ALGORITHMIQUE

AU LYCEE PROFESSIONNEL

OUTIL DE RAISONNEMENT

Compte rendu de travaux

par

Maurice Albet
Henri Capell
Bernard Dunevon
Guy Hacquart

N° 5

Publication de l'I.R.E.M. d'Aix-Marseille

1986

ALGORITHMIQUE
AU LYCEE PROFESSIONNEL
OUTIL DE RAISONNEMENT

ALGORITHMIQUE
AU LYCEE PROFESSIONNEL
OUTIL DE RAISONNEMENT

Maurice Albet
Henri Capell
Bernard Dunevon
Guy Hacquart

Publications de l'IREM d'Aix-Marseille

ISSN : 0297-4347

PREFACE

Pourquoi l'algorithmique au Lycée Professionnel

Il ne s'agit pas pour nous de créer un nouvel enseignement, mais d'introduire une nouvelle démarche (pour le professeur dans son rôle de formateur, pour l'élève dans son rôle d'apprenant) dans le cadre des programmes et des horaires officiels de l'enseignement scientifique des lycées professionnels.

Dans un premier temps nous nous intéressons plus particulièrement aux sections de B.E.P.

Cette démarche consiste à expliciter avec les élèves, un certain nombre de structures algorithmiques rencontrées et choisies avec opportunité.

Ces algorithmes pourront être concrétisés par la réalisation de petits programmes simples (en travaux dirigés de mathématiques), lesquels pourront à leur tour constituer des outils favorisant l'étude de certaines notions.

Les résultats attendus sont :

- la rénovation de l'enseignement de notre discipline en intégrant le phénomène de société qu'est le développement de l'informatique.

- la formation à une méthode de raisonnement structuré/organisé, qui, par transfert pourra s'avérer utile dans la formation professionnelle (cf. GRAFCET, automatismes, commandes numériques, robotique, gestion, etc.).

Dans ce document, nous n'avons pas la prétention de donner un modèle de ce qui peut être fait dans ce domaine. Nous avons simplement voulu recenser et commenter par thèmes un certain nombre d'algorithmes qui apparaissent dans le cours. Ce travail d'inventaire, qui ne peut être exhaustif, nous voudrions qu'il soit, pour nos collègues et pour nous-mêmes, le point de départ, le révélateur d'expériences menées en classe avec les élèves.

C'est dans ce but que nous avons réalisé dans le cadre du P.A.F. une action de 36 heures intitulée : Algorithmique et informatique au lycée professionnel, outils du raisonnement logique, stage qui a regroupé 22 enseignants de mathématiques, et qui a permis ainsi d'enrichir le travail ébauché et d'élargir le champ d'expérimentation.

Equipe "Algorithmique au lycée professionnel"
IREM d'Aix-Marseille

SOMMAIRE

ALGORITHMIQUE
AU LYCEE PROFESSIONNEL
OUTIL DE RAISONNEMENT

PARTIE 1
TRAVAUX
PREPARATOIRES

I. Introduction

1 - "Vous avez dit algorithme ?"

Le mot, en fait n'est pas nouveau. On en trouve les premières traces au Moyen-Age, on disait alors algorismus.
Ce terme, à son origine était lié aux chiffres arabes ainsi qu'à la numération décimale de position, introduite en Europe occidentale par les mathématiciens arabes (école de Bagdad) qui la tenaient eux-mêmes des Indes.

Les historiens semblent s'accorder pour dire que le mot "algorithme" a pour origine le nom du savant perse AL KHOWARIZMI dont l'un des ouvrages d'arithmétique publié au IX ème siècle énonçait, avec les principes de la numération décimale, les règles du calcul écrit.

Initialement donc, un algorithme est une règle de calcul numérique --technique opératoire-- et par là même ne s'applique qu'à des données numériques.

Si la manipulation des algorithmes était, et demeure une chose courante, le mot, lui, fut très peu usité y compris dans l'enseignement des mathématiques. A peine énonçait-t-on l'algorithme d'Euclide (recherche du PGDC de deux entiers), quelquefois l'algorithme d'extraction d'une racine carrée ...

Le développement récent de l'informatique a contribué à accroître considérablement l'usage de ce mot, de même qu'il lui a apporté un sens beaucoup plus large en l'étendant au traitement de données non numériques.
On pourrait accepter la définition suivante :

Un algorithme est une suite ordonnée de traitements élémentaires permettant, à partir d'une situation initiale, (données) d'arriver à la situation finale attendue (résultats) en un nombre fini d'étapes.

Exemple :

La multiplication de 354 par 11, associe au couple (354,11) un produit unique P,

(354,11) ------> P

L'algorithme décrit ci-dessous permet d'obtenir le résultat P par une succession de traitements élémentaires, à savoir : multiplication puis addition de nombres à 1 chiffre (que l'on sait effectuer grace aux tables).

```
   354
x   11
------
   354
  354
  ----
  3894
```

Remarque : Dans ce cas précis, on peut utiliser un autre algorithme

```
3         5         4
  \     /   \     /
    3+5       5+4
     |         |
3    8         9    4
```

2 - Structure rencontrées dans les algorithmes :

On considère qu'il existe 4 types de structures dans les algorithmes élémentaires.

2-1 La structure séquentielle :

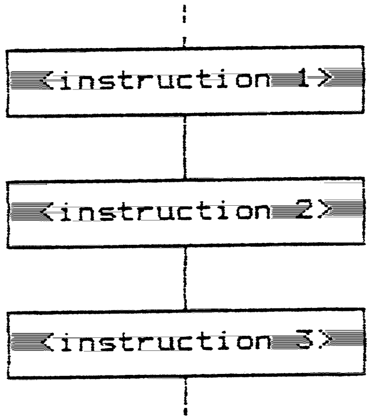
C'est la structure la plus simple, les instructions sont exécutées séquentiellement les unes après les autres.

Exemple :

LIRE A,B,C

D <-- BxB -- 4xAxC

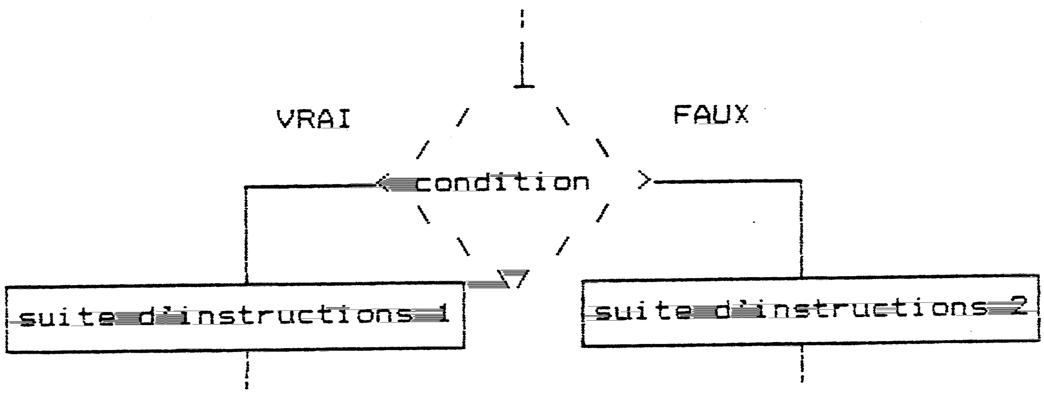
AFFICHER D



2-2 La structure conditionnelle :

Une séquence d'instructions pourra être rompue par la rencontre d'une alternative liée à la valeur de vérité d'un test. Cette instruction alternative qui met en jeu le résultat d'une condition va peut-être orienter le déroulement du programme vers deux directions différentes.

```
! SI <condition vraie>
!     ALORS
!           <séquence d'intructions 1>
!     SINON
!           <séquence d'instructions 2>
! fin de SI
```



Exemple :

```
! SI D<0
!    ALORS
!        AFFICHER 'Pas de racine réelle'
!    SINON
!       ! SI D=0
!       !  ALORS
!       !    AFFICHER 'Une racine double'
!       !  SINON
!       !    AFFICHER 'Deux racines distinctes'
!       ! * fin de SI
! * fin de SI
```

2-3 La structure itérative ou répétitive :

Cette structure sera utilisée quand une séquence d'instructions devra être répétée plusieurs fois. On pourra distinguer plusieurs types de structures répétitives, mais une seule d'entre elles suffirait pour répondre à la plupart des itérations.

C'est la structure dite TANT QUE. (cf. 2-3-2-1)

2-3-1 Nombre d'itérations fixées à l'avance :

Le nombre de répétitions de la séquence d'instructions est fixé avant d'entrer dans la structure, c'est un invariant.

```
! POUR I <-- N1 JUSQU'A N2 FAIRE
!
!       <séquence d'instructions>
!
! * fin de POUR
```

Exemple :

```
! POUR I <-- 1 JUSQU'A 100 FAIRE
!       R <-- I MOD 7
!       Q <-- I DIV 7
!       D <-- I
!       AFFICHER 'Dividende = ',D
!       AFFICHER 'Quotient  = ',Q
!       AFFICHER 'Reste     = ',R
! * fin de POUR
```

(*) avec MOD donnant I modulo 7
avec DIV donnant le quotient entier de I par 7

2-3-2 Le nombre d'itérations est indéterminé :

La séquence d'instructions sera exécutée un certain nombre de fois indéterminé par avance. C'est le résultat d'un test conditionnel qui commandera l'interruption des répétitions.

Deux type d'instructions seront possibles en fonction du moment où se produira le test.

2-3-2-1 Test en début de boucle :

```
! TANT QUE <condition vraie>
!
!          <séquence d'instructions>
!
! * fin de TANT QUE
```

Exemple :

```
LIRE I
! TANT QUE I <= 100
!          LIRE C
!          I <-- I+C
!          AFFICHER I
! * fin de TANT QUE
```

L'itération sera interrompue par l'affectation à I d'une valeur supérieure à 100.

Si la valeur de I était supérieure à 100 avant d'aborder l'instruction TANT QUE, la séquence d'instructions interne à la boucle ne serait pas réalisée une seule fois.

2-3-2-2 Test en fin de boucle :

```
! REPETER
!
!        <séquence d'instructions>
!
! JUSQU'A <condition vraie>
```

Exemple :

```
LIRE I
! REPETER
!        LIRE C
!        I <-- I+C
!        AFFICHER I
! JUSQU'A I>100
```

Ici encore, l'itération sera interrompue par l'affectation à I d'une valeur supérieure à 100. Par contre, même si la valeur de I est supérieure à 100 avant d'aborder l'instruction REPETER ... JUSQU'A, la séquence d'instructions interne à la boucle sera réalisée au moins une fois.

2-4 Les constructions procédurales :

On s'apercevra rapidement que certaines séquences d'instructions peuvent être utilisées plusieurs fois dans un même algorithme. C'est le cas par exemple de certaines fonctions mathématiques. Il sera intéressant de donner un nom ou label à cette séquence d'instructions et de l'appeler si nécessaire par ce nom, en lui transmettant la valeur des paramètres variables qu'elle utilise lors de son exécution.

Ce type de séquence sera appelé procédure.

Exemple :

```
! PROCEDURE delta(A,B,C)
!     D = BxB - 4xAxC
!     RESULTAT D
! * fin de PROCEDURE
```

Cette procédure pourra être utilisée directement dans l'algorithme de résolution de l'équation :

$$3X^2 + 5X - 7 = 0$$

Si delta(3,5,7)<0 ALORS ...

2-5 Les constructions procédurales récursives :

Une procédure mettra en jeu la récursivité, s'il est possible de faire appel à elle même dans sa propre définition.

Exemple :
Une illustration connue de la récursivité est donnée par le calcul de la factorielle d'un entier naturel.

$$n! = n \times (n-1) \times (n-2) \times \ldots \times 1$$

```
PROCEDURE fact(n)
      Si n=0
         ALORS
               f = 1
         SINON
               f = n x fact(n-1)
      * fin de SI
      RESULTAT f
* fin de PROCEDURE
```

2-5 Conclusion :

D'une manière générale, la plupart des algorithmes que nous aurons l'occasion de mettre en oeuvre seront constitués par un assemblage plus ou moins complexe de ces différentes structures élémentaires.

3 - Ecriture des algorithmes :

Il n'existe pas de notation normalisée pour l'écriture des algorithmes. Il semble bien d'autre part, que les mathématiciens et les informaticiens, utilisateurs privilégiés d'algorithmes, ne souhaitent pas qu'il en existe une. Un impératif cependant, c'est que leur explicitation soit dépourvue de toute ambigüité.

Cependant l'usage fait que l'on trouve des algorithmes écrits de deux façons différentes :

- Description à travers la langue naturelle.
- Description à travers un organigramme ou algorigramme. Là encore, malgré un effort d'uniformisation on constate une certaine diversité dans les symboles utilisés.

Exemple : Résolution de l'équation du premier degré.

