Algorithme PanaMaths
→ PGCD de deux entiers non nuls

Introduction : quelques éléments mathématiques

L'algorithme présenté ici est un petit algorithme fournissant le PGCD (Plus Grand Commun Diviseur. Il s'agit du plus grand diviseur commun positif de A et B).

Quels que soient les signes des entiers A et B, nous nous ramènerons toujours au cas où A et B sont deux entiers naturels non nuls en considérant les valeurs absolues des deux entiers. En effet, rappelons que l'on a : $\mathrm{PGCD}(\mathrm{A},\mathrm{B}) = \mathrm{PGCD}(|\mathrm{A}|,|\mathrm{B}|)$.
Dans ce qui suit, les deux entiers A et B sont donc supposés naturels et non nuls.

En effectuant la division euclidienne de A par B, on obtient :

$$\mathrm{A} = \mathrm{B}\times\mathrm{Q} + \mathrm{R} \text{ avec } 0 \leq \mathrm{R} < \mathrm{B}$$

Rappelons que l'on a la propriété fondamentale suivante :

$$\mathrm{PGCD}(\mathrm{A},\mathrm{B}) = \mathrm{PGCD}(\mathrm{B},\mathrm{R})$$

L'algorithme d'Euclide repose sur cette propriété fondamentale : en utilisant cette propriété, nous construisons une suite d'égalités découlant de divisions euclidiennes successives dans lesquelles, les restes décroissent strictement. Notons A_n et B_n les valeurs des entiers à l'étape *n* et R_n le reste de la division euclidienne de A_n par B_n. D'après la propriété fondamentale, on aura : $\mathrm{PGCD}(\mathrm{A}_n,\mathrm{B}_n) = \mathrm{PGCD}(\mathrm{B}_n,\mathrm{R}_n)$.
On a donc : $\mathrm{A}_{n+1} = \mathrm{B}_n$ et $\mathrm{B}_{n+1} = \mathrm{R}_n$.
Comme nous venons de le mentionner, l'élément déterminant relatif à la suite (B_n) est le suivant : comme $\mathrm{B}_{n+1} = \mathrm{R}_n < \mathrm{B}_n$, la suite (B_n) est une suite d'entiers naturels <u>strictement</u> décroissante. Ainsi, l'un des restes sera nul et tous les restes ultérieurs également. Le PGCD cherché sera alors le premier reste non nul, c'est-à-dire le dernier terme non nul de la suite (B_n).

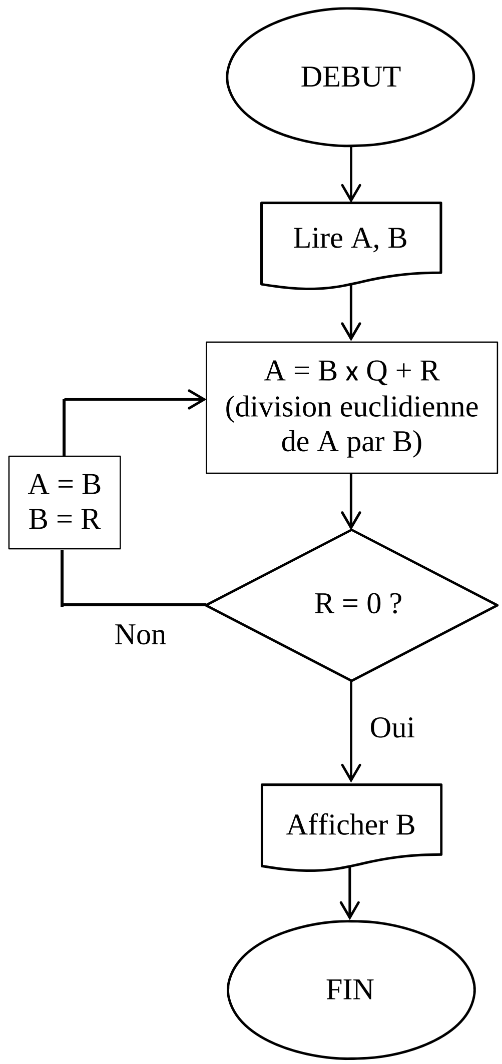
Ainsi, au niveau de l'algorithme, on travaillera essentiellement avec la suite récurrente double :

$$\begin{cases} (\mathrm{A}_0\,;\mathrm{B}_0) = (\mathrm{A}\,;\mathrm{B}) \\ (\mathrm{A}_{n+1}\,;\mathrm{B}_{n+1}) = (\mathrm{B}_n\,;\mathrm{R}_n) \\ \quad \text{où } \mathrm{A}_{n+1} = \mathrm{B}_n\times\mathrm{Q}_n + \mathrm{R}_n \text{ avec } 0 \leq \mathrm{R}_n < \mathrm{B}_n \end{cases}$$

A chaque étape, on testera la nullité éventuelle de R_n.

