

TP2 : la structure d'un programme en C++

1. Premier programme

Problème :

On veut écrire un programme qui fait l'addition de deux nombres entiers.

Solution :

Pour résoudre ce problème on suit les étapes suivantes (voir cours) :

1.1 L'analyse du problème

Dans cette étape on identifie précisément le problème et on définit les *données (les entrées)*, les *résultats (sorties)* et les *opérations* pour résoudre le problème.

- Dans notre problème :
 - ✓ *Données d'entrées*: deux nombres A et B de type entier,
 - ✓ *Données de sorties*: un nombre C de type entier,
 - ✓ *Opérations*: l'addition des deux nombres A et B (A+B).

1.2 L'écriture de l'algorithme

On définit dans cette étape en utilisant le langage algorithmique l'**ordre des opérations** à mettre en œuvre pour obtenir les résultats (sorties) à partir des données (entrées).

- dans notre problème, l'algorithme est le suivant :

```
Algorithme Addition
A,B,C : entier;
Début
A ← 15;
B ← 10;
C ← A + B;
Ecrire(C);
Fin.
```

1.3 L'écriture du programme

Dans cette étape on traduit l'algorithme a un langage de programmation (C, Pascal, Fortran.. etc). Dans notre cas on va utiliser le langage C++, on

obtient alors le *code source* du programme qui est stocké dans un ou plusieurs fichiers dans l'ordinateur.

- L'algorithme précédant est traduit en langage C++ comme suit :

Language algorithmique	Language C++
<pre>Algorithme Addition A,B,C : entier; Début A ← 15; B ← 10; C ← A + B; Ecrire(C); Fin.</pre>	<pre>#include <iostream> using namespace std; int A,B,C; int main () { A=15; B=10; C=A+B; cout<< C; getchar(); return 0; }</pre>