Langue naturelle (Résolution de l'équation aX + b = 0)

```
VARIABLES
          a,b,X : nombres réels
DEBUT
     LIRE a
     LIRE b
     SI a = 0
          ALORS
               SI b = 0
                    ALORS
                    AFFICHER 'Tout nombre est solution'
                    SINON
                    AFFICHER 'Pas de racine'
               * fin de SI
          SINON
               AFFICHER 'La racine est ', -b/a
     * fin de SI

FIN
```

Algorigramme

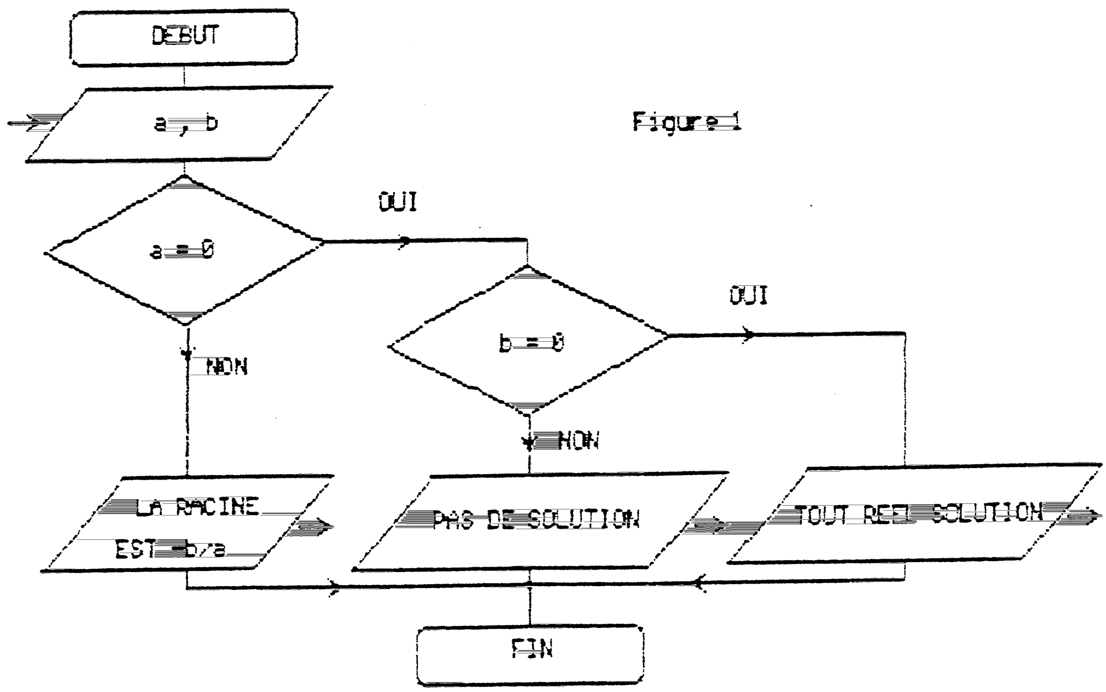


Figure 1

II. Schéma de la recherche

La recherche sur l'introduction des démarches algorithmiques dans les Lycées Professionnels par l'équipe de l'IREM est centrée sur le programme de mathématiques de la classe de seconde B.E.P.

L'étude des algorithmes et leur introduction devra être présentée dans le contexte particulier de la classe, c'est à dire: à partir de l'énoncé du thème de la leçon, de sa place dans la progression annuelle, de sa spécificité en fonction de la section professionnellle concernée.

Chaque notion d'algorithme devra s'insérer dans un canevas de présentation, qui pourra permettre de cerner le plus précisément possible, le problème développé, les différentes étapes de la démarche, les différents algorithmes auxquels elles amènent, et si possible, le compte rendu d'une ou plusieurs exploitations en classe. Enfin, ce canevas devra comporter une étude du travail réalisé en situation avec les élèves, analyse, critiques, et prolongements envisagés.

Le canevas proposé est le suivant :

1 - Présentation du problème :

- Notion de mathématique abordée
- Les difficultés rencontrées lors de son introduction avec les élèves
- Sa place dans la progression annuelle
- Les prérequis
- Les objectifs à atteindre
- Résumé succint des différentes méthodes qui auraient pu être employées pour aborder ce problème avec les élèves
- etc.

2 - Présentation de la démarche algorithmique :

Cette partie permettra de présenter les différents algorithmes rencontrés au cours de l'étude du problème, elle montrera tout d'abord les algorithmes les plus intuitifs pour, en affinant le raisonnement, arriver à des algorithmes plus performants ou plus "beaux".

La présentation de chaque algorithme sera faite en français dans un langage clair et ne prêtant pas à confusion. Elle pourra être illustrée par un algorigramme lorsque celui-ci apportera une clarification.

La rédaction de l'algorithme devra éviter de proposer des instructions de mise en page lorsque celles-ci ne sont pas nécessaires à la compréhension de la résolution du problème.

Chaque algorithme pourra être traduit dans un ou plusieurs langages informatiques. On favorisera l'utilisation du LSE à cause de sa portabilité sur les matériels utilisés dans les classes. La présentation en BASIC impliquera un BASIC "standard" (Microsoft par exemple) en évitant toute instruction particulière à un BASIC (le cas échéant, préciser le type de BASIC employé).

Il serait souhaitable qu'une version en Pascal soit présentée.

3 - Utilisation en classe :

Cette troisième partie devra décrire dans le cas envisagé l'introduction de la démarche algorithmique en classe, quels sont les documents à fournir aux élèves, quels sont les documents qu'ils devront rendre pour l'évaluation des acquis.

4 - Présentation d'une expérience :

Compte rendu détaillé d'une expérimentation en classe.

5 - Ananlyse, critiques, prolongements :

Cette partie devra présenter les observations sur l'expérience, l'analyse du comportement des élèves, envisager les modifications à apporter au déroulement du cours, prévoir les prolongements possibles.

III. Les multiples d'un entier naturel

1 - Présentation du problème :

La notion de multiple d'un entier naturel semble une notion relativement bien acquise par les élèves des classes de B.E.P. . Bien des fois, il est quand même nécessaire de leur rappeler que le mot multiple a la même racine que multiplication. Ceci fait, ils retrouvent rapidement les multiples de 7 en remarquant que cet ensemble débute par la "table de multiplication par 7". La généralisation de la définition sera plus difficile à obtenir d'eux.

Les multiples d'un entier naturel n s'obtiennent en multipliant n par la suite des entiers naturels.

On pourra envisager le cas de la multiplication par 0, qui implique que 0 est multiple de tout entier naturel. Cela pourra poser un problème pour la détermination du Plus Petit Multiple Commun à deux ou plusieurs nombres, qui se trouve être dans ces conditions 0.

Certains élèves pourront remarquer que l'on peut obtenir les multiples de n par additions successives de ce nombre. Cela pourra donner lieu à différents algorithmes de recherche.

L'ensemble des multiples d'un entier naturel étant infini, il sera souhaitable de se limiter dans la recherche à l'ensemble des multiples d'un entier inférieurs à un nombre donné.

2 - Recherche des multiples d'un entier naturel :

2-1 Recherche par addition : Algorithme

Variables : A : nombre dont on recherche les multiples
MAX : borne maximum de recherche
M : multiple obtenu

Remarque: L'affectation d'une valeur à une variable sera notée par le symbole := ou ←

```
DEBUT

LIRE A

LIRE MAX

M := A

TANT QUE M ≤ MAX FAIRE

     AFFICHER M

     M := M+A

% fin de TANT QUE

FIN
```

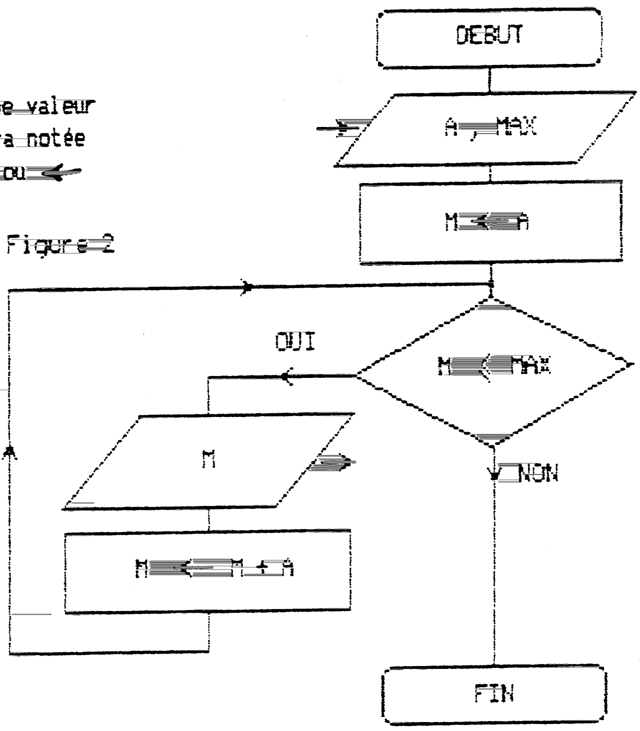


Figure 2

2-2 Recherche par multiplication :

L'algorithme obtenu par la recherche par addition est plus simple que celui qui pourra être écrit plus intuitivement en utilisant la multiplication, qui fera intervenir une variable supplémentaire contenant le coefficient.

Algorithme

Variables : A : entier dont on recherche les multiples
MAX : borne maximum de recherche
N : coefficient entier
M : multiple obtenu

```
DEBUT
LIRE A
LIRE MAX
N := 0
M := A
TANT QUE M ≤ MAX FAIRE
    AFFICHER M
    N := N+1
    M := N x A
* fin de TANT QUE
FIN
```

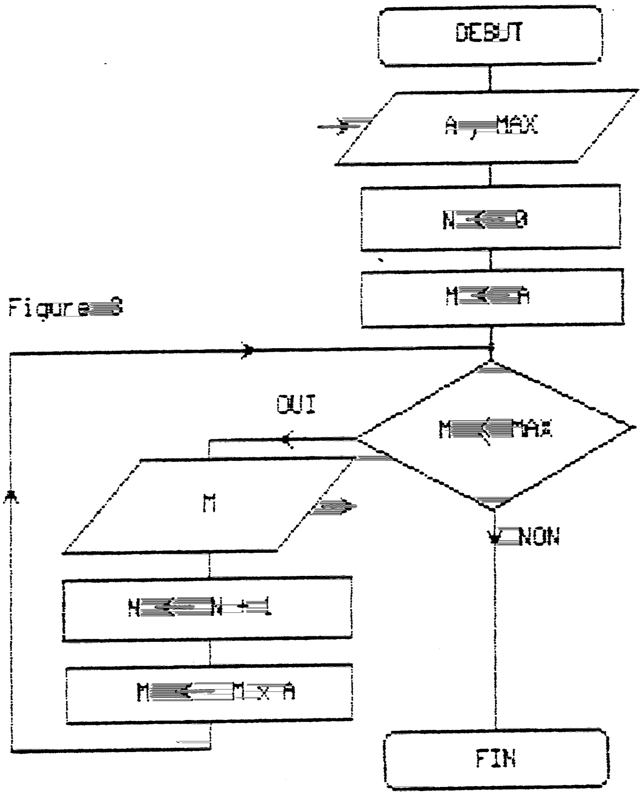


Figure 8

3 - Multiples d'un entier compris entre deux bornes :

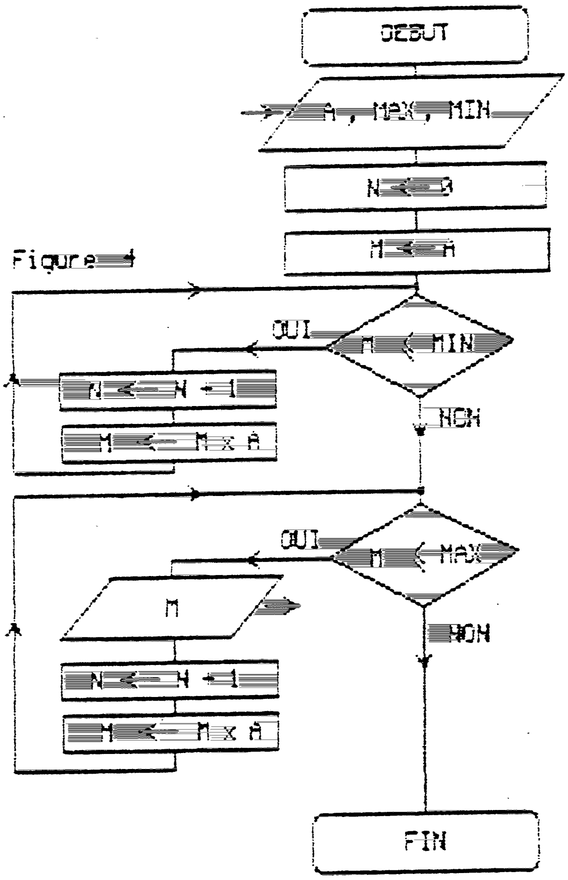
On pourra proposer l'étude de l'algorithme de recherche de l'ensemble des multiples d'un entier compris entre deux entiers donnés.

Algorithme

Variables : A : entier dont on recherche les multiples
MIN : borne minimun de recherche
MAX : borne maximum de recherche
N : coefficient entier
M : multiple obtenu

```
DEBUT
LIRE A
LIRE MIN
LIRE MAX
N := 0
M := A
TANT QUE M ≤ MIN FAIRE
    N := N+1
    M := A x N
‡ fin de TANT QUE
TANT QUE M ≤ MAX FAIRE
    AFFICHER M
    N := N+1
    M := N x A
‡ fin de TANT QUE
FIN
```

Figure 4



IV. Les diviseurs d'un entier naturel

1 - Présentation du problème :

Le cours de mathématiques de BEP commence en général par une nécessaire révision des ensembles numériques. Si, comme on vient de le dire cette révision est nécessaire, elle risque cependant de paraître superflue aux élèves qui ont "subi" ce cours de nombreuses fois. Aussi peut-on apporter un regain d'intérêt par l'introduction d'une démarche intellectuelle différente appuyée sur l'algorithmique et l'informatique en tant qu'outils du raisonnement logique.

