



COURS et TP DE LANGAGE C++

Chapitre 4

Utilisation d'une bibliothèque

Joëlle MAILLEFERT

joelle.maillefert@iut-cachan.u-psud.fr

IUT de CACHAN

Département GEII 2

CHAPITRE 4

UTILISATION D'UNE BIBLIOTHEQUE

Ce petit chapitre vise à expliquer comment se servir d'une bibliothèque de fonctions. On prendra quelques exemples dans la bibliothèque de BORLAND C++.

NOTION DE PROTOTYPE

Les fichiers de type ".h" (conio.h, dos.h, iostream.h etc...), appelés *fichiers d'en tête* contiennent la définition des *prototypes* des fonctions utilisées dans le programme. Le prototype précise la syntaxe de la fonction: son nom, le type des paramètres éventuels à passer, le type de l'expression - la valeur retournée au programme (void si aucun paramètre retourné).

Grâce aux lignes "#include", le compilateur lit les fichiers de type ".h" et vérifie que la syntaxe de l'appel à la fonction est correcte.

FONCTIONS NE RENVOYANT RIEN AU PROGRAMME

Ce sont les fonctions de type **void**.

Exemple: **clrscr**

fonction Efface l'écran de la fenêtre dos.

prototype void clrscr();

prototype dans conio.h et donc bibliothèque à charger.

Une fonction ne renvoyant rien (de type void) s'écrit telle que. C'est une action représentée par une instruction. Ici pas de passage d'arguments.

```
ex:  clrscr();           // efface l'écran
      cout >>"BONJOUR\n";
      cout>>"AU REVOIR\n";
```

FONCTIONS RENVOYANT UNE VALEUR AU PROGRAMME

Ce sont les fonctions de type **autre que void**. Elles renvoient au programme une valeur (expression) dont le type est précisé dans la documentation.

Exemple: **kbhit**

fonction Teste une éventuelle frappe au clavier
prototype int kbhit();
prototype dans conio.h et donc bibliothèque à charger.

La fonction kbhit renvoie un entier au programme appelant. Cet entier vaut 0, tant qu'il n'y a pas eu de frappe clavier. On ne peut donc pas écrire la fonction telle que. Il faut la traiter comme le résultat de calcul d'une expression qui peut être mémorisée dans une variable de type int.

```
ex:   int u;
      do
      {
      .....
      .....
      u = kbhit();
      }
      while(u== 0);
      // boucle exécutée tant qu'il n'y a pas de frappe clavier
```

ou plus simplement

```
do
{
.....
.....
}
while(kbhit()== 0);
```

FONCTIONS AVEC PASSAGE DE PARAMETRE

Exemple: **log**

fonction Fonction logarithme népérien..
prototype double log(double x);
prototype dans math.h et donc bibliothèque à charger.

La fonction log, renvoie un réel au programme appelant. On traite la fonction comme une variable de type double. Il faut lui passer un paramètre effectif de type double. On peut se contenter d'un paramètre effectif de type float (en C et C++ le compilateur fait automatiquement une transformation de type qui n'est pas dégradante dans ce cas précis).

```
ex:   double x, y;
      cout <<"SAISIR x : ";
      cin >> x;
```

```
y = log(x);
cout<<"log(x)= "<<y<<"\n";
```

Exercice IV 1:

En utilisant randomize et random jouer au 421.

```
#include <iostream.h>
#include <time.h>
#include <stdlib.h>
#include <conio.h>

void main()
{
    char c;
    int n1, n2, n3;
    cout<<"JEU DU 421\n";
    randomize();
    do{
        clrscr();
        // LANCEMENT DES DES
        cout<<"LANCER LES DES EN FRAPPANT UNE TOUCHE : ";
        getch();
        n1 = random(6) + 1;
        n2 = random(6) + 1;
        n3 = random(6) + 1;
        cout<<"\n VOICI LES DES : "<< n1 <<" " << n2 <<" " << n3 <<"\n";
        // TEST
        if(((n1==4) && (n2==2) && (n3 ==1)) ||
            ((n1==4) && (n2==1) && (n3 ==2)) ||
            ((n1==2) && (n2==4) && (n3 ==1)) ||
            ((n1==2) && (n2==1) && (n3 ==4)) ||
            ((n1==1) && (n2==2) && (n3 ==4)) ||
            ((n1==1) && (n2==4) && (n3 ==2))) cout<<"GAGNE !\n";
        else cout <<"PERDU !\n";
        cout<<"\nPOUR REJOUER FRAPPER O SINON UNE TOUCHE QUECONQUE\n";
        c = getch();
    }
    while((c=='O') || (c=='o'));
}
```