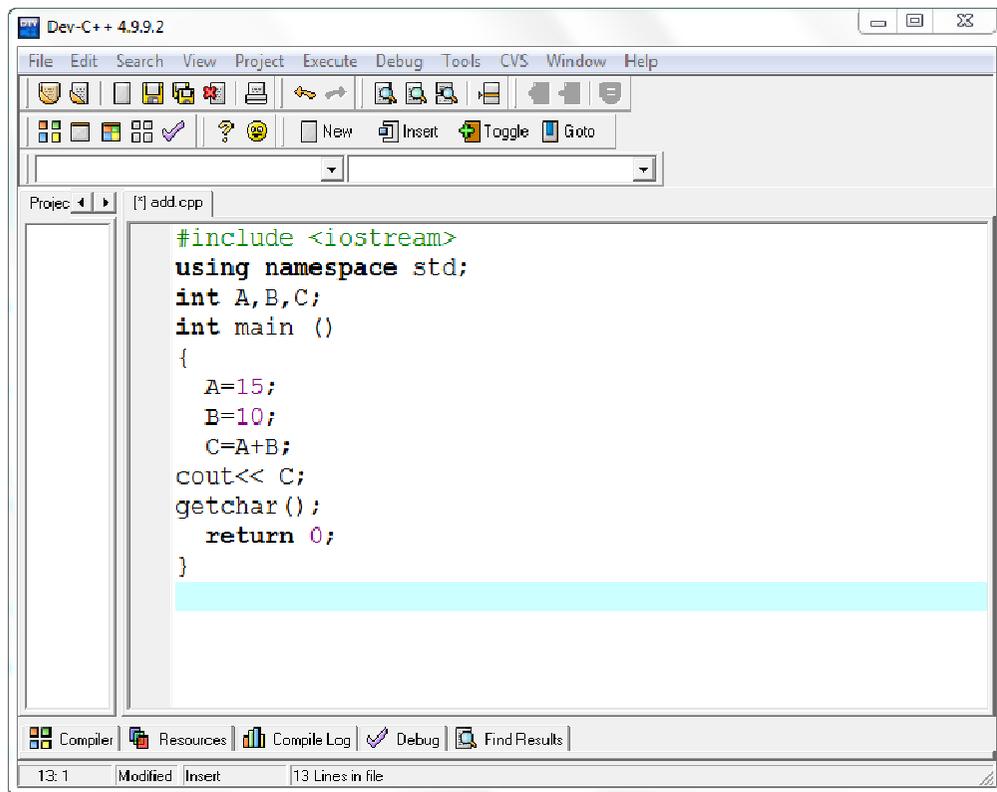
1.4 Compiler le programme

Dans cette étape on traduit le *code source* du programme en *langage machine* en utilisant un logiciel spéciale appelé *compilateurs*, on obtient alors le *code machine* du programme, ce *code machine* est ensuite exécuté directement par l'ordinateur.

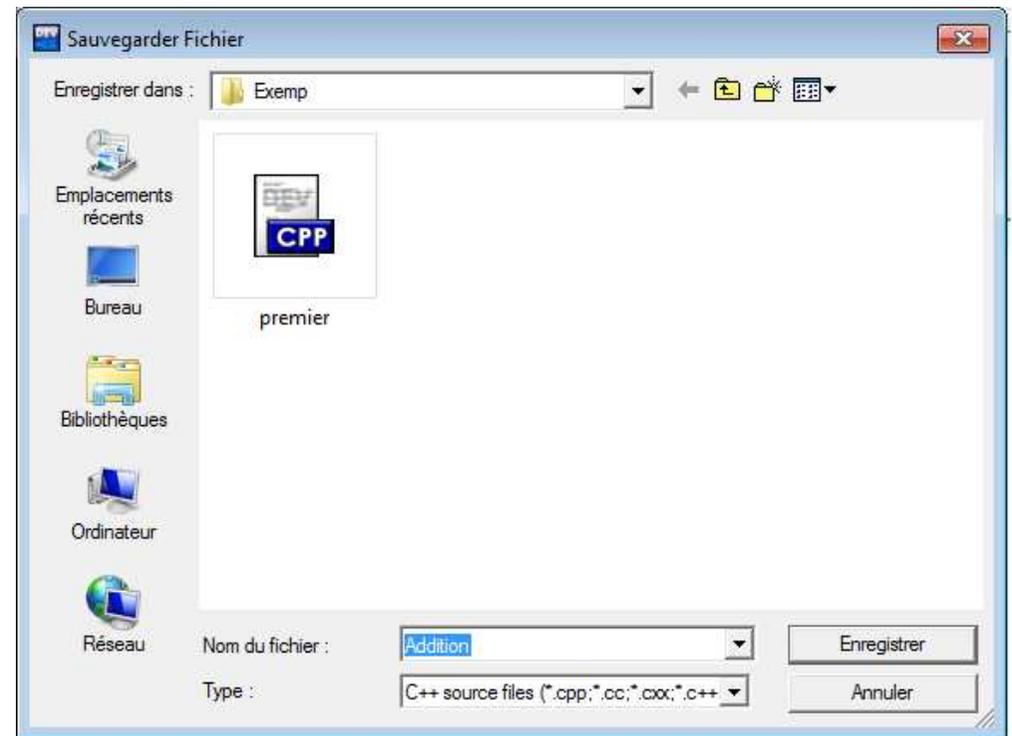
Il existe plusieurs logiciels qui intègrent des compilateurs pour le langage C++comme : Dev C++, Turbo C++, Microsoft Visual C++ Express, ..etc. Ces logiciels intègrent aussi un éditeur du texte pour écrire les programmes et plusieurs autres utilitaires qui facilitent la programmation. Dans notre cas on va utiliser *Dev C++*.

- Pour compiler notre programme on suit les étapes suivantes :

1. On crée un nouveau fichier source en cliquant sur le bouton "*Fichier Source*" (*Fichier* → *Nouveau* → *Fichier Source*),
2. On écrit le programme en utilisant l'éditeur de texte de *Dev C++*,



3. On enregistre le code source du programme dans un fichier en cliquant sur le bouton "**Sauvegarder**" (**Fichier** → **Sauvegarder**),



4. On compile le code source du programme en cliquant sur le bouton "**Compiler**" (**Exécuter** → **Compiler**). Le compilateur va générer le code machine du programme et l'enregistrer dans un fichier sur l'ordinateur (Le code machine du programme est enregistré dans le même emplacement que le code source).

