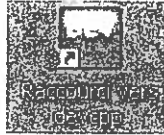


Ouvrir une session de nom "maths" et de mot de passe "Smaths\$"

I Présentation du système Dev-C++

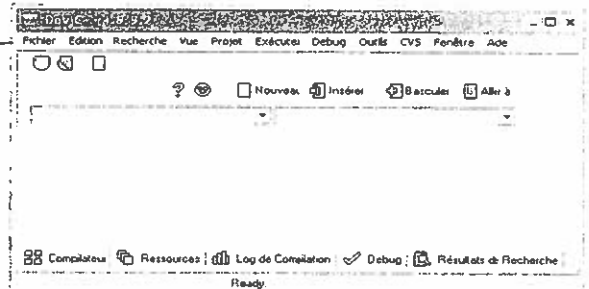
1) Lancement de Dev-C++

"Double-cliquer" sur l'icône :



, la fenêtre de Dev-C++ apparaît :

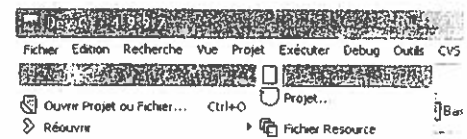
La barre des menus :



2) Menus importants :

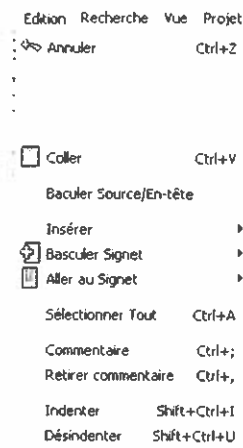
Menu Fichier :

- Ouvrir Projet ou Fichier : ouverture d'un sous-Menu
- Fichier Source.
- Sauvegarder: copier sur le disque le contenu de la fenêtre de texte active (mise à jour du fichier)
- Sauvegarder Sous: copier le texte de la fenêtre active dans un nouveau fichier.
- Fermer : fermer la fenêtre active
- Quitter: quitter Dev-C++



Menu Edition :

- Annuler
- Répéter
- Couper
- Copier
- Coller
- Sélectionner Tout



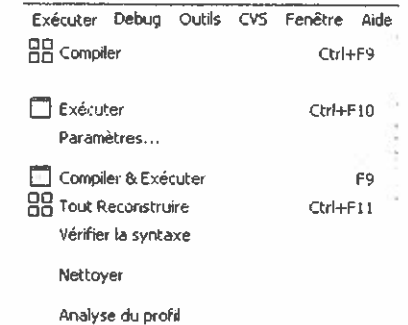
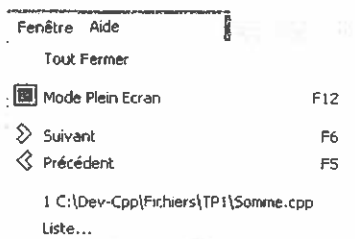
Menu Exécuter :

