

LIFAP1 – TD 3 : Fonctions et procédures

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**.

1. Rappeler en quelques mots la différence entre une fonction et une procédure. Donnez 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.
4. Écrire l'algorithme d'un sous-programme 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].
5. Écrire l'algorithme d'un sous-programme qui calcule la somme des n premiers entiers.
Rappel : $1 + 2 + 3 + \dots + n = n(n+1) / 2$
6. 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.
7. Écrire l'algorithme d'un sous-programme qui dessine un carré de côté N à l'écran. L'utilisateur pourra choisir le caractère du contour du carré lors de l'appel du sous-programme.

Pour s'entraîner

1. Écrire **en un minimum de lignes** l'algorithme d'un sous-programme permettant de dessiner le motif ci-contre. Le nombre de motifs et la longueur du motif seront passés en paramètres du sous-programme. Dans cet exemple, le motif de base est répété 3 fois et la base d'un triangle est de longueur 4.

```
*
**
***
****
*
**
***
****
*
**
***
****
*
**
***
****
```