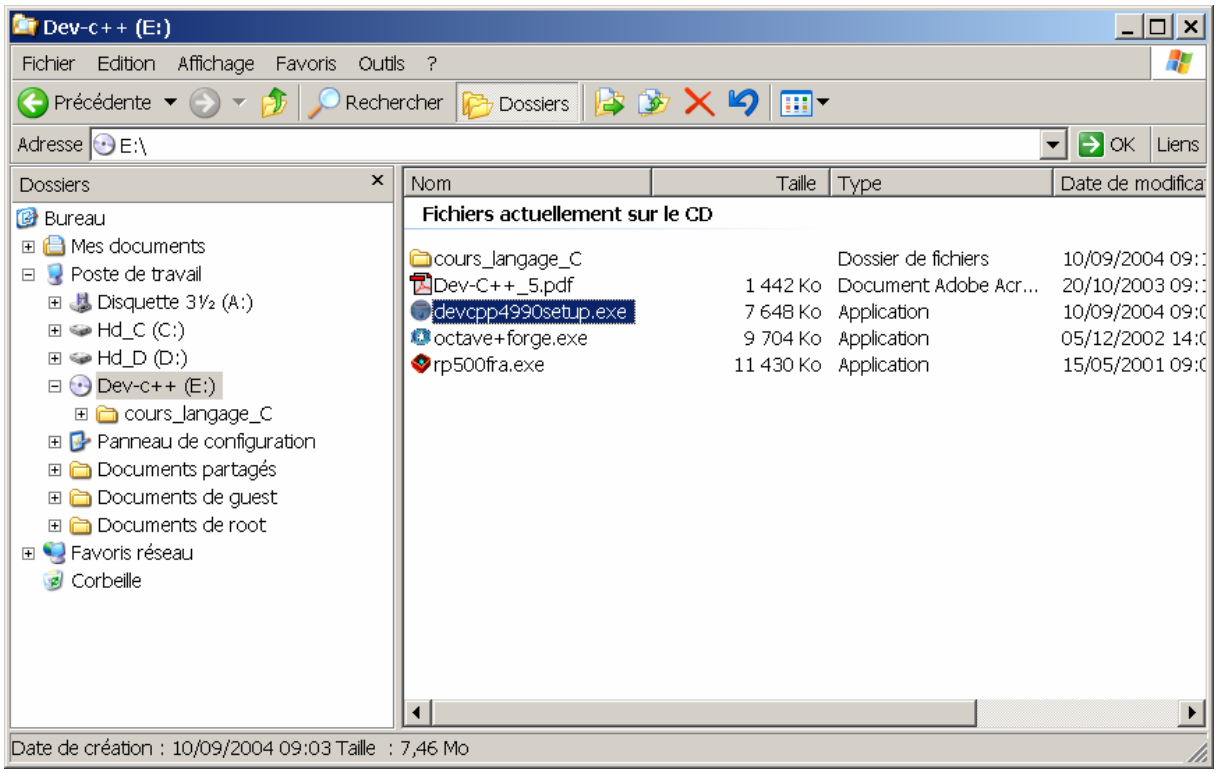
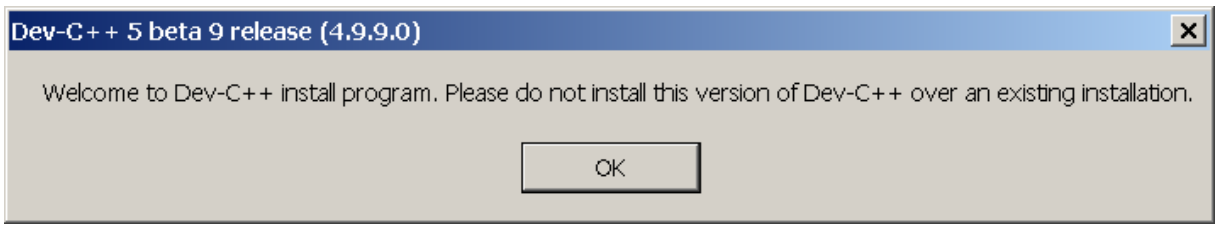
1) Installation de Dev-C++

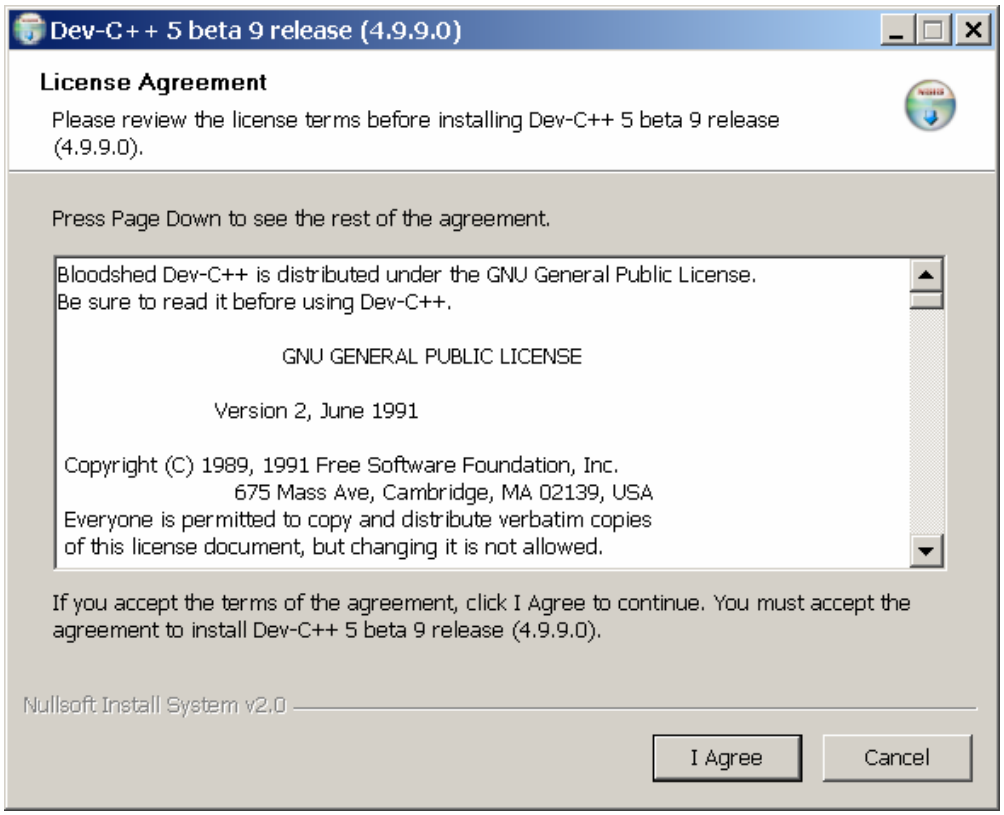
Téléchargez le fichier `devcpp4990setup.exe` dans un répertoire de votre PC, puis double-cliquez dessus :



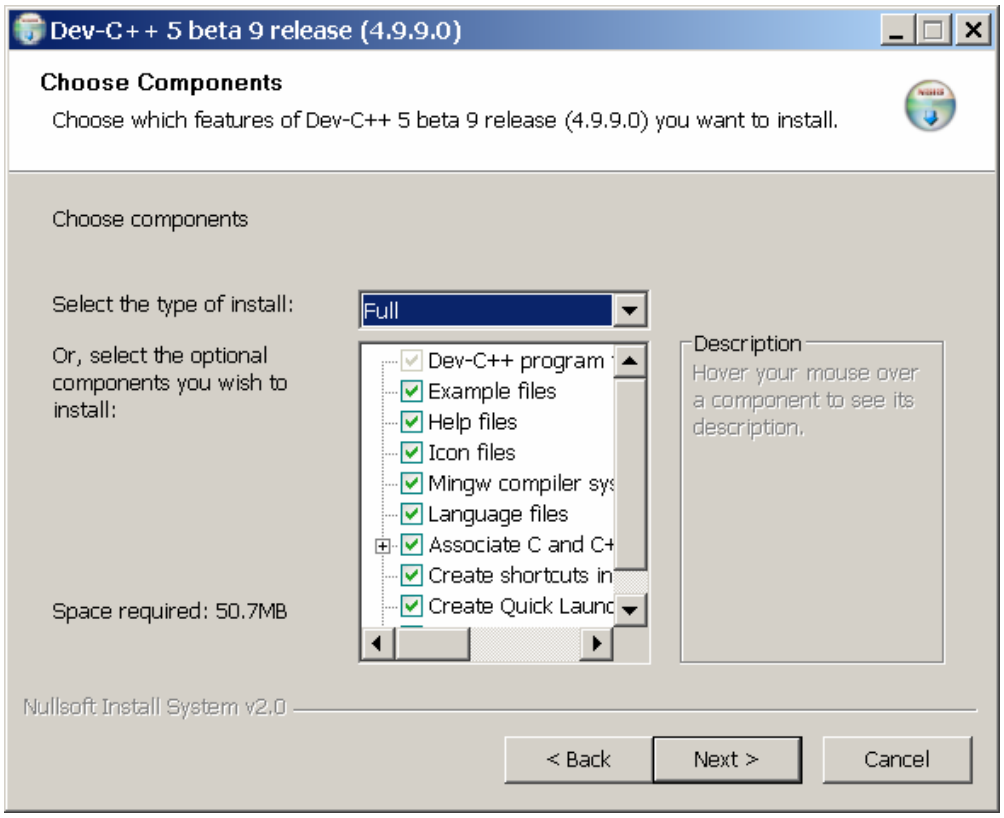
La procédure d'installation démarre. La fenêtre suivante vous indique qu'il faut désinstaller les versions précédentes de Dev-C++ avant d'installer celle-là. Cliquez sur le bouton « OK » :



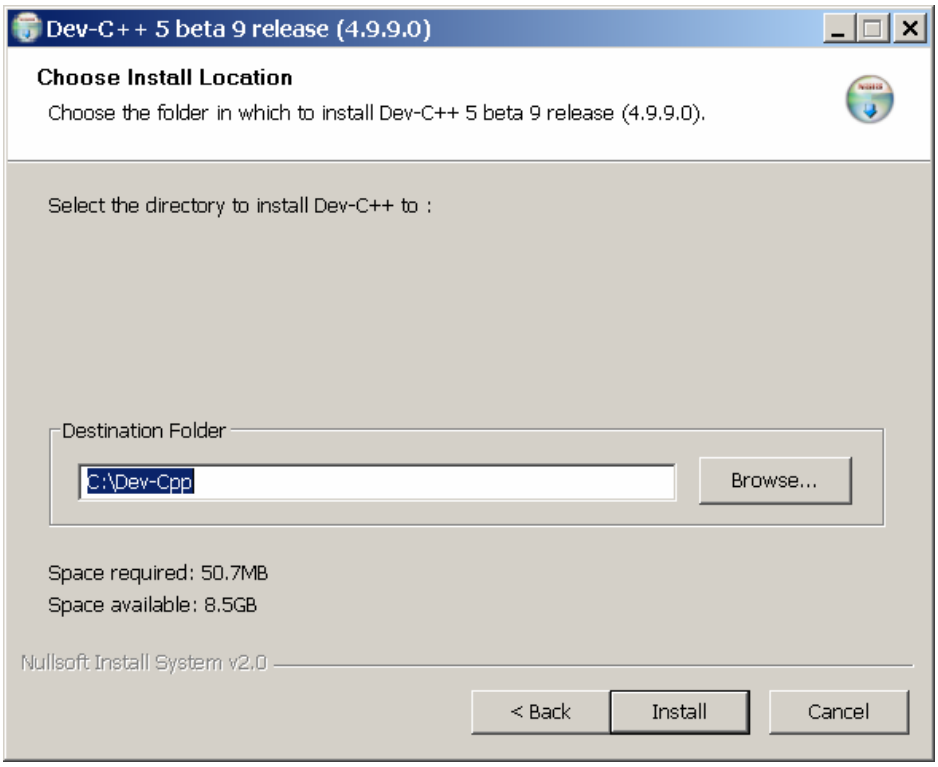
Dans la fenêtre suivante, cliquez sur le bouton « I Agree » pour accepter les termes de la licence GNU :



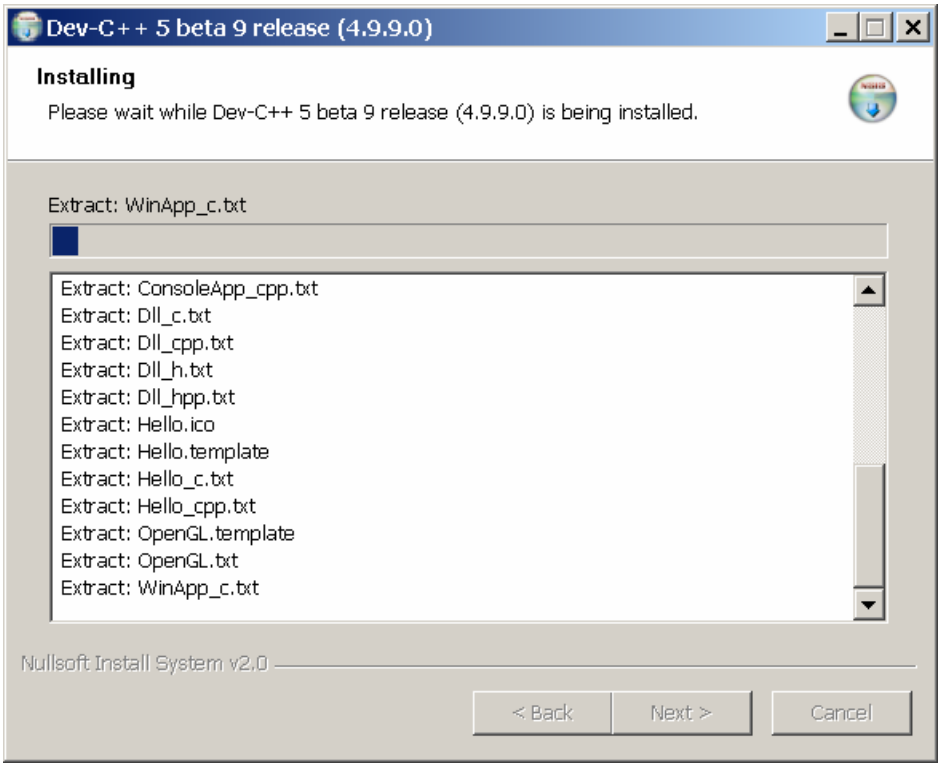
Dans la fenêtre suivante, laissez le type d'installation sur « Full » et cliquez sur le bouton « Next > » :



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

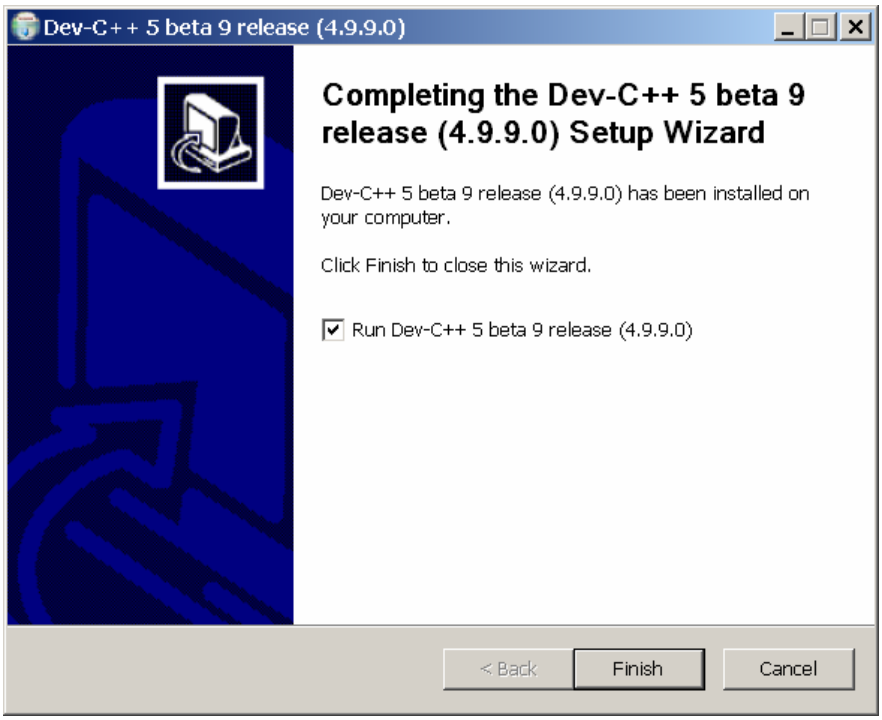


L'installation démarre :

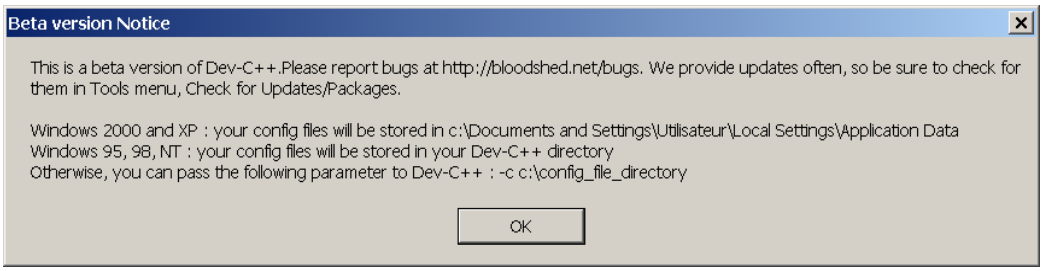


La fenêtre suivante vous demande si vous souhaitez que tous les utilisateurs de la machine puissent utiliser Dev-C++. Cliquez sur le bouton « Oui ».

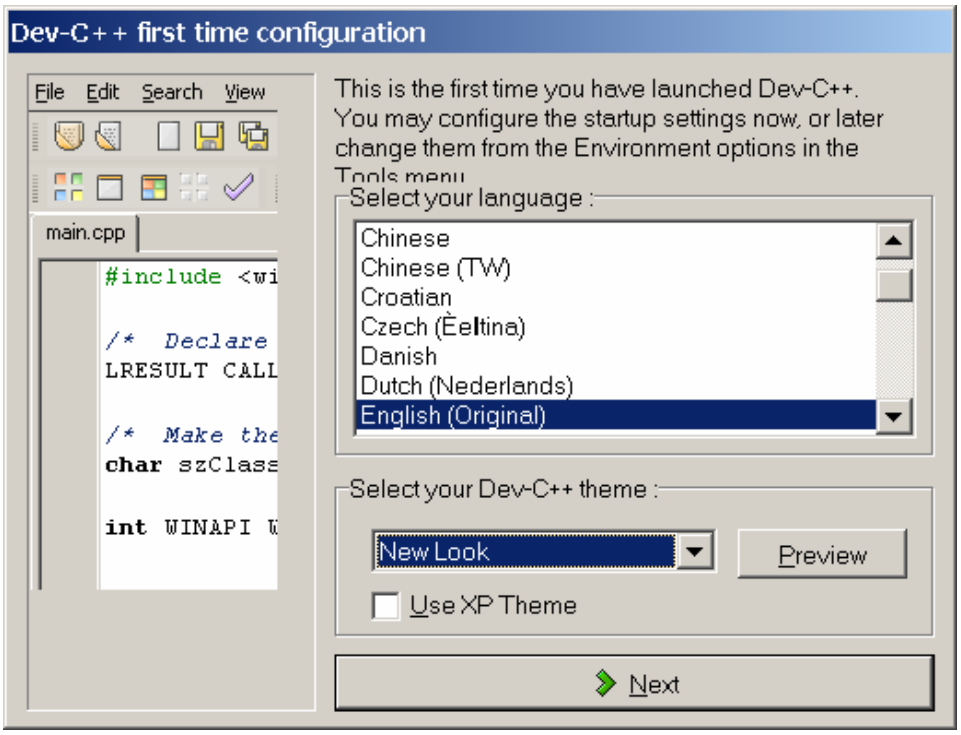
L'installation est terminée. Quand vous cliquerez sur le bouton « Finish » dans la fenêtre suivante, Dev-C++ sera automatiquement lancé.