Une petite remarque pour finir : traditionnellement, on pose le problème de recherche du PGCD de deux entiers sous la forme $\mathrm{PGCD}(a,b)$ avec $a \geq b > 0$. Cette dernière contrainte est en fait superflue au niveau de la mise en œuvre de l'algorithme. En effet, si on cherche $\text{PGCD}(a,b)$ avec $0 < a \leq b$, la première étape de l'algorithme d'Euclide consiste à effectuer la division euclidienne de a par b et on obtient simplement, a étant inférieur à b : $a = 0 \times b + a$. D'après la propriété fondamentale, on a alors : $\text{PGCD}(a,b) = \text{PGCD}(b,a)$. Ainsi, dès la deuxième étape, on se retrouvera dans la situation « standard » ($a \geq b > 0$).

Organigramme



Au niveau de la mise en œuvre de cet algorithme simple, on peut ajouter à la lecture des variables A et B un test pour garantir, avant d'entrer dans la boucle principale, que les nombres saisis sont bien des entiers non nuls (cf. l'algorithme AlgoBox fourni ci-après).

L’algorithme AlgoBox

Voici l’algorithme que vous pouvez tester en ligne :

```
PGCD  -  04.08.2012

******************************************
Cet algorithme détermine le PGCD de deux nombres entiers non nuls.
******************************************

VARIABLES
  A EST_DU_TYPE NOMBRE
  B EST_DU_TYPE NOMBRE
  R EST_DU_TYPE NOMBRE
DEBUT_ALGORITHME
  //Première saisie de la valeur de la variable A.
  AFFICHER "Saisir la valeur du premier entier."
  LIRE A
  TANT_QUE (A-floor(A)!=0 OU A==0) FAIRE
    DEBUT_TANT_QUE
    AFFICHER "ATTENTION ! Vous devez saisir un entier non nul !"
    LIRE A
    FIN_TANT_QUE
  AFFICHER "Premier entier considéré : "
  AFFICHER A
  //Première saisie de la valeur de la variable B.
  AFFICHER "Saisir la valeur du deuxième entier."
  LIRE B
  TANT_QUE (B-floor(B)!=0 OU B==0) FAIRE
    DEBUT_TANT_QUE
    AFFICHER "ATTENTION ! Vous devez saisir un entier non nul !"
    LIRE B
    FIN_TANT_QUE
  AFFICHER "Deuxième entier considéré : "
  AFFICHER B
  //Les valeurs des variables A et B sont valides. On démarre la détermination de leur PGCD.
  //Préparation de l'affichage final.
  AFFICHER "Le PGCD de "
  AFFICHER A
  AFFICHER " et de "
  AFFICHER B
  AFFICHER " vaut "
  A PREND_LA_VALEUR abs(A)
  B PREND_LA_VALEUR abs(B)
  R PREND_LA_VALEUR A%B
```

```
36    TANT_QUE (R!=0) FAIRE
37      DEBUT_TANT_QUE
38      A PREND_LA_VALEUR B
39      B PREND_LA_VALEUR R
40      R PREND_LA_VALEUR A%B
41      FIN_TANT_QUE
42    AFFICHER B
43    AFFICHER "."
44    SI (B==1) ALORS
45      DEBUT_SI
46      AFFICHER " Les deux entiers sont premiers entre eux."
47      FIN_SI
48  FIN_ALGORITHME
```

Remarques :

- Quelques commentaires ont été ajoutés pour rendre l’algorithme plus lisible.
- Un test double est effectué sur les variables A et B puisque celles-ci doivent être :
  - Non nulles.
  - Entière (A–floor(A) correspond à la différence entre A et sa partie entière et est nulle si, et seulement si, A est entière. Il en va bien sûr de même pour B).
- La boucle principale de l’algorithme correspond aux lignes 36 à 41.