- Compiler
- Exécuter

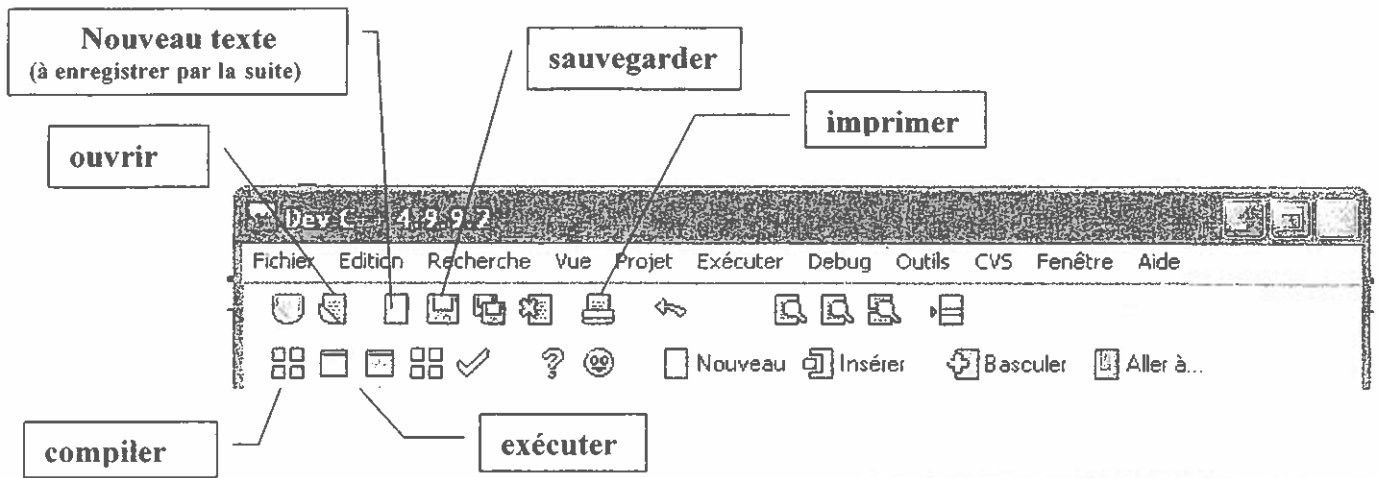


Menu Aide :

Menu Fenêtre :



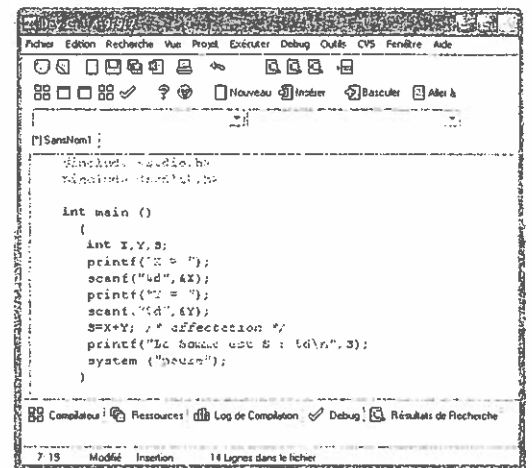
Les raccourcis intéressants (icônes) :



3) Taper le texte du programme suivant :

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

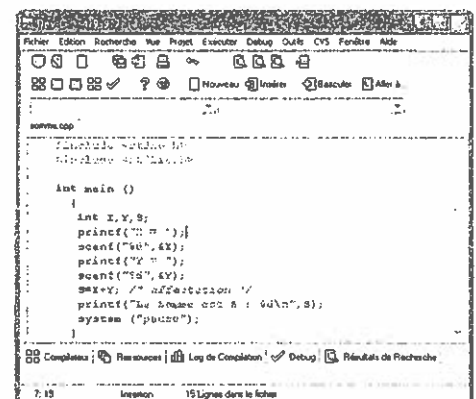
int main ()
{
    int X,Y,S;
    printf("X = ");
    scanf("%d",&X);
    printf("Y = ");
    scanf("%d",&Y);
    S=X+Y; /* affectation */
    printf("La Somme est S : %d\n",S);
    system ("pause");
}
```



3-1) Sauvegarder le texte dans le fichier somme.c

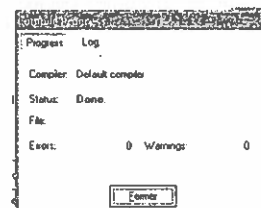
Commande Sauvegarder du menu Fichier :

Taper le nom *somme.c*, le fichier est créé dans le dossier FICHIERS du dossier Dev-C++ .



3-2) Compiler le programme

Le compilateur crée un programme exécutable qui est placé dans le dossier FICHIERS (en fait on réalise 2 étapes : la compilation et l'édition de liens).



3-3) Exécuter le programme

La fenêtre d'exécution s'ouvre, taper les valeurs demandées .
Les entrées et les sorties réalisées au cours de l'exécution du programme apparaissent dans la fenêtre



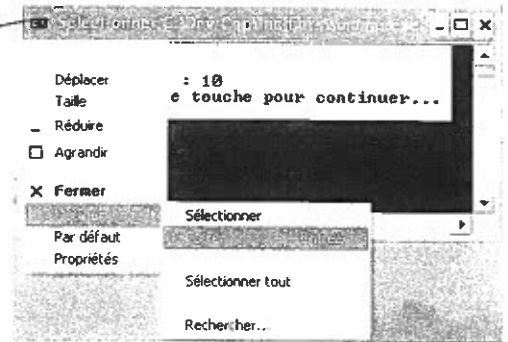
3-4) La fenêtre d'exécution

bouton

Un bouton de menu permet de récupérer le contenu de la fenêtre :

Exemple de texte récupéré :