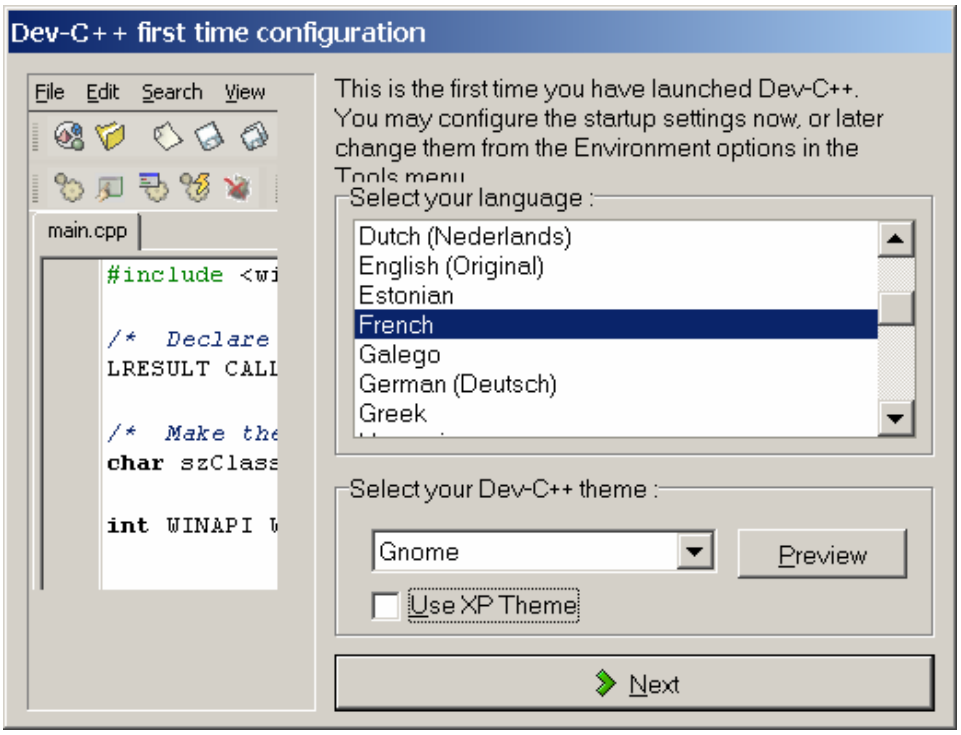
Au cours de ce premier lancement, certaines questions concernant la configuration du logiciel vont vous être posées. Elles n'apparaîtront plus lors des lancements ultérieurs. Cliquez sur le bouton « OK » dans la fenêtre suivante :



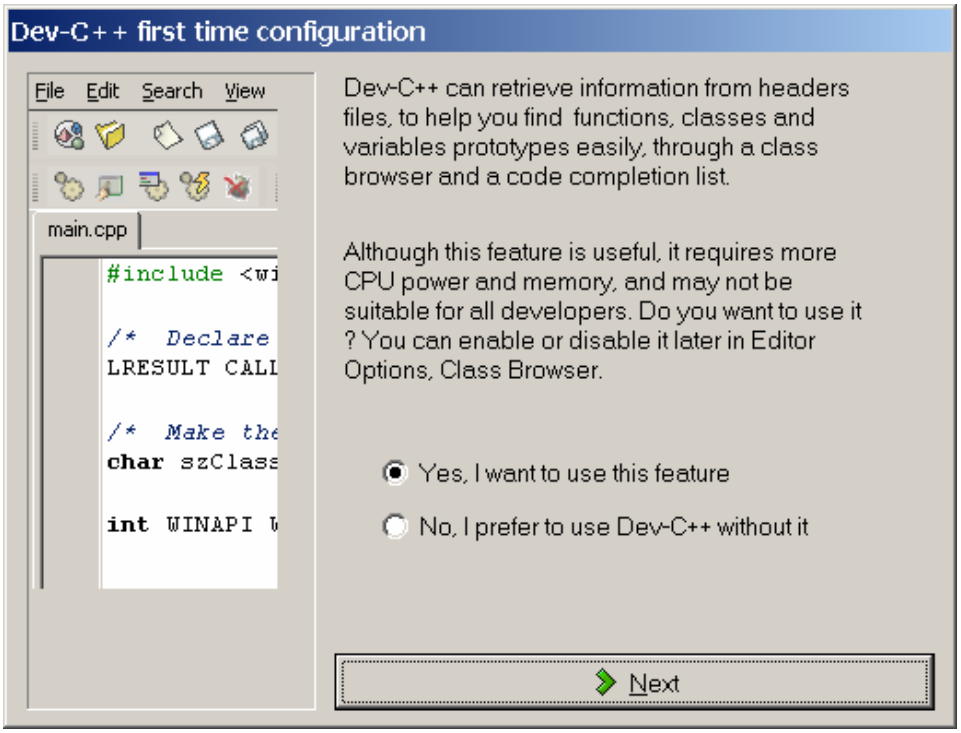
La fenêtre ci-dessous apparaît :



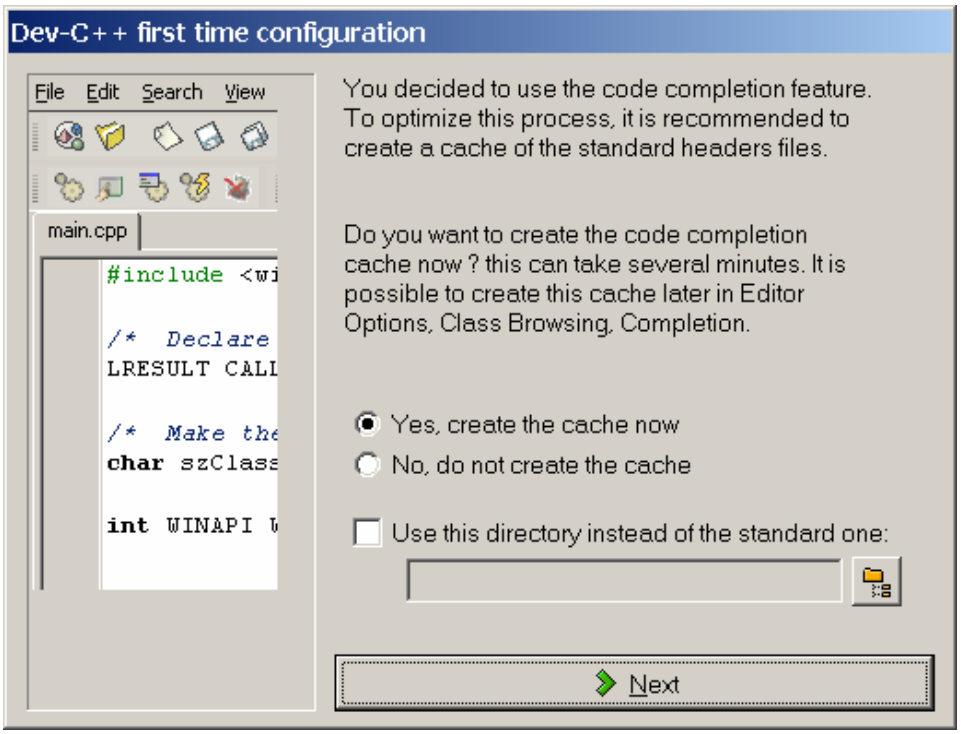
Sélectionnez le langage « French » et le thème « Gnome » avant de cliquer sur le bouton « Next » :



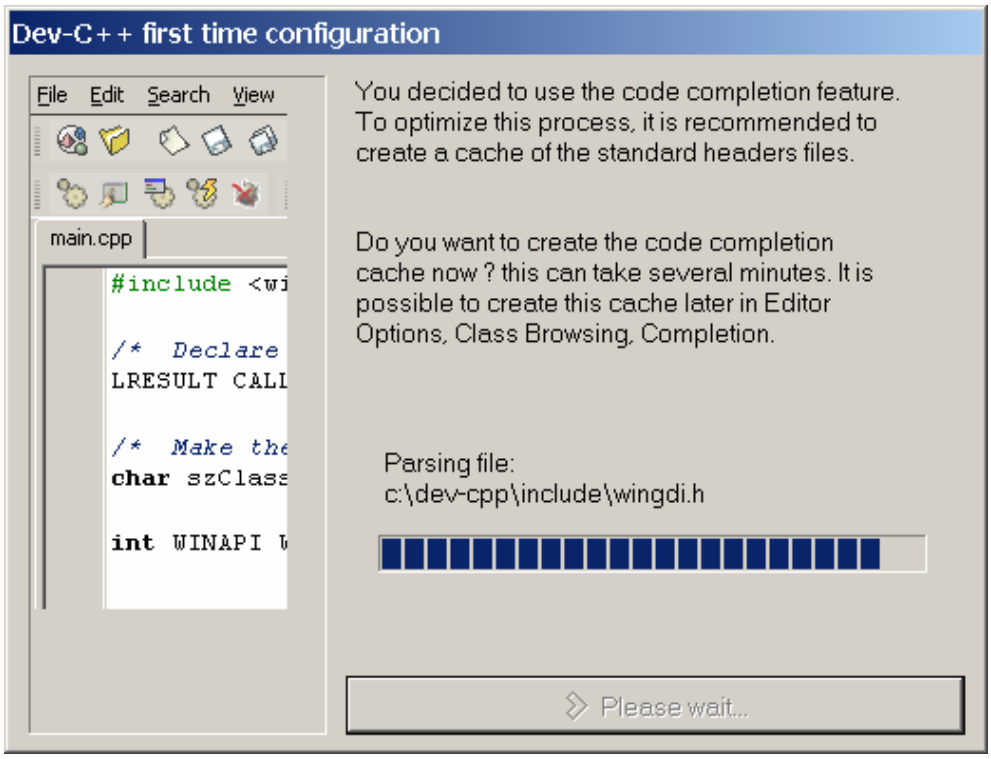
Cliquez sur le bouton « Next » dans la fenêtre suivante :



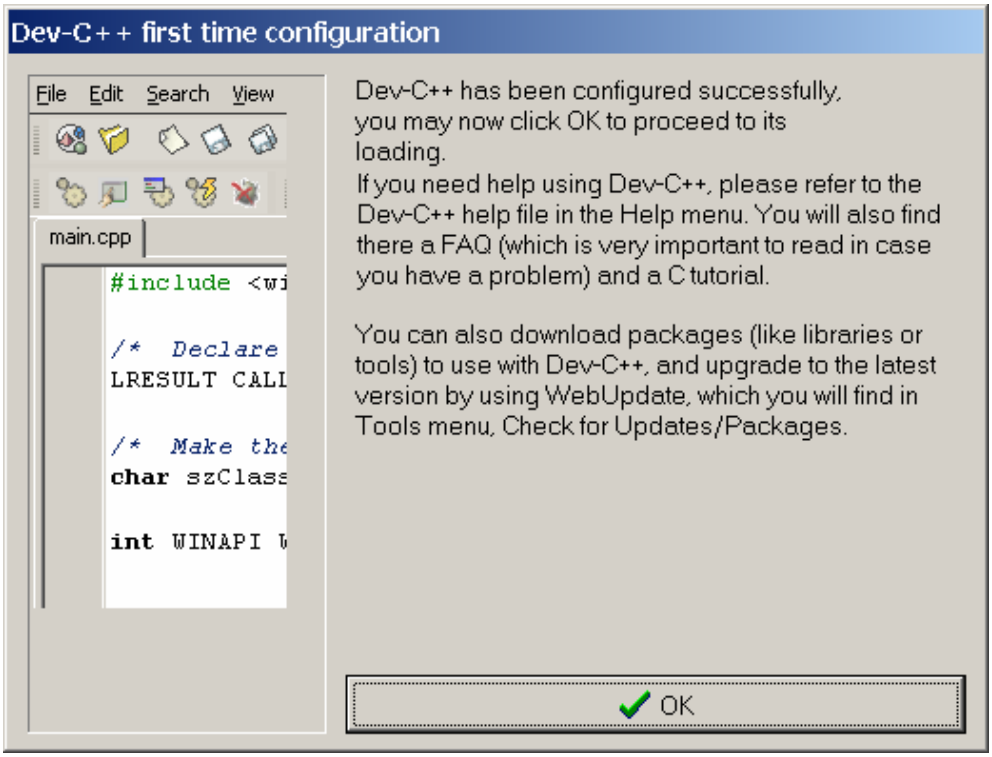
Cliquez sur le bouton « Next » dans la fenêtre suivante :



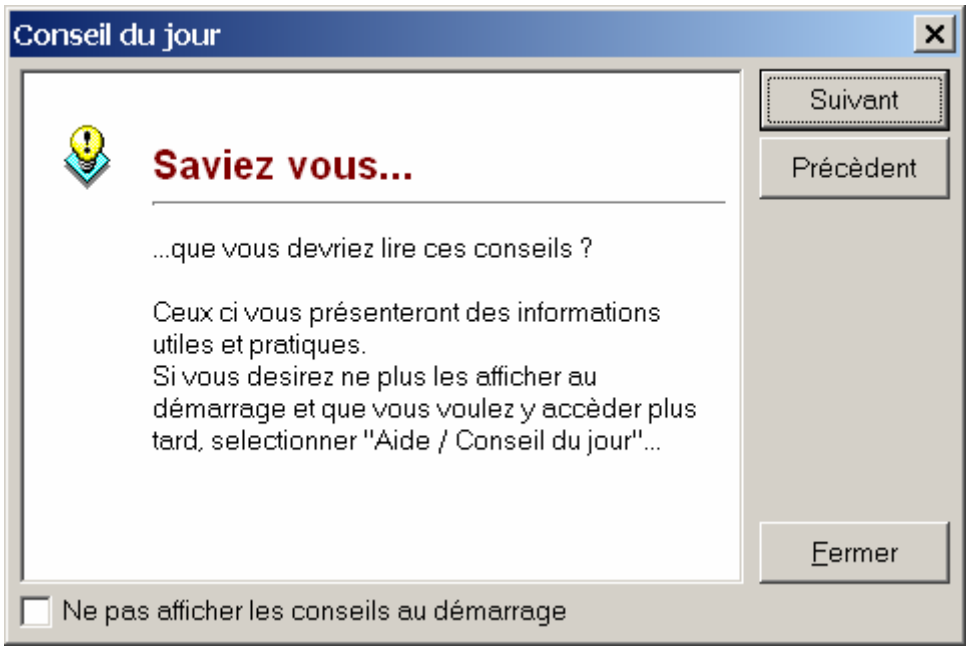
La fenêtre suivante apparaît à l'écran :



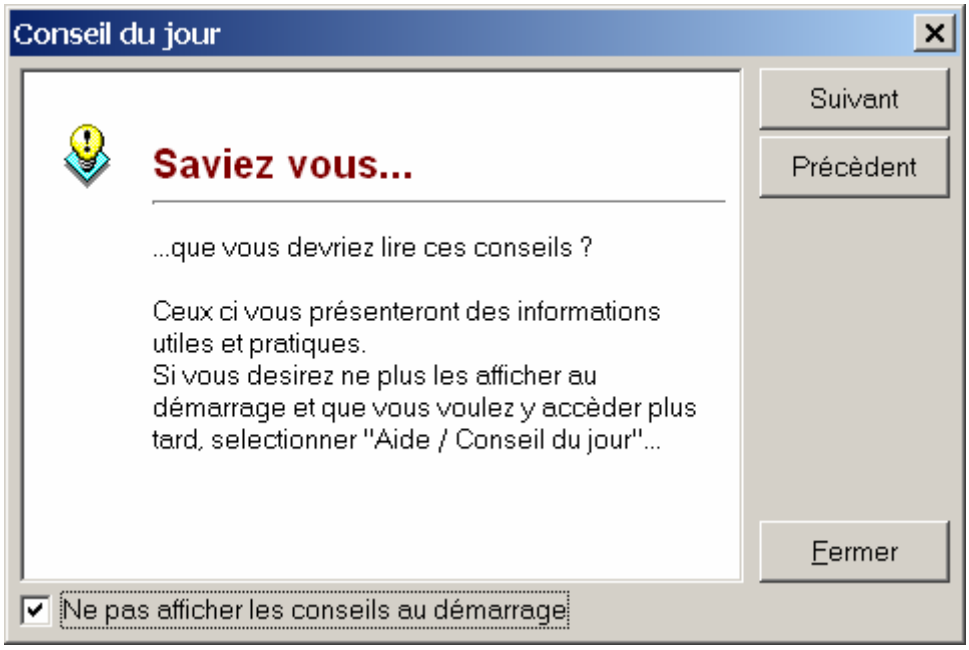
Après un moment, la fenêtre ci-dessous vous indique que la configuration est terminée. Cliquez sur le bouton « OK » :



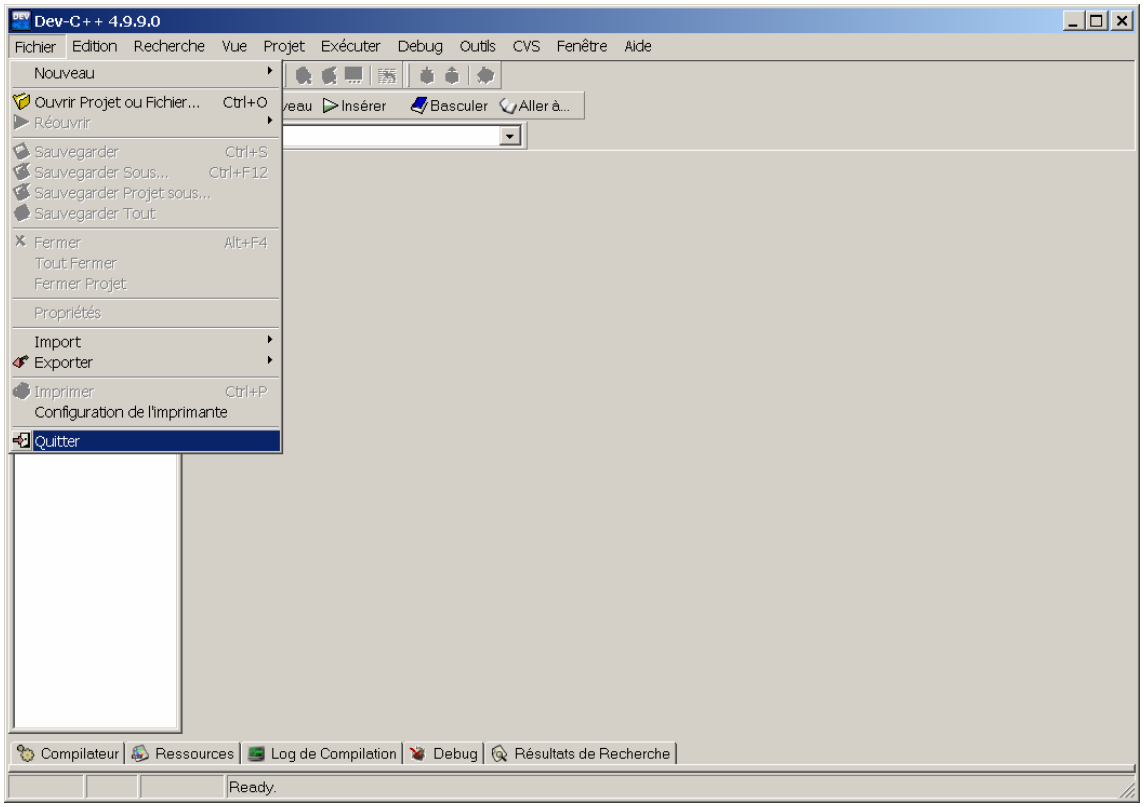
Dev-C++ apparaît enfin à l'écran. La fenêtre suivante apparaît à chaque lancement du logiciel



Cochez la case du bas (Ne pas afficher les conseils au démarrage) pour supprimer cette option puis cliquez sur le bouton « Fermer ».