1.5 Exécuter le programme

Dans cette étape, on exécute et on teste notre programme. Pour exécuter le programme on fait un double clic avec le bouton gauche de la souris sur le fichier du programme en **code machine**.

- Pour exécuter le programme on peut aussi utiliser le bouton "**Exécuter**" (**Exécuter** → **Exécuter**) dans le menu de **Dev C++**.



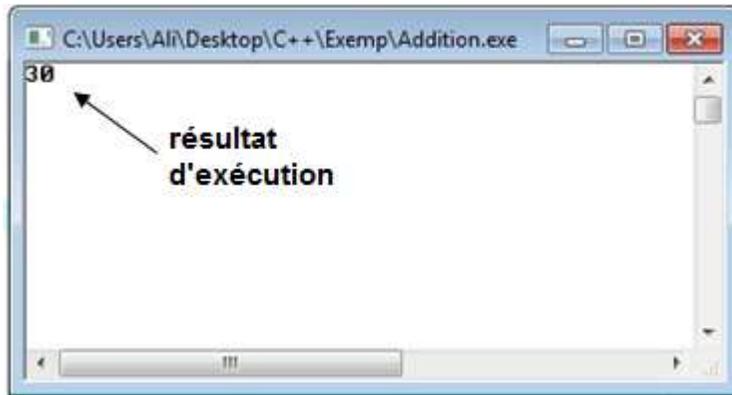
Addition

fichier code source



Addition

fichier exécutable



- Pour exécuter le programme on peut aussi utiliser le bouton "**Exécuter**" (*Executer* → *Executer*) dans le menu de *Dev C++*.

Remarques :

- Pour créer un nouveau fichier source, on clique sur le bouton "**Fichier Source**": (*Fichier* → *Nouveau* → *Fichier Source*), ou on utilise les touches clavier **Ctrl + N**,
 - Pour enregistrer un fichier source, on clique sur le bouton "**Sauvegarder**" (*Fichier* → *Sauvegarder*) et on choisit le langage utilisé (dans notre cas le langage C++) et l'emplacement du fichier, puis on clique sur le bouton "**Enregistrer**".
- On peut aussi utiliser les touches clavier **Ctrl + S**,
- Pour compiler un fichier source, on clique sur le bouton "**Compiler**" (*Executer* → *Compiler*), ou on utilise les touches clavier **Ctrl + F9**,
 - Pour exécuter un fichier source, on clique sur le bouton "**Exécuter**" (*Executer* → *Exécuter*), ou on utilise les touches clavier **Ctrl + F10**.

- Pour directement compiler et exécuter un fichier source, on clique sur le bouton "**Compiler & Exécuter**" (*Executer* → *Compiler&Executer*), ou on utilise la touche clavier **F9**.

TP :

Ecrire un programme qui calcul la somme de 3 nombres entiers.

2. Modification d'un programme :

- On veut modifier le programme précédant pour permettre la saisie (au clavier) des données.