Le premier chapitre porte sur l'ensemble des entiers naturels, où les notions de diviseurs d'un entier et de nombres premiers sont loin d'être maîtrisées.

La notion de diviseur est en général définie à partir de la division euclidienne de deux nombres dont le reste est nul. A savoir, B est un diviseur de A si et seulement si le reste de la division euclidienne de A par B est nul.

Une autre méthode pour reconnaître si B divise A consiste à vérifier que le quotient de A par B est un nombre entier.

Quel algorithme peut-on développer pour reconnaître si B divise A ?

```
DEBUT

LIRE A

LIRE B

SI B = 0 ALORS
         AFFICHER "Division par 0 impossible"

     SINON
         SI B ≤ A ALORS
                     Q := A/B
                     SI Q = ENT(Q) ALORS
                                 AFFICHER " B divise A "

                              SINON
                                 AFFICHER " B ne divise pas A "

                     * fin de SI

                  SINON
                     AFFICHER " B ne divise pas A "

         * fin de SI


* fin de SI

FIN
```

Dans les algorithmes à venir, les conditions de divisibilité développées ci-dessus seront considérées comme implicites afin de ne pas alourdir inutilement le raisonnement.

2 - Recherche des diviseurs d'un entier naturel :

Algorithme

```
DEBUT

LIRE A

X := 1

TANT QUE X ≤ A FAIRE

    Q := ENT(A/X)

    R := A - X x Q

    SI R = 0 ALORS
                  AFFICHER X

    * fin de SI

    X := X+1

* fin de TANT QUE

FIN
```

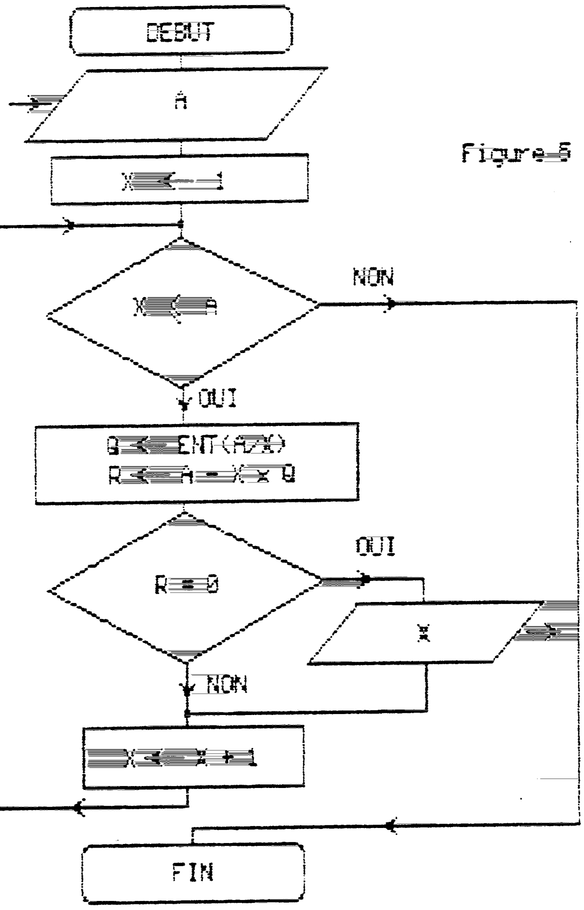


Figure 5

Programmes

LSE

```
1 * RECHERCHE DES DIVISEURS D'UN ENTIER NATUREL
10 LIRE A
20 X <-- 1
30 FAIRE 80 TANT QUE X<=A
40          Q <-- ENT(A/X)
50          R <-- A-X*Q
60          SI R=0 ALORS AFFICHER X
70          X <-- X+1
80 * fin de TANT QUE 30
90 TERMINER
```

BASIC

```
1 REM RECHERCHE DES DIVISEURS D'UN ENTIER NATUREL
10 INPUT A
20 X=1
30 WHILE X<=A
40          Q = INT(A/X)
50          R = A-X*Q
60          IF R=0 THEN PRINT X
70          X = X+1
80 WEND
90 END
```

Dans le cas où le BASIC ne possède pas l'instruction WHILE

```
1 REM RECHERCHE DES DIVISEURS D'UN ENTIER NATUREL
10 INPUT A
20 X=1
30        Q = INT(A/X)
40        R = A-X*Q
50        IF R=0 THEN PRINT X
60        X = X+1
70 IF X<=A THEN GOTO 30
80 END
```

L'instruction WHILE ... WEND est simulée par un test en fin de boucle qui permet grace à un GOTO de recommencer la séquence d'instructions.

La deuxième méthode de recherche qui consiste à vérifier que le quotient d'un nombre par un de ses diviseurs est entier donnera un algorithme et des programmes semblables.

Algorithme

```
DEBUT
LIRE A
X := 1
TANT QUE X ≤ A FAIRE
        SI A/X = ENT(A/X) ALORS
                        AFFICHER X
        * fin de SI
        X := X+1
* fin de TANT QUE
FIN
```

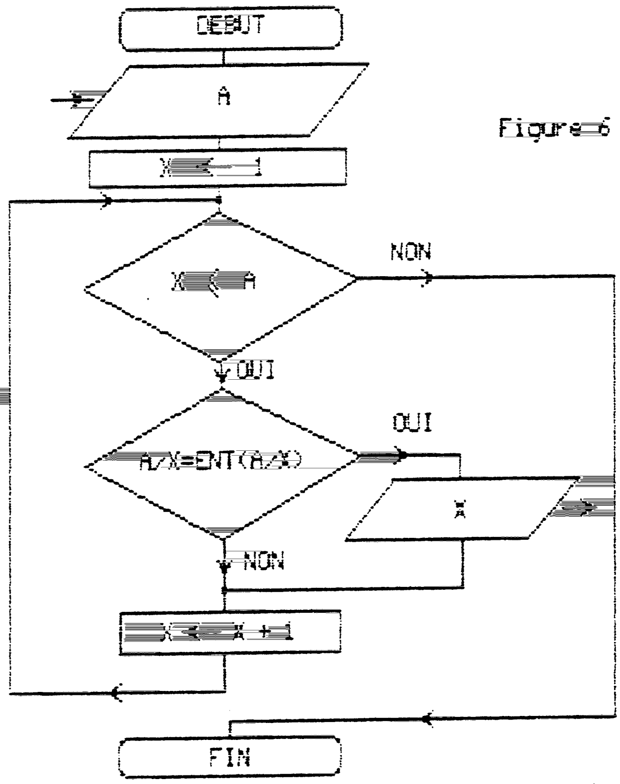


Figure 6

3 - Un entier naturel est-il premier? :

La recherche des diviseurs d'un nombre pourra nous amener à déterminer le nombre de diviseurs de ce nombre, et par là à amener la recherche des nombres premiers.

On utilisera pour cela la définition même de la primarité d'un entier naturel, à savoir :
Un entier naturel est dit premier, si et seulement si il admet deux diviseurs 1 et lui-même.

Au lieu d'afficher la suite des diviseurs de l'entier considéré, il suffira de compter ses diviseurs.

Si le nombre de diviseurs est 2, l'entier considéré sera un nombre premier.

Algorithme

```
DEBUT

LIRE A

X := 1

Y := 0

TANT QUE X ≤ A FAIRE

        SI A/X = ENT(A/X) ALORS
                          Y := Y+1

        * fin de SI

        X := X+1

* fin de TANT QUE

SI Y=2 ALORS
           AFFICHER "A est premier"

        SINON
           AFFICHER "A non premier"

* fin de SI

FIN
```

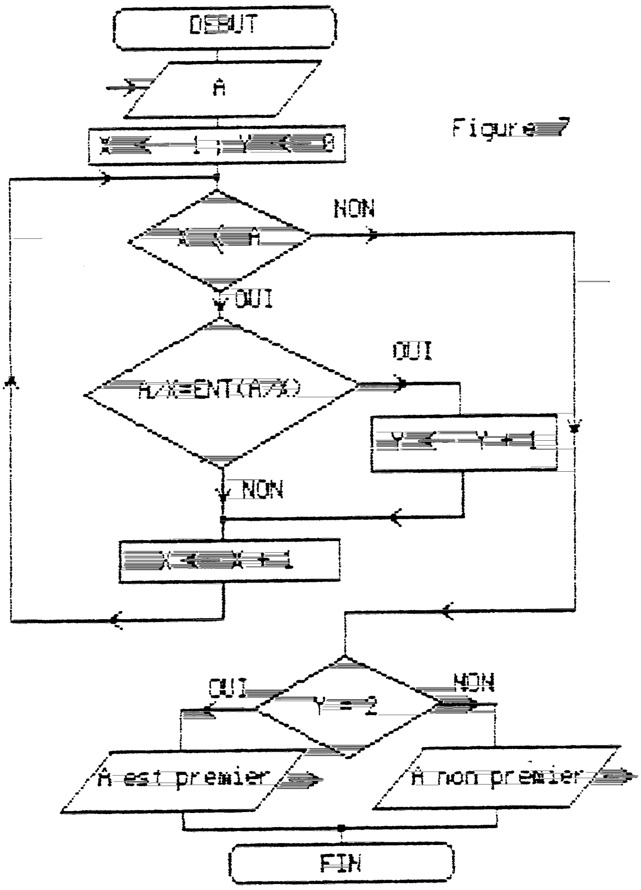


Figure 7

<u>Remarque</u> : Si l'on considère que 1 est un nombre premier, il suffira de remplacer le test Y=2 par le test Y≤2.

Pour rendre l'algorithme plus performant (plus rapide), on peut se limiter à faire varier X de 1 à racine carrée de A, ce qui s'obtient en modifiant le test sur X.

On remplace X ≤ A par X ≤ √A et Y = 2 par Y = 1.

Programmes

LSE

```
1 * UN NOMBRE EST-IL PREMIER ?
10 LIRE A
20 X <-- 1 ; Y <-- 0
30 FAIRE 60 TANT QUE X <= A
40        SI A/X=ENT(A/X) ALORS Y <-- Y+1
50        X <-- X+1
60 *fin de TANT QUE
70 SI Y=2 ALORS AFFICHER "A est premier"
          SINON AFFICHER "A n'est pas premier"
80 TERMINER
```

BASIC

```
1 REM UN NOMBRE EST-IL PREMIER ?
10 INPUT A
20 X=1:Y=0
30 WHILE X<=A
40         IF A/X=INT(A/X) THEN Y=Y+1
50         X=X+1
60 WEND
70 IF Y=2 THEN PRINT "A est premier"
           ELSE PRINT "A n'est pas premier"
80 END
```

BASIC SANS WHILE ... WEND

```
1 REM UN NOMBRE EST-IL PREMIER ?
10 INPUT A
20 X=1:Y=0
30         IF A/X=INT(A/X) THEN Y=Y+1
50         X=X+1
60 IF X<=A THEN GOTO 30
70 IF Y=2 THEN PRINT "A est premier"
           ELSE PRINT "A n'est pas premier"
80 END
```

POURQUOI NE PAS ESSAYER LE LANGAGE PASCAL ?

```
PROGRAM PREMIER (INPUT,OUTPUT);

VAR
    A,X,Y:INTEGER;

BEGIN
    READ(A);
    X:=1;Y:=0;
    WHILE X<=A DO
          BEGIN
               IF A/X=TRUNC(A/X) THEN Y:=Y+1;
               X:=X+1
          END;
    IF Y=2 THEN WRITELN(A,' est premier')
           ELSE WRITELN(A,' non premier')
END.
```

Pour compléter et terminer l'étude sur les diviseurs d'un entier naturels et sur les nombres premiers, nous pouvons envisager la recherche des nombres premiers inférieurs à un nombre donné N.

En dehors de l'approche précédente, il semble qu'il n'y ait pas d'algorithme simple et à la portée de nos élèves.

Il suffira d'inclure l'algorithme de la recherche sur la primarité d'un nombre dans une itération en faisant varier A de 2 à N. Pourquoi 2 ? Pour éliminer le test sur 1.