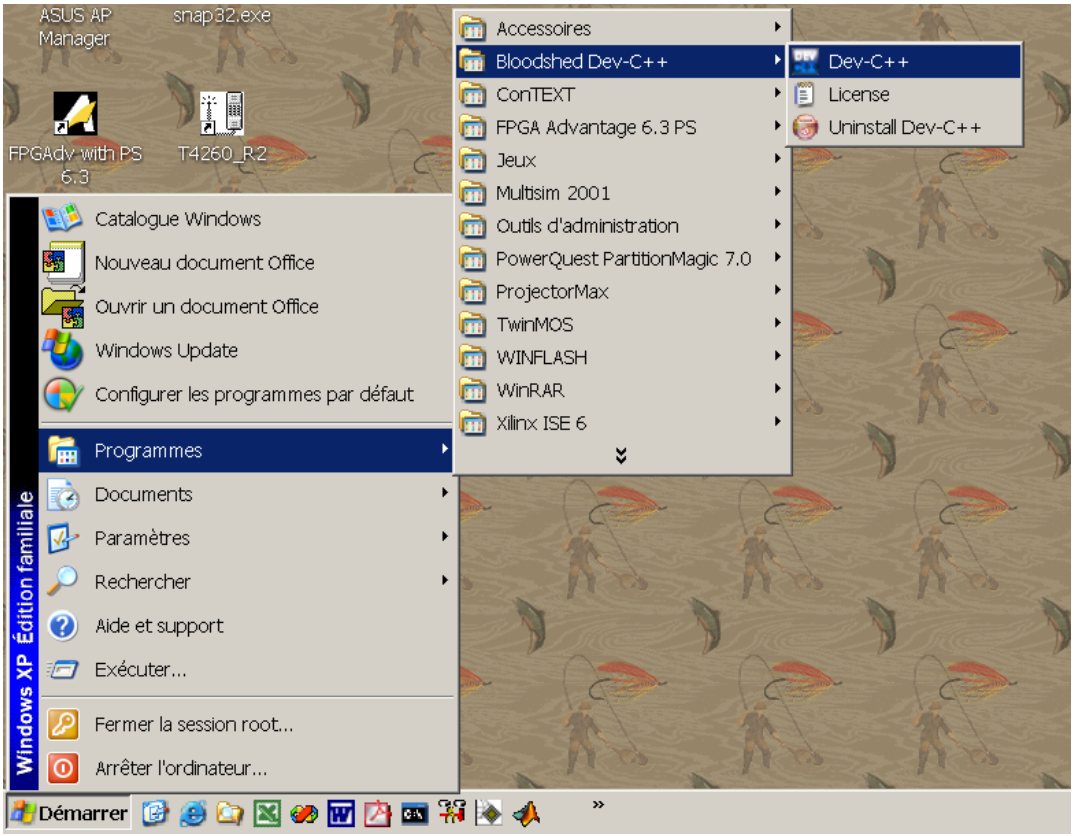


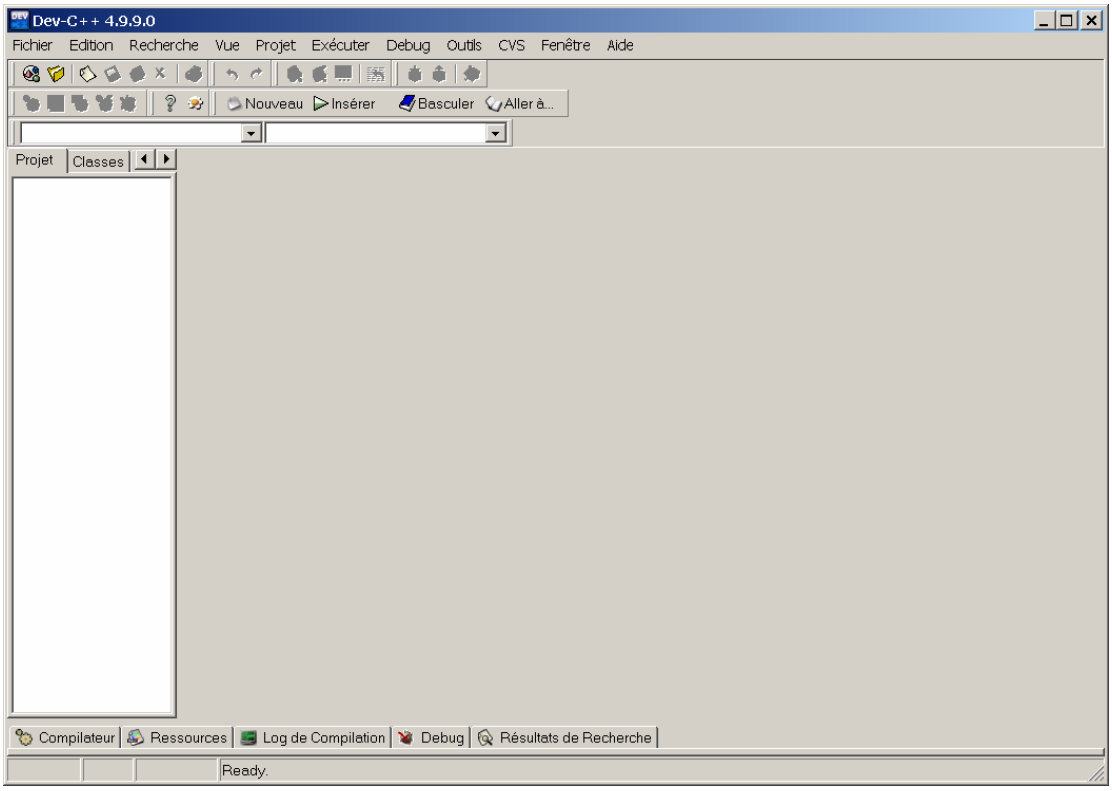
Quittez Dev-C++ de la manière suivante :



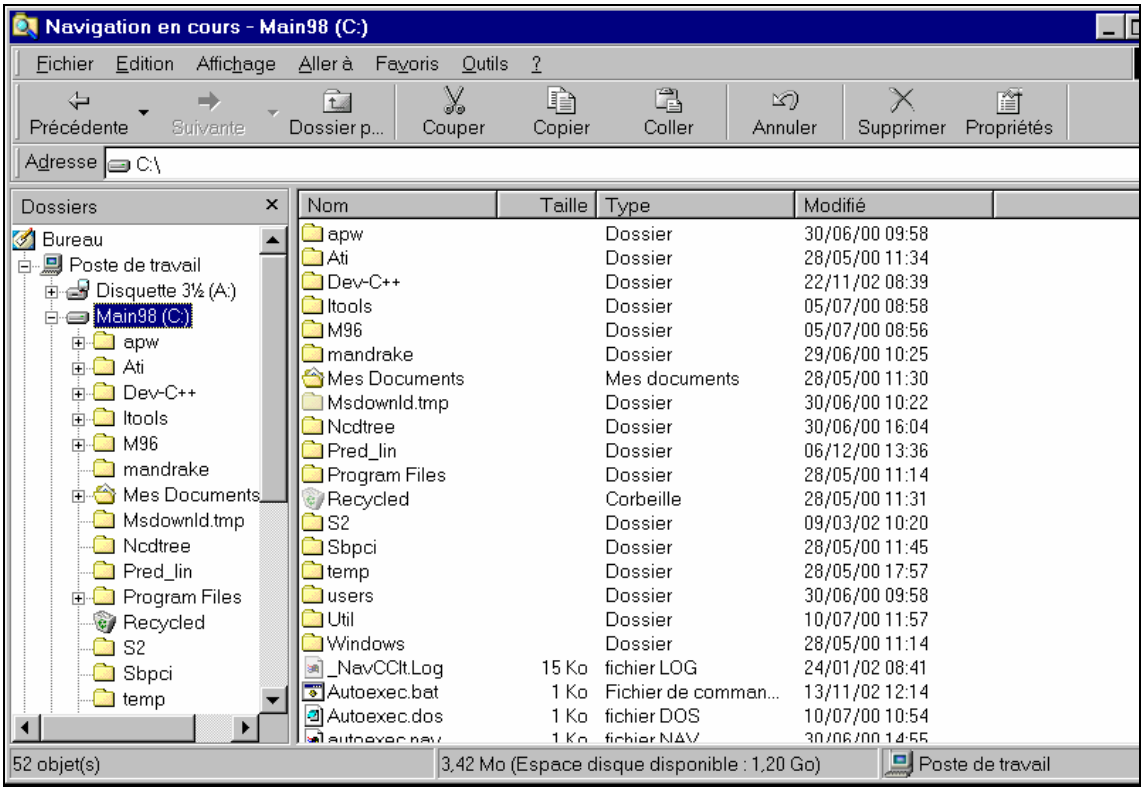
2) Premier programme

Vous pouvez maintenant relancer Dev-C++ :

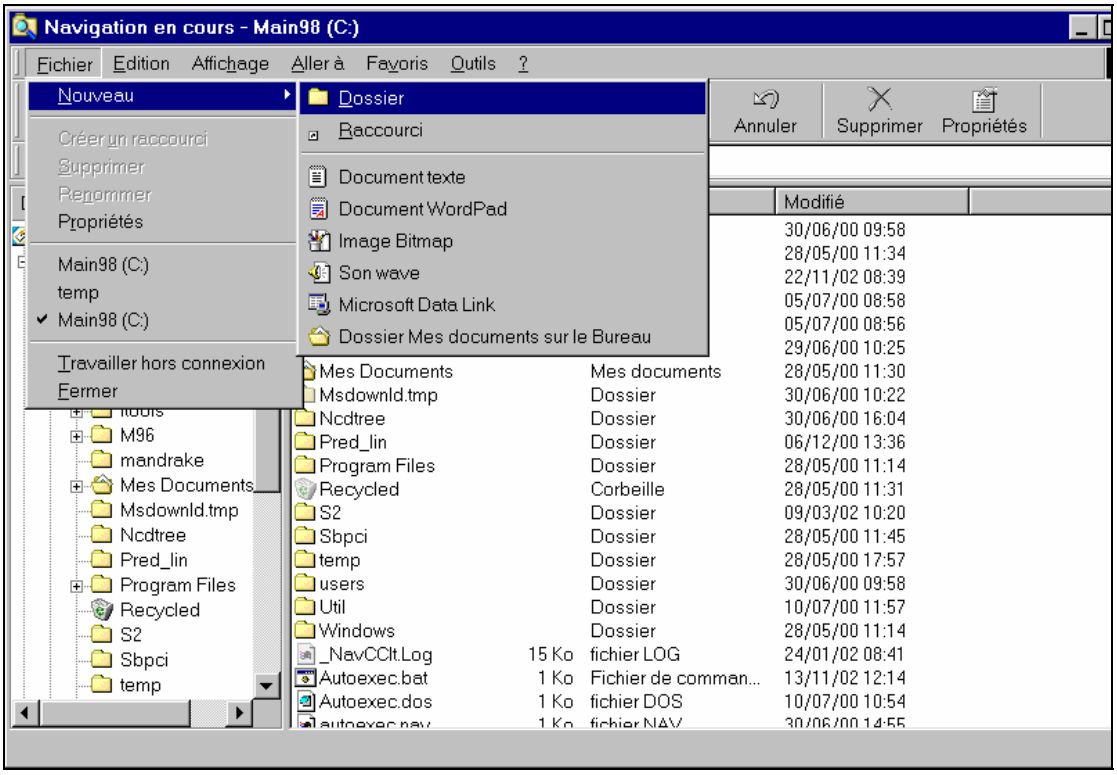




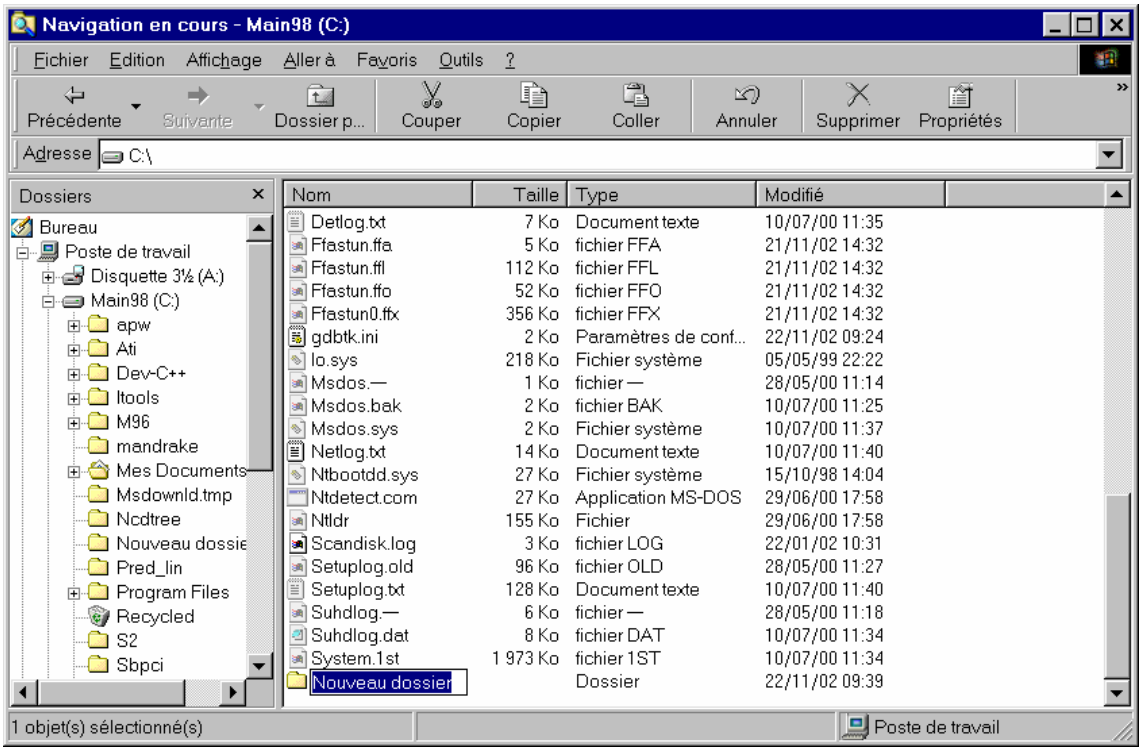
Avant d'écrire votre premier programme, nous allons créer un répertoire de travail où vous allez enregistrer vos fichiers. Pour cela, ouvrez l'explorateur Windows et sélectionnez le disque C :



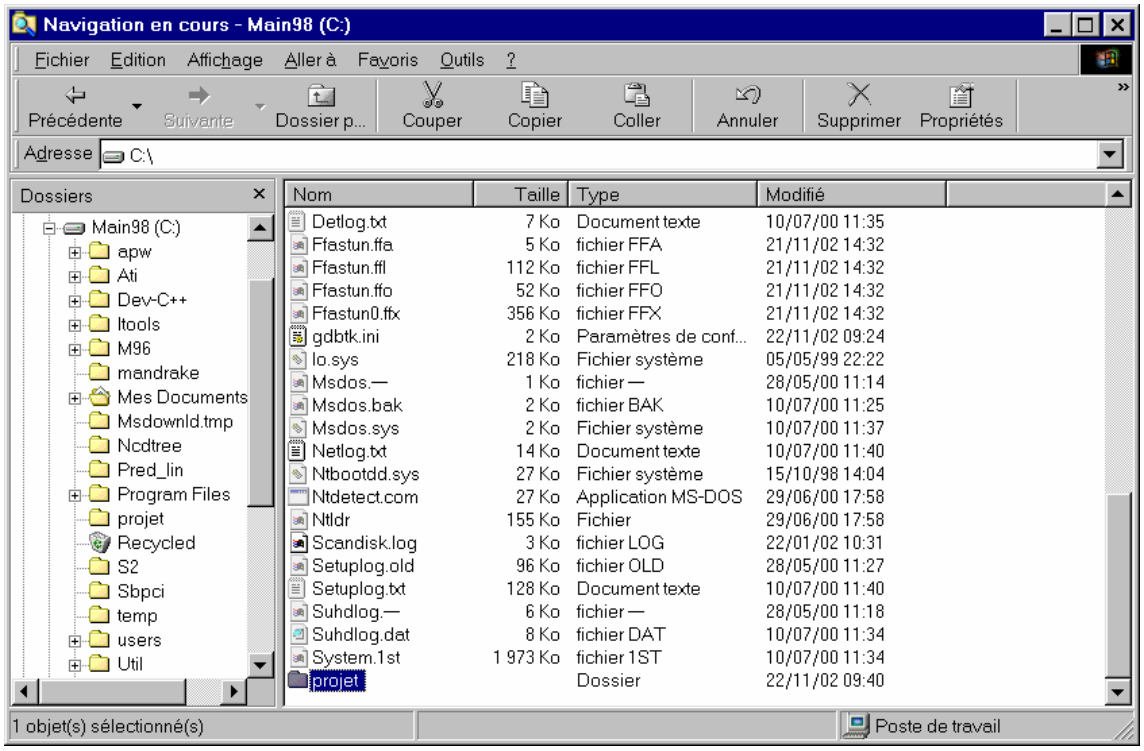
Créez ensuite un nouveau répertoire (on appelle cela un dossier sous Windows) en cliquant sur le menu « Fichier », sur le sous-menu « Nouveau » puis sur l'icône « Dossier » :



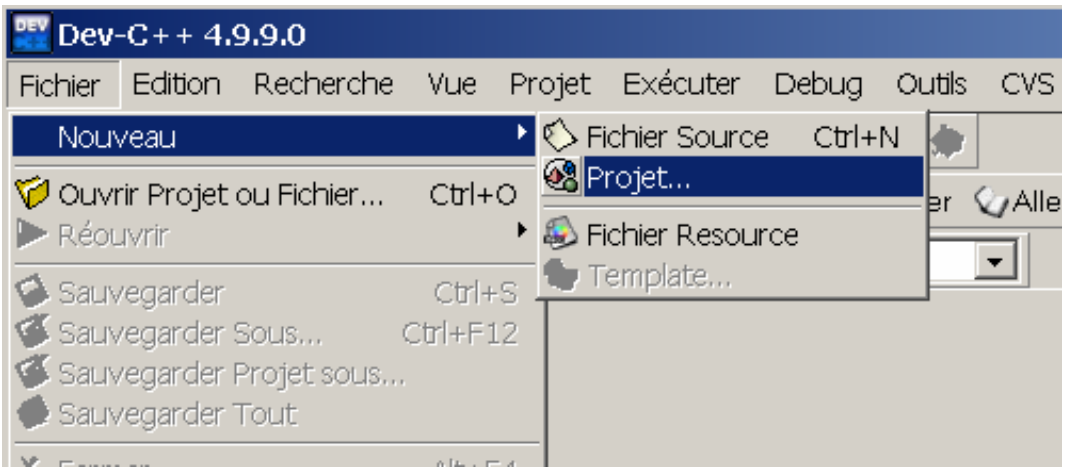
Le nouveau dossier apparaît en bas de la fenêtre avec son nom sélectionné en bleu :



