



ATELIER 1: PREMIER PROGRAMME C

OBJECTIFS :

Prise en main de l'environnement de programmation Dev-C++ et découvrir quelques fonctions de base en langage C.

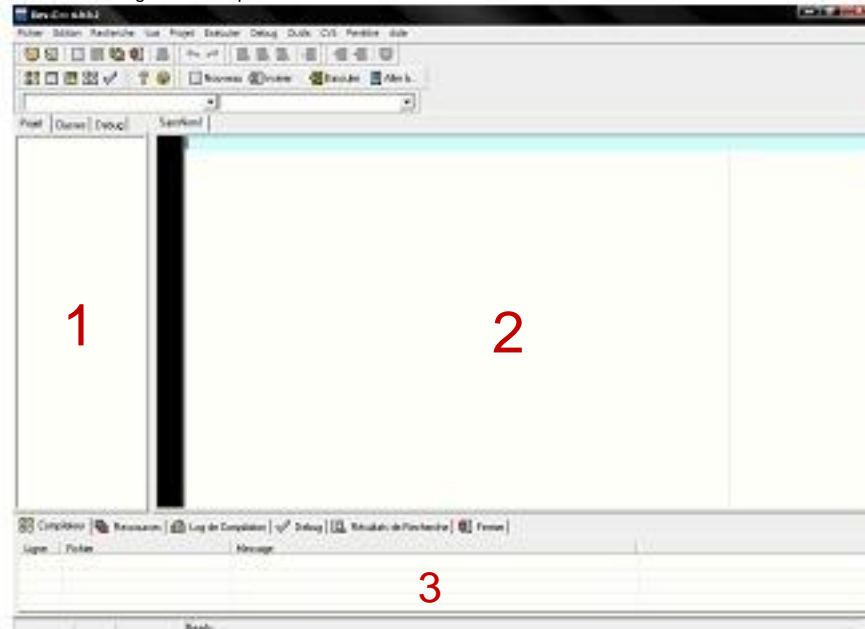
A. Lancement du Dev-C++ :

Le compilateur qu'on va utiliser dans ces TP's est le DEV C++ *9, ce n'est pas le Meilleur mais, il est très simple et préféré par la majorité des débutants.

- Tout d'abord, lancer Dev C++ (sous Windows XP):



- Ensuite, lancer un nouveau projet : **Fichier -> Nouveau -> Fichier Source**.
- L'interface générale se présente sous cette forme :



- L'interface de développement de Dev C++, en dehors de ses menus et barres d'outils, possède trois zones :

1. **Projet**: Regroupe l'ensemble des projets et ces différentes ressources.
2. **Affichage** : zone d'affichage et d'édition des programmes.
3. **Sortie (output)** : fenêtre d'affichage des résultats de compilation, de débogage et de recherches.

B. Premier programme C:

- Crée un dossier qui porte votre **nom** dans *Mes documents*.
- Dans l'éditeur écrivez les lignes de codes suivants:

```
#include <stdio.h>
#include <conio.h>
main()
{
    printf("C'est mon premier programme C \n");
}
```

- A partir du menu **Fichier** → **Enregistrer sous ...**, enregistrer le code sous le nom **programme1.c** dans votre dossier.

Passer dans votre dossier. Commentaire!

- Compiler le programme à partir du menu **Exécuter** → **Compiler (Ctrl+F9)**

Passer dans votre dossier. Commentaire!



- Exécuter le programme avec **Exécuter** → **Exécuter (Ctrl+F10)**
Commentaire!
- Ajouter **getch()**; avant le dernier accolade. Exécuter et commenter.
- Mettre le **m** de **main** en majuscule. Compiler et commenter.
- Supprimer le point-virgule après **getch()**. Compiler et commenter.
- Ajouter la ligne suivante dans le programme : **/* Premier programme en C */**
Commentaire!

Exercice 1:

Editer les lignes de code C suivantes :

```
#include <stdio.h>
#include <conio.h>
main()
{
    int x, y, z;
    x= 3;
```

```

y= 2;
z= x+y;
printf("%d", z);
}

```

- Exécuter le programme et commenter son résultat d'exécution.
- Modifier ce programme en remplaçant `x=3` par `scanf("%d",&x);` et `y=2` par `scanf("%d",&y);`

- Améliorer ce programme en ajoutant les deux lignes suivantes :

```
printf ("donner un entier :");
```

```
avant scanf("%d",&x);
```

```
printf ("donner un deuxième entier :");
```

```
avant scanf("%d",&y) ;
```

- Déduire le rôle des deux fonctions : `printf` et `scanf`.

Exercice 2:

Ecrire un programme en C permettant la saisie de la valeur du côté d'un carré (donné par l'utilisateur) et l'affichage du périmètre et de la surface.

Exercice 3:

Editer les lignes de code C suivantes :

```

#include <stdio.h>
#include <conio.h>
main()
{
    int x, y, z;
    printf ("donner un réel:\n") ;
    scanf ("%d",&x) ;
    printf ("donner un deuxième réel:\n") ;
    scanf ("%d",&y) ;
    z = x/y ;
    printf ("%d / %d= %d",x,y,z) ;
    getch() ;
}

```

- Exécuter le programme et commenter son résultat d'exécution.
- Ajouter une variable `w` de type `int` , et insérer la ligne suivante : `w = x % y ;` après `z = x / y ;`
- Ajouter les deux lignes suivantes à la fin du programme (avant `getch()`).

```
printf("Le quotient de la division de %d par %d = %d",x,y,z);
printf("Le reste de la division de %d par %d = %d",x,y,w);
```
- Exécuter et commenter!
- Modifier dans le premier programme le terme `int` par `float`. Et tous les `%d` par `%f`. commenter !

Exercice 4:

On désire permuter deux entiers (c.à.d le contenu de `x` dans `y` et inversement).

Un étudiant a proposé la solution suivante :

```

#include <stdio.h>
#include <conio.h>
main()
{
    int x, y;
    x = 3;
    y = 2;
    printf("le contenu de x et y avant la permutation:\n");
    printf("x = %d \ny = %d\n",x,y);
    x = y;
    y = x;
    printf("le contenu de x et y apres la permutation:\n");
    printf("x = %d \ny = %d",x,y);
    getch() ;
}

```

Tester cette solution. Commenter !!

Résoudre ce problème.

Exercice 5:

Traduire l'algorithme suivant en langage C :

```

Algorithme algo
Variables nbr, S : Entier
Début
    S ← 0
    Ecrire ('donner le nombre 1 :')
    Lire (nbr)
    S ← S + nbr
    Ecrire ('donner le nombre 2 :')
    Lire (nbr)
    S ← S + nbr
    Ecrire ('donner le nombre 3 :')
    Lire (nbr)
    S ← S + nbr
    Ecrire (S)
Fin

```

Que fait le programme obtenu.