

LIFAPI – TD 5 : Passage de paramètres

Objectifs : Comprendre la différence entre les modes de passage des paramètres : donnée ou donnée / résultat
Comprendre la différence entre paramètres formels et paramètres effectifs

Recommandations : Pour chacun des algorithmes que vous écrirez, vous préciserez le mode de passage des paramètres (**donnée** ou **donnée / résultat**) et vous écrirez le programme principal appelant les sous-programmes que vous aurez écrits.

Données (passage par valeur)

- Le sous-programme dispose d'une copie de la valeur.
- Il peut la modifier, mais l'information initiale dans le code appelant n'est pas affectée par ces modifications.

Syntaxe en C/C++ : type nom ;

Résultats ou **données / résultats** (passage par adresse)

- Le sous-programme dispose d'une information lui permettant d'accéder en mémoire à la valeur que le code appelant cherche à lui transmettre.
- Il peut alors modifier cette valeur, le code appelant aura accès aux modifications faites sur la valeur.

Syntaxe en C/C++ : type & nom ;

1. Soit le programme suivant. Identifiez et notez :
 - a. le(s) paramètre(s) formel(s) / le(s) paramètre(s) effectif(s)
 - b. le(s) paramètre(s) en donnée / le(s) paramètre(s) en donnée / résultat
 - c. Qu'est censé faire ce programme ?
 - d. Quelle(s) modification(s) faudrait-il apporter pour obtenir un résultat plus logique ?

```
#include <iostream>
using namespace std ;

void mystere (int a, int b, int &c, int d)
{c=a+b;
 d=a*b;
}

int main (void)
{int e,f,g,h;
 cout<<"donnez une valeur";
 cin>>e;
 cout<<"donnez une valeur";
 cin>>f;
 mystere(e,f,g,h);
 cout<<" valeur "<<g<<" valeur : "<<h<<endl;
 return 0;
}
```

2. Écrivez l'algorithme d'une procédure effectuant la permutation circulaire de trois variables : a=5 b=8 et c=2 donne après exécution : a=2 b=5 et c=8.
3. Écrire l'algorithme d'une **fonction** saisie_valeur qui demande à l'utilisateur et retourne au programme principal une valeur entière comprise entre 0 et 20. La saisie sera recommencée tant que la valeur choisie n'appartient pas à l'intervalle [0 ; 20]. Transformez la fonction précédente en une procédure. Utiliser ces deux sous-programmes dans un programme principal et constatez les différences.
4. Écrire l'algorithme d'une fonction factorielle qui calcule et retourne la valeur de la factorielle d'un entier passé en paramètres. Transformer la fonction en procédure. Utiliser ensuite les deux sous-programmes dans un programme principal.