Algorithme

```
DEBUT

LIRE N

A := 2

TANT QUE A ≤ N FAIRE

        X := 1

        Y := 0

        TANT QUE X ≤ √A FAIRE

            SI A/X = ENT(A/X)
                    ALORS Y := Y+1

            * fin de SI

                X := X+1

        * fin de TANT QUE

        SI Y=1 ALORS
                AFFICHER A

        * fin de SI

        A := A+1

* fin de TANT QUE

FIN
```

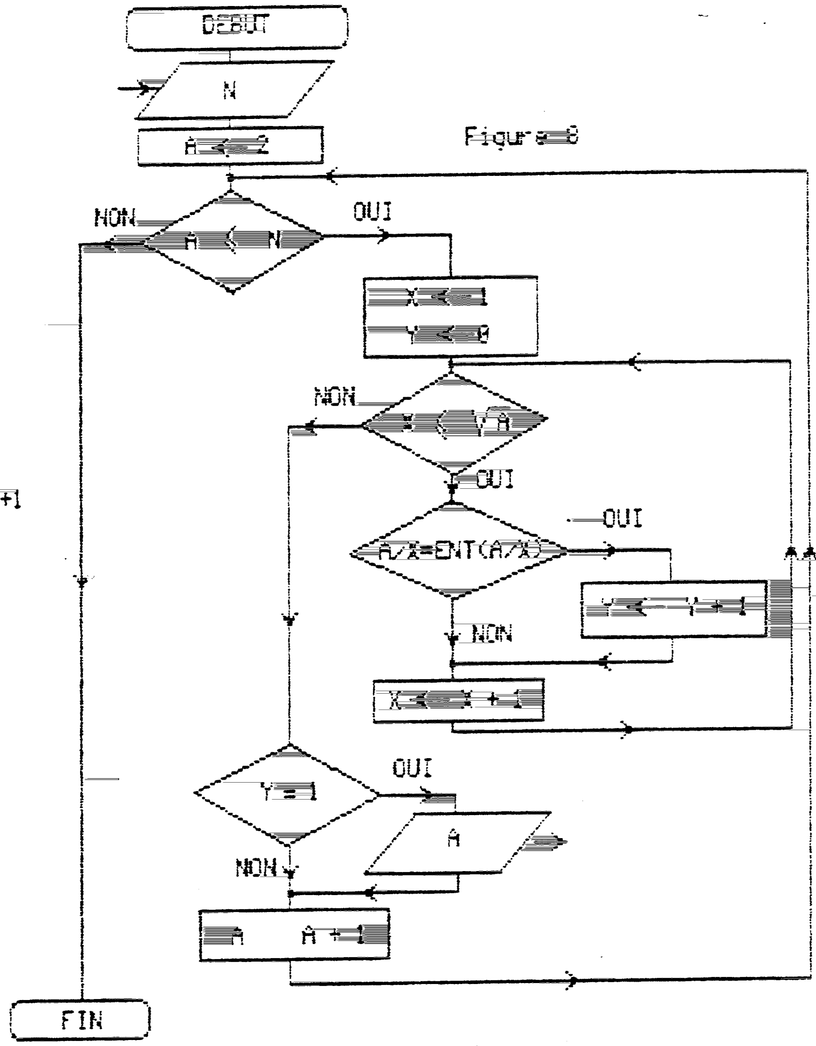


Figure 8

Programmes

LSE

```
1 * NOMBRES PREMIERS INFERIEURS A UN NOMBRE DONNE
10 LIRE N
20 A <-- 2
30 FAIRE 110 TANT QUE A <= N
40          X <-- 1 ; Y <-- 0
50          FAIRE 80 TANT QUE X <= RAC(A)
60                  SI A/X=ENT(A/X) ALORS Y <-- Y+1
70                  X <-- X+1
80          * fin de TANT QUE 50
90          SI Y=1 ALORS AFFICHER A
100         A <-- A+1
110 *fin de TANT QUE 30
120 TERMINER
```

BASIC

```
1 REM NOMBRES PREMIERS INFERIEURS A UN NOMBRE DONNE
10 INPUT N
20 A = 2
30 WHILE A <= N
40          X = 1 : Y = 0
50          WHILE X <= SQR(A)
60                   IF A/X=INT(A/X) THEN Y = Y+1
70                   X = X+1
80          WEND
90          IF Y=1 THEN PRINT A
100         A = A+1
110 WEND
120 END
```

BASIC SANS INSTRUCTION WHILE ... WEND

```
1 REM NOMBRES PREMIERS INFERIEURS A UN NOMBRE DONNE
10 INPUT N
20 A = 2
30          X = 1 : Y = 0
40                   IF A/X=INT(A/X) THEN Y = Y+1
50                   X = X+1
60          IF X<=SQR(A) THEN GOTO 40
70          IF Y=1 THEN PRINT A
80          A = A+1
90 IF A<=N THEN GOTO 30
100 END
```

PASCAL

```
PROGRAM PREMIERS(INPUT,OUTPUT);

VAR
    A,X,Y:INTEGER;

BEGIN
    READLN(N);
    A:=2;
    WHILE A<=N DO
          BEGIN
               X:=1;
               Y:=0;
               WHILE X<=SQRT(A) DO
                     BEGIN
                          IF A/X=TRUNC(A/X) THEN Y:=Y+1;
                          X:=X+1
                     END;
               IF Y=1 THEN WRITELN(A);
               A:=A+1
          END
END.
```

V. Plus grand diviseur commun

1 - Présentation du problème :

La notion de diviseurs communs à deux nombres prend toute son importance lorsque l'on aborde les simplifications dans l'ensemble des rationnels exprimés sous forme de fractions.

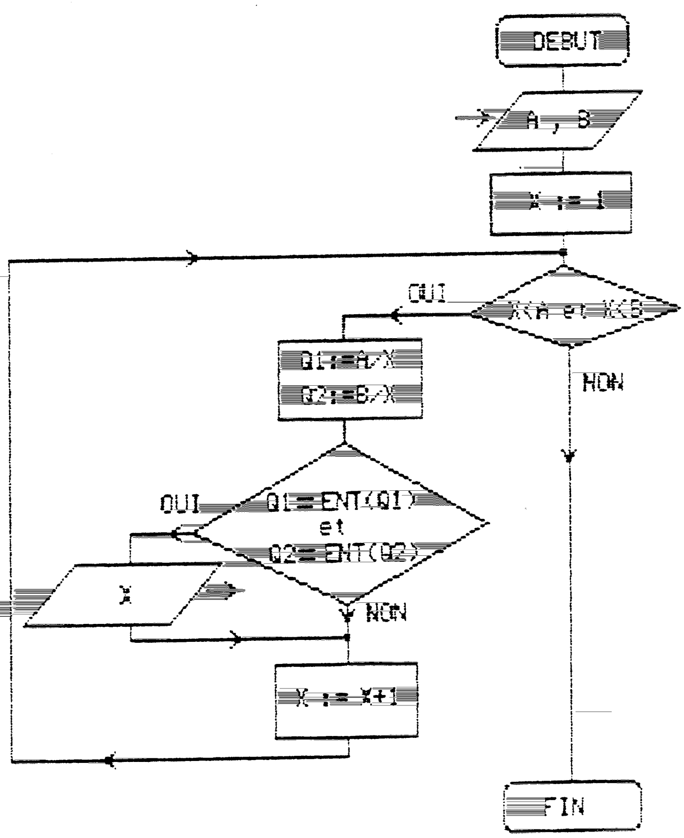
Le plus souvent ces simplifications sont "oubliées" ou bien incomplètes, plus encore lorsque les nombres sont grands. Il est nécessaire alors de faire appel à la notion de plus grand diviseur commun et il peut être intéressant à cette occasion de déboucher sur la recherche d'algorithmes. Pour beaucoup, le terme d'algorithme n'est connu qu'associé à Algorithme d'Euclide pour la recherche du PGDC, il est donc évident qu'il faudra en parler, cependant, l'on n'est pas certains que les connaissances mathématiques qu'il nécessite soient compatibles avec le savoir de nos élèves.

Des démarches plus simples peuvent leur être plus accessibles et même induites par eux (avec un minimum d'aide).

2 - Ensemble des diviseurs communs à deux nombres :

On pourra utiliser une démarche dérivant de la recherche des diviseurs d'un nombre.

```
DEBUT
LIRE A
LIRE B
X := 1
TANT QUE X<A ET X<B FAIRE
    DEBUT
    Q1 := A/X
    Q2 := B/X
    SI (Q1 = ENT(Q1)) et (Q2 = ENT(Q2))
                        ALORS AFFICHER X
    * fin de SI
    X :=X+1
* fin de TANT QUE
FIN
```



Quelques remarques sont possibles quant aux pré-requis nécessaires pour aborder l'algorithme précédent :

a - Comprendre que tout diviseur commun à deux nombres A et B est obligatoirement inférieur au plus petit des deux (mais on a déjà vu les diviseurs d'un nombre).

b - Avoir quelques notions des connecteurs logiques ET et OU (Les élèves de certaines sections les ont traitées, logique et automatismes. Il sera peut être nécessaire de les expliciter un peu).

En dehors de ces deux points, les connaissances mathématiques nécessaires sont généralement acquises par nos élèves.

Programmes

LSE

```
  1 * Ensemble des diviseurs communs à deux nombres
 10 LIRE A
 20 LIRE B
 30 X <-- 1
 40 FAIRE 80 TANT QUE X<=A et X<=B
 50         Q1 <-- A/X ; Q2 <-- B/X
 60         SI Q1=ENT(Q1) ET Q2=ENT(Q2) ALORS AFFICHER X
 70         X <-- X+1
 80 *fin de tant que
 90 TERMINER
```

BASIC

```
  1 REM Ensemble des diviseurs communs à deux nombres
 10 INPUT A
 20 INPUT B
 30 X = 1
 40 WHILE X<=A AND X<=B
 50         Q1 = A/X : Q2=B/X
 60         IF Q1=INT(Q1) AND Q2=INT(Q2) THEN PRINT X
 70         X = X+1
 80 WEND
 90 END
```

Pascal

```
Program DiviseursCom(Input,Output);
Var
    A,B,X       :Integer;
    Q1,Q2       :Real;
Begin
    Read(A);
    Read(B);
    X := 1;
    While (X<=A) And (X<=B) do
         Begin
         Q1 := A/X;
         Q2 := B/X;
         If (Q1=Trunc(Q1)) And (Q2=Trunc(Q2))
            Then
               Writeln(X);
         X := X+1
         End
End.
```

<u>Remarque</u> : Si le BASIC utilisé ne possède pas l'instruction WHILE ... WEND, et qu'il faut utiliser un test conditionnel associé avec un GOTO, il sera peut être nécessaire d'inverser la condition.

Le X<=A AND X<=B deviendra X>A OR X>B il faudra donc veiller à donner l'algorithme dans ce sens pour ne pas accentuer les difficultés des élèves.

<u>3 - Recherche du Plus Grand Diviseur Commun à deux entiers</u>

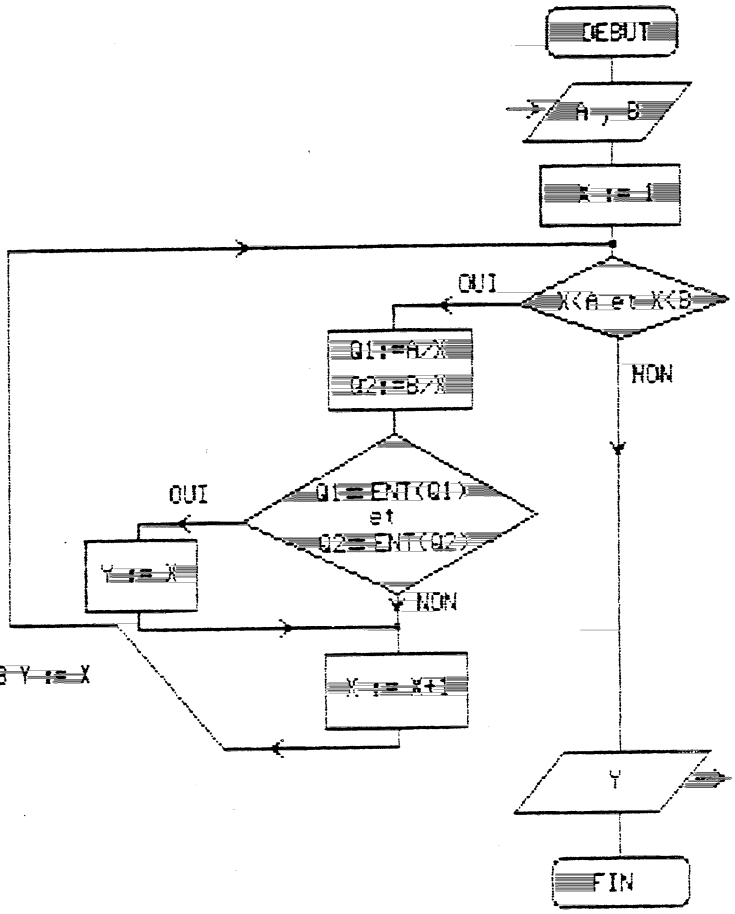
<u>3-1 Application de l'algorithme précédent :</u>

Lors de la simplification d'une fraction A/B, l'ensemble des diviseurs communs de A et de B ayant été trouvés, il va devenir évident que pour simplifier cette fraction nous allons avoir besoin du plus grand de ces diviseurs que l'on désignera par PGDC.

Nous allons donc modifier l'algorithme précédent pour qu'il nous donne uniquement le PGDC des nombres A et B.

Il suffira pour cela de mémoriser chaque diviseur rencontré dans une variable Y qui aura en fin de traitement la valeur cherchée.

```
DEBUT
LIRE A
LIRE B
X := 1
TANT QUE X<A ET X<B FAIRE
    DEBUT
    Q1 := A/X
    Q2 := B/X
    Si (Q1 = ENT(Q1)) et (Q2 = ENT(Q2))
                                  ALORS Y := X
    * fin de SI
    X :=X+1
* fin de TANT QUE
AFFICHER Y
FIN
```



Les programmes seront semblables aux précédent et s'en déduisent par des modifications minimes.