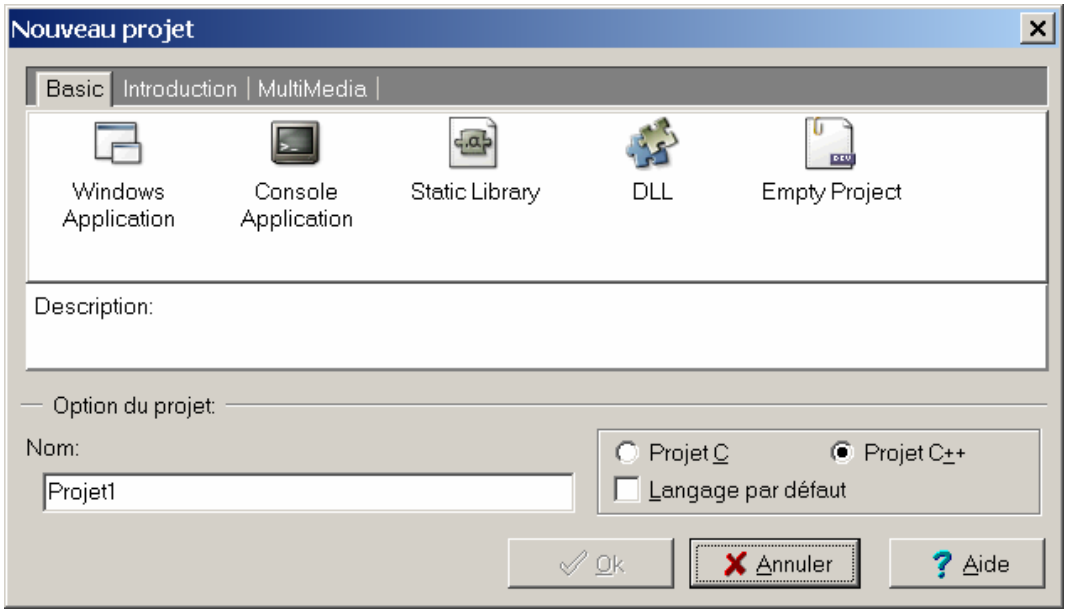
Tapez le nom `projet` puis tapez sur la touche Entrée du clavier. Le répertoire est créé :



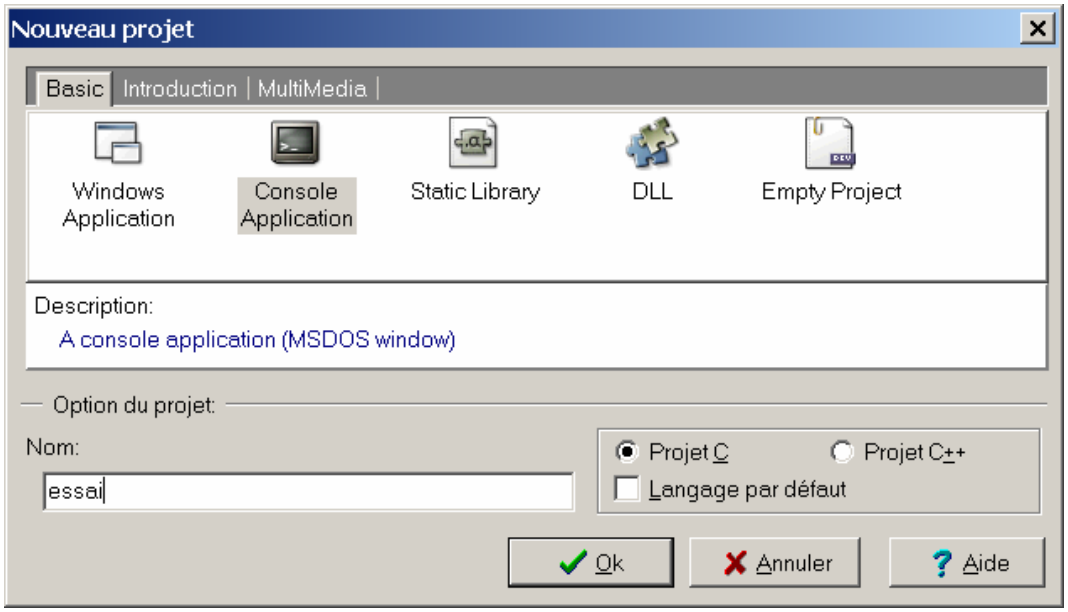
Fermez l'explorateur Windows et revenez sur la fenêtre de Dev-C++. Cliquez sur le menu « Fichier » puis sur le sous-menu « Nouveau » et enfin sur « Projet » :



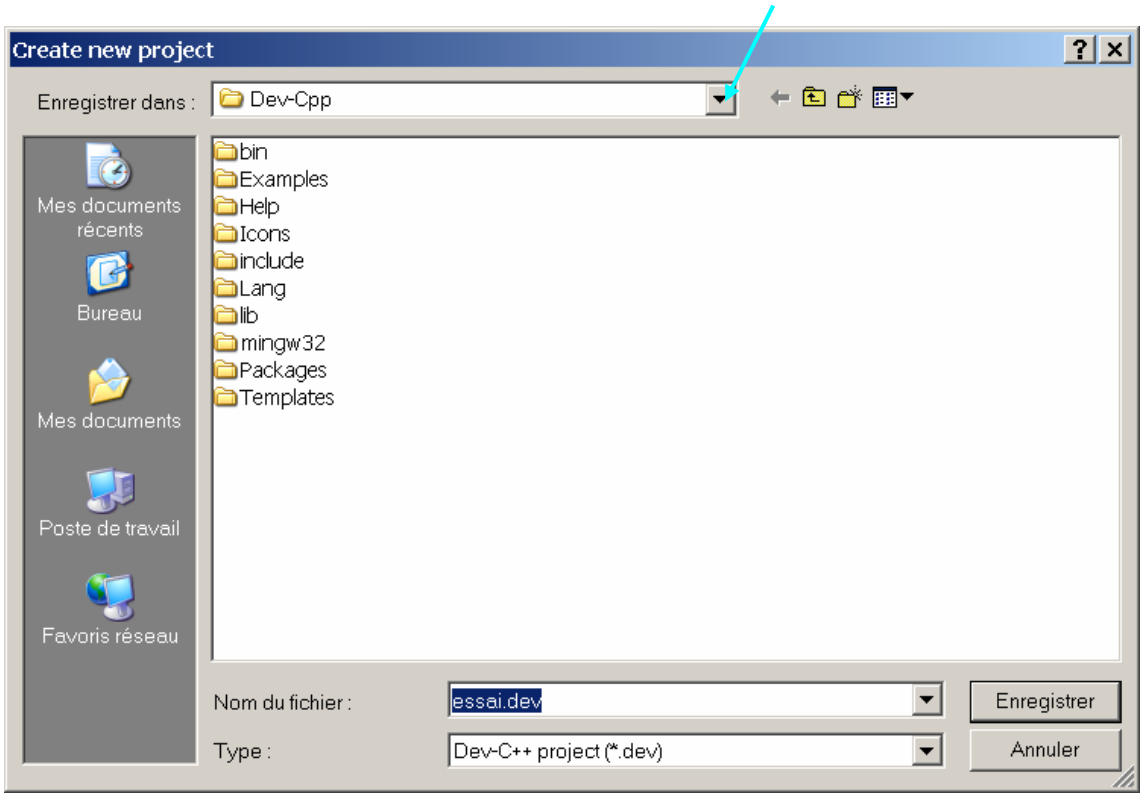
La fenêtre de création de projet apparaît à l'écran.



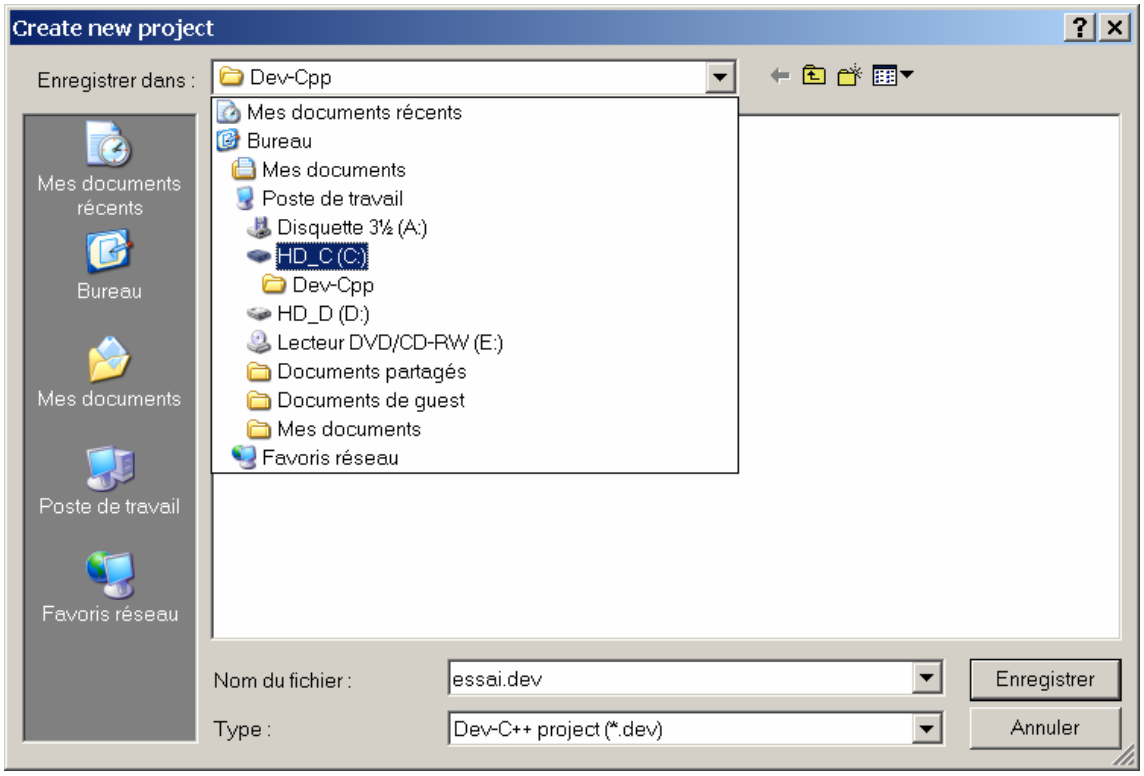
Cliquez sur l'icône « Console Application » pour créer une application non-graphique puis sélectionnez la case « Projet C » pour travailler en langage C. Tapez le nom du projet dans la zone « Nom », par exemple `essai`. Vous **devez** obtenir la fenêtre suivante avant de cliquer sur le bouton « OK » :



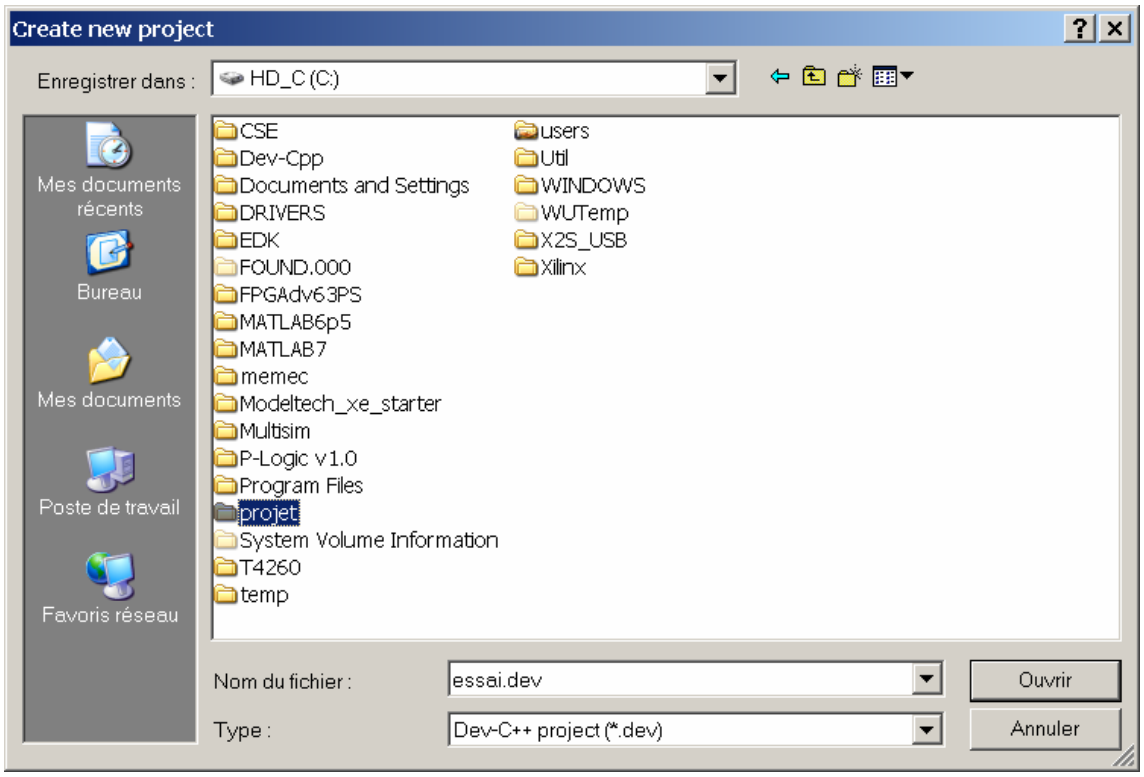
Le programme vous demande maintenant dans quel répertoire vous voulez enregistrer votre projet. Dans notre exemple, il s'agit du répertoire `C:\projet`. Pour lui indiquer, cliquez sur le petit triangle noir à droite de « Dev-Cpp »  indiqué par la flèche ci-dessous :



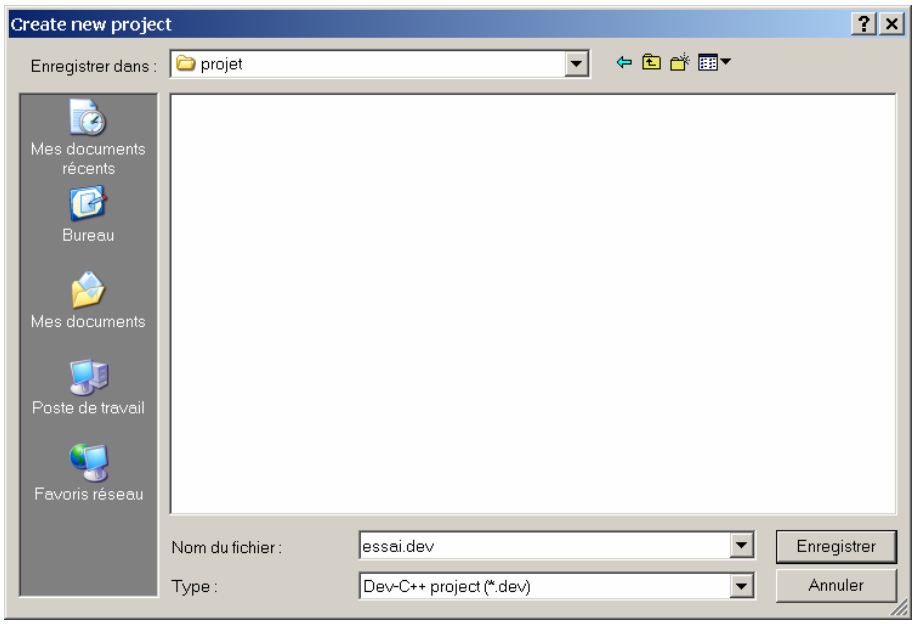
Dans la petite fenêtre qui apparaît alors, cliquez sur le disque `C:`



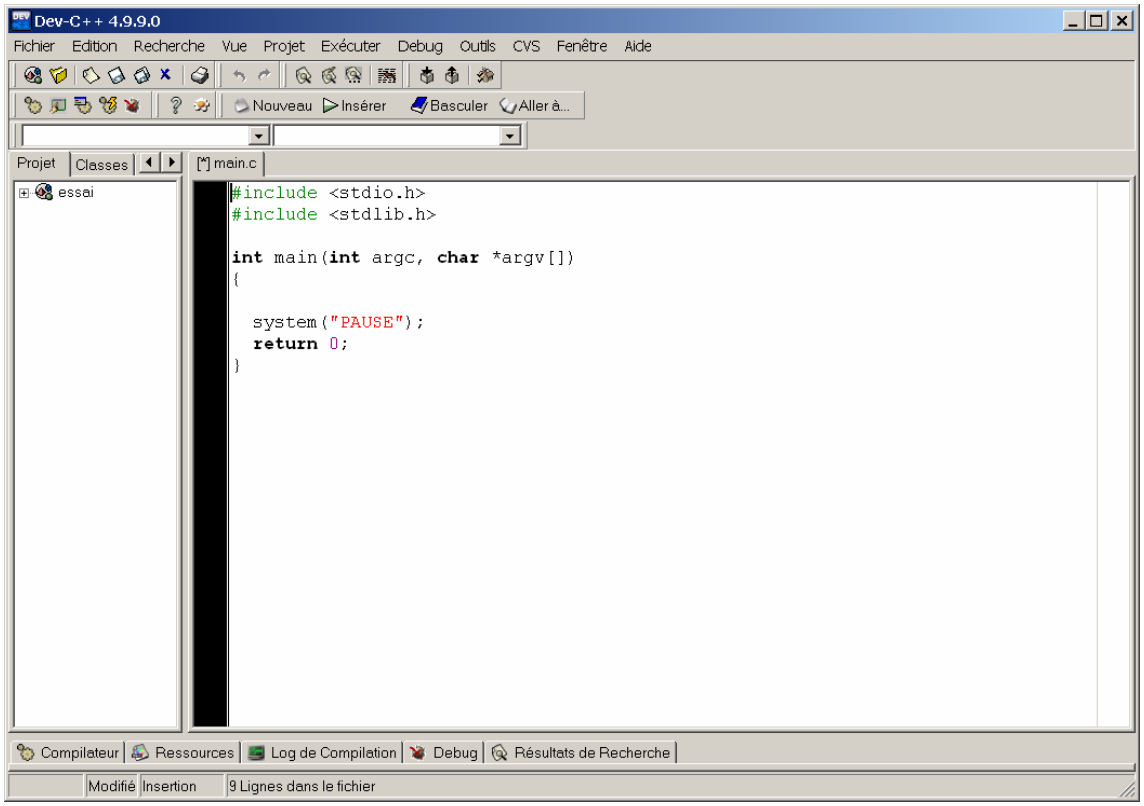
Puis double-cliquez ensuite sur le répertoire `projet` :



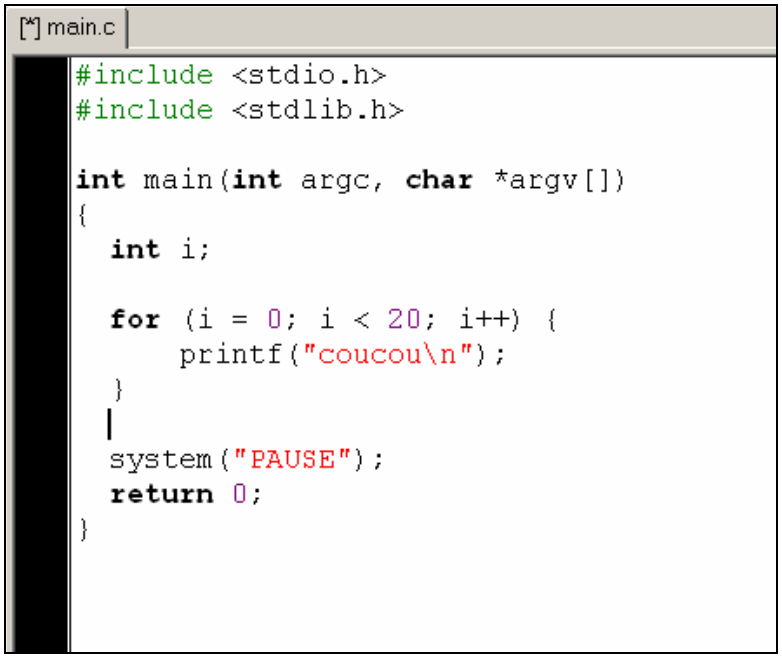
Vous êtes maintenant dans le répertoire `projet`. Vous devez obtenir la fenêtre suivante avant de cliquez sur le bouton « Enregistrer » :



La fenêtre de Dev-C++ a changé. Le projet a été créé ainsi qu'un fichier source par défaut que nous allons compléter.

