

## Compte rendu TD algorithme de Kaprekar en 1°S

**Situation :** classe de 1°S de 34 élèves ; travail en demi-classe par groupes de 2 ou 3.

### Première séance : questions 1 et 2 du T.D

#### Objectifs :

- Ecrire un premier algorithme de tri ;
- Utilisation de « si...sinon ».

#### Déroulement de la séance :

Les élèves testent l'algorithme sur deux exemples chacun et remarquent qu'on finit toujours par obtenir 495. Peut-être pourrait-on le démontrer par l'algorithmique, en testant tous les cas possibles ?

Seuls 4 groupes répondent entièrement à la question 2 au cours de la séance et trois autres ont l'algorithme en langage naturel mais pas sous algobox.

Deux groupes pensent répondre à la question en écrivant un algorithme qui ne donne que le plus grand des trois nombres.

Dans l'ensemble, les élèves ne comprennent pas bien l'énoncé : certains essaient de partir d'un nombre de trois chiffres (j'ai donc rajouté le commentaire en italique après la question 1 dans l'énoncé); d'autres ne voient pas du tout ce qu'il faut faire ni comment démarrer. Il est vrai que c'est la première fois que je leur demande de créer un algorithme avec autant de cas à tester sans modèle. Dans les exercices précédents, j'ai souvent donné un algorithme à lire puis à modifier pour répondre petit à petit au problème posé.

D'autre part, la plupart des groupes ne suit pas la consigne : ils ne passent pas par l'étape « écriture sur le papier d'un algorithme en langage naturel » ; ils préfèrent ouvrir tout de suite le logiciel.

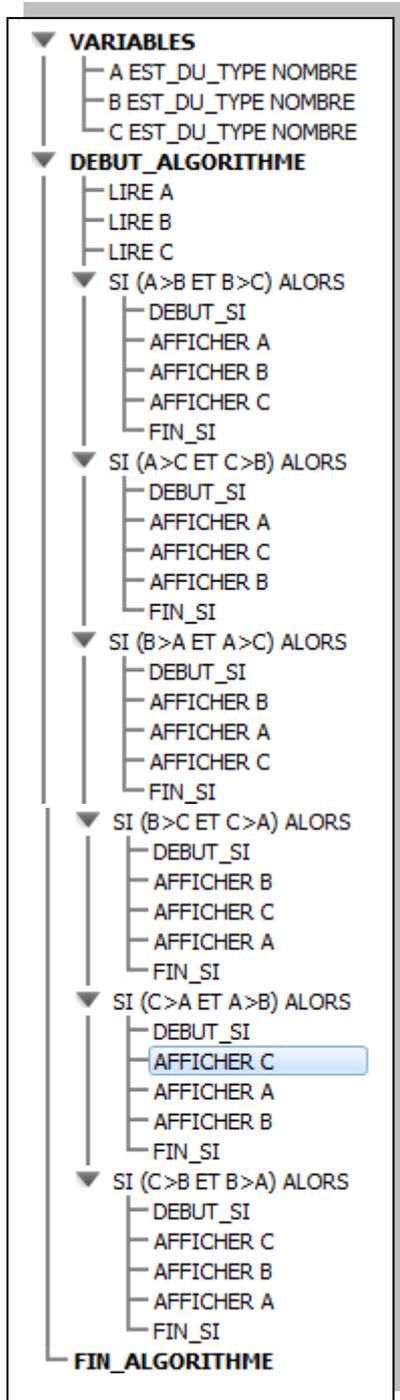
Ceux qui ont transposé leur travail sur algobox, confondent souvent « déclarer la variable » et « lire la variable » et même « lire la variable » et « afficher la variable » ce qui révèle sans doute un manque de pratique.

Les algorithmes proposés testent les 6 cas indépendamment les uns des autres ; un seul groupe utilise le « sinon ». En fait, ils l'ont écrit mais ne se sont pas servi de ce que cela leur apportait ; ils ont donc aussi testé les 6 cas les uns après les autres.

