Exercice 1

Tâche

Après saisie de V1, V2 et V3, variables entières, échanges de leurs valeurs pour obtenir V1 ≤ V2 ≤ V3, puis affichage des trois variables pour contrôle

Description des données, schéma algorithmique, algorithme, jeu d'essai

Correction possible

Données

Variables

v1, v2, v3,temp :Flottant

Schéma algorithmique

Structure linéaire

↓ saisie des valeurs

[v1 ≤ v2 et v2 ≤ v3] rien à faire

[v2 ≤ v3 et v3 ≤ v1] échange(v1, v3) et échange(v1, v2)

[v3 ≤ v1 et v1 ≤ v2] échange(v1, v3) et échange(v1, v2)

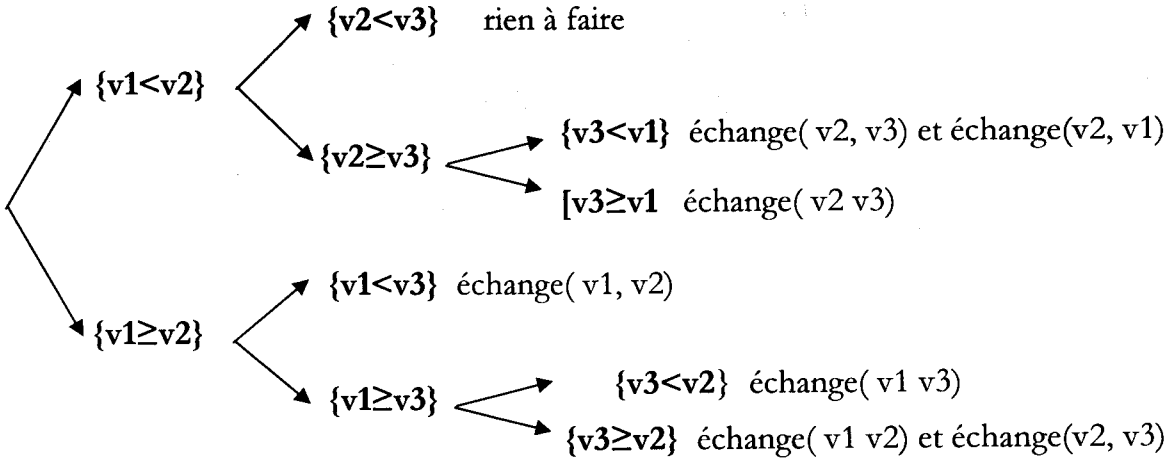
[v2 ≤ v1 et v1 ≤ v3] échange(v2, v3) et échange(v1, v2)

[v1 ≤ v3 et v3 ≤ v2] échange(v2, v3)

[v3 ≤ v2 et v2 ≤ v1] échange(v1, v3)

↓ affichage des variables

Structure hiérarchique



Exercices 2

Tâche

Calcul de la solution de ax + b = 0, a et b nombres réels saisis.

Description des données, schéma algorithmique, algorithme, jeu d'essai

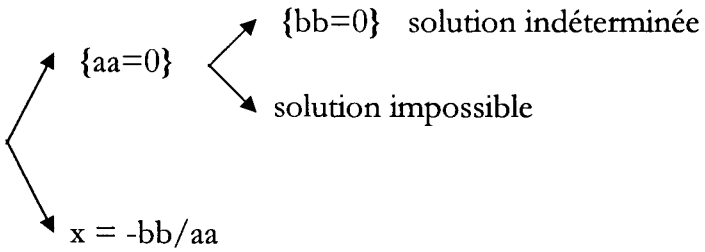
Correction possible

Données

Variables

aa,bb,xx :Flottant

Schéma algorithmique



Code

```
.....
Si aa= 0
      alors si bb = 0
           alors afficher("solution indéterminée")
           sinon afficher("solution  impossible")
      sinon début
      xx :=-bb/aa
      afficher("x = ", xx)
      fin
afficher("programme terminé")
```

Exercices 3

Tâche

Calcul des nombres d'Armstrong inférieurs à 1000. (nombre d'Armstrong, nombre tel qu'il est égal à la somme des cubes de ses chiffres).