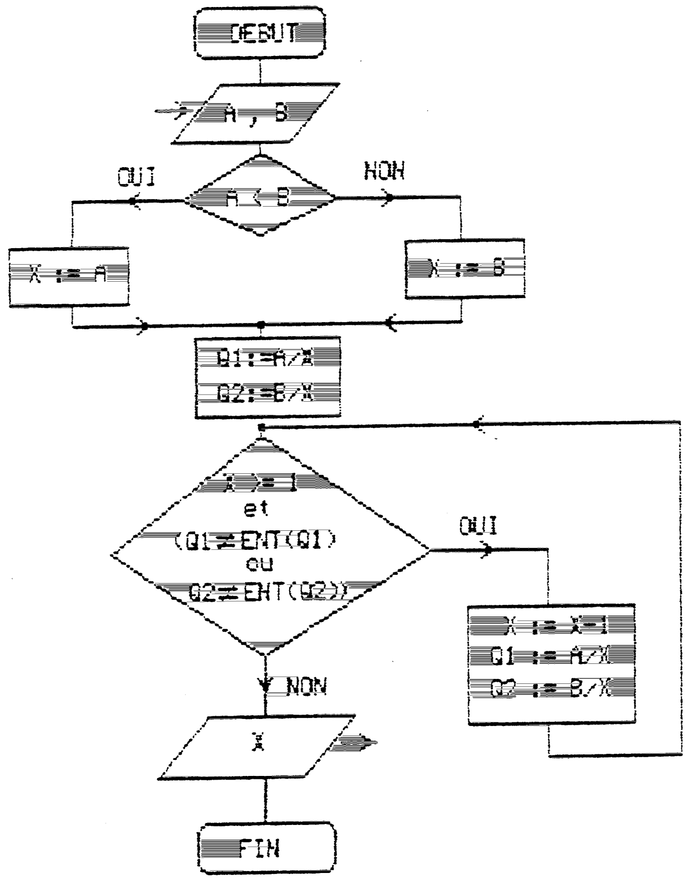
3-2 Autre algorithme intuitif :

Un diviseur d'un nombre est obligatoirement inférieur ou égal à ce nombre, si nous cherchons le PGDC de deux nombres A et B, celui-ci ne pourra être supérieur au plus petit d'entre eux.

Pour trouver le PGDC de A et B, il faudra déterminer le plus petit de A et B, soit A ce nombre.

On divise alors A et B par A si le quotient est entier A est le PGDC, sinon on prend A-1 et on recommence.

```
DEBUT

LIRE A

LIRE B

SI A<B ALORS
          X := A
      SINON
          X := B

* fin de SI

Q1 := A/X

Q2 := B/X

TANT QUE          (X>=1) ET
   ((Q1<>ENT(Q1) OU (Q2<>ENT(Q2)) FAIRE

   Q1 := A/X

   Q2 := A/X

   X := X-1

* fin de TANT QUE

AFFICHER X

FIN
```



Les deux algorithmes présentés ci-dessus sont relativement simples, bien que le dernier présente une difficulté certaine au niveau de la logique. On pourra montrer le parenthèsage nécessaire des expressions booléennes à cause de la priorité du ET sur le OU.

Pour éviter cette difficulté, on pourra faire appel à des tests successifs imbriqués.

Ces algorithmes ne peuvent être utilisés pour la détermination du PGDC que si l'on dispose de calculateurs rapides permettant de faire les nombreuses divisions demandées. On remarquera que leur exécution demande un certain temps. Comment améliorer les performances.

PARTIE 2

LE STAGE

MAT-04 A

VI. Le programme du stage

Vendredi 7 février   Animateurs: H. CAPELL
B. DUNEVON

Matin :
- Présentation de l'algorithmique
- Les structures algorithmiques
- Les notations utilisées
- Rôle de l'ordinateur
- Présentation du travail de recherche de l'équipe IREM-Lycées professionnels sur l'introduction de l'algorithmique dans les cours de B.E.P.

Après-midi:
- Fin de la présentation des travaux
- Exemple d'une utilisation de la démarche algorithmique dans un cours, étude de la progression d'un cours, "Des diviseurs d'un entier aux nombres premiers".
- Présentation du travail de recherche proposé aux stagiaires au cours du stage:
  * constitution des équipes
  * Proposition et répartition de thèmes
  * Présentation de la documentation et du matériel disponible.

Lundi 17 février   Animateurs: B. DUNEVON
G. HACQUART

Travail en équipe toute la journée:

- Structuration des équipes; répartition des tâches
- La fin de la journée pourra être consacrée à une première mise au net des travaux réalisés.

Jeudi 20 mars   Animateurs: H. CAPELL
G. HACQUART

Matin:
- Mise en forme définive des documents produits

Après-midi:
- Présentation des travaux de chaque équipe à l'ensemble des stagiaires
- Discussion
- Evaluation du stage.

VII Objectifs et contenu du stage

Historique

Ce stage a été proposé par le groupe IREM Lycées professionnels de l'Académie d'Aix-Marseille pour répondre à un besoin ressenti et souvent exprimé des collègues.

Il fait suite à des travaux de recherche menés depuis quelques années par des collègues de l'IREM en particulier par Joël CORBIN qui avait commencé ces travaux à l'IREM de Paris-Nord, et qui a été le moteur, pendant deux ans, de cette recherche dans l'Académie d'Aix-Marseille. Sa disparition tragique et brutale a laissé un grand vide parmi l'équipe de l'IREM, mais nous avons décidé de poursuivre son oeuvre en essayant de respecter l'esprit que Joël avait insufflé à l'équipe.

Objectifs

L'objectif de ce stage est de promouvoir un travail de recherche au sein d'équipes de professeurs de mathématiques, autour d'un thème qui est le développement du raisonnement logique des élèves en utilisant comme outil la démarche algorithmique.

L'équipe de l'IREM a entrepris un travail d'exploration des programmes de mathématiques des classes de B.E.P. pour essayer d'en extraire tous les thèmes pouvant se prêter à une exploitation en classe par l'algorithmique. Ces premiers travaux serviront de point de départ aux recherches proposées aux stagiaires au cours du stage.

Des documents de travail leur seront fournis pour les aider dans leur tâche, la présence de deux animateurs pendant toutes les séances pourra permettre un encadrement des différentes équipes constituées.

Le stage devrait permettre d'atteindre à deux objectifs:

1 - Impulser le travail en équipe et mettre en évidence les compétences de ses différents membres, qui sont souvent complémentaires. Il devrait permettre de sensibiliser les stagiaires à la nécessité de se regrouper pour créer les outils indispensables à l'enseignement des mathématiques dans les Lycées professionnels et profiter les uns et les autres des compétences de chacun en partageant les tâches.

Le travail en équipe amène nécessairement un travail de réflexion plus important et une critique plus approfondie des produits réalisés.

2 - Créer les bases d'une bibliothèque de séquences de cours qui introduisent l'outil algorithmique dans la classe de mathématiques.

Les travaux réalisés par les stagiaires ainsi que les documents réalisés par l'équipe de l'IREM pourront être publiés et mis à la disposition des professeurs qui dans leurs établissements voudront participer à l'expérimentation de cette recherche et à son développement.

Contenu du stage :

Le nombre de candidats inscrits régulièrement à ce stage dans le cadre de la Mission Académique à la Formation des Personnels de l'Education Nationale est de 16.
Ce nombre doit permettre de mettre en place quatre équipes de recherche.

La durée de l'action est de trois journées non consécutives espacées de 10 jours pour les deux premières et de un mois pour la dernière. L'étalement dans le temps doit permettre aux participants de mener à bien les travaux initialisés au cours des journées passées à l'IREM.

La première journée du vendredi 7 février sera employée à la présentation de l'algorithmique et à son utilisation comme outil du raisonnement logique.

La fin de l'après-midi sera consacrée à :

- la constitution des équipes de travail autour de thèmes proposés par les animateurs.
- la répartition des tâches au sein de chaque équipe en fonction des aspirations et des possibilités de chacun.
- la prévision de l'état d'avancement des travaux pour la séance suivante du lundi 17 février.

La deuxième journée du lundi 17 février sera consacrée entièrement au travail en équipes.

- mise en commun des recherches réalisées par chaque stagiaire
- critiques
- préparation de la mise au net des travaux.

Pendant cette journée des matériels informatiques seront mis à la disposition des équipes en exprimant le besoin. Ils pourront bénéficier de l'infrastructure informatique du CREFI (pratiquement tous les matériels rencontrés dans les établissements scolaires).

La dernière journée du jeudi 20 mars devra permettre aux équipes de terminer la mise en forme des documents réalisés, (photocopies, listings de programmes, ...). Ces travaux devront être terminés en fin de matinée pour que l'après-midi puisse être consacrée à la présentation des réalisations de chaque équipe à l'ensemble du groupe.

La fin de l'après-midi, qui sera aussi la fin du stage débouchera sur la discussion des travaux, l'évaluation du stage et la détermination de la suite possible à donner à ce type de formation.

VIII. Les thèmes proposés aux stagiaires

1 - Le P.G.D.C. de deux entiers naturels

2 - Le P.P.M.C. de deux entiers naturels

3 - Les fractions: simplification, opérations

4 - La résolution des équations du premier et du second degré

5 - Les systèmes d'équations du premier degré à deux inconnues

6 - Les fonctions, détermination des valeurs numériques

7 - Algorithmes dans les calculs commerciaux

8 - Algorithmes de représentations de figures géométriques simples sur l'écran d'un ordinateur.

Les travaux réalisés sur ces thèmes ne devront pas se ramener à l'énumération de tous les algorithmes possibles susceptibles d'y être rencontrés.

La partie la plus importante est nous semble-t-il la description de l'environnement pédagogique, l'utilisation des algorithmes dans le développement d'une notion mathématique :

- la structure du cours
- la description du déroulement du cours
- l'exploitation possible des algorithmes dans les leçons suivantes.

32

IX. Un exemple : diviseurs et recherche du PGCD (G. Hacquart)

FICHE PROFESSEUR

Présentation et structure du cours :

Ce cours prend place au début de première année de BEP industriel ou commercial.

Il est précédé :

* d'un rappel sur les notions de logique et d'ensembles :
  - connecteurs logiques
  - ensemble, sous-ensembles, intersection et réunion
  - produit cartésien de deux ensembles
  - relations binaires et leurs propriétés
  - fonctions, applications, bijections.

* d'un rappel sur l'ensemble N des entiers naturels :
  - définitions, opérations internes et leurs propriétés.

Il est suivi immédiatement :

* d'un chapitre sur les multiples d'un entier naturel et le P.P.M.C de deux entiers.

Il constitue une révision des notions vues de nombreuses fois et ne nécessite pour prérequis que des notions simples :

- les quatre opérations dans l'ensemble des décimaux
- quelques notions de calcul littéral.

Objectifs :

Ce document est réalisé dans l'optique d'un enseignement individualisé à rythme variable dans le cadre du contrôle continu. Pour ces raisons, il a été nécessaire tout en conservant la rigueur du fond, de simplifier et de restreindre la forme.

Le temps nécessaire à la formation en utilisant ce document variera selon les élèves, on peut penser (après expérimentation) qu'un élève moyen mais appliqué pourra le remplir en deux heures sans grosses difficultés et avec des interventions limitées du professeur.

DOCUMENT ELEVE

1 - DIVISEURS D'UN ENTIER NATUREL :

1-1 Définition :

b divise a <==> la division de a par b a pour quotient un entier naturel

Exemple:

D(120) = { 1,2,3,4,5,6,10,12,15,20,24,30,40,60,120 }

1-2 Nombres premiers :

Ce sont les entiers naturels qui admettent uniquement deux diviseurs 1 et lui-même (1 exepté).

Exemple: 2,3,5,7,11,13,17,19,23,29,...

1-3 Décomposition d'un nombre en produit de nombres premiers:

Exemple:

$$
\begin{aligned}
120 &= 2 \times 60 \\
&= 2 \times 2 \times 30 \\
&= 2 \times 2 \times 2 \times 15 \\
&= 2 \times 2 \times 2 \times 3 \times 5 = 2^3 \times 3 \times 5
\end{aligned}
$$

Disposition pratique : on divise par la suite croissante des nombres premiers autant de fois qu'on le peut.

$$
\begin{array}{r|l}
120 & 2 \\
60 & 2 \\
30 & 2 \\
15 & 3 \\
5 & 5 \\
1 &
\end{array}
$$

$120 = 2^3 \times 3 \times 5$

Rappel sur la divisibilité:

par 2 : si le nombre est pair

par 5 : si le chiffre des unités est 0 ou 5

par 3 : si la somme des chiffres est multiple de 3

1-4 Recherche des diviseurs :

$120 = 2^3 \times 3 \times 5$

$5 \to D(5) = 1,5$ L1

$3 \to D(3) = 1,3$ L2

$2^3 \to D(2^3) = 1,2,4,8$ L3

D(3,5) = {1,3,5,15}   (L1 X L2)

D(120)=D(2^3x3x5)={1,2,4,8,3,6,12,24,5,10,20,40,15,30,60,120}

soit ((L1 X L2) X L3)

<u>Exercice</u> : En utilisant la même méthode déterminer les diviseurs de 100.

```
100 = ___ x ___
       |      └──────────> Diviseurs: ___________ L1
       └─────────────────> Diviseurs: ___________ L2
```

D(100) = {_____________________________________}

1-6 Et si on utilisait l'ordinateur ? :

C'est pratique, c'est lui qui fait les calculs.

1-6-1 Exemple: Recherche des diviseurs de 120 :

Méthode: Comme l'ordinateur calcule rapidement, il suffit de diviser 120 par tous les entiers X tels que 1 <= X <= 120 et de voir si le quotient est un entier.