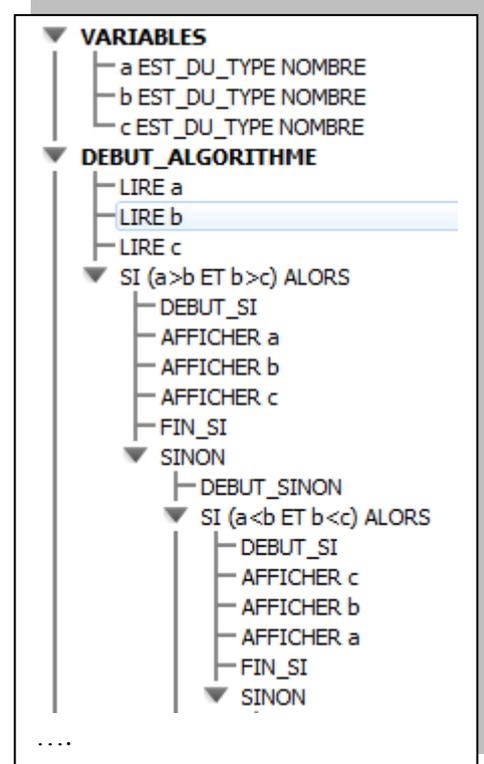
Nous commençons alors ensemble un algorithme privilégiant l'utilisation de « sinon » que les élèves termineront seuls. Je leur distribue un corrigé au début de la séance suivante pour qu'ils puissent tous répondre à la question 4.

## Travaux d'élèves :

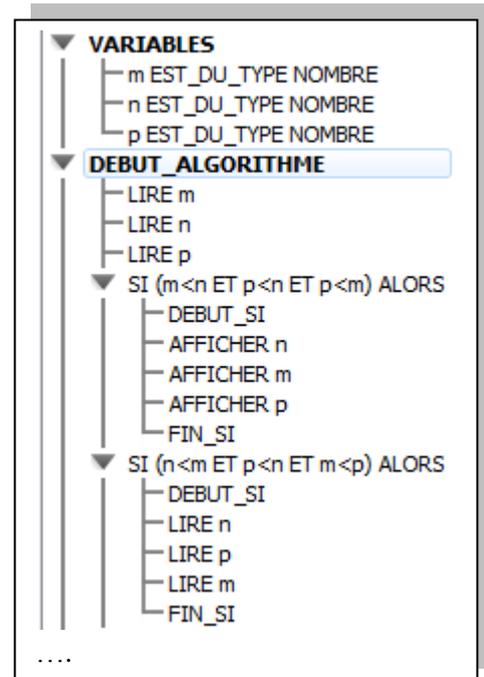
Algorithme de tri décroissant testant les 6 cas indépendamment :



Algorithme (partiel) de tri décroissant dans lequel les élèves pensent utiliser « sinon » :



Confusion entre « lire » et « afficher » :



## Deuxième séance : question 3 du T.D

### Objectifs :

- Décomposition d'un nombre de trois chiffres ;
- (Re)découvrir la fonction partie entière ou celle donnant le reste dans une division euclidienne.

### Déroulement de la séance :

Beaucoup de difficultés pour isoler le chiffre des unités d'un nombre à trois chiffres... Les élèves ne voient pas quelle démarche adopter, quelle opération proposer. Nombreux sont ceux qui pensent qu'il suffit d'entrer les chiffres  $a$ ,  $b$  et  $c$  puis de nommer «  $abc$  » le nombre à trois chiffres pour que le logiciel distingue les trois chiffres.

Deux idées différentes émergent finalement :

- « tant que  $x > 10$ ,  $x$  prend la valeur  $x - 10$  » ;
- Utiliser la partie entière pour trouver le chiffre des centaines qui est alors  $E\left[\frac{n}{100}\right]$ .

