

TP 1 : Les sous-programmes (fonctions et procédures)

1. Introduction :

- Les sous-programmes permettent de découper les programmes en modules pour faciliter la compréhension et la maintenance des programmes et aussi pour éviter les répétitions dans le code.
- Comme on a vu dans le cours, il existe deux types de sous-programmes : les procédures et les fonctions.
- Contrairement au langage algorithmique, le langage C++ généralise la structure de la fonction sur les deux types de sous-programmes.
- Pour cela, on va commencer par l'étude des fonctions en langage C++, ensuite on va voir la structure des procédures.

2. Les fonctions :

La structure générale d'une fonction en langage C++ est comme suit :

```
type_de_retour_fonction nom_de_la_fonction ( type_1 variable1, type_2 variable2....typeN variableN )  
{  
<Déclaration des variables de la fonction>  
<instructions1>;  
<instructions2>;  
...  
return (résultat); /* Pour retourner le résultat */  
}
```

Exemple :

Un algorithme qui lie trois nombres positifs non nuls A, B, et C et calcule et affiche la somme suivante : **A! + B! + C!**

Langage Algorithmique	Langage C++
<p>Algorithme Exemple A, B, C, Somme: entiers</p> <hr/> <p><i>/* Définition de la fonction factorielle */</i> Fonction Factorielle (N: entier):entiers ; Resultat, fact, i : entiers Début fact ← 1; Pour i allant de N a 1 pas -1 faire fact ← fact * i ; Fin pour ; Resultat ← fact ; Retourner(Resultat) ; Fin ;</p> <hr/> <p><i>/* Programme principale */</i> Début Lire (A, B, C) ; Somme ← Factorielle(A) + Factorielle(B) + Factorielle(C) ; Ecrire(Somme) ; Fin.</p>	<pre>#include <iostream> using namespace std; ----- /* Définition de la fonction factorielle */ int Factorielle (int N) { int Resultat, fact, i ; fact =1; for (i=N; i>=1; i= i-1) { fact = fact * i; } Resultat = fact; return (Resultat); } ----- /* Programme principale main()*/ int main () { int A, B, C, Somme; cin >>A>>B>>C; Somme = Factorielle(A) + Factorielle(B) + Factorielle(C); cout << Somme ; getchar(); return 0; }</pre>

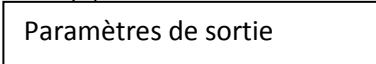
TP:

Ecrire un programme C++ qui lie trois nombres positifs non nuls A, B, et C et calcule et affiche la somme suivante : $A! + B^C$

3. Les procédures :

Une procédure dans le langage C++ est représentée par une fonction avec un type de retour **void** (rien). Les paramètres de sortie des procédures sont passés par variable (adresse) en utilisant l'opérateur '**&**' et sont définis en utilisant l'opérateur '*****'.

Exemple 1 : Un programme qui affiche la somme et le produit de deux nombre entiers A et B.

Langage Algorithmique	Langage C++
<pre> Algorithme Exemple1 A,B,C, Somme, Produit: entiers ----- /* Définition de la procédure Calcul */ Procédure calcul (X1, X2 : entiers; var S, P : entiers) Début S ← X1 + X2; P ← X1 * X2; Fin ; ----- /* Programme principale */ Début Lire (A,B); calcul (A, B, Somme, Produit) ; Ecrire(Somme, Produit) ; Fin. </pre>	<pre> #include <iostream> using namespace std; ----- /* Définition de la procédure Calcul */ void calcul (int X1, int X2, int *S, int *P) { *S = X1 + X2; *P = X1 * X2; } ----- /* Programme principale <i>main()</i> */ int main () { int A, B, Somme, Produit; cin >>A>>B>>C; calcul (A, B, &Somme, &Produit) ; cout << Somme << Produit ; getchar(); return 0; } </pre> 

TP:

Ecrire un programme C++ qui fait la permutation de deux variables entières A et B.

4. La réutilisation de sous-programmes

On peut mettre le programme principal et les fonctions et les procédures dans des fichiers séparés, ce qui permet à plusieurs programmes d'utiliser le même sous-programme.

Par exemple on peut mettre le programme principal dans le fichier "**TP.cpp**" et le programme factorielle dans le fichier "**fonctions_tp.cpp**" comme suit :

Le fichier "TP.cpp"	Le fichier "fonctions_tp.cpp"
<pre>#include <iostream> #include "fonctions_tp.cpp" using namespace std; /* Programme principale main() */ int main () { int A, B, C, Somme; cin >>A>>B>>C; Somme = Factorielle(A) + Factorielle(B) + Factorielle(C); cout << Somme ; getchar(); return 0; }</pre>	<pre>/* Définition de la fonction factorielle */ int Factorielle (int N) { int Resultat, fact, i ; fact =1; for (i=N; i>=1; i= i-1) { fact = fact * i; } Resultat = fact; return (Resultat); }</pre>

Pour indiquer où se trouve la fonction Factorielle

Remarque :

Il faut mettre les fichiers "**TP.cpp**" et "**fonctions_tp.cpp**" dans le même répertoire avant de compiler le programme "**TP.cpp**".