

## 9/4.5

# Instructions arithmétiques

---

### PGCD - PPCM - FRACTIONS DE NOUVELLES FONCTIONS ARITHMÉTIQUES POUR VOTRE CPC

Votre AMSTRAD est un mathématicien hors pair, vous n'en doutez certainement pas. Grâce à des programmes plus ou moins étoffés, il vous est possible de résoudre quasiment tous les problèmes qui peuvent se poser.

Nous avons pensé dans ce chapitre, en plus d'un rapide rappel sur les notions de PGCD et PPCM, vous proposer de nouvelles fonctions mathématiques permettant de calculer automatiquement le PGCD, le PPCM de deux nombres, plus une fonction permettant de simplifier des fractions.

Ces fonctions s'appelleront : **PGCD**, **PPCM** et **FAC**, et auront la syntaxe suivante :

**PGCD,@x,@a,@b** pour obtenir  $x = \text{PGCD}(a,b)$  ;

**PPCM,@x,@a,@b** pour obtenir  $x = \text{PPCM}(a,b)$  ;

**FAC,@a,@b** qui donnera dans a la simplification du numérateur, et dans b la simplification du dénominateur, de la fraction a/b.

Pour ces trois instructions, les variables a, b et x devront être de type ENTIER.

## 9/4.5.1

### Le PGCD

---

#### DÉFINITION

Le PGCD de plusieurs nombres est l'abréviation de Plus Grand Commun Diviseur, et correspond par définition au plus grand nombre

qu'il est possible de trouver, qui les divisent tous exactement, c'est-à-dire qui produit un quotient entier, et aucun reste.

Prenons pour exemple les deux nombres 30 et 45 : le chiffre 3 est un diviseur de 30 ( $30 / 3 = 10$ ) et 45 ( $45 / 3 = 15$ ), mais ce n'est pas le plus grand. En effet, 5, qui est supérieur à 3 est aussi un diviseur de 30 et 45. 10 ne peut pas être le PGCD car 10, divisant bien 30, ne divise pas 45. Or il est possible de trouver un nombre plus grand encore, qui divise exactement 30 et 45 : il s'agit de 15. Comme on ne trouvera pas de nombre supérieur divisant exactement ces deux nombres (tout nombre supérieur donne un reste non nul), 15 est bien le PGCD de 30 et 45.

On notera :  $\text{PGCD}(30,45) = 15$

Il est bien sûr possible de trouver les PGCD de plusieurs nombres, mais nous nous limiterons ici à deux.

### RECHERCHE DU PGCD MANUELLEMENT

Il est hors de question, comme dans l'exemple ci-dessus, de rechercher le PGCD de deux nombres par essais successifs de tous les diviseurs possibles des nombres.

La méthode habituelle se décompose en deux étapes :

1) Décomposer chacun des nombres en facteurs premiers : on divise chacun d'eux par tous les nombres premiers possibles (sauf 1), en commençant par le plus petit : 2, 3, 5, 7, 11, 13, 17, 19, 23 ... etc.

2) Effectuer le produit de chacun des nombres premiers qui sont communs, en les affectant de leur plus petit exposant.

Reprenons nos deux nombres 30 et 45 :

$$30 = 2 * 15 = 2 * 3 * 5 = 2^1 * 3^1 * 5^1$$

$$45 = 3 * 15 = 3 * 3 * 5 = 3^2 * 5^1$$

Les nombres communs sont donc 3 et 5, et les plus petits exposants sont donnés par  $3^1$  et  $5^1$ .

On retrouve ainsi le  $\text{PGCD}(30,45) = 3^1 * 5^1 = 3 * 5 = 15$ .

Nous vous proposons ci-après la liste des nombres premiers de 1 à 1 000, pour vous permettre d'effectuer ce type de calcul :

1 2 3 5 7

11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89

97

101 103 107 109 113 127 131 137 139 149 151 157 163 167 173 179  
181 191 193 197 199

211 223 227 229 233 239 241 251 257 263 269 271 277 281 283

293

307 311 313 317 331 337 347 349 353 359 367 373 379 383 389

397

401 409 419 421 431 433 439 443 449 457 461 463 467 479 487 491 499

503 509 521 523 541 547 557 563 569 571 577 587 593 599

601 607 613 617 619 631 641 643 647 653 659 661 673 677 683 691

701 709 719 727 733 739 743 751 757 761 769 773 787 797

809 811 821 823 827 829 839 853 857 859 863 877 881 883 887

907 911 919 929 937 941 947 953 967 971 977 983 991 997

A titre d'exercice de programmation, vous pourrez vérifier si un nombre est premier ou non, en utilisant l'algorithme qui consiste à diviser ce nombre par tous les nombres premiers, en commençant par le plus petit, et en vous arrêtant au nombre juste supérieur à la racine carré du nombre traité. Vous pouvez ainsi calculer des nombres premiers supérieurs à 1 000, dans la limite de la capacité de calcul de votre AMSTRAD.

### INTÉRÊT DU PGCD

Le calcul du PGCD de deux nombres est utile pour réduire une fraction de deux nombres. Il suffit de diviser chacun des nombres par leur PGCD, la fraction ainsi trouvée est dite irréductible.