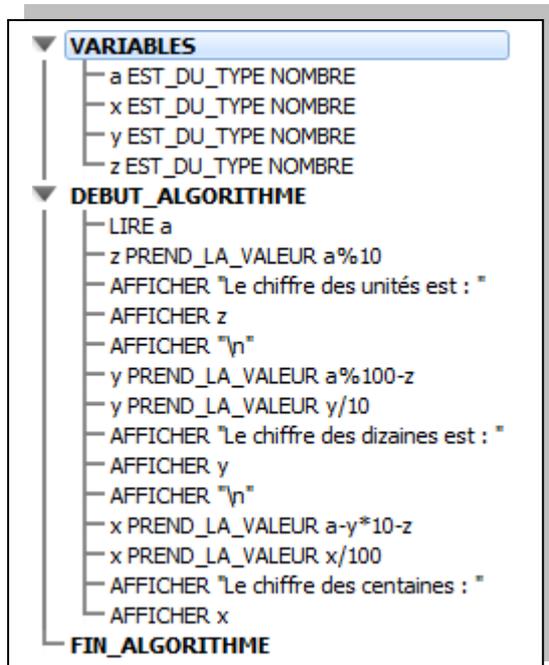
C'est une fonction que nous avons découverte l'année précédente lors du travail sur l'exercice « enchainement d'entiers »

Je propose d'utiliser une division euclidienne. On découvre ensemble la fonction «  $x\%y$  » d'algoBox.

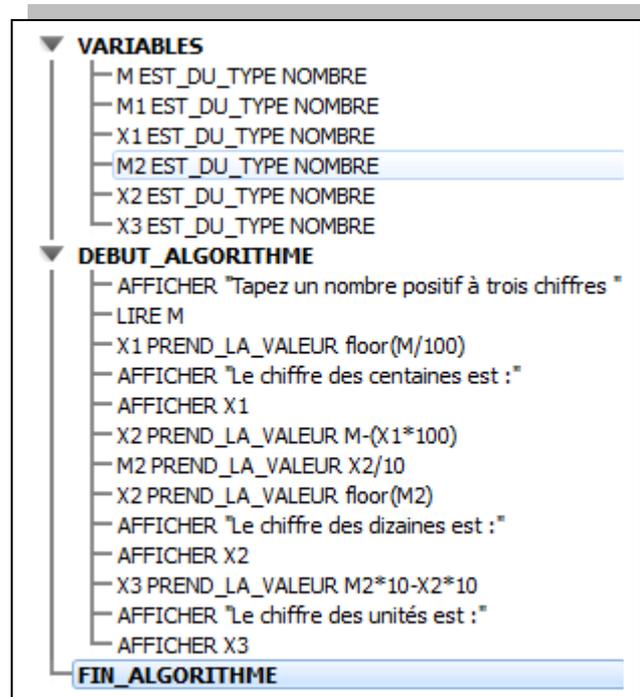
Les élèves choisissent une des trois méthodes citées pour répondre à la question 3. Le travail est à terminer, si besoin, pour la semaine suivante.

### Travaux d'élèves :

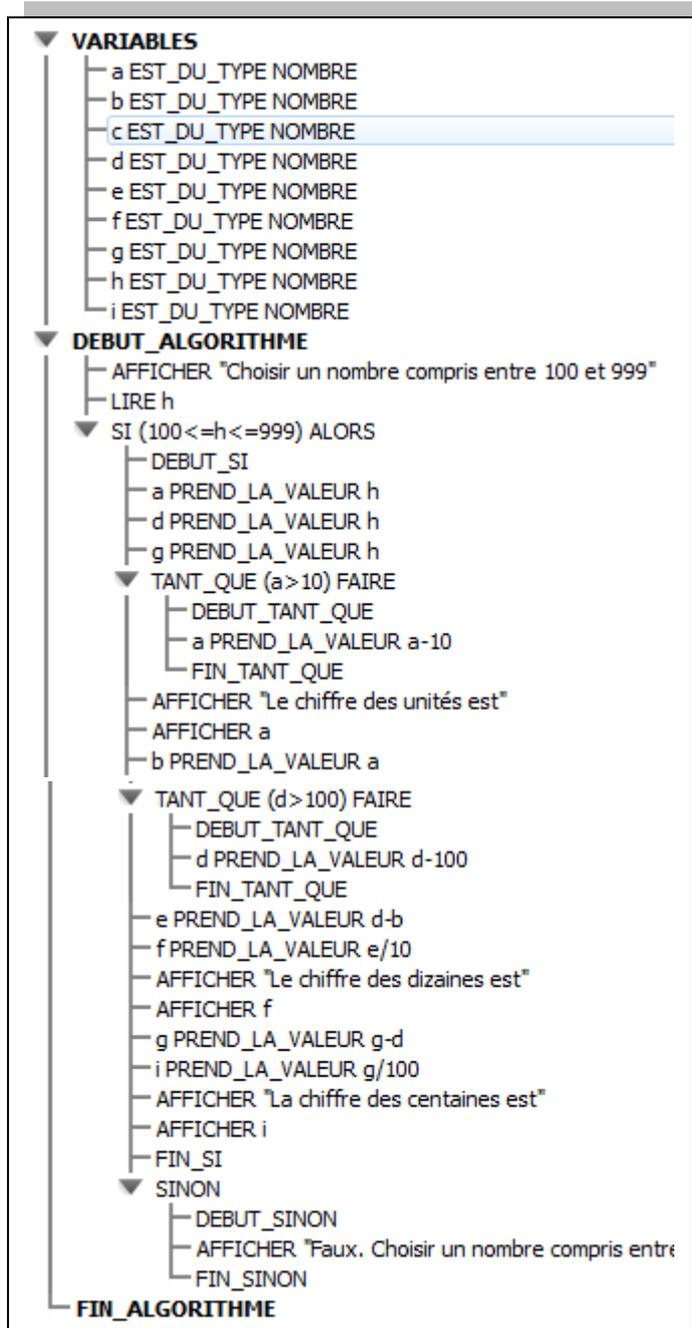
Algorithme utilisant la division euclidienne :