2.1 L'algorithme

```

Algorithme Addition2
A,B,C : entier;
Début
Lire(A);
Lire(B);
C ← A + B;
Ecrire(C);
Fin.

```

2.2 Le programme C++ :

```

#include <iostream>
using namespace std;
int main()
{
int a, b, c;
cin>> a;
cin>> b;
    c = a + b;
cout<< c;

getchar();
return 0;
}

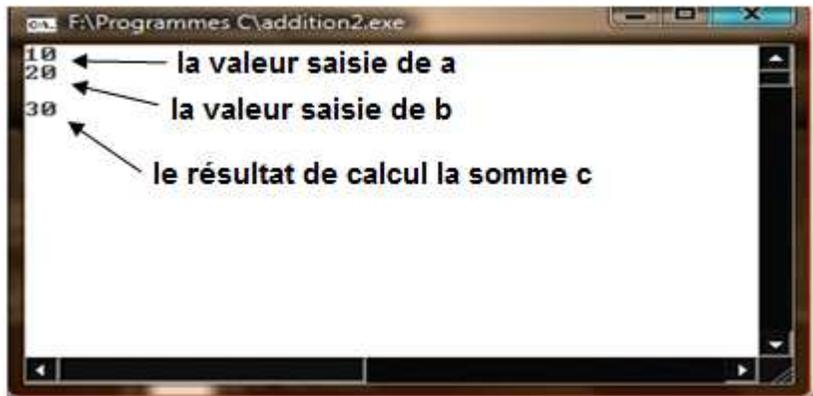
```

2.3 Compilation du programme



addition2

2.4 Exécution du programme



3 Ajouter des messages au programme

- On veut ajouter quelques messages au programme pour faciliter son utilisation.

3.1 L'algorithme

```
Algorithme Addition3  
A,B,C : entier;  
Début  
Ecrire(' Tapez un nombre entier svp :');  
Lire(A);  
Ecrire('Tapez un deuxieme nombre entier svp :');  
Lire(B);  
C ← A + B;  
Ecrire('La somme des deux nombres est : ');  
Ecrire(C);  
Fin.
```

3.2 Le programme C

```
#include <iostream>  
using namespace std;  
int main()  
{  
int a, b, c;  
cout<< "Tapez un nombre entier svp :";  
cin>> a;  
cout<< "Tapez un deuxieme nombre entier svp :";  
cin>> b;  
c = a + b;  
cout<< "La somme des deux nombres est :";  
cout<< c;  
  
getchar();  
return 0;  
}
```

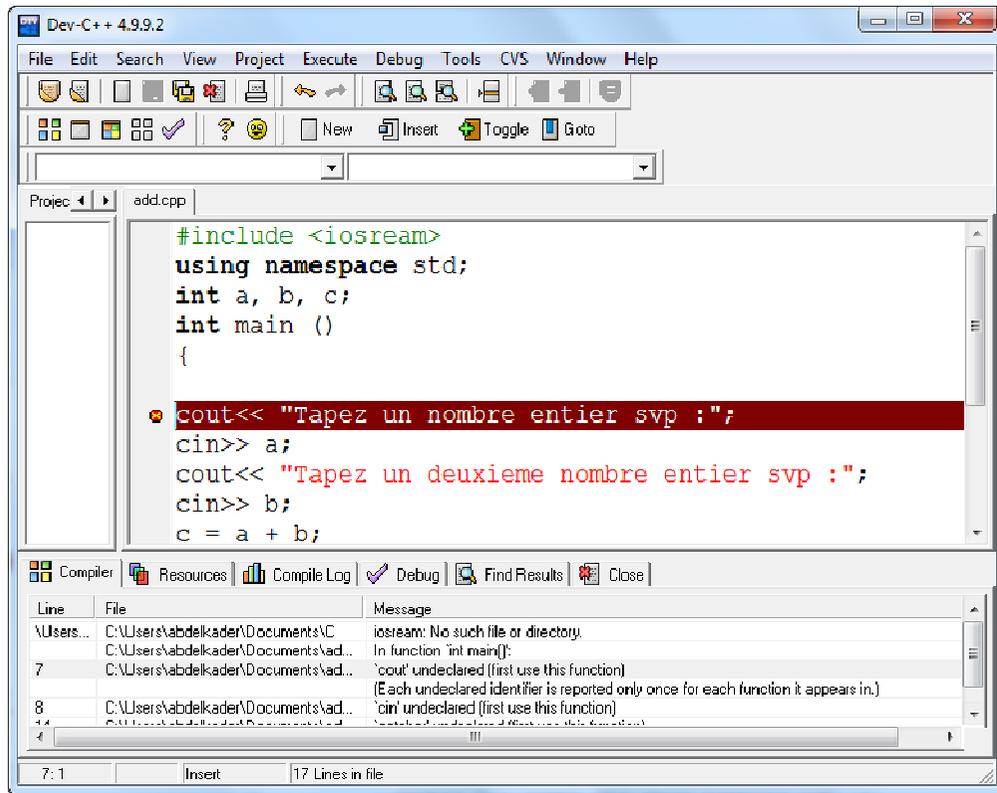
3.3 Compilation du programme



addition3

3.3.1. Correction des erreurs de programmation

Dev C ++ affiche les messages d'erreurs du programme dans une fenêtre séparée au-dessous de l'éditeur.