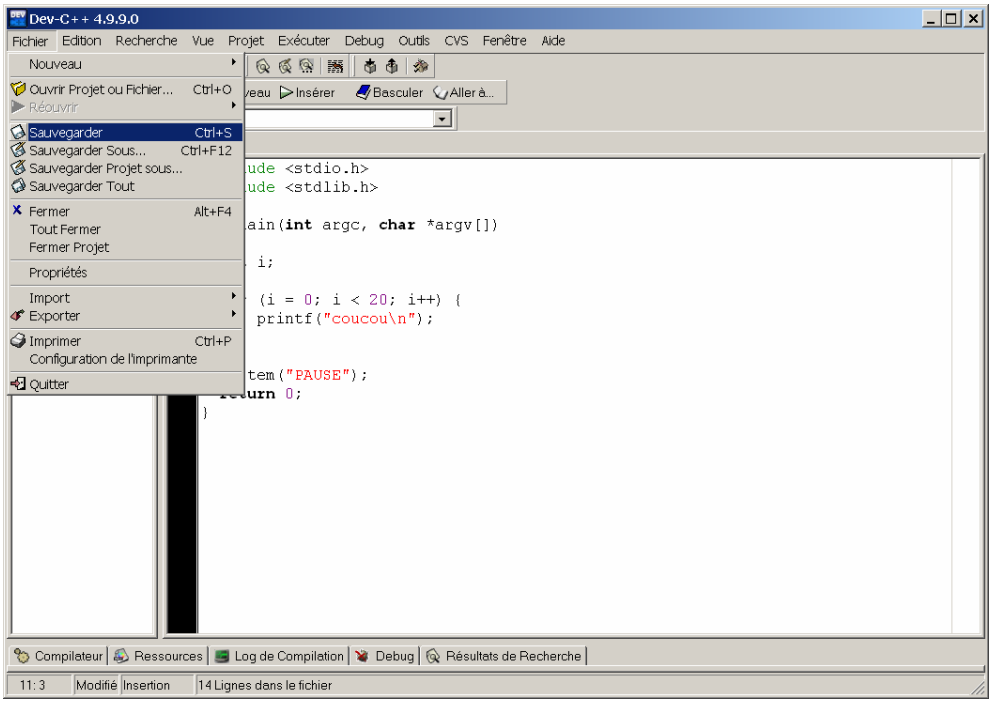


Dans la fenêtre d'édition appelée « main.c », saisissez le texte suivant :

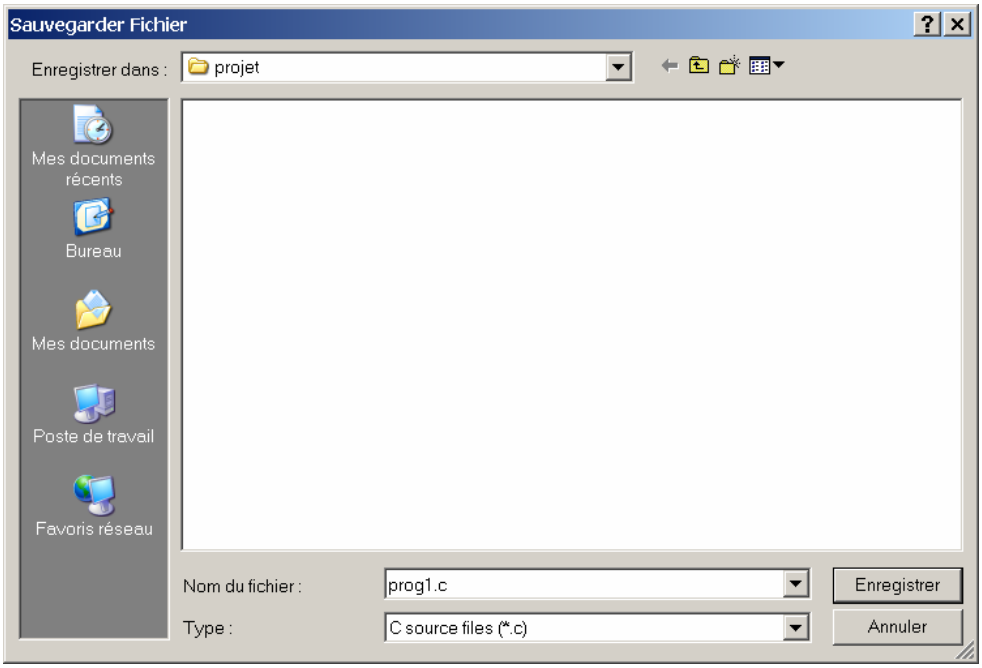


La déclaration de la fonction `main()` est plus compliquée que celle que vous connaissez, mais laissons cela de coté pour l'instant. La ligne `system("pause");` permet de maintenir la fenêtre de commande ouverte quand votre programme s'arrête. Si vous ne la

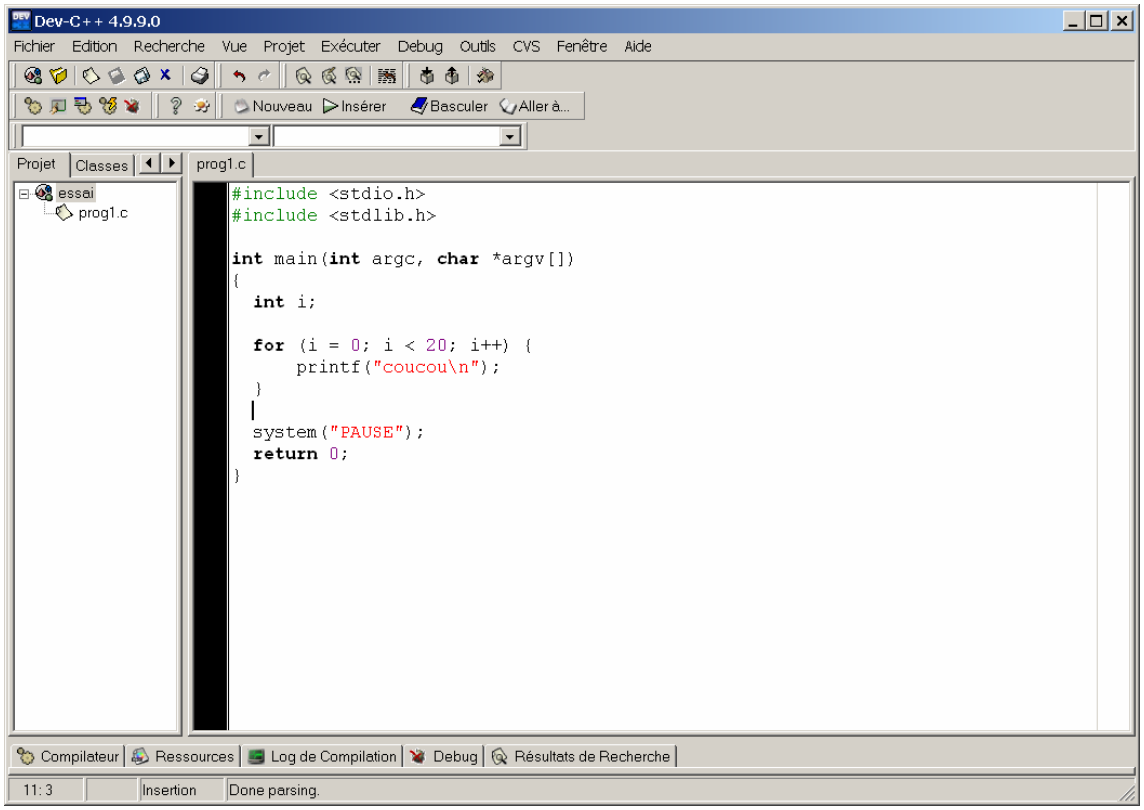
mettez pas, la fenêtre se referme sans que l'on ait pu voir la fin du programme. Enregistrons maintenant ce fichier. Cliquez sur le menu « Fichier » puis sur le sous-menu « Sauvegarder » :



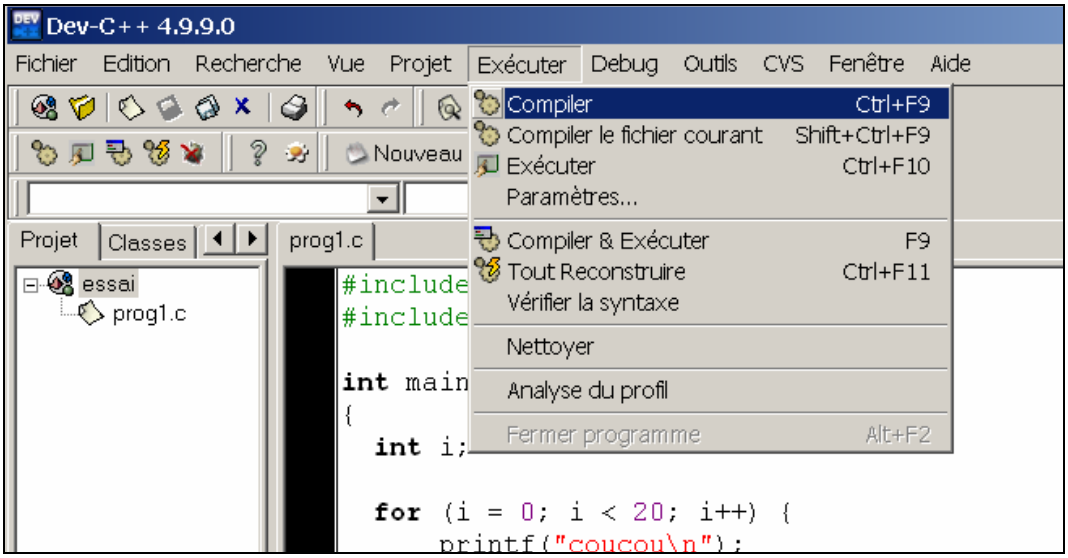
Dans la fenêtre qui apparaît, tapez le nom `prog1.c`. Vous **devez** obtenir cette fenêtre avant de cliquer sur le bouton « Enregistrer » :



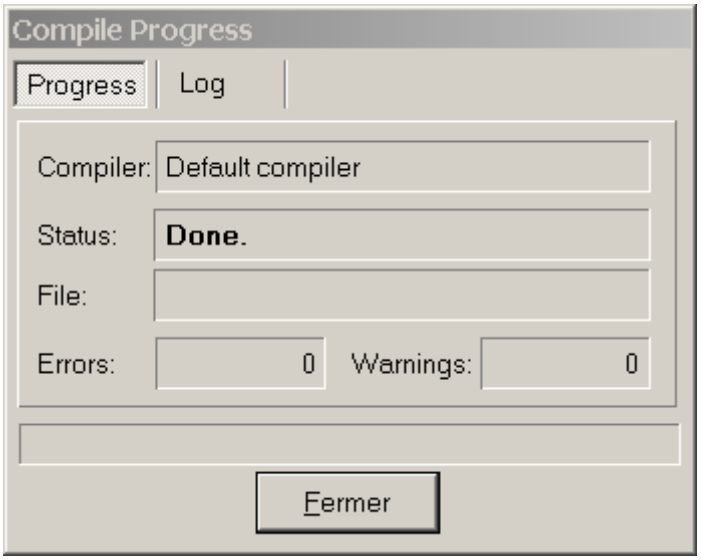
La fenêtre d'édition a maintenant pris le nom du fichier `prog1.c`.



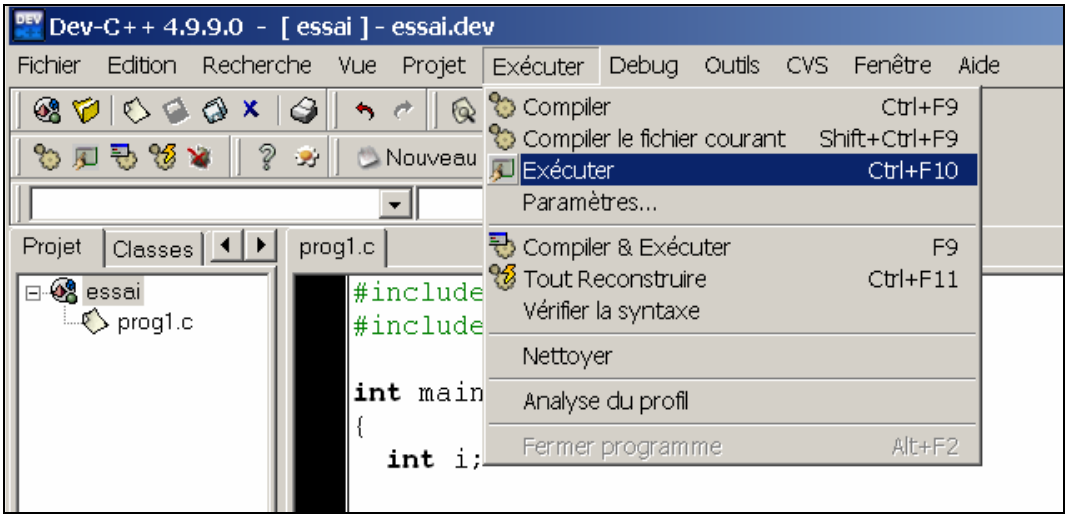
Pour compiler ce programme, cliquez sur le menu « Exécuter » puis sur le sous-menu « Compiler » :



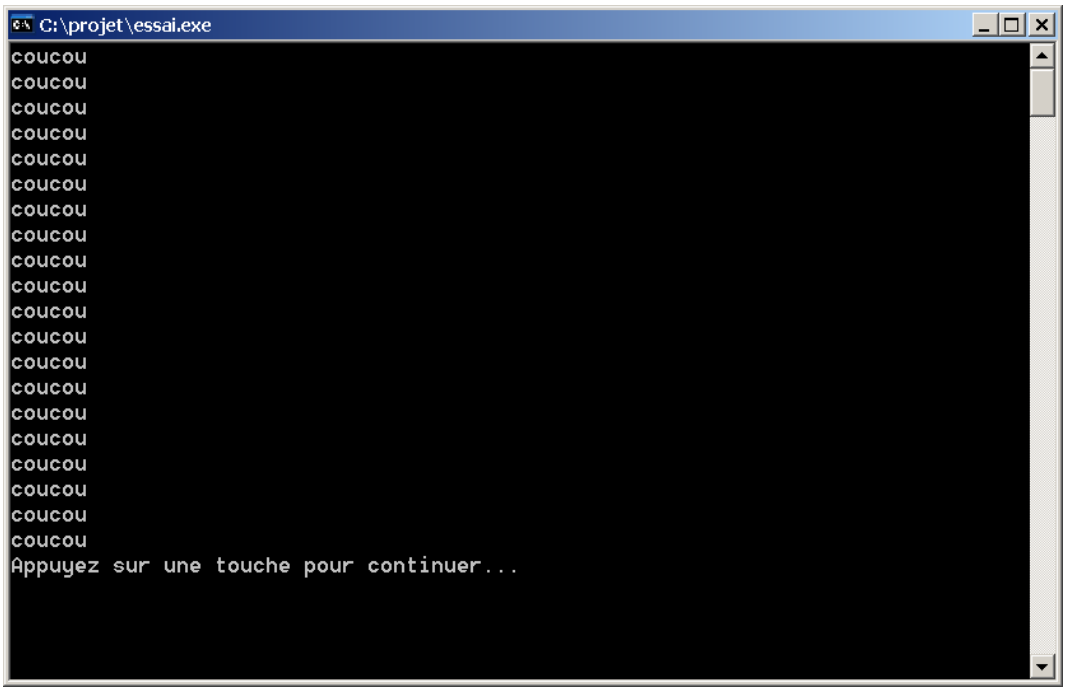
Une petite fenêtre (la fenêtre de compilation) apparaît à l'écran qui vous informe sur déroulement de la compilation. A l'issue de la compilation, vous devez obtenir la fenêtre suivante (avec 0 erreur et 0 warning). Nous verrons plus loin comment corriger les erreurs de compilation. Cliquez sur le bouton « Fermer » :



Pour lancer le programme, cliquez sur le menu « Exécuter » puis sur le sous-menu « Exécuter » :



La fenêtre de commande suivante apparaît à l'écran :



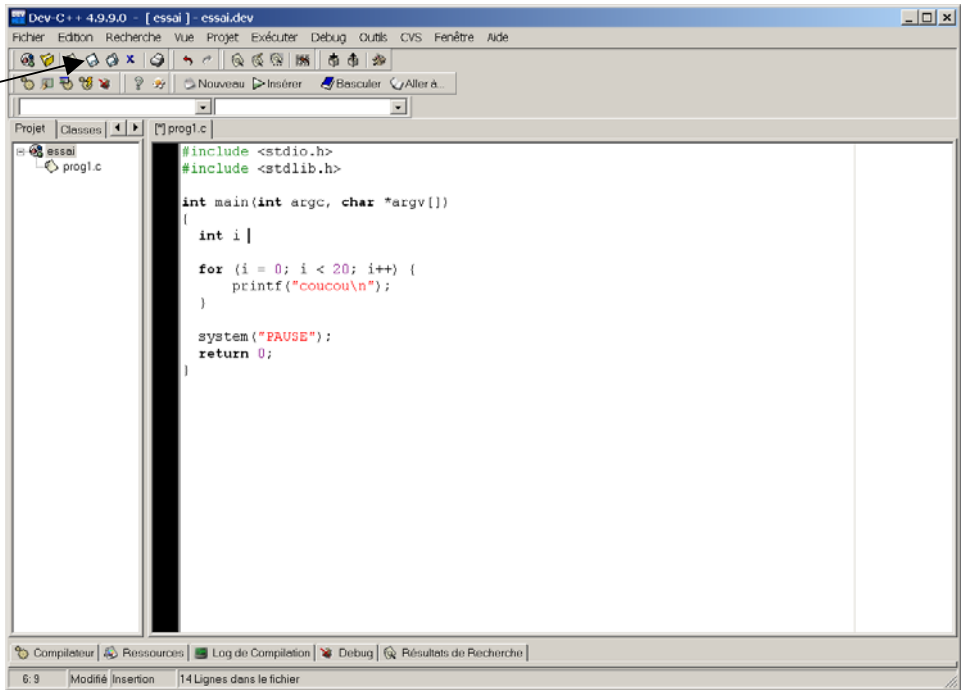
Appuyez sur une touche du clavier pour fermer cette fenêtre. Voyons maintenant ce qu'il se passe en cas d'erreur de compilation. Pour cela, nous allons créer une erreur de syntaxe en supprimant le point virgule à la fin de la ligne de déclaration de la variable i :