```
X = 3
Y = 7
La Somme est S : 10
Appuyez sur une touche pour continuer...
```

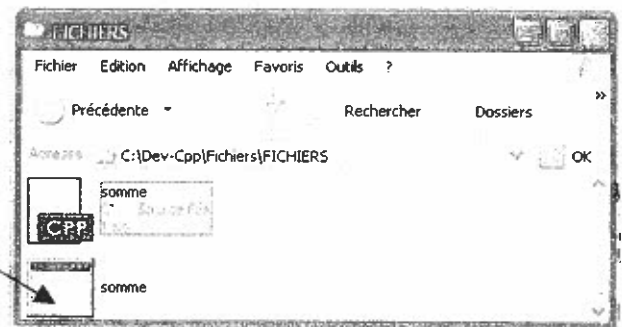


Appuyer sur une touche, la fenêtre se referme.

3-5) Exécution directe du programme (sans utiliser Dev-C++)

Sortir de Dev-C++ et exécuter le programme en "double-cliquant" sur l'icône de somme.exe. du dossier Fichiers (le dossier Fichiers se trouve dans le dossier Dev-Cpp situé à la racine de l'unité C :

(C:\Dev-Cpp\Fichiers\))



II Quelques programmes simples sur des nombres particuliers

Définitions :

- Un nombre entier est dit "**premier**" s'il est différent de 1, et s'il n'admet comme diviseurs que 1 et lui même.
- Un nombre entier est dit "**parfait**" s'il est égal à la somme de ses diviseurs autres que lui même.
- Un nombre entier est dit "**doublon**" si le produit de ses diviseurs est un multiple de la somme de ses diviseurs.

Remarques : Tout entier positif n n'admet pas de diviseur dans l'intervalle $]n/2, n[$.

On peut trouver tous les diviseurs d'un entier n en se limitant aux diviseurs $\leq \sqrt{n}$

1) Les nombres premiers :

Écrire l'algorithme puis le programme en langage C qui détermine si un nombre tapé au clavier est premier.

2) Les nombres parfaits :

Écrire l'algorithme puis le programme en langage C qui détermine si un nombre tapé au clavier est parfait.

3) Les nombres doublons :

Écrire l'algorithme puis le programme en langage C qui détermine si un nombre tapé au clavier est un doublon.

III Fonctions sur des nombres particuliers

- a) Écrire une fonction qui détermine si un nombre est premier ou pas.
- b) Écrire un programme C qui affiche les k premiers nombres premiers. La valeur k sera fournie par l'utilisateur au clavier.

- a) Écrire une fonction qui détermine si un nombre est parfait ou pas.
- b) Écrire un programme C qui affiche tous les nombres parfaits compris entre 2 limites fournies par l'utilisateur au clavier.

- a) Écrire une fonction C qui détermine si un nombre donné est un doublon ou pas.
- b) Écrire un programme C qui affiche tous les doublons compris entre 2 limites fournies par l'utilisateur au clavier.

IV Programme de test

Écrire un programme qui permet de tester les 3 fonctions précédentes .

Le programme principal est une boucle qui, à l'aide d'un menu , propose à l'utilisateur de taper :

- 0 pour arrêter le programme
- 1 pour calculer et afficher les k premiers nombres premiers
- 2 pour calculer et afficher tous les nombres parfaits compris entre 2 limites.
- 3 Pour calculer et afficher tous les doublons compris entre 2 limites.

Exemple d'exécution :

```
MENU
----
0 : arreter
1 : test nombres premiers
2 : test nombres parfaits
3 : test nombres doublons

Votre choix (0|1|2|3) : 1
Recherche des k premiers nombres premiers
k= 10
Les 10 premiers nombres premiers sont :

    2    3    5    7   11   13   17   19   23   29
      MENU
      ----
0 : arreter
1 : test nombres premiers
2 : test nombres parfaits
3 : test nombres doublons

Votre choix (0|1|2|3) : 0
Appuyez sur une touche pour continuer...
```