help » :



Début de la modification Vista (Seven ?). Les fichiers CHM et CHI sont maintenant bloqués par défaut sous Vista. Pour les débloquer, procédez de la manière suivante. Dans l'explorateur Windows, sélectionnez chaque fichier, puis cliquez avec le bouton droit de la souris. Dans le menu contextuel qui apparait, cliquez sur « Propriétés » :

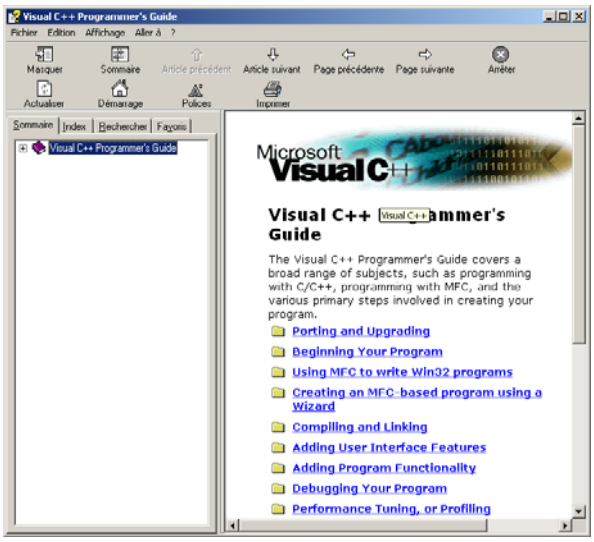


Dans la fenêtre qui s'ouvre, cliquez sur le bouton « Débloquer » puis sur le bouton OK.