Décomposons le problème :

- se fixer un nombre : A := 120
- écrire "les diviseurs de 120 sont : "
- donner à X la valeur 1
- Tant que X <= 120 on exécute les opérations suivantes:
  - -- on divise 120 par X   (B := A/X)
  - -- si B = partie entière de B (B est un entier)
    alors X est un diviseur de A, on l'écrit.
  - -- on donne à X la valeur X+1.
  - -- et on recommence.

Nous venons d'écrire un <u>algorithme</u> c'est à dire :
Une suite ordonnée de traitements élémentaires permettant, à partir d'une situation initiale, d'arriver à la situation finale attendue en un nombre fini d'étapes.

1-6-2 <u>Généralisation</u> à tout nombre autre que 120

```
Début
    lire A
    X := 1
    Afficher "Les diviseurs de A sont"
    Tant que X < A faire
            B := A/X
            Si B = ENT(B)
               alors afficher X
            X := X + 1
    * fin de Tant que
Fin
```

1-6-3 Et l'ordinateur ? :

Il nous faut un langage pour communiquer avec lui , par exemple le L.S.E.

Traduit sous forme d'un programme L.S.E. notre algorithme devient:

```
 1 * Recherche des diviseurs d'un nombre
10 LIRE A
20 X <-- 1
30 AFFICHER 'Les diviseurs de A sont :'
40 FAIRE 80 TANT QUE X<A
50        B <-- A/X
60        SI B = ENT(B) ALORS AFFICHER X
70        X <-- X+1
80 * fin de TANT QUE
90 TERMINER
```

2 - PLUS GRAND DIVISEUR COMMUN :

2-1 Déterminer l'ensemble des diviseurs communs aux nombres 100 et 120 :

D(100) ∩ D(120) = {______________________}

Le plus grand d'entre eux est le P.G.D.C. de 100 et 120

P.G.D.C. (100,120) = ________

2-2 Recherche du P.G.D.C. :

100 = ____ x ____	20
120 = ____ x ____ x ____	
20 = ____ x ____	

2-3 Conclusions :

Le P.G.D.C. de deux ou plusieurs nombres s'obtient en faisant le produit

☐ de tous les nombres premiers ☐ plus grand exposant

affectés du

☐ des nombres premiers communs ☐ plus petit exposant

Reprenons l'ordinateur :

Algorithme de recherche des diviseurs communs de deux entiers naturels A et B

Compléter

Début

_______ A

_______ B

X <- _____

Tant que X < ___ et X < ___ faire

Q1 <-- A / ___

Q2 <-- B / ___

Si Q1 = ____ et Q2 = ____

alors Afficher _____

X <- ____

fin de ________

Fin

2-4 Et le P.G.D.C. ? :

On donne à une variable Y les valeurs successives des diviseurs communs de A et B. En fin de calcul, Y a la valeur du plus grand d'entre eux, le P.G.D.C. de A et B.

L'algorithme s'obtient par une légère modification du précédent.

Début

_______ A

_______ B

X <- _____

Tant que X < ___ et X < ___ faire

Q1 <-- A / ___

Q2 <-- B / ___

Si Q1 = ____ et Q2 = ____

alors Y <-- X

X <- ____

fin de ________

Fin

2-5 Programme en LSE :

```
1 * Recherche du ________ de deux nombres entiers
10 _________
20 _________
30 X _______
40 FAIRE ____ TANT QUE _____________
50       _____________
60       _____________
70       _____________________________________________
80       X _____________
90 * fin de __________
100 AFFICHER 'Le P.G.D.C. de ',A,' et ',B,' est ',Y
110 ______________
```

Et maintenant essayez-le.

PARTIE 3
LES PRODUCTIONS DES STAGIAIRES

X. Conversions système décimal/système sexagésimal
(M. Séméria)

FICHE PROFESSEUR

Présentation :

Le problème de la conversion des nombres exprimés sous forme décimale ou sous forme sexagésimale se rencontre du fait de l'utilisation des calculatrices pour effectuer les opérations sur les angles ou les temps.

Les élèves ne sont pas habitués à effectuer ces conversions, et il nous parait que l'utilisation de l'algorithmique pour résoudre ce problème peut aider l'élève.

Les algorithmes développés sont très simples et n'utilisent que la structure séquentielle.

Prérequis nécessaires :

Les notions qu'il serait souhaitable que les élèves possèdent avant d'aborder ce thème sont :

- proportionalité
- système sexagésimal, unités

Objectifs :

Permettre à l'élève de donner le résultat d'une mesure d'angle ou de temps indifféremment sous forme décimale ou sexagésimale, en fonction du résultat obtenu.

DOCUMENT ELEVE

CONVERSION
SYSTEME DECIMAL <---> SYSTEME SEXAGESIMAL

1 - La mesure d'un angle est écrite dans le système sexagésimal; on veut l'écrire dans le système décimal.

Exemple : 45° 40' 48"

A partir d'un exemple, on va élaborer un algorithme permettant le calcul dans tous les cas.

Remarques préalables : Compléter

1° = ... '
1' = ... "
1° = ... "

Compléter : 1,5° = 1° + 0,5°

convertir en secondes

.... " + ... "

1,5° = ... "

Quel est le coefficient K qui permet directement de calculer le nombre de secondes ? K

1,5° x .... = ... "

Vérifiez réciproquement : K

...... : .... = 1,5°

Cette méthode peut être généralisée. A partir du nombre total de secondes, (SECT) on obtient la mesure de l'angle A dans le systèmedécimal en divisant le nombre total de secondes par le nombre K.

Calculons SECT

45° 40' 48"

D° MIN' SEC"

45° x .... = ... SECD

SECD = D° x ....

Pour les minutes
40' x .... = ... SECM

SECM = MIN' x ....

SECT SECD SECM SEC

...... = ... + ... + ...

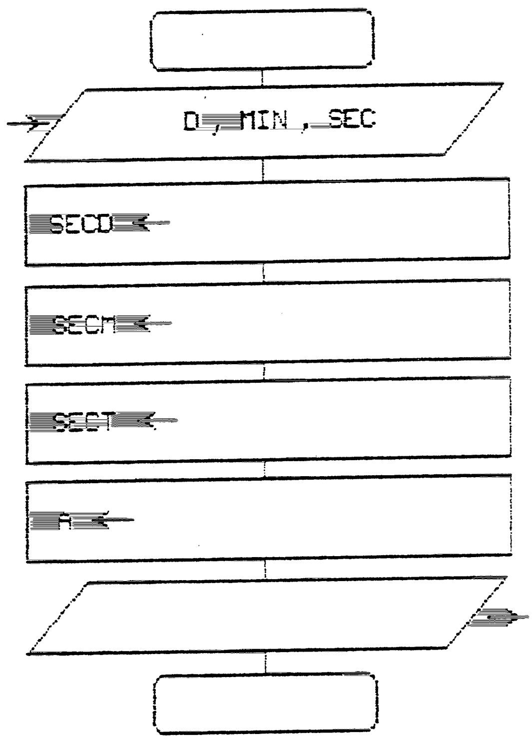
SECT = SECD + SECM + SEC

A = ... : ....

A = SECT : K

Voir l'organigramme figure 1

ORGANIGRAMME POUR LA CONVERSION
SYSTEME SEXAGESIMAL ---> SYSTEME DECIMAL



2 - La mesure d'un angle est écrite dans le système décimal, on veut l'écrire dans le système sexagésimal.

Exemple : 45,680 ° peut s'écrire 45° 40' 48"

Nous allons conduire le raisonnement à partir d'un exemple et élaborer un algorithme applicable dans tous les cas.

A = 45,68°

Ecrire le nombre entier de degrés :

D = ...

Ecrire la partie décimale de A :

DEC = 0,......

C'est cette partie qu'il faut convertir en minutes et secondes

Remarque:

1° = 60'
0,5° = ...'
0,5 x K° = ...'

Le coefficient K est K = ...

Faisons de même avec la partie décimale de A

DEC		K		MIN
......	x	....	=	.....

Le nombre entier de minutes est

MIN = ...

Ecrire la partie décimale de MINDEC

DECMIN = ...

Le nombre de secondes est obtenu de la même manière que précédemment.

DECMIN		K		SEC
......	x	....	=	.....

Résultat final:

A = D° MIN' SEC"

..... = ...° ....' ...."

Voir l'organigramme figure 2

.... A

D : = ...

Ecrire l'opération donnant DEC en fonction de D et A

DEC : = ...

MINDEC est le nombre de minutes contenues dans la partie décimale

K

MINDEC : = DEC x ....

MIN : = ...

Ecrire l'opération donnant la partie décimale de MINDEC appelée DECMIN

DECMIN : = ...

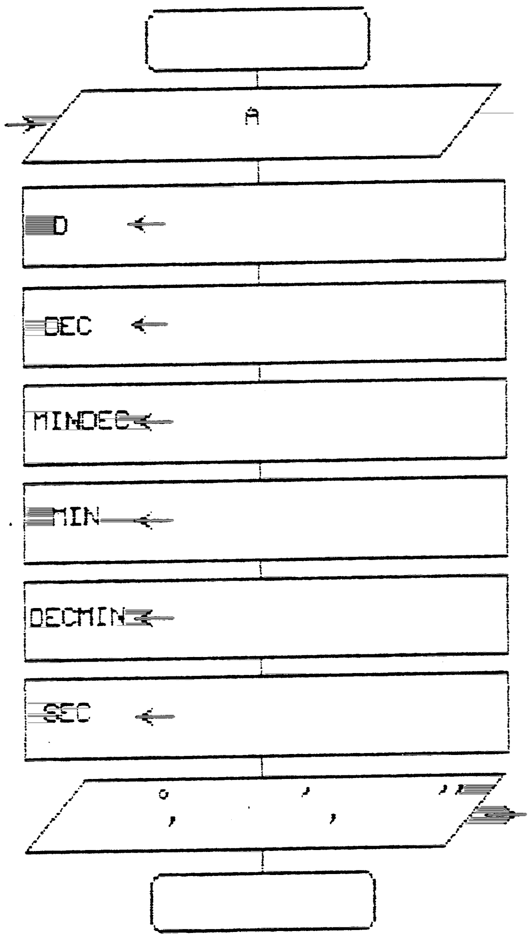
K

SEC : = DECMIN x ...

Résultat final:

D° MIN' SEC"

ORGANIGRAMME POUR LA CONVERSION
SYSTEME DECIMAL ------> SYSTEME SEXAGESIMAL



XI. Triangle rectangle : existence et calcul des angles
(A. Bonafoux, P. Foulquier, A. Moulet, A. Peyre)

FICHE PROFESSEUR

Présentation :

Le problème posé ici consiste à reconnaitre l'existence d'un triangle connaissant la mesure de la longueur de ses trois côtés, de reconnaitre ensuite si le triangle proposé est rectangle et de donner la longueur de l'hypoténuse ainsi que la mesure des angles aigus.

Elèves concernés :

BEP industriels
Niveau d'intervention : fin de première année ou début de deuxième année.

Prérequis nécessaires :

Pour aborder cette séance les élèves doivent connaitre :

- Propriété de Pythagore
- Rapports trigonométriques dans le triangle rectangle
- Première approche de l'algorithmique

Objectifs :

1 - Vérification de l'existence d'un triangle à partir de trois nombres donnés a, b, c susceptibles d'être les mesures des trois côtés.

2 - Vérification de la nature d'un triangle rectangle ou quelconque.

3 - Calcul des angles aigus d'un triangle rectangle à partir de la mesure de ses côtés.

DOCUMENT ELEVE

EXISTENCE D'UN TRIANGLE

1 - Rappel de la condition d'existence d'un triangle :

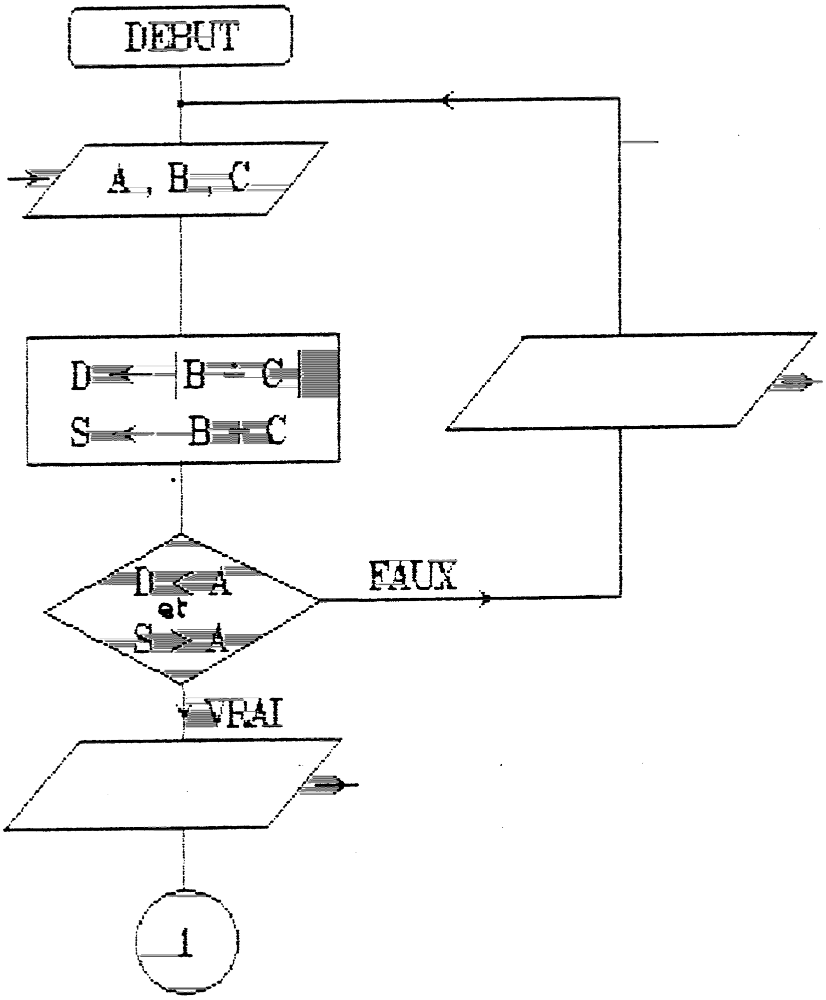
Trois nombres a, b, c représentent les mesures des trois côtés d'un triangle si :

$$|b - c| < a < b + c$$

2 - Analyse de la double condition :