[*] prog1.c

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

int main(int argc, char *argv[])
{
  int i

  for (i = 0; i < 20; i++) {
      printf("coucou\n");
  }

  system("PAUSE");
  return 0;
}
```

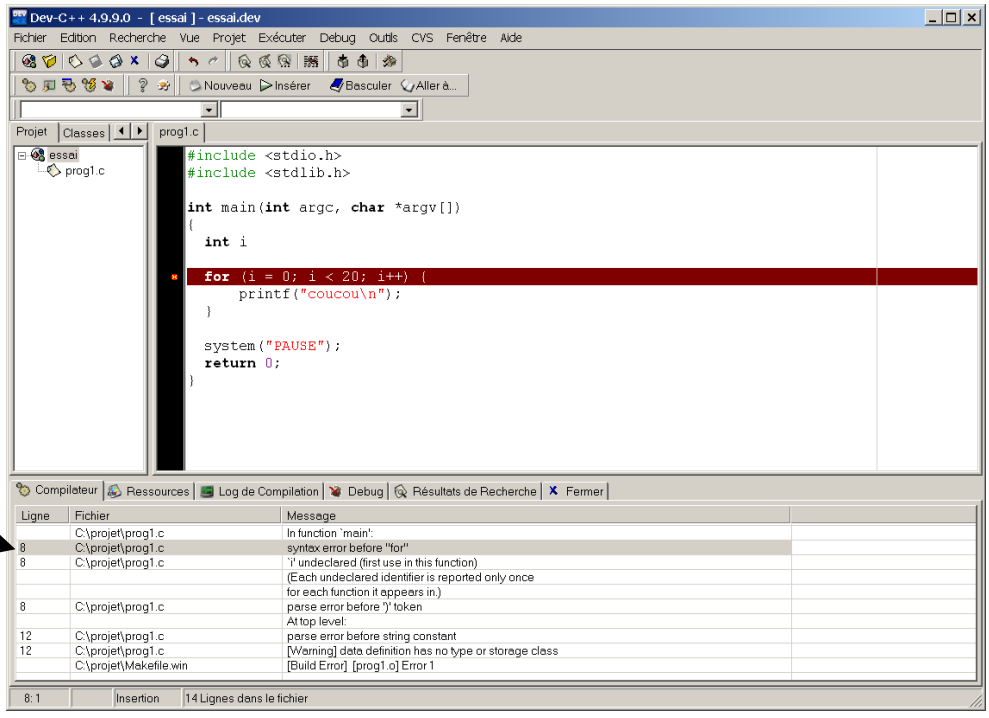
N'oubliez pas d'enregistrer le fichier en cliquant sur la disquette  dans la barre d'outils.

Enregistrement fichier

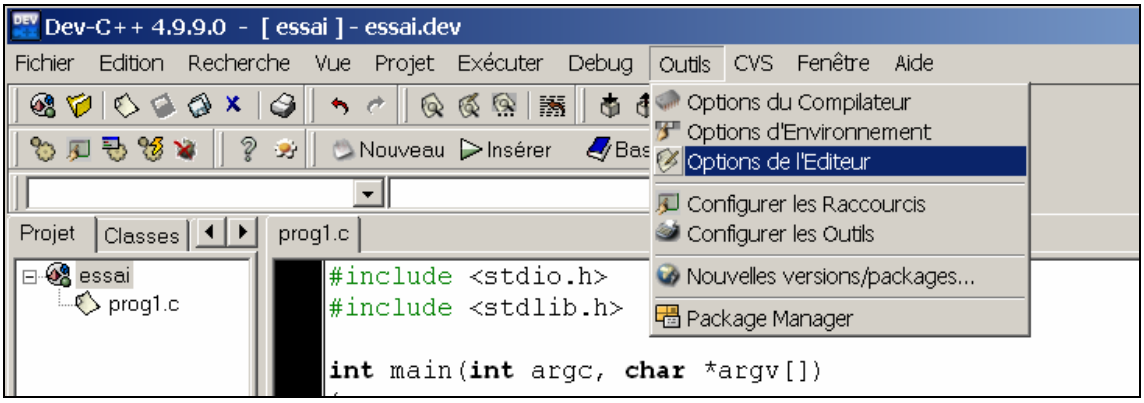


Relancez la compilation. Le compilateur voit maintenant plusieurs erreurs de compilation. La fenêtre inférieure de Dev-C++ vous indique la liste des erreurs. La ligne correspondant à la première erreur est surlignée en rouge dans l'éditeur. D'une manière générale, pour identifier la ligne la plus proche de l'erreur, il suffit de double-cliquer sur le message dans la fenêtre inférieur et la ligne apparaît surlignée en rouge dans l'éditeur.

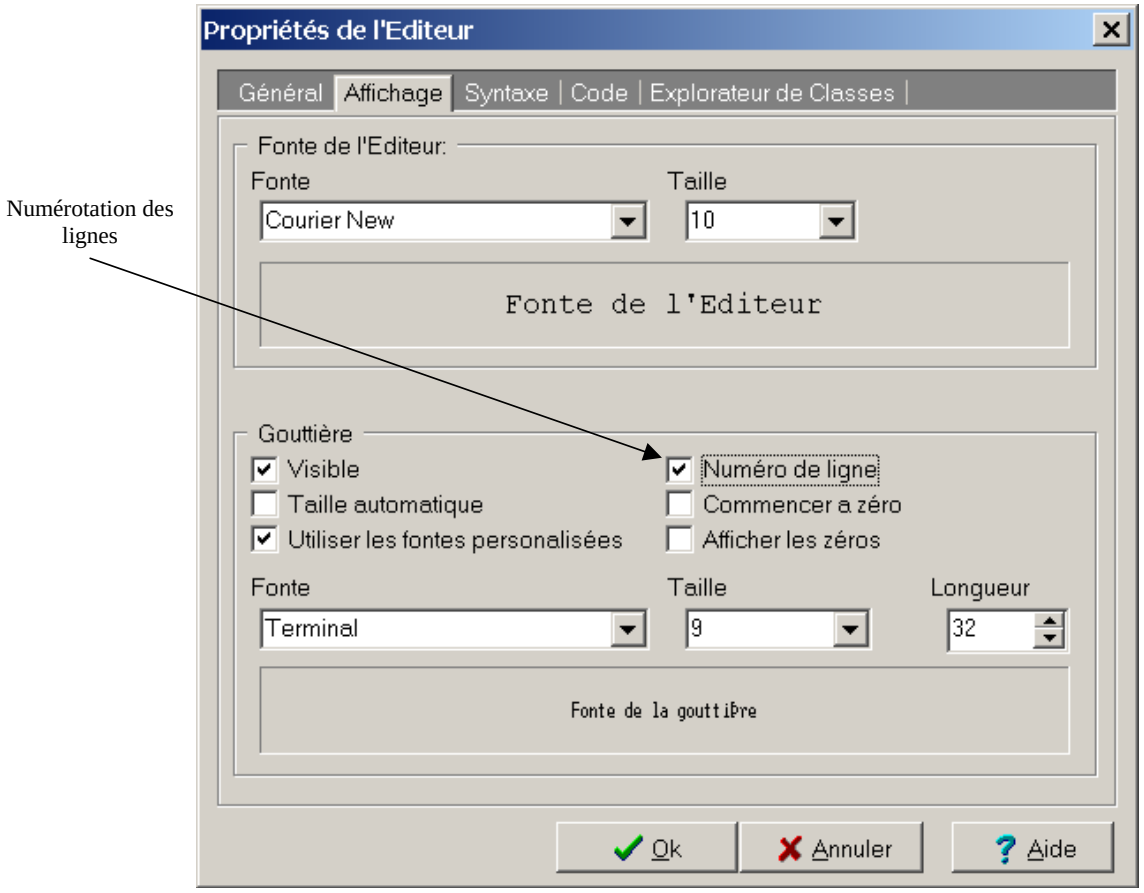
Premier message d'erreur



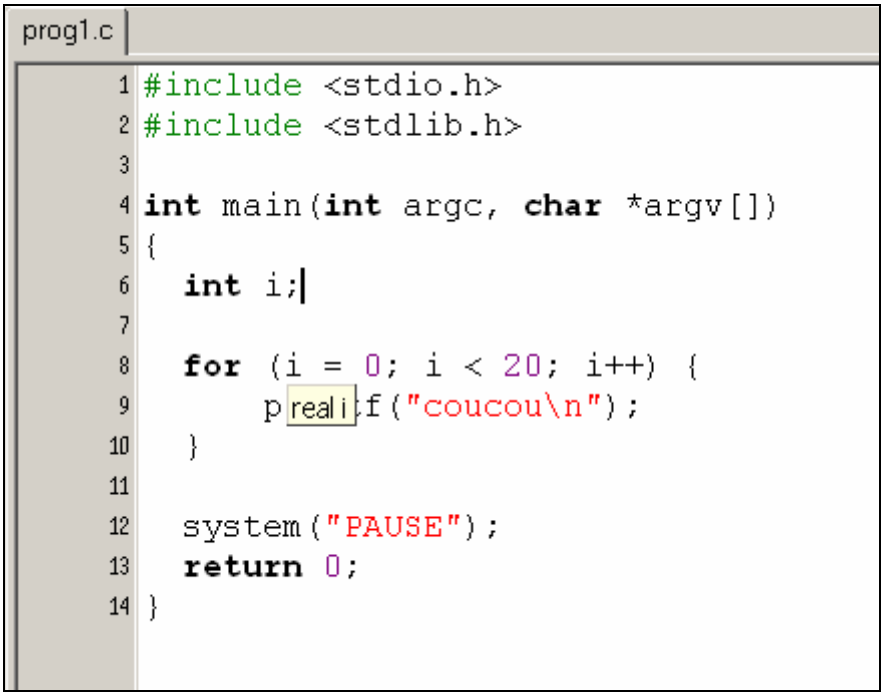
Dans notre exemple, le message vous indique qu'il y a une erreur de syntaxe avant le for. Rajoutez le point-virgule manquant sur la ligne précédente, enregistrer le fichier puis recompilez le programme. Il y a une astuce pratique pour corriger plus facilement les erreurs de compilation, c'est de demander à l'éditeur de numéroter les lignes de votre programme. Pour cela, cliquez sur le menu « Outils » puis sur le sous-menu « Options de l'éditeur » :



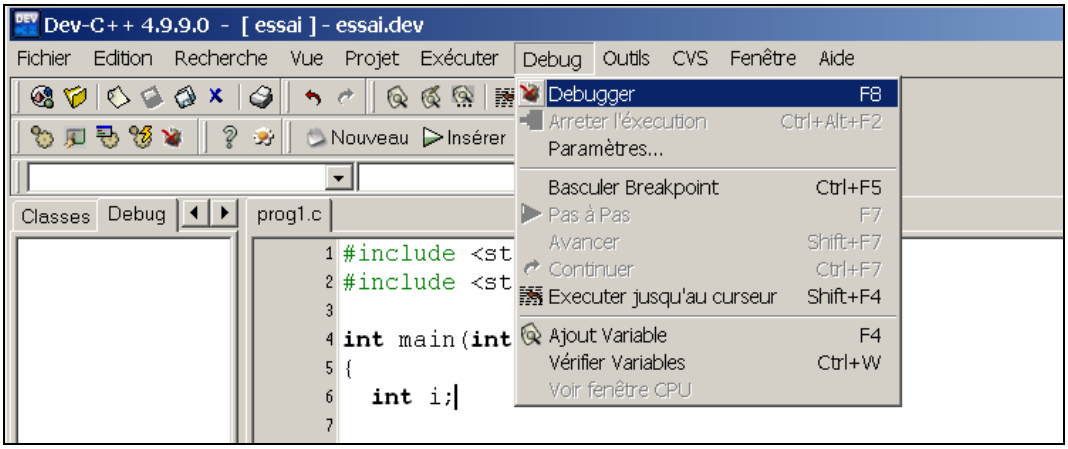
Dans la fenêtre qui apparaît, cliquez sur l'onglet « Affichage », puis cochez la case « Numéro de ligne » :



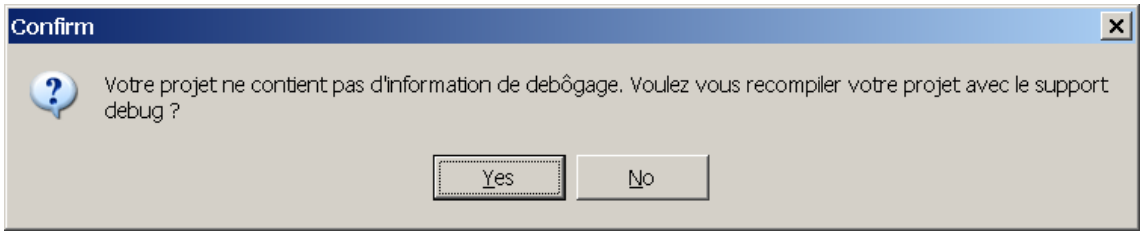
Cliquez sur le bouton « OK ». La fenêtre se ferme et des numéros de ligne apparaissent maintenant à gauche de la fenêtre d'édition :



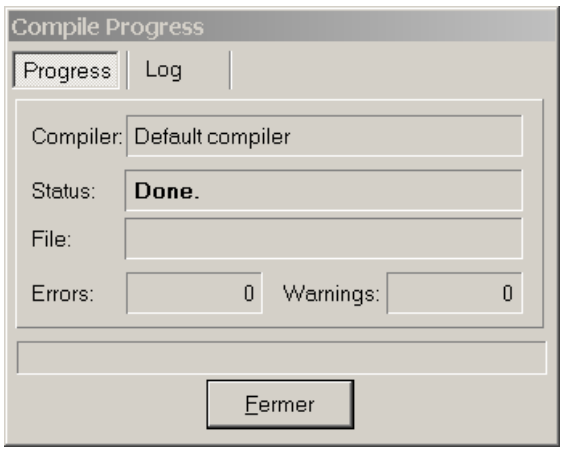
Après avoir corrigé vos erreurs de compilation, il se peut que votre programme ne fonctionne pas comme vous le souhaitez. Pour corriger ces erreurs de fonctionnement, vous pouvez utiliser le debugger. cliquez sur le menu « Debug » puis sur le sous-menu « Debugger » (raccourci clavier, touche F8):



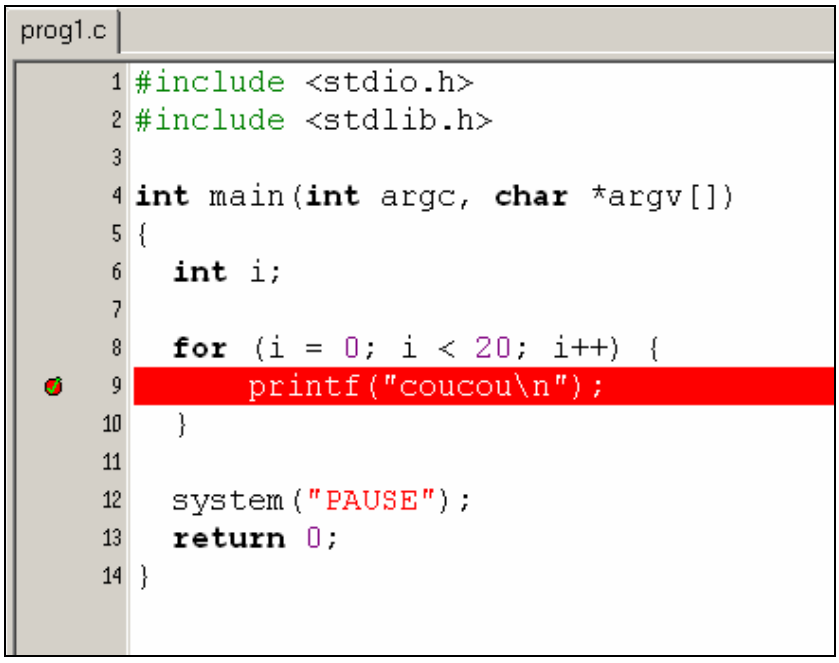
La fenêtre suivante vous indique que votre projet ne contient pas les informations nécessaires au debug de votre programme. Cliquez sur le bouton « Yes » pour recompiler le projet avec les bonnes options :



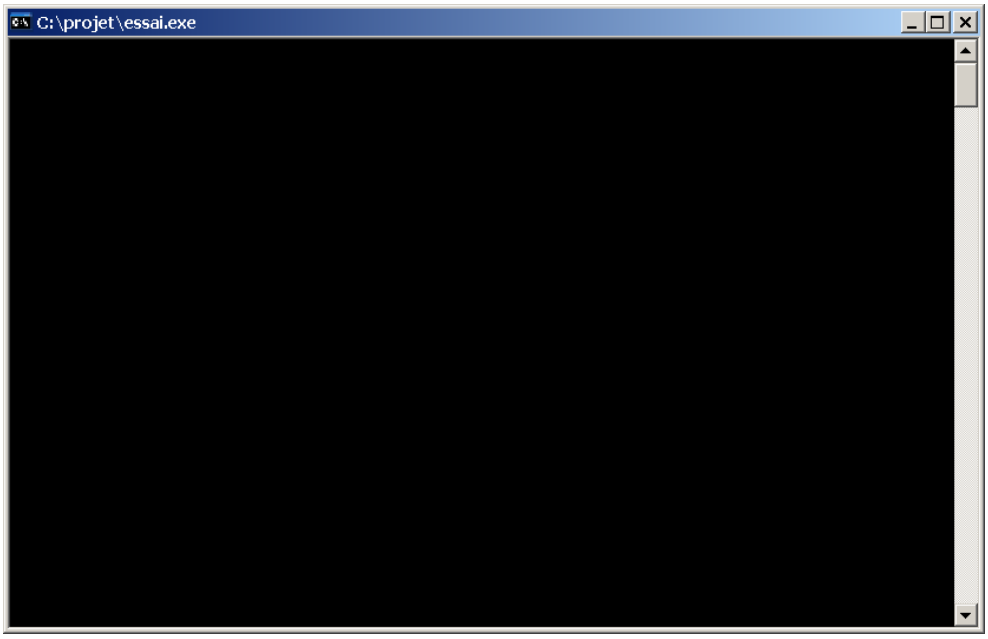
puis fermez la fenêtre de compilation.

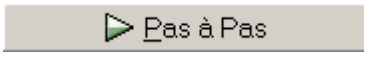


Si vous relancez le debugger (F8), le programme s'exécute comme avant. Quel est donc l'intérêt du mode debug ? Pour le comprendre, il faut placer un point d'arrêt dans le programme. Pour cela, cliquez dans la fenêtre d'édition à gauche du numéro de ligne 9. Un point d'arrêt (breakpoint en anglais) apparaît sur la ligne 9 qui devient rouge :

