

# LIFAPI – TD 4 : Les sous-programmes

*Objectifs :* Assimiler la différence entre une fonction et une procédure  
Savoir déclarer et utiliser un sous-programme

*Recommandations :*

Pour chaque **algorithme** demandé, vous préciserez (en justifiant) s'il s'agit d'une **procédure** ou d'une **fonction**.

Vous écrirez **l'entête du sous-programme** (sans oublier les préconditions, les données et résultats, les déclarations des variables locales...) ainsi qu'un **exemple d'appel au sous-programme**.

<b>Fonction</b> nom(liste paramètres):type retourné Préconditions : Données : Résultat : Description : Variables locales : <b>Début</b> instruction(s) <b>retourner</b> valeur (ou <b>renvoyer</b> ) <b>Fin</b> nom  <b>Appel</b> : variable ← nom (paramètres) <b>Exemple</b> : f ← factorielle (6)	<b>Procédure</b> nom (liste des paramètres) Préconditions : Données : Description : Variables locales :  <b>Début</b> instruction(s) <b>Fin</b> nom  <b>Appel</b> : nom (paramètres) <b>Exemple</b> : mention (12)
--	---

1. Rappeler en quelques mots la différence entre une fonction et une procédure. Donner un exemple caractéristique pour chaque.
2. Écrire l'algorithme d'un sous-programme qui retourne la moyenne de deux réels a et b donnés en paramètre. Écrire le programme principal qui utilise le sous-programme précédent et affiche le résultat produit.
3. Écrire l'algorithme d'un sous-programme qui affiche les dix nombres suivants la valeur n donnée en paramètre. Par exemple, si l'utilisateur entre le nombre 17, le programme affichera les nombres de 18 à 27. Écrire le programme principal qui utilise le sous-programme précédent.
4. Écrire l'algorithme d'un sous-programme qui calcule et retourne la somme des n premiers entiers. Rappel :  $1 + 2 + 3 + \dots + n = n(n+1) / 2$   
Écrire le programme principal qui utilise le sous-programme précédent.
5. Un nombre **parfait** est un nombre naturel n non nul qui est égal à la somme de ses diviseurs stricts (n exclus). Exemple :  $6 = 1 + 2 + 3$ 
  - a. Écrire en langage algorithmique une fonction booléenne qui retourne vrai si un entier n passé en paramètre est un nombre parfait, faux sinon.
  - b. Écrire en langage algorithmique le programme principal permettant d'afficher la liste des nombres parfaits compris entre 1 et 10000. On utilisera le résultat renvoyé par la fonction précédente.