Description des données, schéma algorithmique, algorithme, jeu d'essai

Correction possible

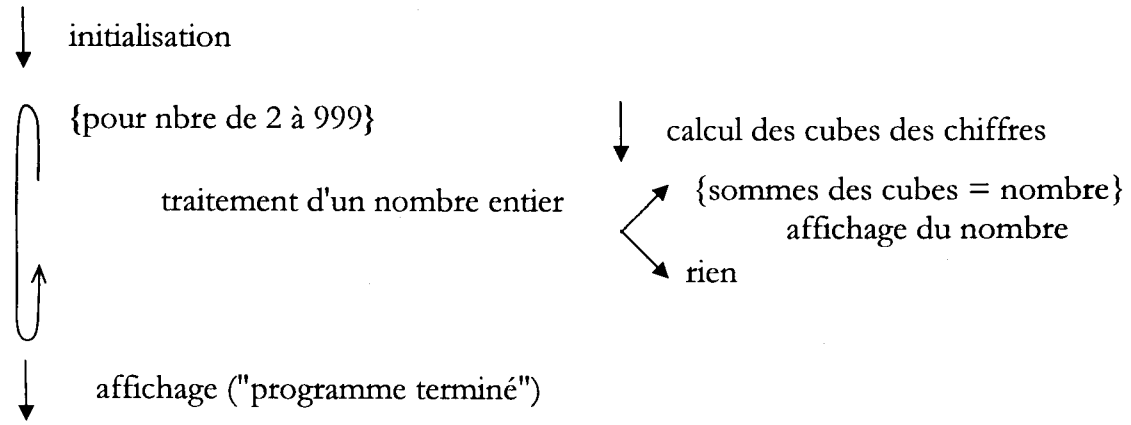
Données

Variables

nbre : Entier

cent, diz, unite : Entier

Schéma algorithmique



Code

```
nbre, nbrew : Entier
cent, diz, unite : Entier

afficher("Calcul des nombres d'Armstrong")
afficher("nombres d'Armstrong : ")
pour nbre 2 à 999 pas 1 faire
      //traitement d'un nombre
      debut
      cent ← nbre div 100
      diz ← nbre mod 100 div 10
      unite ← nbre mod 10
      si    cent * cent * cent
         + diz * diz * diz
         + unite * unite *unite = nbre

          alors afficher (nbre, ", ")
      fin
afficher("programme terminé")
```

Exercices 4

Tâche

Recherche de la première occurrence d'un entier dans un tableau préalablement rempli de 150 entiers aléatoires inférieurs à 300 et en **désordre**. Le nombre entier saisi peut ne pas figurer dans le tableau.

Description des données, schéma algorithmique, algorithme, jeu d'essai

Correction possible

Données

Constantes

Taille = 150

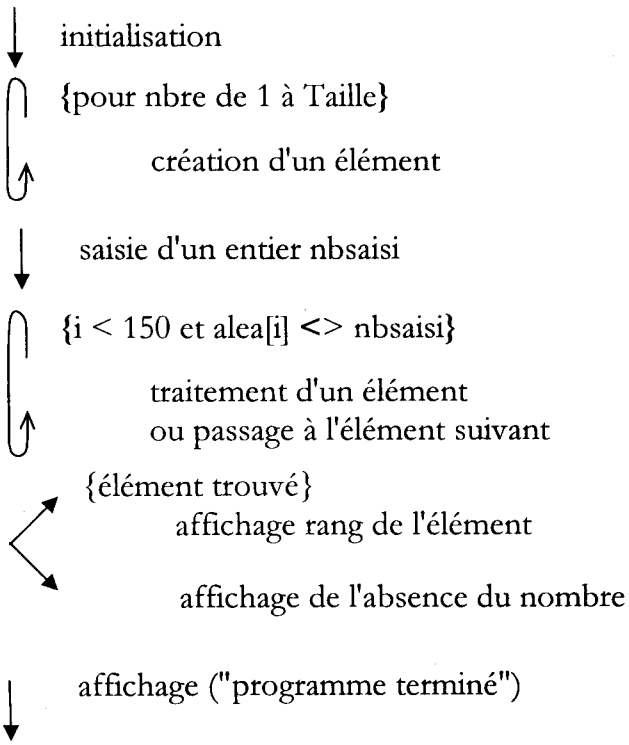
Variables

nbre : Entier

alea[Taille] : Entier

i :Entier

Schéma algorithmique



Algorithme

```
Taille = 150

nbreSaisi : Entier

entalea[Taille] : Entier
i : Entier

afficher("Recherche d'une valeur dans un tableau")
pour i de 0 à Taille - 1 pas 1 faire
      //création d'un élément
      debut
      entalea[i] ← alea(300)
      fin
afficher("Tapez un nombre entier : ")
entrer(nbreSaisi)
i ← 0
tant que i < Taille et entalea[i] <> nbreSaisi faire
      i ← i + 1
si entalea[i] = nbreSaisi  // attention.
si i < 150
      alors afficher("Rang de ", nbreSaisi, " : ", i)
      sinon afficher (nbreSaisi, " n'est pas dans le tableau")
afficher ("programme terminé")
```

Jeu d'essai

Exercices 5

Tâche

Recherche de la première occurrence d'un entier dans une tableau préalablement rempli de 150 entiers aléatoires inférieurs à 300 et en **ordre**. Le nombre entier saisi peut ne pas figurer dans le tableau.