$|b - c| < a < b + c$ est équivalent à $|b - c| < a$ et $a < b + c$

3 - Organigramme :



Programme en L.S.E.

```
1 * Existence d'un triangle

10 LIRE  _

20 LIRE  _

30 LIRE  _

40 D <- ABS(______)

50 S <- _______

60 SI D <__ ET S >__ ALORS AFFICHER '___________________'

                     SINON DEBUT AFFICHER '________________________'

                                 ALLER EN _____

                           FIN

70 TERMINER
```

4 - Nature du triangle et calcul des angles :

1 - Rappel de la propriété de Pythagore

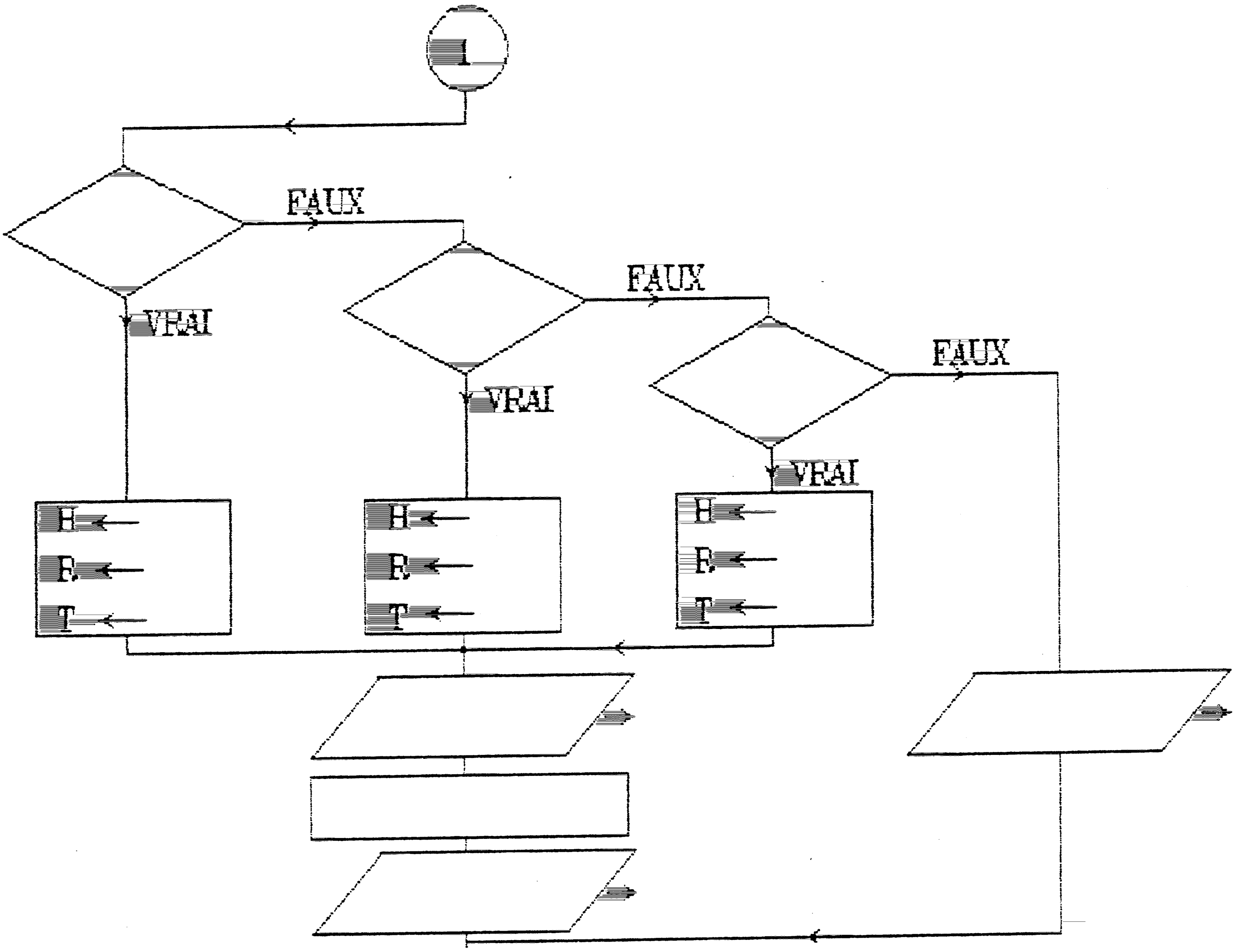
---

2 - Vérifier la propriété de Pythagore

3 - Recherche de l'Hypoténuse dans le cas où le triangle est rectangle.

4 - Calcul de la mesure des angles

5 - Organigramme



5 -- Traduction de l'algorithme en L.S.E. :

1 * Existence et recherche des angles d'un triangle rectangle

```
10 ____ __

20 ____ __

30 ____ __

40 D __ _______

50 S __ _______

60 SI ______ ET ______ ALORS AFFICHER '___________________'

                      SINON DEBUT AFFICHER '_______________________'

                                  ALLER EN _____

                            FIN

70 SI _________ ALORS DEBUT __________

                            __________

                            __________

                    FIN

              SINON SI _________ ALORS DEBUT _________

                                             _________

                                             _________

                                   FIN

                            SINON SI ________ ALORS DEBUT _________

                                                          _________

                                                          _________

                                                FIN

                                             SINON DEBUT AFFICHER '_____________'

                                                         ALLER EN ____

                                                   FIN

80 AFFICHER '_____________________________'

90 D __ ATG(________)
```

```
100 D1 __ ______________

110 D2 __ ______________

120 AFFICHER ______________

130 ____________
```

Remarque: La ligne 70 du programme est très complexe et surtout trop longue et trop lourde. Sa transcription en mémoire de l'ordinateur risque d'amener des erreurs.

Il est souhaitable d'utiliser une procédure sous-programmes pour effectuer les différentes affectations rencontrées dans cette ligne. Il suffira de lui transmettre les paramètres convenables.

La ligne 70 devient:

```
70 SI ________ ALORS &AFF(A,B,C)
               SINON SI ________ ALORS &AFF(B,A,C)
                                 SINON SI _______ ALORS &AFF(C,A,B)
                                                  SINON DEBUT AFFICHER '___________'
                                                              ALLER EN ____
                                                        FIN
```

Il suffira ensuite d'écrire la procédure après la fin du programme principal:

```
150 PROCEDURE &AFF(X,Y,Z)

160 __ __ X

170 __ __ Y

180 __ __ Z

200 RETOUR
```

XII. Fonction linéaire et calculs commerciaux
(G. Goll, A. Liégeois, J. Gaggero, E. Davin)

FICHE PROFESSEUR

Présentation et structure du cours :

Ce cours doit permettre la redécouverte de la fonction linéaire dans les calculs commerciaux.

Il s'adresse aux élèves de Seconde BEP commerciaux.

Prérequis nécessaires :

- proportionnalité
- fonction linéaire
- formules sur les intérêts simples

Objectifs :

Ce cours vise à faire découvrir par les élèves, dans des applications professionnelles courantes l'utilisation des "outils mathématiques" comme la fonction linéaire. Ceci est appliqué ici dans les calculs commerciaux , mais il peut être envisagé le même type de cours pour des applications dans les spécialités industrielles, ou les sciences physiques.

DOCUMENT ELEVE

1 - Calcul de I suivant le nombre de jours :

1-1 Méthode logique de raisonnement :

$$I = \frac{C \times t \times n}{36000}$$

* Trouvons les valeurs de I suivant le nombre de jours n

Que faut-il connaître pour cela ? ____ et ____

Donnons leurs valeurs

____ = 7200

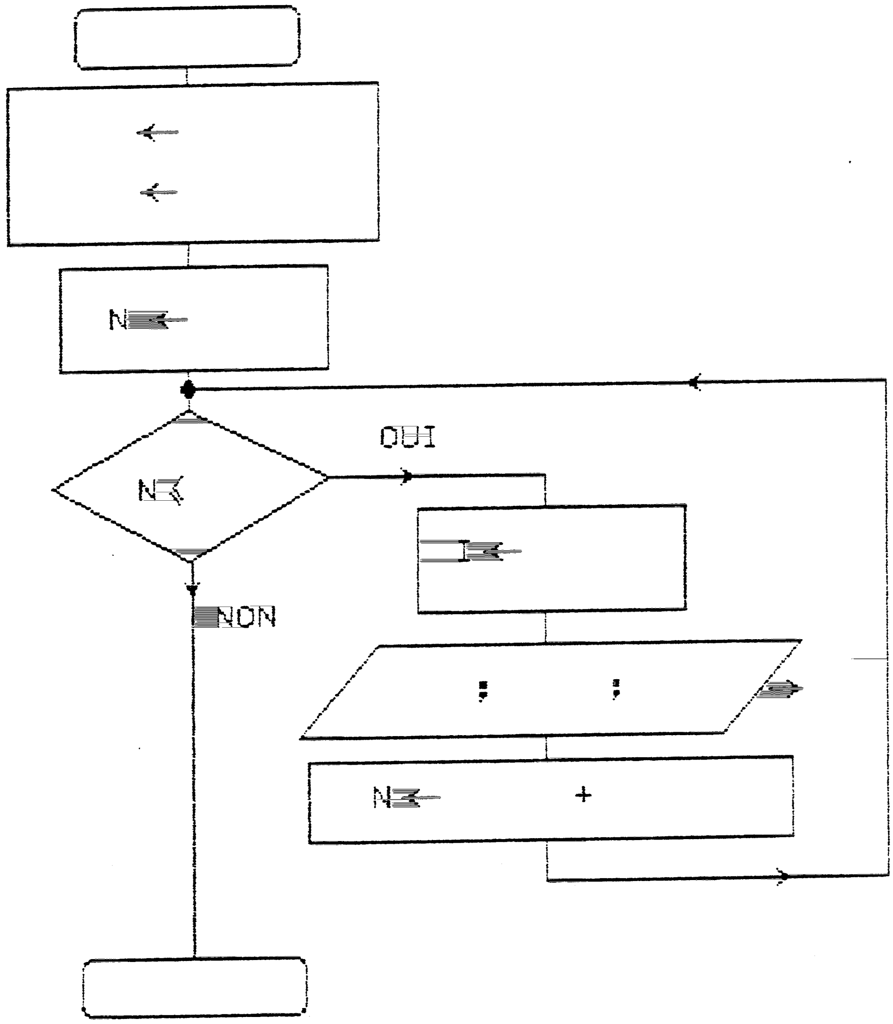
____ = 12

* On va calculer I pour n variant de 20 en 20.

- Début
- On commence par n=20
- Quelle est la valeur maximale de n ? _____
- Tant que n ≤ ______ on exécute l'opération I = _____
- On affiche les valeurs I , n , I/n
- On donne à n la valeur suivante n : = n + ____
- On recommence au Tant que.
- Fin

Nous venons d'écrire l'algorithme qui permet de réaliser les calculs prévus.

1-2 Ecrivons l'organigramme correspondant :



1-3 Utilisons l'ordinateur pour faire les calculs rapidement

1-3-1 Ecrivons le programme en LSE :

Testons le programme

1-3-2 Améliorons le programme pour que l'utilisateur puisse choisir C et t

Testons le programme, on prendra des valeurs telles que:

$$100 \leq C \leq 100000$$
$$0 < t \leq 25$$

1-3-3 Facultatif : incorporons au programme des tests de validité sur C et t

Testons le programme

Conclusion : Dans chacun des cas précédents, que peut-on dire du rapport I/n ?

il est ____________

LES GRANDEURS I ET n SONT ________________

1-4 Représentation graphique :

1-4-1 Utilisation du programme de visualisation de la représentation graphique des variations de l'intérêt I en fonction du nombre de jours n.

1-4-2 Etude de la représentation graphique :

Nous remarquons que sur la représentation graphique:

____

Conclusion :

C'est la représentation graphique d'une fonction _________

La relation qui lie n et I est : I = _______ x n

elle est de la forme : y = __________

COMPLEMENTS A CE TRAVAIL

2 - Calcul de I en fonction de C et t :

3 - Valeur acquise et fonction affine :

QUELQUES EXTENSIONS POSSIBLES

- Dans le secteur industriel : U = R I
  U = E - r I

- Dans le secteur industriel : U = R I^2
  (but a x^2)

- Distance - temps - vitesse : (but : 1/x) etc.