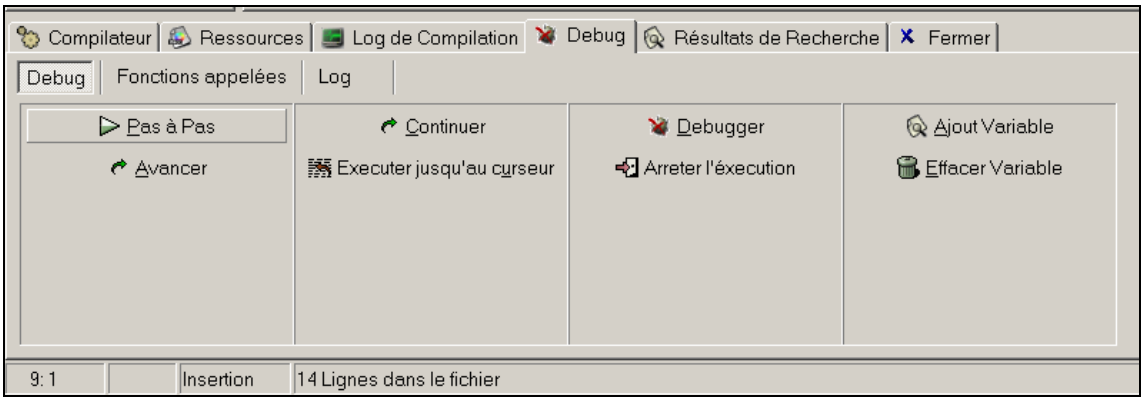


Relancez le debugger (F8). Le programme est arrêté sur la ligne 9. La ligne qui va être exécutée est maintenant bleue et la fenêtre de commande nommée `essai` où va s'afficher la sortie du programme est ouverte à l'écran :

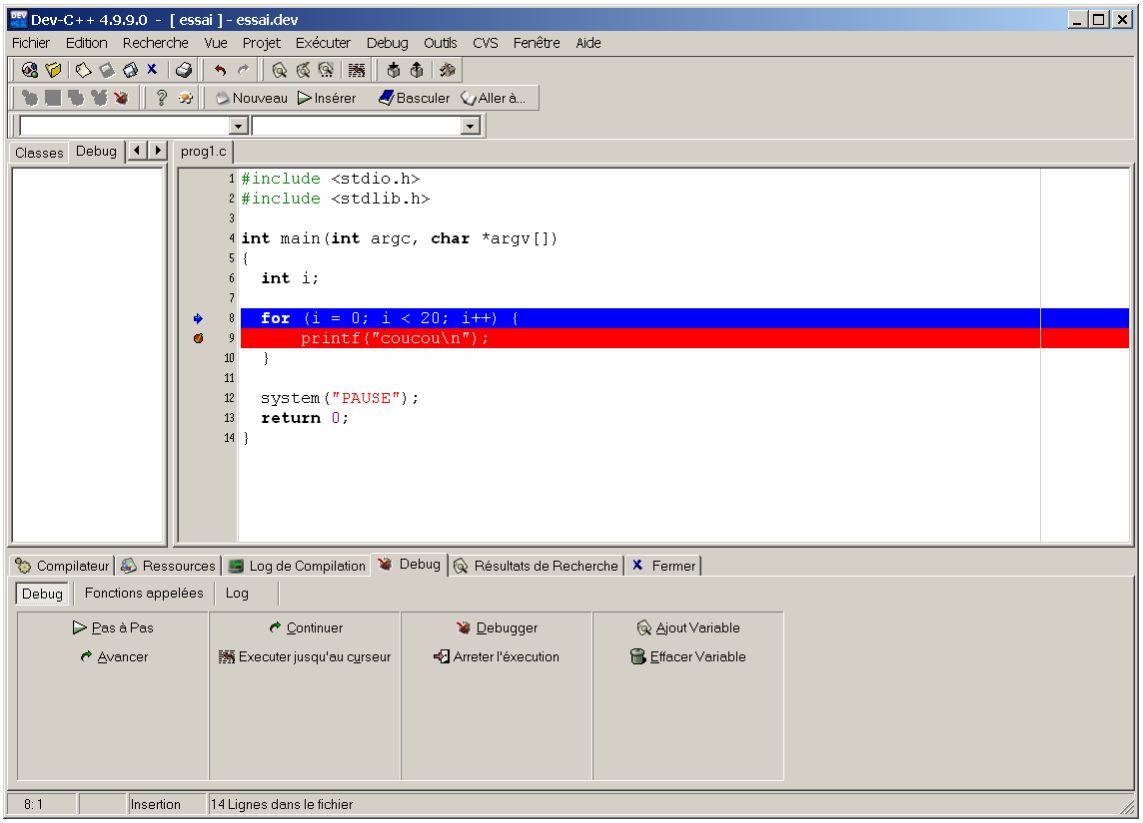


Vous pouvez maintenant avancer dans le programme ligne par ligne. On appelle cela le mode pas à pas. Il suffit pour cela de cliquer sur l'icône « Pas à Pas » 

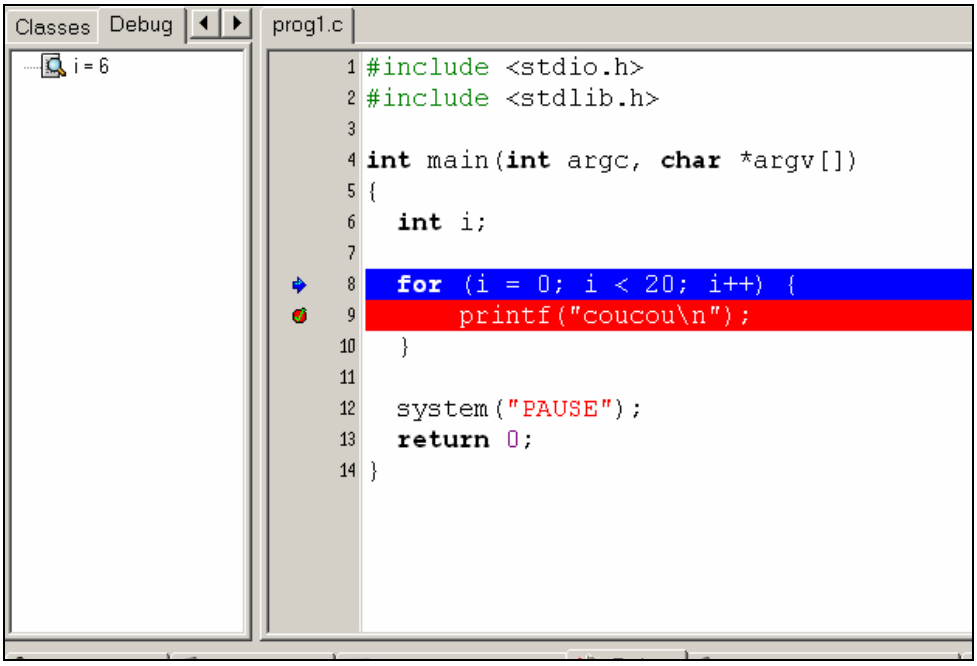
dans la partie inférieure de Dev-C++ :



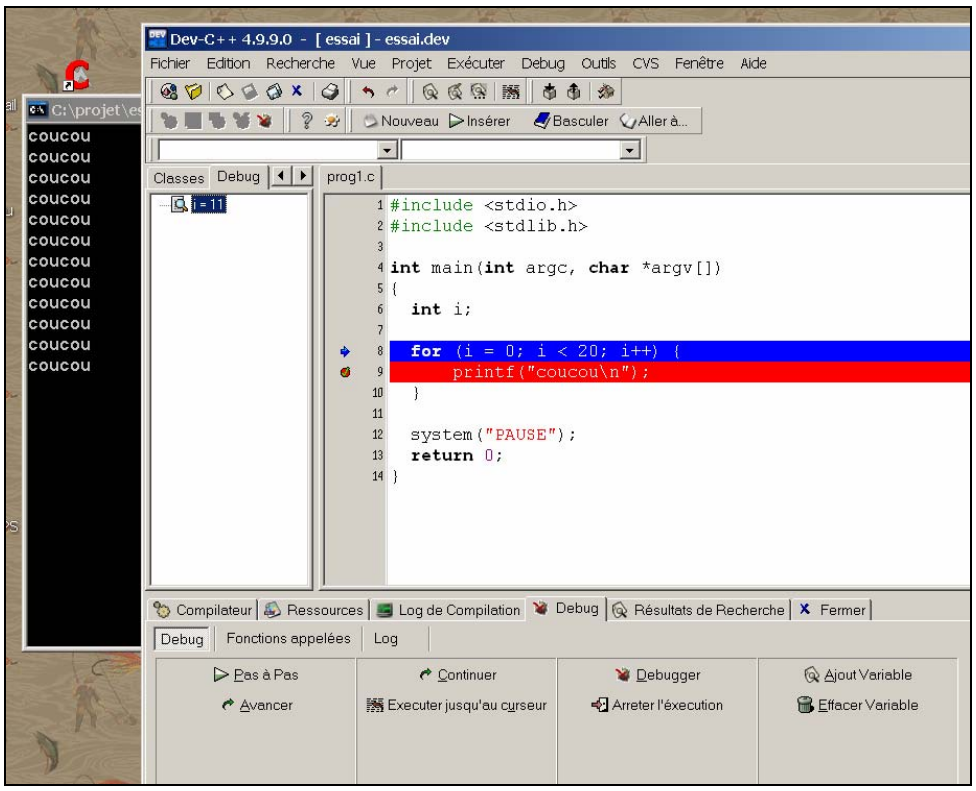
A chaque fois que vous cliquez sur « Pas à Pas », le programme exécute une ligne supplémentaire, ce qui fait avancer la ligne bleue.



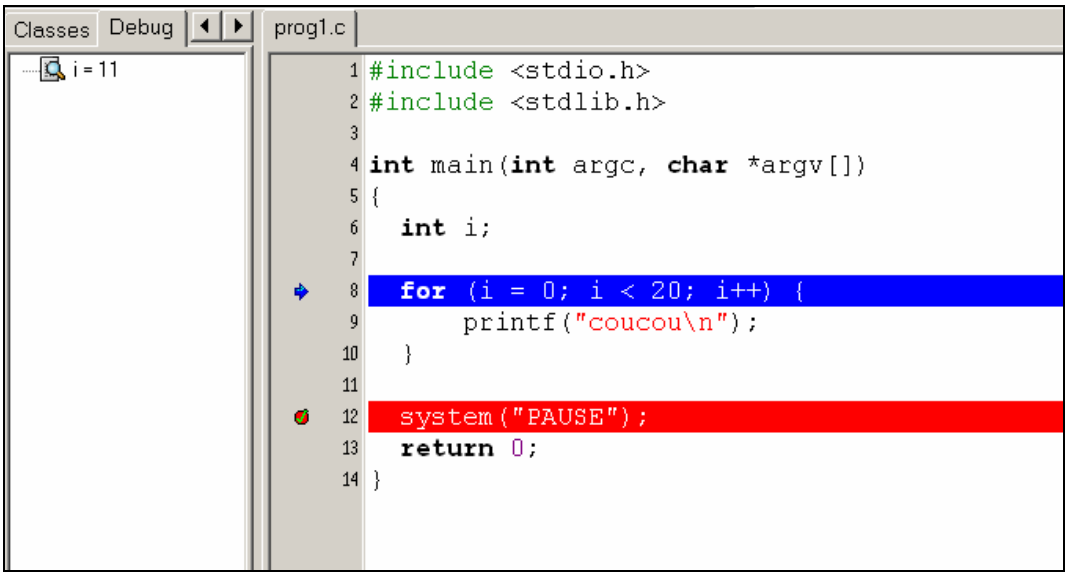
Vous pouvez inspecter (to watch en anglais) simplement le contenu d'une variable en plaçant le curseur de la souris sur le nom de cette variable dans le programme. La valeur apparaît alors dans la fenêtre « Debug » à gauche de l'éditeur. Dans notre exemple, il s'agit de la variable `i`.



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

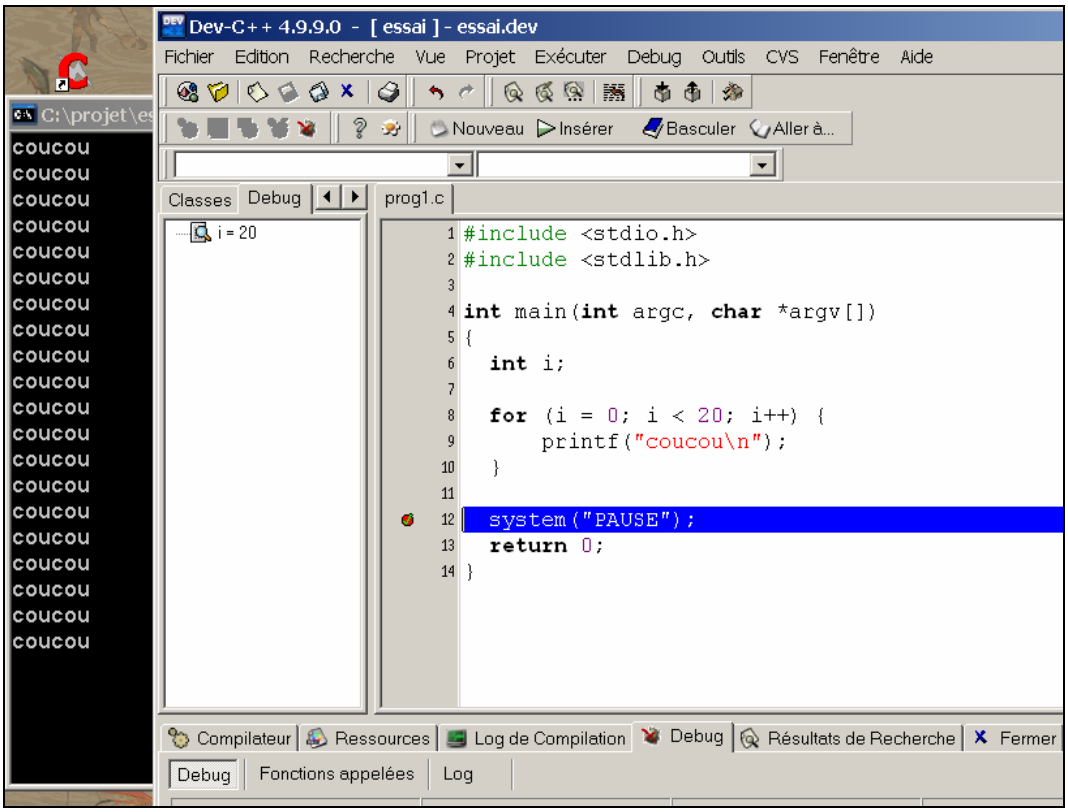


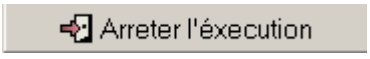
Vous pouvez supprimer le point d'arrêt en cours de debug en cliquant à gauche du numéro de la ligne et en placer un autre sur la ligne 12 :



Vous pouvez ensuite lancer le programme jusqu'au nouveau point d'arrêt en cliquant sur le bouton « Continuer » 

. Le programme a fini la boucle `for`. `i` est égal à 20 et 20 lignes ont été affichées dans la fenêtre de commande.

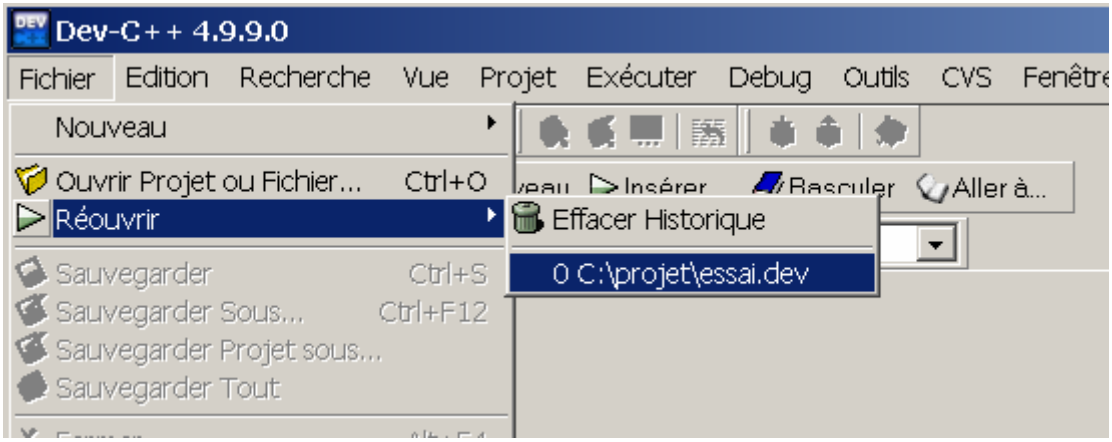


Pour arrêter le debugger, cliquez sur le bouton « Arrêter l'exécution » 

. La fenêtre de commande se ferme. Fermez Dev-C++.

3**) Deuxième programme**

Ouvrez Dev-C++. Vous pouvez ouvrir à nouveau le projet `essai` en cliquant sur le menu « Fichier » puis sur le sous-menu « Reouvrir ». Cliquez alors sur la ligne « c:\projet\essai.dev » qui correspond au projet que nous avons créé précédemment :

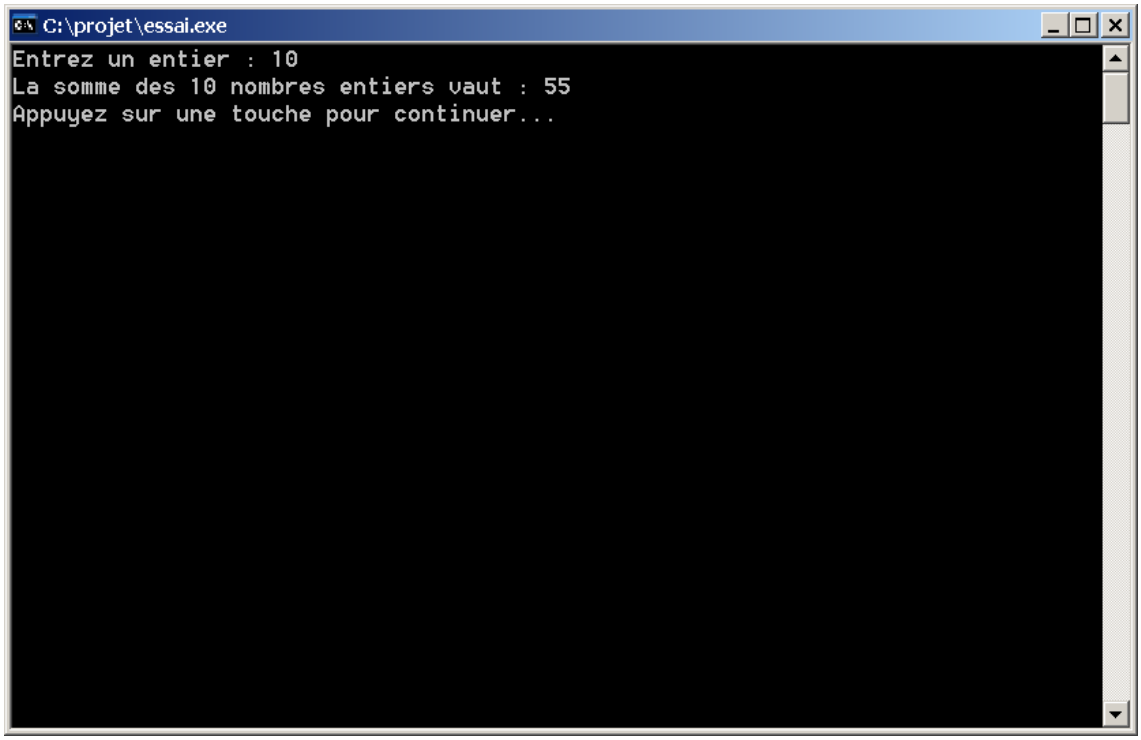


Le projet précédent s'ouvre dans Dev-C++. Modifiez `prog1.c` pour obtenir le programme suivant :

prog1.c

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

int main(int argc, char *argv[])
{
  int i, somme, N;

  printf("Entrez un entier : ");
  scanf("%d", &N);

  i = 1;
  somme = 0;

  while (i <= N) {
      somme = somme + i;
      i++;
  }

  printf("La somme des %d nombres entiers vaut : %d\n", N, somme);

  system("PAUSE");
  return 0;
}
```

