

LIFAPI – TD 6 : Passage de paramètres

Objectifs : Approfondir les notions vues dans le TD précédent (différence fonction / procédure, mode de passage des paramètres)

Rappels paramètres

- **Paramètre donnée (entrée)**
C'est une **valeur fournie au sous-programme** (procédure ou fonction) pour qu'il puisse effectuer son travail.
→ Elle est **utilisée** mais **pas modifiée** par le sous-programme.
- **Paramètre résultat (sortie)**
C'est une **valeur produite par le sous-programme**.
→ Elle est **initialisée ou modifiée** dans la fonction pour **transmettre un résultat** à l'extérieur.
- **Paramètre donnée/résultat (entrée/sortie)**
C'est une **valeur transmise au sous-programme**, qui peut être **modifiée et renvoyée**.
→ Elle est **utilisée comme donnée** et mise à jour **comme résultat**

1. Écrivez l'algorithme d'une procédure permettant d'effectuer la division euclidienne de deux entiers a et b (sans utiliser la division !). On appellera q le quotient et r le reste de cette division. On rappelle la formule de la division : $a = b \cdot q + r$, avec $r < b$.
2. Un nombre entier est dit "**doublon**" si le produit de ses diviseurs est multiple de la somme de ses diviseurs.
Exemple : $n = 6$. Les diviseurs de n sont : 1, 2, 3, 6. La somme des diviseurs est 12 et le produit des diviseurs est 36 ($= 3 \cdot 12$). Le produit des diviseurs de n est donc un multiple de la somme des diviseurs de n donc n est un nombre doublon.
 - a. Ecrire l'algorithme d'un sous-programme `somme_produit` permettant de calculer et "renvoyer" au programme principal la **somme des diviseurs et le produit des diviseurs** d'un nombre n passé en paramètre.
 - b. Ecrire l'algorithme d'une **fonction booléenne** `verifie_doublon` qui retourne vrai si un entier passé en paramètre est un doublon, faux sinon. On utilisera pour cela le sous-programme écrit dans la question précédente.
3. Un entier admettant exactement 3 diviseurs est le carré d'un nombre premier. Si p est premier, alors ses diviseurs sont 1 et p, donc les diviseurs de p^2 sont 1, p et p^2 .
Par exemple 49 a pour diviseurs 1, 7 et 49, donc 7 est un nombre premier.
 - a. Ecrire un sous-programme `diviseurs` qui calcule et "retourne" le nombre de diviseurs **et** la valeur du deuxième diviseur rencontré.
Pour $n = 49$ nombre de diviseurs = 3 (1, 7 et 49) et valeur du deuxième diviseur = 7.
 - b. Ecrire une fonction booléenne `est_premier` qui retourne vrai si un entier p passé en paramètre est premier, faux sinon. Pour cela on testera si le sous-programme `diviseurs` appelé avec p^2 comme paramètre "retourne" bien 3 comme nombre de diviseurs et p comme valeur de deuxième diviseur.
 - c. Ecrire le programme principal qui affiche tous les nombres premiers inférieurs à 50.

Pour s'entraîner

Ecrire l'algorithme d'une procédure permettant à partir des trois coefficients a, b et c d'un polynôme du second degré, de calculer et retourner (si elles existent) les racines.