Algorithme utilisant la partie entière :

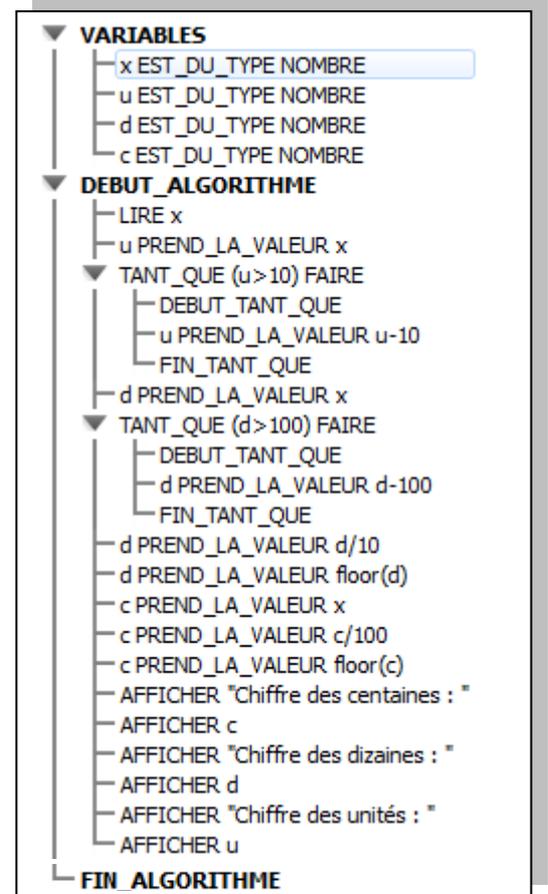


Algorithme utilisant des soustractions successives :



Algorithme mélangeant 2

Méthodes :



### Troisième séance : question 4 du T.D

Objectif :

- Utiliser les questions 2 et 3 pour trouver  $K(n)$ .

Déroulement de la séance :

Reconstituer le nombre de trois chiffres connaissant ses trois chiffres pose peu de problème.