Compilez-le puis exécutez-le (F9) :



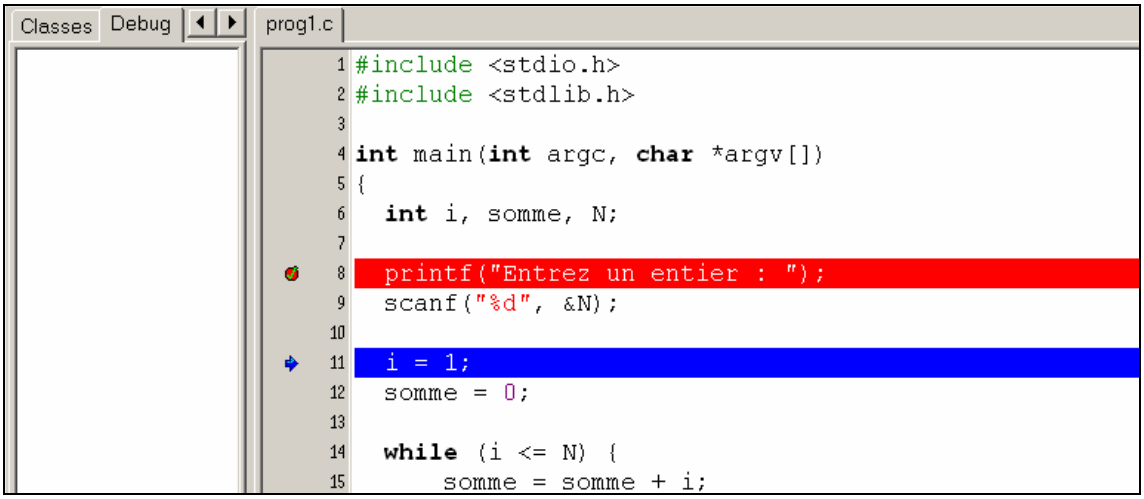
Tapez sur une touche du clavier pour fermer la fenêtre. Placez un point d'arrêt sur la ligne 8 :

prog1.c

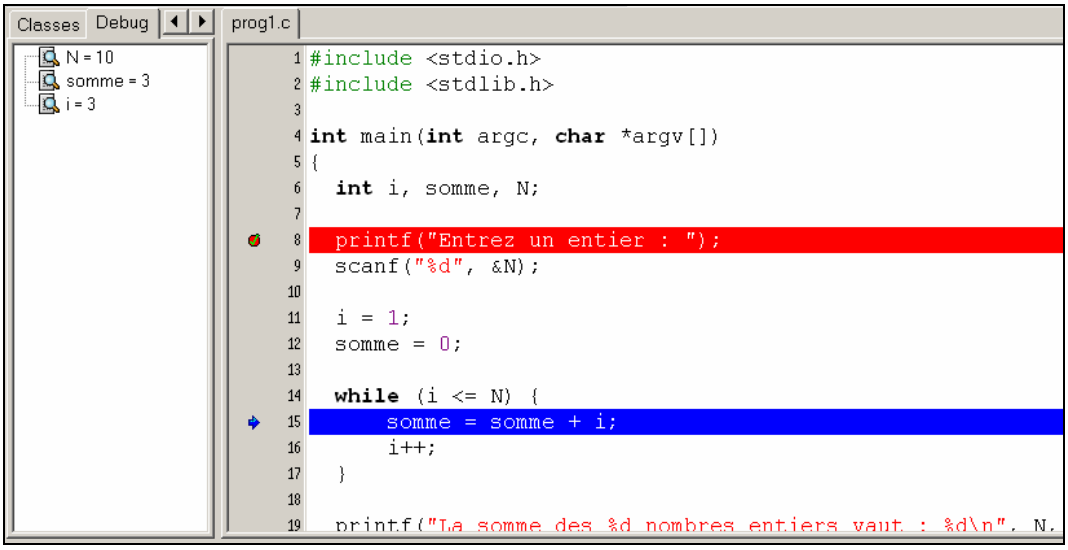
```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(int argc, char *argv[])
5  {
6    int i, somme, N;
7
8    printf("Entrez un entier : ");
9    scanf("%d", &N);
10
11   i = 1;
12   somme = 0;
13
14   while (i <= N) {
15       somme = somme + i;
16       i++;
17   }
18
19   printf("La somme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

puis lancez le debugger (F8). Après recompilation, le debugger s'arrête sur le premier printf du programme. Avancez en mode pas à pas. Quand vous essayez de dépasser la ligne du `scanf`, le debugger cesse d'avancer.

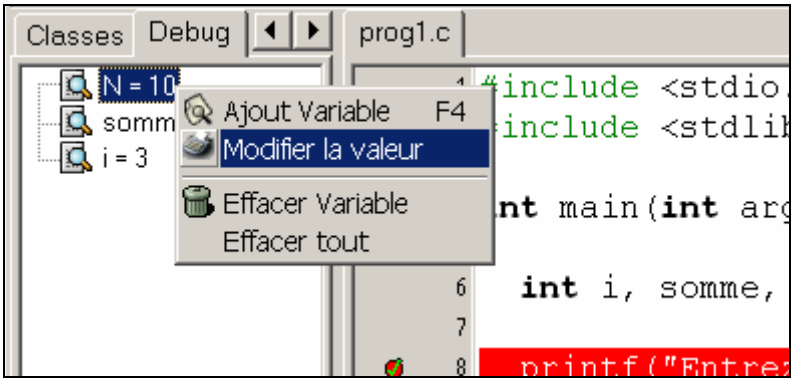
Le `scanf` attend que vous saisissiez une valeur dans la fenêtre de commande. Cette fenêtre doit être active pour que vous puissiez taper une valeur à l'intérieur. Pour rendre une fenêtre active, il faut cliquer dessus de façon à ce que la bande supérieure (le bandeau) de la fenêtre devienne bleue. Cliquez 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.



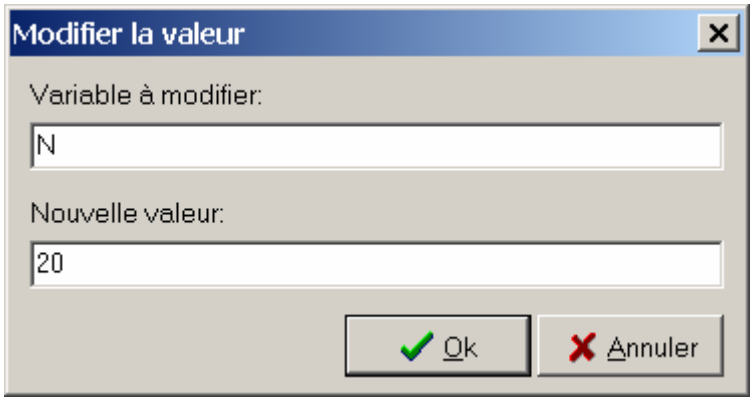
Vous pouvez maintenant examiner les variables `i`, `somme` et `N` en mode pas à pas.



Vous pouvez changer la valeur d'une variable en la sélectionnant, en cliquant avec le bouton droit de la souris puis en cliquant sur le sous-menu « Modifier la valeur » :



Dans la fenêtre qui s'ouvre, tapez une nouvelle valeur puis cliquez sur le bouton « OK » :

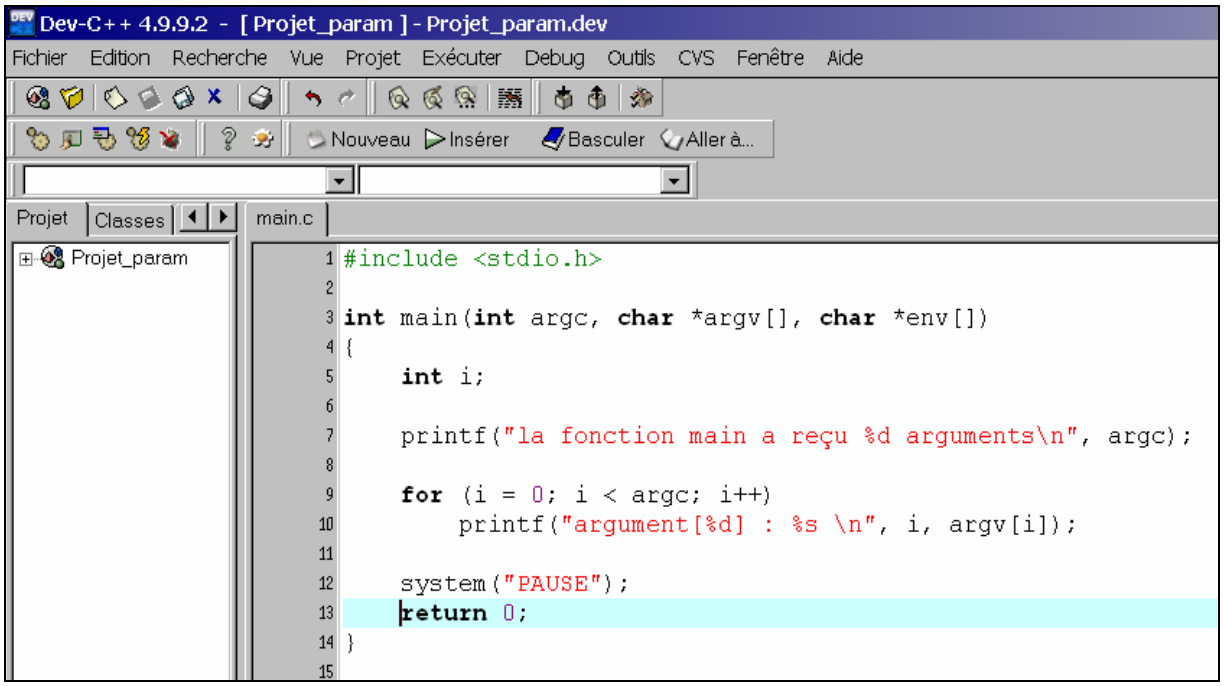


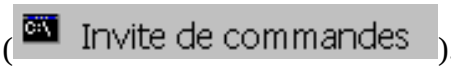
Continuez l'exécution en mode pas à pas. N est maintenant égale à 20.

Vous pouvez maintenant fermer le debugger et Dev-C++. Vous avez tous les éléments nécessaires pour commencer à programmer.

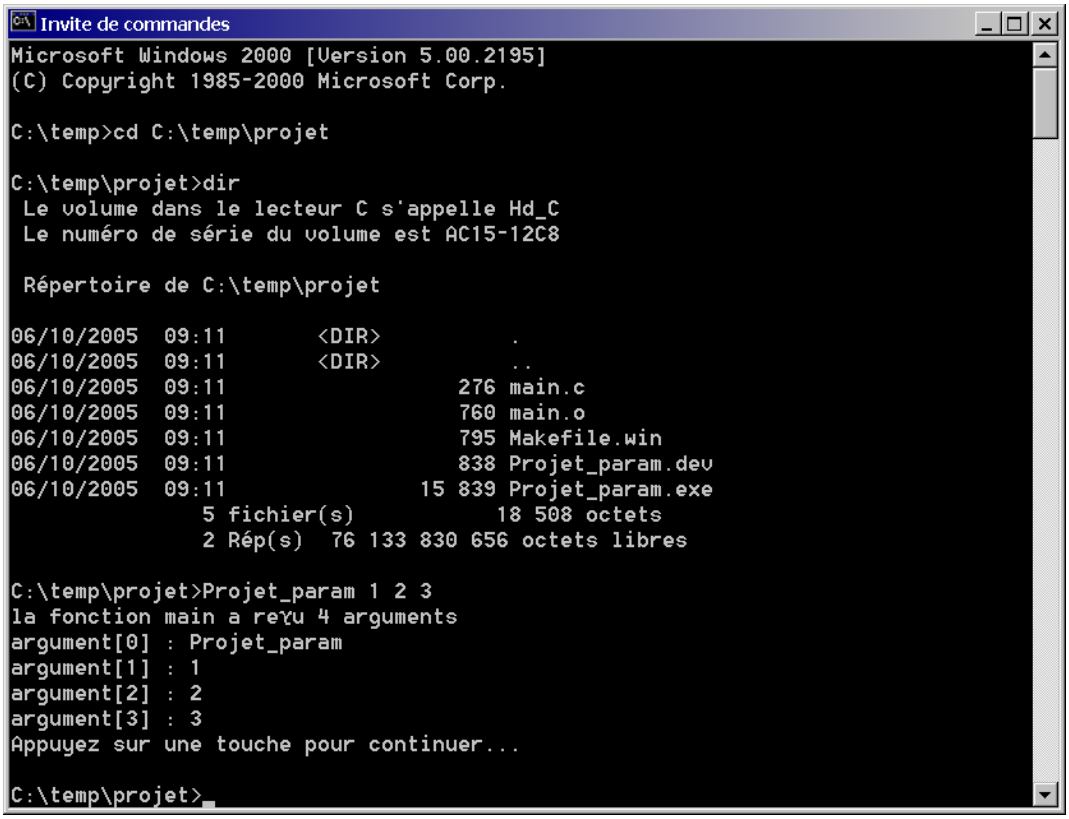
4) troisième programme : passage de paramètres à la fonction main

Pour passer des paramètres à la fonction main, il existe deux méthodes. Créez un nouveau projet et tapez le programme suivant :

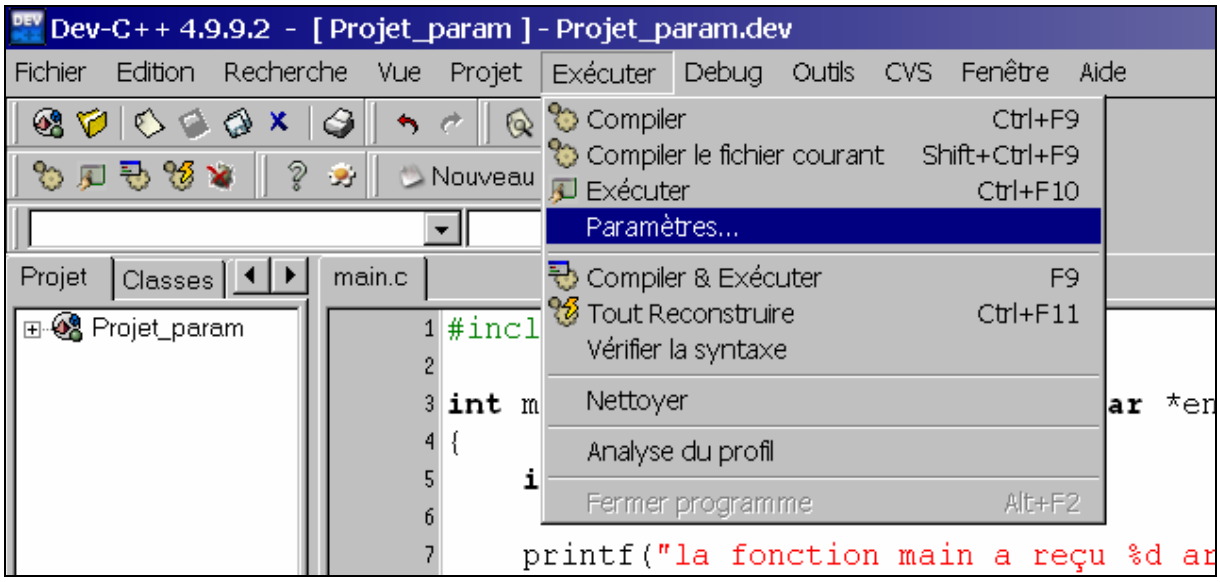


Ce programme accepte des paramètres sur la ligne de commande (voir cours cbim3). Mais comment les faire passer à la fonction main avec Dev-C++ ? La première méthode utilise la console de Windows (

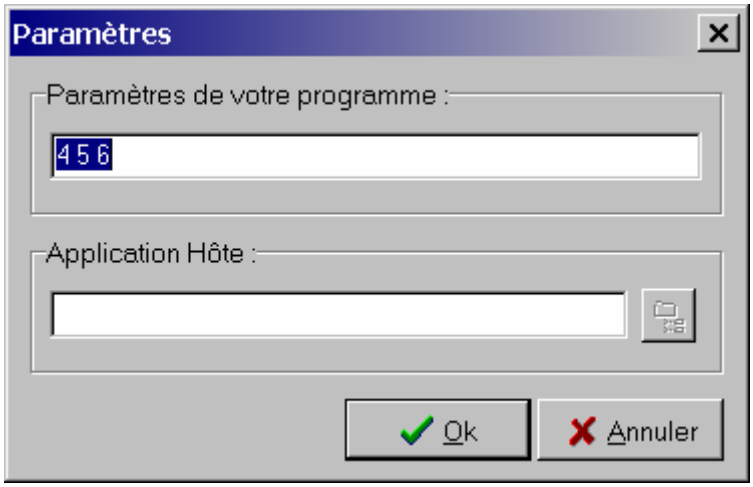
). Il suffit de l'ouvrir, de se placer dans le répertoire du projet (dans cet exemple, `C:\temp\projet`) grâce à la commande `cd c:\temp\projet`, puis de tapez le nom du programme suivi des paramètres :



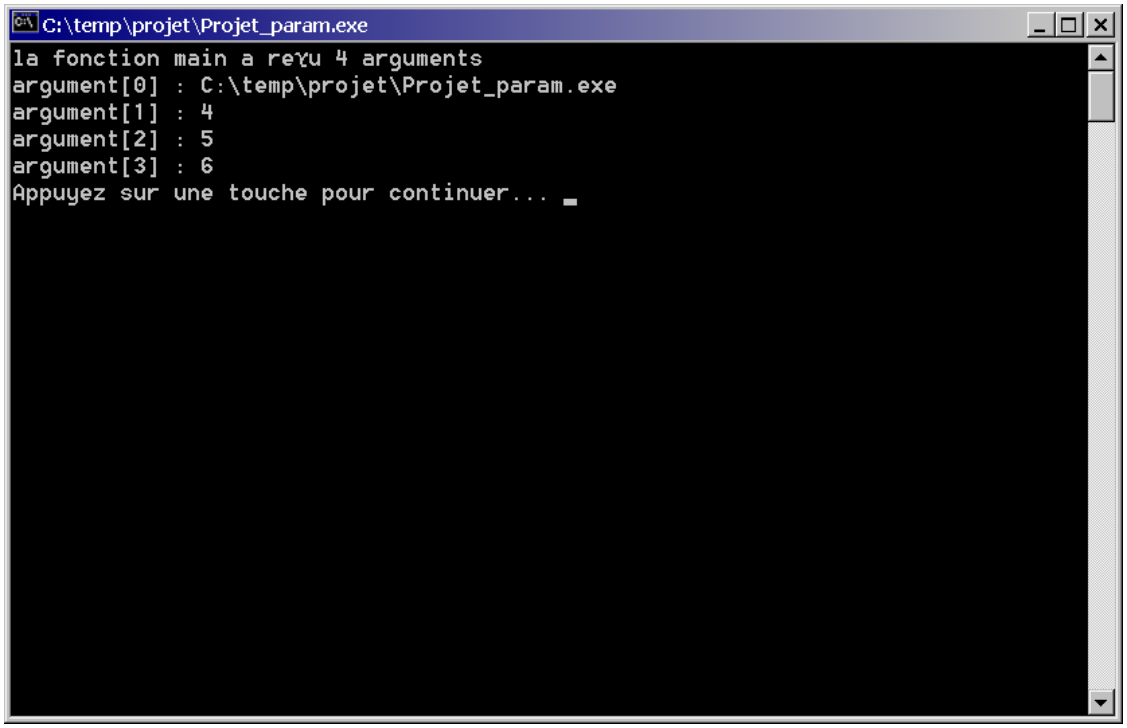
C’est la méthode traditionnelle que l’on utilise sous Linux. Si vous souhaitez passer les paramètres sans utiliser la console en restant sous Dev-C++, il faut aller dans le menu « Exécuter », cliquer sur « Paramètres » :



taper les paramètres de la fonction main puis cliquer sur « Ok » :



et enfin lancer le programme :



Cette méthode fonctionne aussi avec le debugger :