AP1 TD1 – Variables, entrée-sortie, séquence

Correction partielle

Exercice 4 – L'algorithme mystère

Soit l'algorithme suivant :

```
ALGORITHME mystère
CONSTANTE (PIF : réel) ← 3.14159
VARIABLES paf, pof, puf : réels
DEBUT
	saisir paf
	pof ← 2 × PIF
	puf ← pof × paf
	afficher puf
FIN
```

1. Quelle principale critique peut-on formuler à l'encontre de cet algorithme ?
   Les noms des variables sont farfelus car pas du tout significatifs.
2. Quel est le rôle de cet algorithme ?
   Calcul et affichage du périmètre du cercle de rayon *paf*.
3. Ré-écrire cet algorithme plus clairement.

```
ALGORITHME Perimetre_cercle
CONSTANTE (PI : réel) ← 3.14159
VARIABLES rayon, perimetre : réels
DEBUT
	saisir rayon
	perimetre ← 2 × PI × rayon
	afficher perimetre
FIN
```

Exercice 5 – Mon premier algorithme

Concevoir un algorithme qui calcule et affiche l'aire d'un rectangle. La longueur et la largeur du rectangle sont demandées à l'utilisateur.

```
ALGORITHME Aire_rectangle
VARIABLES larg, long, aire : réels
DEBUT
	saisir larg
	saisir long
	aire ← larg × long
	afficher aire
FIN
```

Exercice 6 – Mon second algorithme

Concevoir un algorithme qui calcule le nombre de radiateurs nécessaires pour chauffer une pièce. On sait qu'un radiateur est capable de chauffer 8 m^3. L'utilisateur donnera la longueur, la largeur et la hauteur de la pièce (en mètres).

```
ALGORITHME Nombre_radiateurs
CONSTANTE (CAPACITE: entier) ← 8
VARIABLES
	larg, long, haut, volume : réels
	nbRad : entier
DEBUT
	saisir larg
	saisir long
	saisir haut
	volume ← larg × long × haut
	nbRad ← volume div CAPACITE		{ attention : division entière }
	SI volume mod CAPACITE <> 0 ALORS
		nbRad ← nbRad + 1
		{ si volume n'est pas un multiple de CAPACITE, on ajoute 1 car, sinon, le
		nombre de radiateurs serait trop juste }
	FINSI
	afficher nbRad
FIN
```

Exercice 7 – Analyse descendante

Définition. On appelle **nombre de Armstrong** un entier naturel qui est égal à la somme des cubes des chiffres qui le composent.
Par exemple, 153 est un nombre de Armstrong car : $1^3+5^3+3^3=153$.

Réaliser l'analyse descendante d'un algorithme qui demande à l'utilisateur un entier et qui indique si cet entier est un nombre de Armstrong.

Décomposition de premier niveau :

1. Saisir un entier *nb*
2. Décomposer *nb*
3. Conclure s'il s'agit d'un nombre de Armstrong

Décomposition de second niveau. Les étapes 2 et 3 peuvent se décomposer en sous étapes plus simples :

- 2.1. Extraire le chiffre des centaines *c*
- 2.2. Extraire le chiffre des dizaines *d*
- 2.3. Extraire le chiffre des unités *u*

3.1. Calculer $s = c^3 + d^3 + u^3$
3.2. Comparer s et nb

Suite à cette analyse descendante, l'algorithme vient aisément.

```
ALGORITHME Armstrong
VARIABLES nb, c, d, u, s, temp : entiers
DEBUT
    saisir nb

    c ← nb div 100

    temp ← nb mod 100
    d ← temp div 10

    u ← temp mod 10

    s ← c³ + d³ + u³

    SI s = nb ALORS
        afficher nb, "est un nombre de Armstrong"
    SINON
        afficher nb, "n'est pas un nombre de Armstrong"
    FINSI
FIN
```