Ainsi, si l'on avait à effectuer la réduction de 30/45, on divise 30 par 15, ce qui donne 2, et 45 par 15 pour trouver 3, ce qui nous permet d'écrire :

$$30 / 45 = 2 / 3$$

Il vous est peut être apparu que l'utilité du PGCD est faible avec les nombres donnés en exemple, essayez donc de réduire 3256 / 12210 (la réponse est 4 / 15).

### PGCD ET PROGRAMMATION

Calculer le PGCD avec l'AMSTRAD-CPC n'est pas une mince affaire. Nous n'allons pas utiliser l'algorithme vu ci-dessus, car il gaspille beaucoup de temps, notamment pour la décomposition en facteurs premiers.

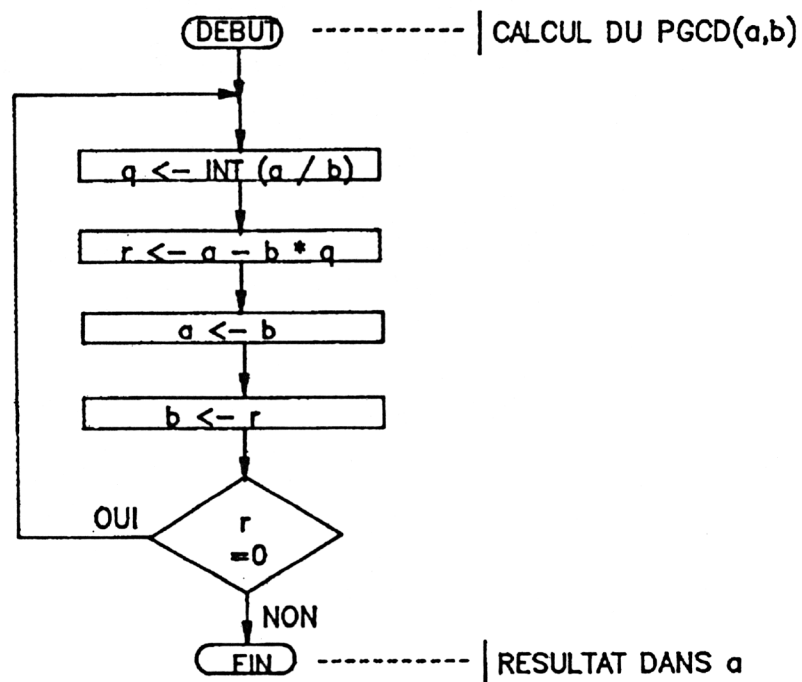
Nous utiliserons un algorithme énoncé par un mathématicien célèbre : EUCLIDE.

Pour trouver le PGCD de deux nombres a et b, effectuer la division entière de a par b. Si le reste est nul alors b est le PGCD, sinon remplacer a par b, et b par le reste, et recommencer, jusqu'à ce que le reste soit nul.

Cet algorithme peut être traduit sous une forme codée :

- DEBUT
  - REPETER
    - Effectuer la division entière de a par b
      - CO
        - Pour obtenir la forme  $a = b * q + r$   
avec a = dividende, b = diviseur  
q = quotient, r = reste
      - FIN CO
    - Remplacer a par b
    - Remplacer b par r
  - JUSQU'A ce que  $r = 0$
- FIN

Nous pouvons représenter cet algorithme sous la forme de l'algorithme suivant :



Dans un langage évolué tel que Turbo-Pascal, la traduction s'effectuera quasi-immédiatement à l'aide de l'algorithme, et sous la forme d'une procédure :

```
PROCEDURE PGCD (a,b : INTEGER);
  {variables locales}
  var
  q,r : integer;
  BEGIN
    REPEAT
      q:= int (a/b);
      r:= a - b * q;
      a:= b;
      b:= r;
    UNTIL r = 0;
  END;
```

Il suffira, après avoir déclaré la variable réponse **x**, de programmer **x := PGCD(nb1,nb2)** pour obtenir le PGCD de **nb1** et **nb2** dans la variable **x**. Il faudra tout de même prendre soin auparavant d'écrire une procédure permettant d'inverser les valeurs **nb1** et **nb2** au cas où **nb2** est supérieur à **nb1**.

En Basic, on préférera l'algorithme qui permettra d'écrire le programme suivant :

```
10 INPUT "Premier nombre";a
20 INPUT "Deuxième nombre":b
30 IF a < b then c = b:b = a: a = c : REM permutation
40 q : INT (a / b):REM ici débute l'algorithme
50 r : a - b * q
60 a = b : b = r
70 IF r < > 0 THEN GOTO 40
80 PRINT "Le PGCD est";a:REM le résultat est dans a
90 END
```

Toute cette étude ne nous permet pas encore d'utiliser l'instruction **!PGCD,@x,@a,@b**, mais nous avons déterminé la façon de résoudre le problème. Nous vous proposerons dans un paragraphe suivant l'instruction **!PGCD**, car nous allons auparavant étudier le PPCM, pour nous apercevoir de sa liaison avec le PGCD.