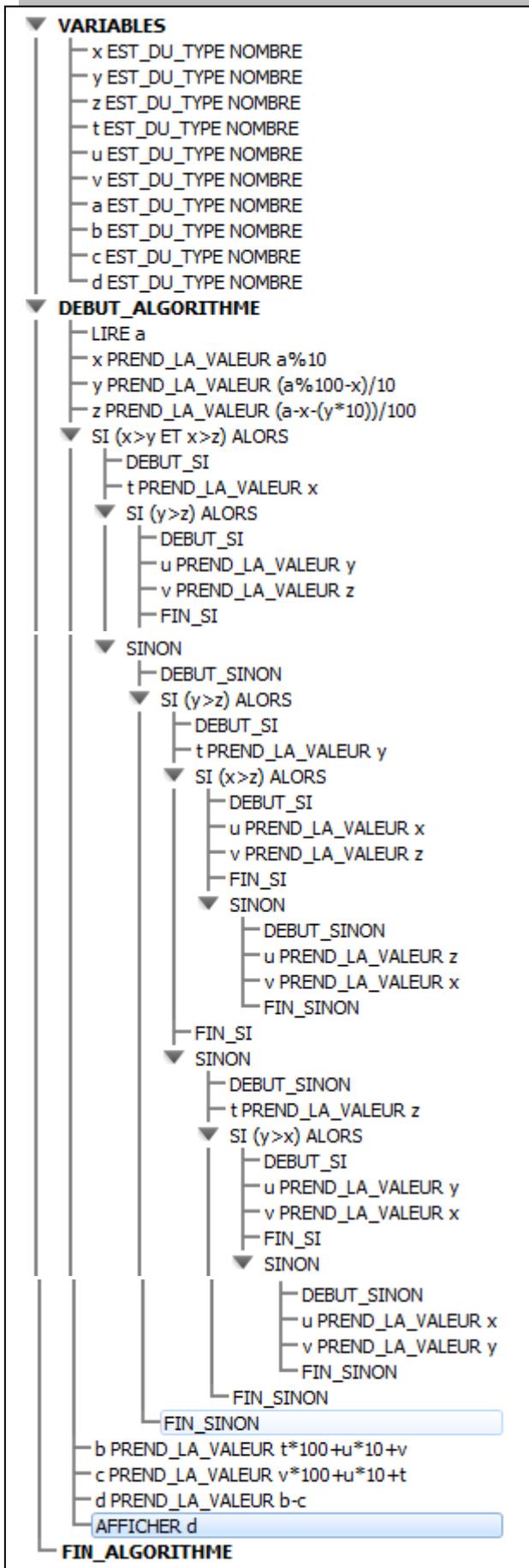
Certains réalisent que, selon les choix de méthodes faits aux questions précédentes, la réponse à la question 4 est plus ou moins rapide ; ils envisagent parfois de modifier leurs algorithmes pour qu'ils soient moins fastidieux à recopier. Ils abordent pour la première fois la question de l'efficacité d'une méthode.

Plusieurs groupes donnent rapidement une réponse à la question 4 ; je leur propose alors de modifier ce dernier algorithme pour montrer que, quelque soit le nombre de 3 chiffres que l'on entre, on obtient 495 par itérations successives.

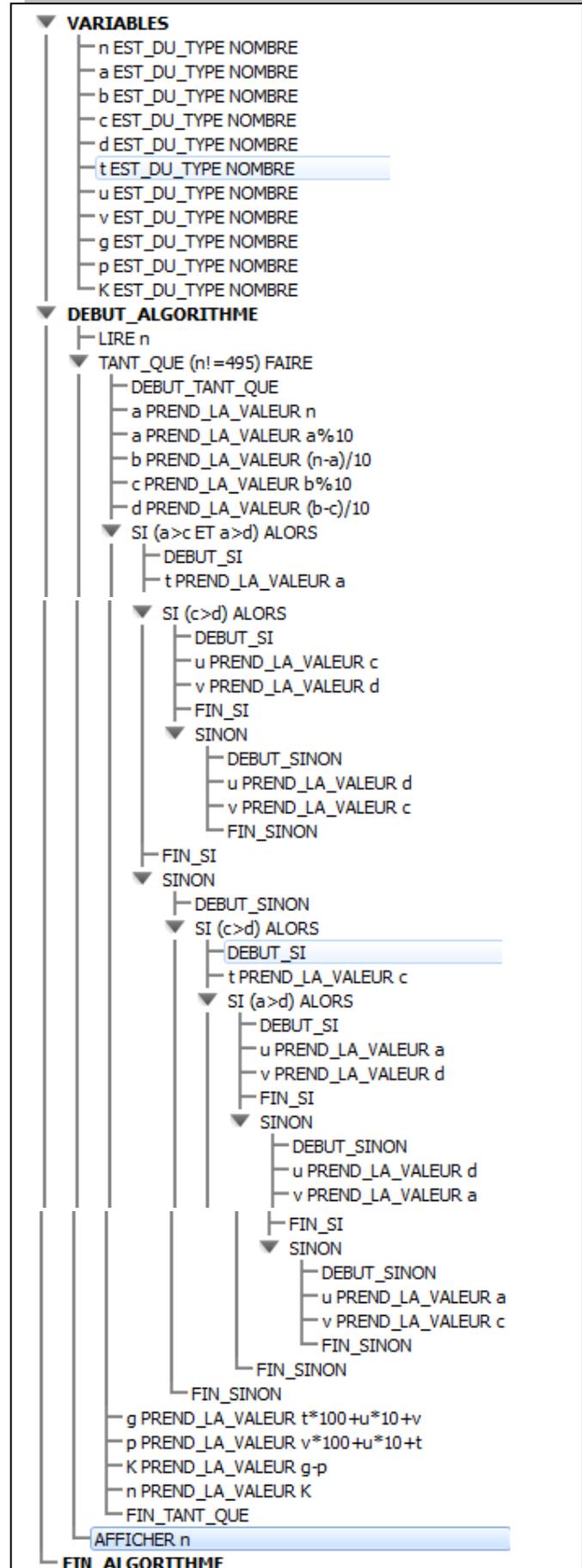
Quelques groupes trouvent une réponse en utilisant comme test d'arrêt «  $K(n) \neq 495$  » mais quand je leur fait remarquer qu'il vaudrait mieux trouver un test d'arrêt qui n'utilise pas ce que l'on cherche à montrer, un seul groupe modifie correctement son algorithme.