PROGRAMMES

Calcul de l'intérêt, le capital C et le taux t étant fixes, le nombre de jours n variant de 20 à 360 avec un pas de 20.

```
 1 * Programme intérêt 1
10 C <-- 7200
20 T <-- 12
30 N <-- 20
40 AFFICHER "N          I          I/N"
50 FAIRE 90 TANT QUE N<=360
60          I <-- N*C*T/36000
70          AFFICHER N,I,I/N
80          N <-- N+20
90 * Fin de Tant que
100 TERMINER
```

Même programme, mais l'utilisateur a le choix du montant du capital C, et du taux d'intérêt t.

Il suffit de modifier les lignes 10 et 20 du programme précédent.

```
10 LIRE "Capital          : ",UJC
20 LIRE "Taux d'intérêt : ",UJT
```

XIII. Equation du premier degré à une inconnue
(H. Bonne, J. Peyre)

FICHE PROFESSEUR

Présentation et structure du cours :

Ce cours est destiné aux classes de première année BEP industriel ou commercial et se situe dans la progression en milieu du 2 ème trimestre.

Prérequis nécessaires :

- Addition des réels
- Règle des signes dans les réels
- Distributivité de la multiplication sur l'addition
- Addition, multiplication et division des rapports
- Propriété des égalités:
  - additionner un même terme aux deux membres d'une égalité.
  - Multiplier et diviser par un même terme non nul les deux membres d'une égalité.
- Notions sur les algorithmes

Objectifs :

Ce cours concerne les équations rationnelles qui se ramènent à une équation du premier degré à une inconnue réelle.

- équations sous forme de produits de facteurs
- fonctions rationnelles

Le document élève permet à partir du cas général de découvrir les cas particuliers qui peuvent se présenter.

DOCUMENT ELEVE

1 - Notion d'équation du premier degré à une inconnue :

1-1 Définition : C'est un ensemble de deux membres séparés par le signe =, dans lequel il y a une inconnue X dont l'exposant est 1.

Sa forme réduite est :

$$aX + b = 0 \quad \text{avec} \quad a \in R^{*} ,\ b \in R ,\ X \in R$$

2-2 Exemples :

$2X + 3 = 0$; $X - 5 = 0$; $\frac{7}{2} X = - \frac{2}{5}$; $-5X = 2$

Donnez trois autres exemples :

-

-

-

2 - Résolution de l'équation du premier degré à une inconnue

2-1 But de la résolution :

Soit $X \in E$ (un ensemble), résoudre dans E l'équation $aX + b = 0$, c'est trouver l'ensemble S des éléments de E qui vérifient l'égalité.

2-2 Algorithme de résolution :

A partir d'une expression donnée sous une forme quelconque, il faut appliquer des règles de calcul pour :

- trouver une expression équivalente ayant la forme:
$$aX + b = 0$$

- déterminer l'ensemble S1 des valeur de X vérifiant l'égalité

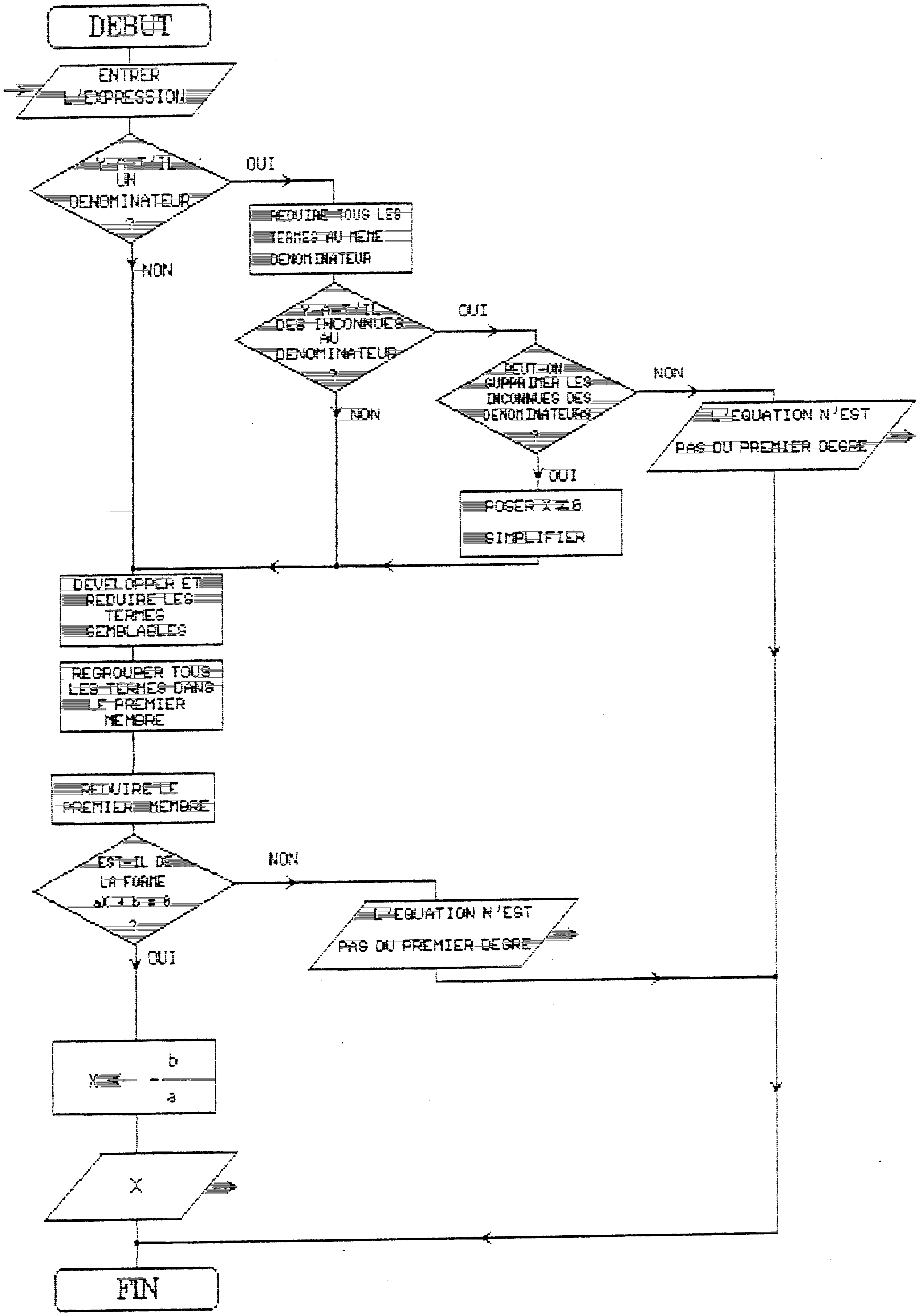
- donner l'ensemble solution S tel que:

$$S = S1 \cap E$$

Pour celà, nous allons étudier les algorithmes de résolution nécessaires.

Un premier algorithme nous permettra de trouver l'expression équivalente sous forme réduite, elle doit être de la forme aX + b = 0. Si ce n'est pas le cas, l'équation n'est pas du premier degré, et nous ne pouvons pas la résoudre pour l'instant.

ALGORITHME GENERAL DE RESOLUTION



2-3 Appliquer cette méthode aux exercices suivants

$$17X - 1 = 3 + 15X$$

$$4(X + 1) = 7(X - 3)$$

$$\frac{2x - 1}{3} = \frac{3x + 2}{5}$$

$$\frac{3X + 2}{2} - \frac{2X}{5} = \frac{5X - 2}{10}$$

$$\frac{4X^2 - 1}{4X - 2} = \frac{3X}{2} - \frac{6X^2 - X}{5X}$$

2-4 Généralisation :

La traduction de l'organigramme précédent dans un langage informatique n'est pas aisée, surtout pour la première partie de l'organigramme qui nous permet de mettre l'expression algébrique sous la forme $aX + b = 0$.

Par contre, nous allons pouvoir écrire l'algorithme et le programme qui nous permettront de passer de cette expression réduite à la solution de l'équation.

Pour $a \in \mathbb{R}^{*}$, $aX + b = 0 \iff X = -\frac{b}{a}$

2-5 Compléter l'organigramme et l'algorithme :

DEBUT

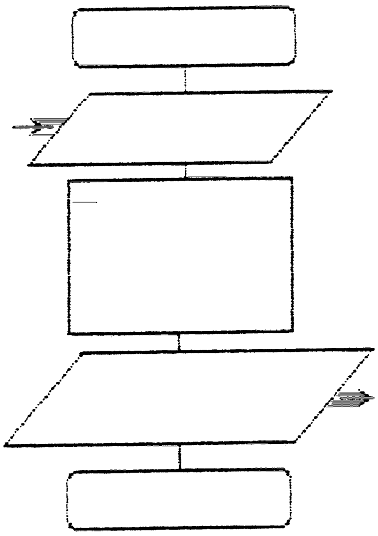
LIRE ____

LIRE ____

____ <- ______

AFFICHER ____

FIN



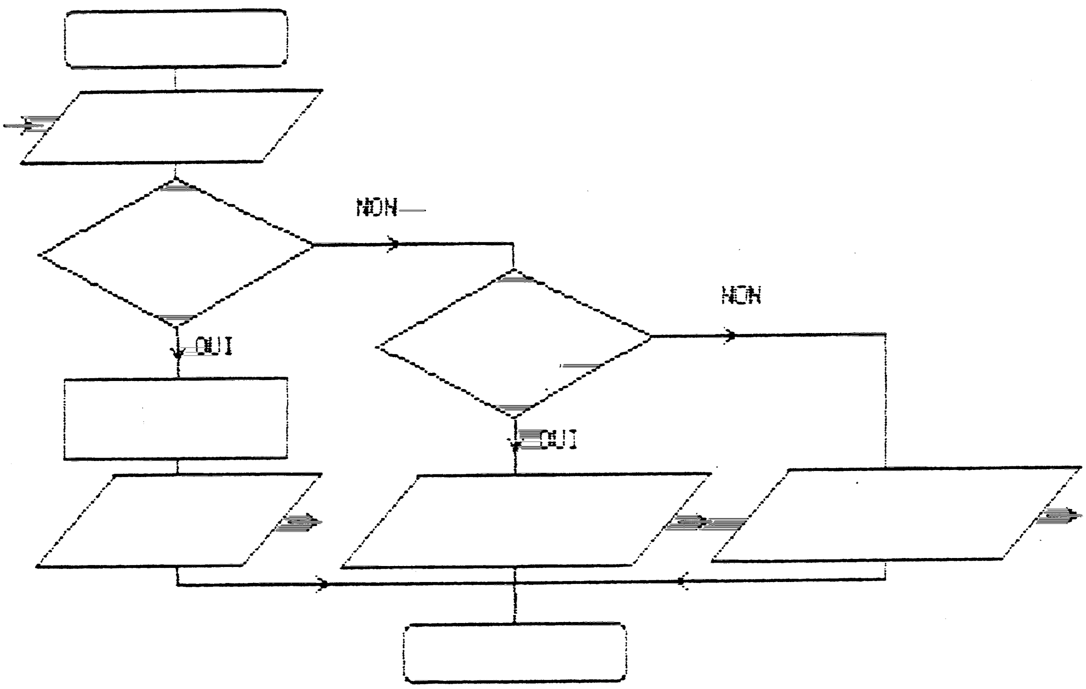
3 - APPLICATIONS :

Vérifiez l'algorithme avec les exercices suivants

Résoudre dans	l'équation	S1	S
$\mathbb{R}$	$5X + 2 = 0$		
$\mathbb{N}$	$\frac{3}{2}X + \frac{4}{2} = 0$		
$\mathbb{R}$	$2X + \frac{2}{3} = \frac{4}{6}$		
$\mathbb{R}$	$2(3X + 7) = 3(2X - 4)$		
$\mathbb{R}^{*}$	$3X - 4 = \frac{6}{2}X$		
$\mathbb{R}$	$\frac{8}{2}X + 3 = 4X + \frac{12}{4}$		

Compléter l'algorithme et l'organigramme suivants en tenant compte des différents cas rencontrés.

```
DEBUT
    LIRE __
    LIRE __
    SI ___
          ALORS
              DEBUT
              ___  <- ____
              AFFICHER ___
              FIN
          SINON
              SI ___
                  ALORS
                      AFFICHER "___________"
                  SINON
                      AFFICHER "___________"
              * fin de SI
    * fin de SI
FIN
```



LISTE DES PARTICIPANTS AU STAGE

ALBET Maurice	LP Gustave	EIFFEL AUBAGNE
BONNAFOUX Alain	LP VAUVENARGUES	AIX en PROVENCE
BONNE Hélène	LP F. MISTRAL	MARSEILLE
CAPELL Henri	LP La Viste	MARSEILLE
DAVIN Emmanuelle	LP GAMBETTA	AIX en PROVENCE
DUNEVON Bernard	LP Bd LEAU	MARSEILLE
FOULQUIER Pierrette	LT DIDEROT	MARSEILLE
GAGGERO Josiane	LP Bd LEAU	MARSEILLE
GOLL Gérard	LP J. FERRY	GARDANNE
HACKART Guy	LP Jean MOULIN	PORT de BOUC
LIEGEOIS Aimée	LP Ph. de GIRARD	AVIGNON
MOULET Alain	LP L. BLERIOT	MARIGNANE
PEYRE Annie	LT DIDEROT	MARSEILLE
PEYRE Jacqueline	LP J.H. FABRE	CARPENTRAS
ROUX Jacques	LP L'Argensol	ORANGE
RUBIEN M.Hélène	LP B. PASCAL	MARSEILLE
SEMERIA Michel	LP J. LURCAT	MARTIGUES

Dépôt légal deuxième trimestre 1986