Description des données, schéma algorithmique, algorithme, jeu d'essai

Correction possible

Données

Idem exercice précédent

Schéma algorithmique

Avec le précédent,même sructure mais différence dans la création des nombres en ordre (qui peut être sautée) et dans la condition de la recherche

Algorithme

```
entalea[0] ← entalea[300 - Taille]
pour i de 1 à Taille - 1 pas 1 faire
      //création d'un élément
      debut
      entalea[i] ← ent alea[i -1] + alea(300 - taille + i)
      fin


i ← 0
tant que i < Taille et entalea[i] < nbreSaisi faire
      i ← i + 1
si i < 150 et entalea[i] = nbreSaisi
      alors afficher("Rang de ", nbreSaisi, " : ", i)
      sinon afficher (nbreSaisi, " n'est pas dans le tableau")
```

Exercices 6

Tâche : Multiplication égyptienne

Soient X et Y, deux nombres entiers, le programme doit calculer le produit des deux en n'utilisant que l'addition, la soustraction et la division par 2.

On remarque que :

Si X est pair, XY = (X/2) (2Y).
Si X est impair, XY = (X-1)Y +Y.

D'où par exemple :

15 x 53 = 14 x 53 + 53
= 7 x 106 + 53
= 6 x 106 + 106 + 53
= 3 x 212 + 159
= 2 x 212 + 212 + 159 = 2 x 212 + 371
= 1 x 424 + 371
= **795**

Remarque : Il est souhaitable que X soit inférieur à Y. Le programme agira en conséquence.

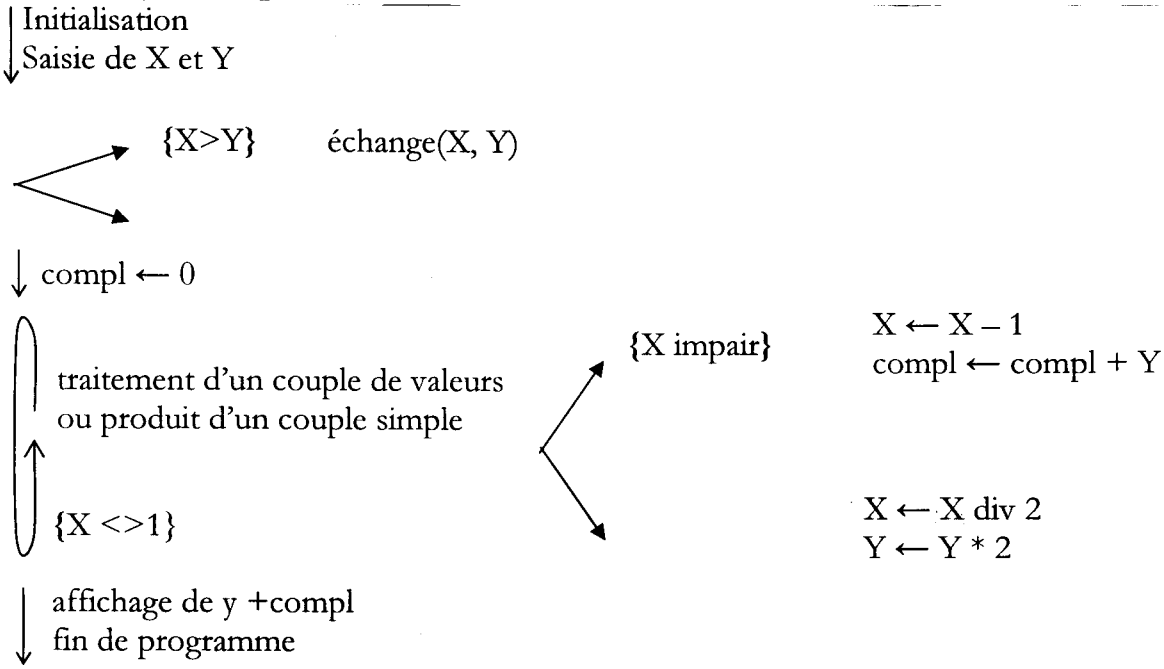
Description des données, schéma algorithmique, algorithme, jeu d'essai

Correction possible

Données

xx , yy , compl : entier

Schéma algorithmique



Algorithme

```
xx , yy , compl    : entier .

afficher("multiplication égyptienne")
afficher("Tapez X :")
entrer(xx)
afficher("Tapez y :")
entrer(yy)
compl ← 0
si X > Y
      alors debut
      temp ← xx
      xx ← yy
      yy ← temp
      fin
faire
      //traitement d'un couple de valeurs
      debut
      si xx mod 2 = 1
           alors debut
           xx ← xx - 1
           compl ← compl + yy
           fin
           sinon debut
           xx ← xx div 2
           yy ← yy * 2
           in
      fin
      tant que xx<> 1
yy ← yy + compl
afficher("le produit est : " ; yy)
afficher("Programme terminé")
```