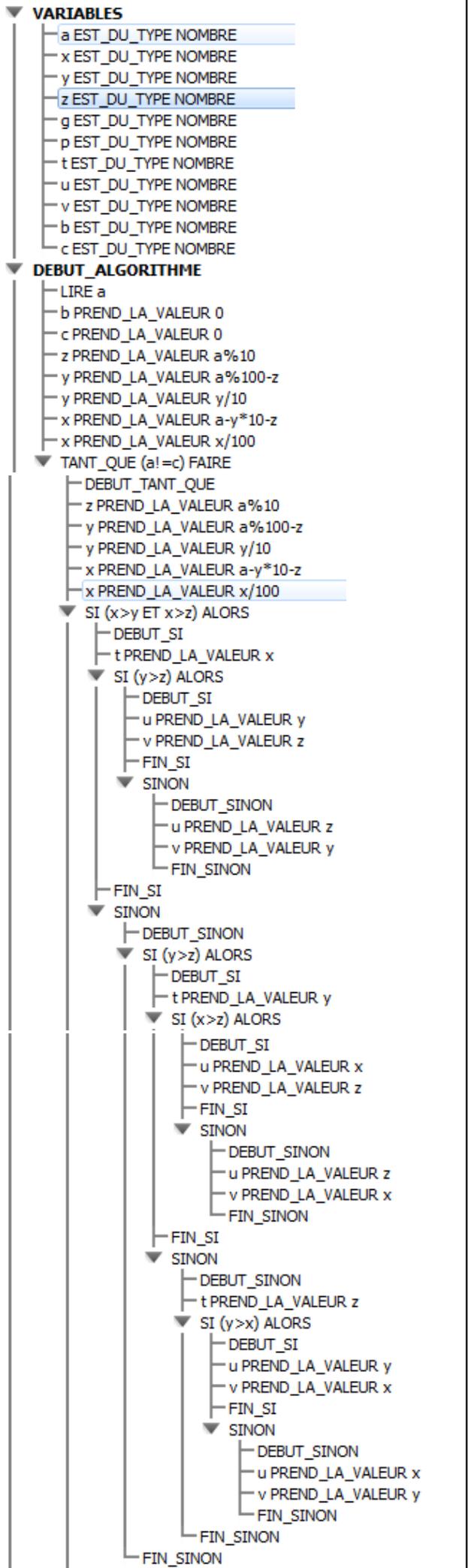
## Travaux d'élèves :

Algorithme répondant à la question 4 :



Algorithme répondant partiellement à la question subsidiaire :





Algorithme répondant à la question Subsidaire.