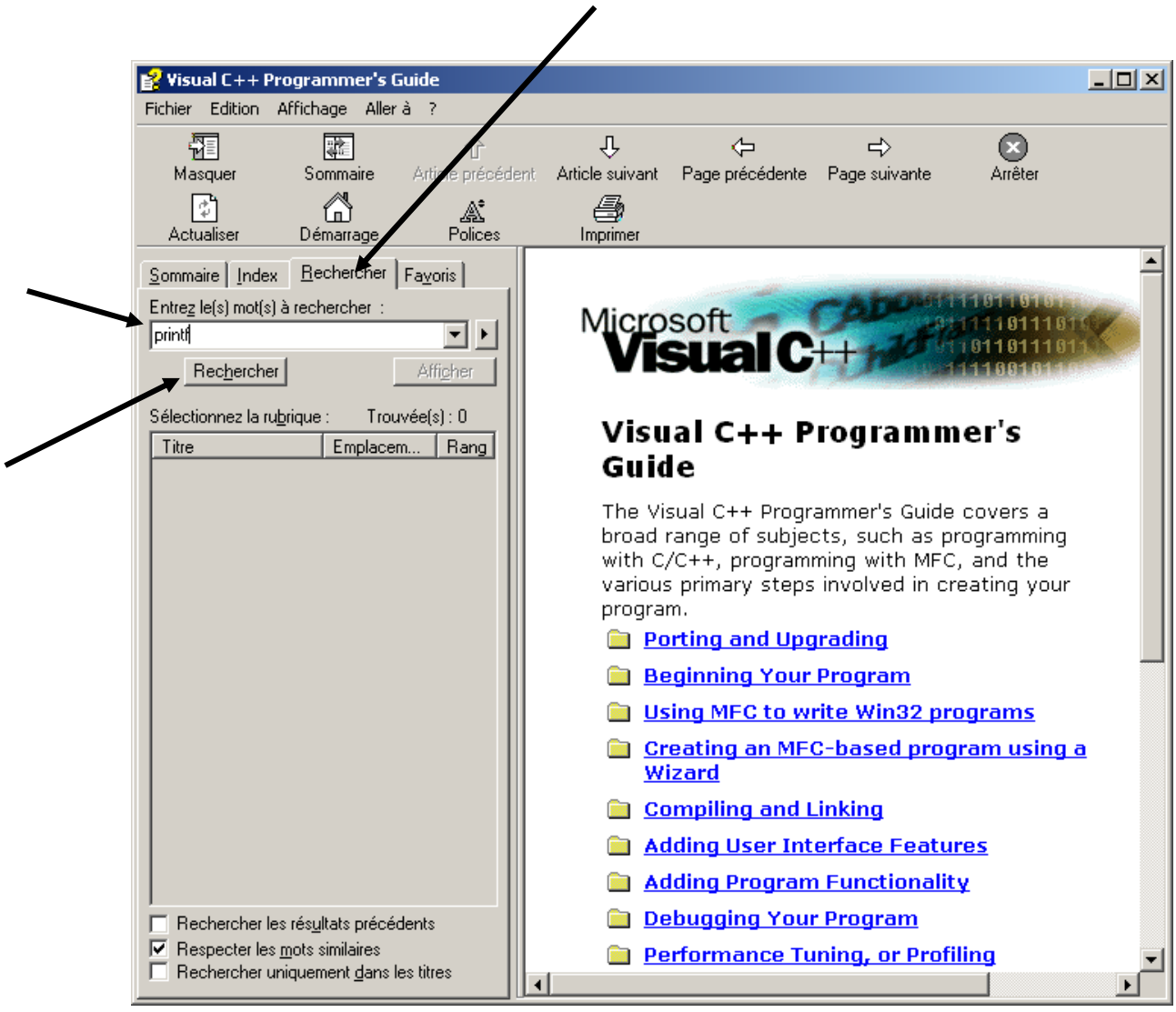


Fin de la modification Vista (Seven ?).

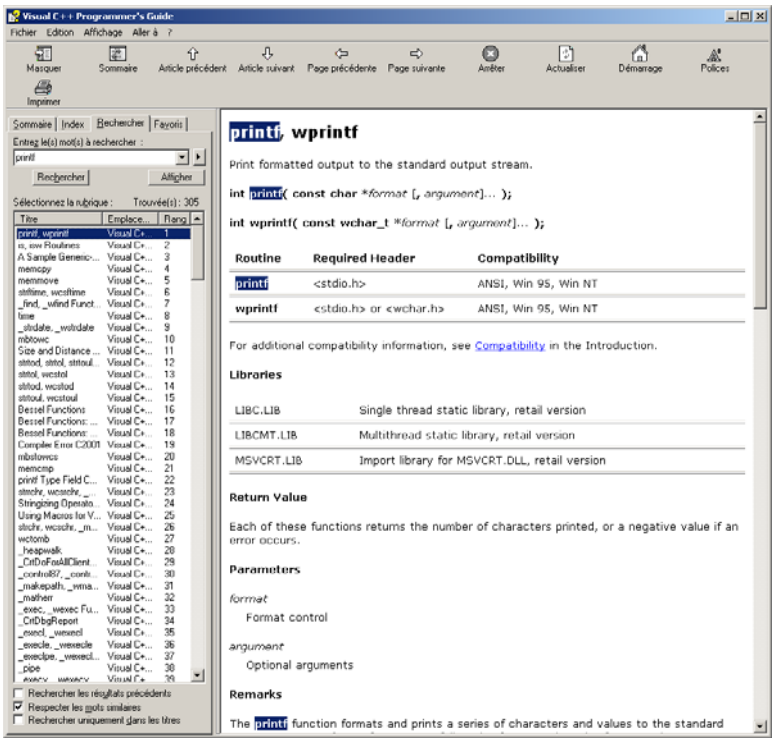
Appuyez sur la touche F1 du clavier. La fenêtre d'aide apparaît :