Le message d'erreur contient la ligne de l'erreur et une description qui permet de corriger l'erreur. Les messages précédés par un **[Warning]** sont des avertissements et le code source du programme peut être compilé même s'ils ne sont pas corrigés.

Exemples :

Programme 1 :

```
#Include <iostream>
using namespace std;
int main()
{
    int a, b, c;
    cout<< "Tapez un nombre entier svp :";
    cin>> a;
    cout<< "Tapez un deuxieme nombre entier svp :";
```

```
cin>> b;
c = a + b;
cout<< "La somme des deux nombres est :";
cout>> c; // avec cout on utilise « << » au lieu de « >> »
getchar();
return 0;
}
```

C:\...\add.cpp no match for 'operator>>' in 'std::cout >> c'

Programme2 :

```
#Include <iostream> // iostream et non pas iosream
using namespace std;
int main()
{
    int a, b, c;
    cout<< "Tapez un nombre entier svp :";
    cin>> a;
    cout<< "Tapez un deuxieme nombre entier svp :";
    cin>> b;
    c = a + b;
    cout<< "La somme des deux nombres est :";
    cout<< c;
    getchar();
    return 0;
}
```

C:\...\Ciosream: No such file or directory.

Programme 3 :

```
include<iostream>
using namespace std;
intmain()
{
    int a, b, c
    cout<< "Tapez un nombre entier svp :";
    cin>> a;
    cout<< "Tapez un deuxieme nombre entier svp :";
    cin>> b;
    c = a + B; // B est différent du b (Majuscule et minuscule)
    cout<< "La somme des deux nombres est :";
```

```
cout<< c;

getchar();
return 0;
}

10 C:\...\addition1.cpp `B' undeclared (first use this function)
```

Programme 4 :

```
#include <iostream>
using namespace std;
int main()
{
int a, b, c
coutt<< "Tapez un nombre entier svp :"; // cout et pas coutt
cin>> a;
cout<< "Tapez un deuxieme nombre entier svp :";
cin>> b;
c = a + B;
cout<< "La somme des deux nombres est :";
cout<< c;

cin.get();
return 0;
}
```

```
6 C:\...\addition1.cpp `coutt' undeclared (first use this
function)
```

```
#include <iostream>
using namespace std;
int main()
{
int a, b, c
cout<< "Tapez un nombre entier svp :";
cin>> a;
cout<< "Tapez un deuxieme nombre entier svp :";
cin>> b;
c = a + B;
```

```
cout<< "La somme des deux nombres est :";
cout<< c;

getchar();
return 0;
// } est obligatoire à la fin du programme
```

```
15 C:\...\addition1.cpp expected `}' at end of input
```

Remarque :

Le nombre des messages d'erreurs affichés ne signifie pas obligatoirement qu'il y a un tel nombre de fautes dans le programme ; par exemple : une faute dans la ligne `#include <iostream>` , implique des erreurs dans les lignes du code qui contient les mots clés « cin » ou « cout ». Une fois la faute du première ligne est corrigée, les autres erreurs seront disparus.

TP :

1) Ecrire un programme C++ qui calcul la somme de 3 nombres entiers. Le programme doit utiliser des messages pour faciliter son utilisation.

2) Modifier le programme précédant pour qu'il en affichant le message suivant :

La somme de **4** et **-12** et **+7** est : **-1**

Où : 4, -12, et +7 sont les valeurs saisies par l'utilisateur des variables respectives a, b, c, et -1 est le résultat.