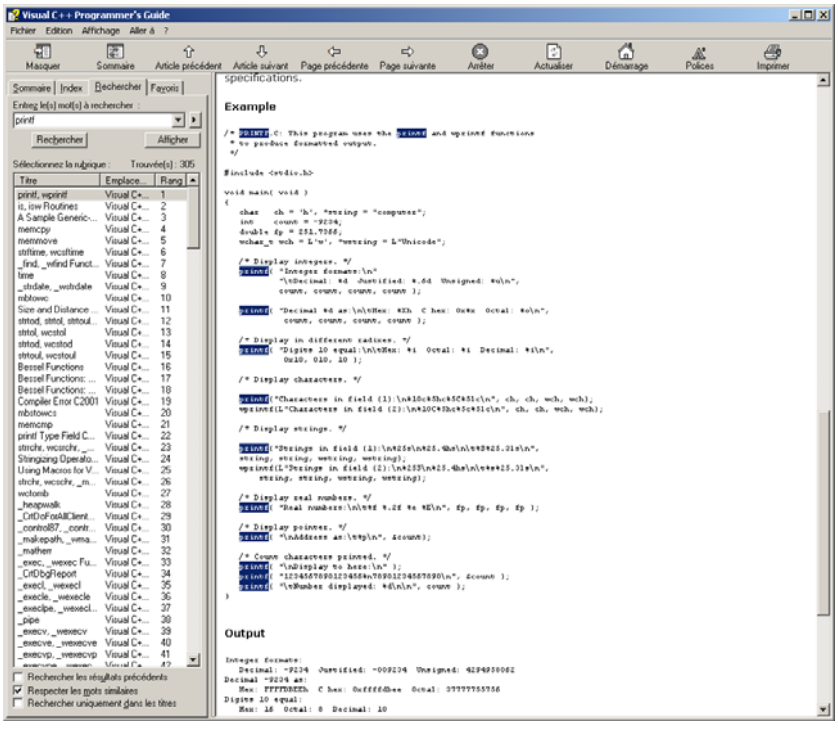
Le plus simple pour s'en servir, c'est de cliquer sur l'onglet « Rechercher » :



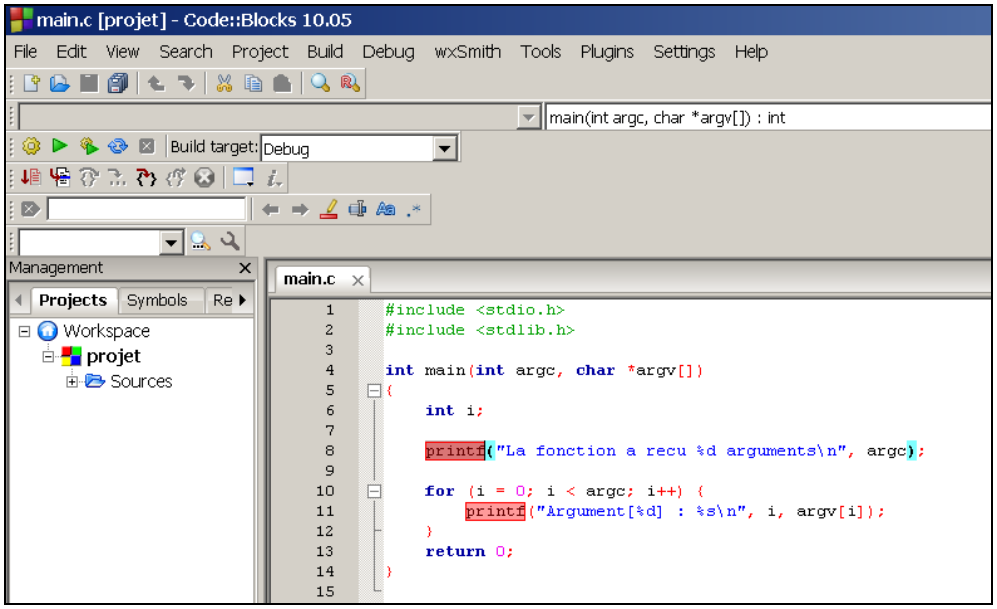
Tapez `printf` dans le champ de saisie puis cliquez sur le bouton « Rechercher ». Il y a beaucoup de réponses. Double-cliquez sur la première, son contenu apparaît dans la fenêtre de droite :



La documentation sur la fonction `printf` est très complète. Il y a même des exemples d'utilisation à la fin du document :



Il y a encore mieux. Vous pouvez sélectionner un mot-clef du langage C dans votre source (en double-cliquant dessus par exemple) :



puis appuyer sur la touche F1 du clavier. L'aide apparaît alors et pointe automatiquement sur le mot-clef sélectionné.