

QUELQUES EXEMPLES
D'ALGORITHMES
POUR LA
CLASSE DE SECONDE

Frédéric MARTIN
Lycée la Herdrie - Basse-Goulaine (44)
martinfrederic44@hotmail.fr

16/11/ 2009

I - Algorithme

C'est l'ensemble des actions nécessaires à l'accomplissement d'une tâche.

1. Caractéristiques d'un algorithme

Il doit se terminer après un nombre fini d'opérations.

Chaque instruction doit être défini sans ambiguïté.

Il doit aboutir à au moins un résultat.

2. Variables

Les instructions s'appliquent à des variables

Une variable est caractérisée par :

son identificateur (son nom) ;

son type (par exemple numérique) ;

son contenu (valeur prise par la variable à un niveau donné de l'algorithme).

L'identificateur est le nom de la case réservée en mémoire, le type est la catégorie d'information qu'elle peut contenir, son contenu est l'information que l'on a mise dans la case.

Par exemple la case appelée *PI* peut être de type réel et contenir le décimal 3,14.

3. Expression d'un algorithme

Un algorithme peut s'exprimer

en langage clair.

par un organigramme.

4. Instructions d'entrée

Affectation :

A reçoit 15, noté $A \leftarrow 15$ ou $A := 15$,

A reçoit le contenu de *B*, noté $A \leftarrow B$ ou $A := B$.

Lecture d'une donnée :

Lire une donnée entrée au clavier notée, LIRE *X* (met dans la case appelée *X* les données entrées au clavier).

5. Instructions de traitement

Opérateurs arithmétiques :

+, -, *, /.

Opérateurs de comparaison :

<, >, ≤, ≥, =, ≠.

Opérateurs logiques :

et, ou, non (\wedge , \vee , \neg).

6. Instructions de sortie

Ecrire un texte ou le contenu d'une variable, ECRIRE "Le résultats est :" (Ecrit sur l'écran *Le résultat est :*), Ecrire *X* (Ecrit sur l'écran le contenu de la variable *X*).

II - Structure d'un algorithme

1. La Séquence

Instructions dans l'ordre dans lequel elles apparaissent (énumération).

Exemple :

Objet : calculer l'image d'un nombre par la fonction f définie sur \mathbb{R} par $f(x) = 3x^2 + 2x + 5$.

```
DEBUT
  LIRE X
   $Y \leftarrow 3 * X * X - 2 * X + 5$ 
  ECRIRE Y
FIN
```

2. La structure conditionnelle (ou alternative)

SI (condition) **ALORS** (instructions 1) **SINON** (instructions 2) **FIN SI**
SINON est facultatif.

Si la condition énoncée est réalisée faire instructions 1 sinon faire instructions 2.

IF ... THEN ... ELSE ... IFEND

Exemples :

Objet : Connaissant a , b et c , déterminer si le trinôme $ax^2 + bx + c$ a des racines.

```
DEBUT
  LIRE A
  LIRE B
  LIRE C
   $D \leftarrow B * B - 4 * A * C$ 
  SI  $D < 0$  ALORS ECRIRE "Pas de racine"
  SINON ECRIRE "Au moins une racine"
  FIN SI
FIN
```

Objet : Connaissant a , b et c , déterminer l'existence et le nombre des racines du trinôme

```
 $ax^2 + bx + c$ .
DEBUT
  LIRE A
  LIRE B
  LIRE C
   $D \leftarrow B * B - 4 * A * C$ 
  SI  $D < 0$  ALORS ECRIRE "Pas de racine"
  SINON SI  $D = 0$  ALORS ECRIRE "Une racine double"
  SINON ECRIRE "Deux racines distinctes"
  FIN SI
  FIN SI
FIN
```

3. Les structures itératives (ou boucles)

TANT QUE (condition) **FAIRE** (instructions) **FIN DE TANT QUE**

Tant que la condition énoncée est réalisée faire les instructions.

WHILE ... WHILEEND

REPETER (instructions) **FIN DE REPETER** **JUSQU'A** (condition)

Répéter les instructions jusqu'à ce que la condition énoncée soit réalisée.

DO ... WHILE ...

POUR (variable) **DE** (valeur) **A** (valeur) **ITERER** (instructions) **FIN D'ITERATION**

Ici le nombre de boucles est connu à priori.

FOR ... TO ... DO ... NEXT

Exemple :

Objet : Soit la suite (u_n) définie par $u_n = 2u_{n-1} - 4$ et $u_0 = 5$.

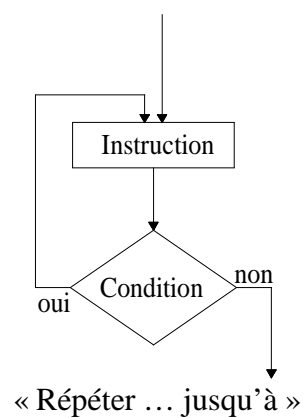
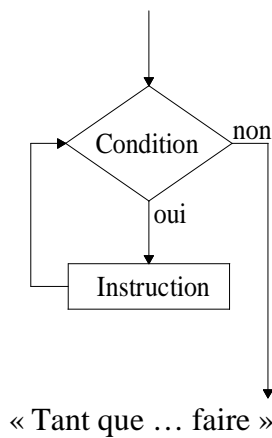
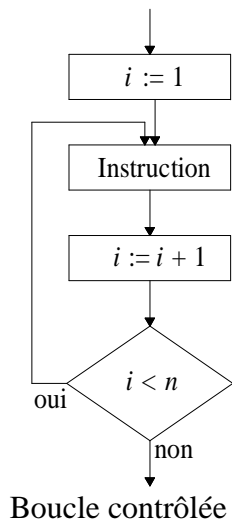
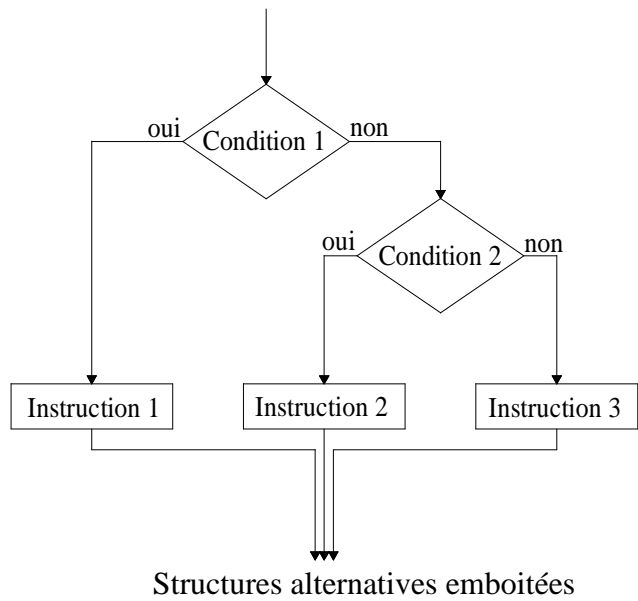
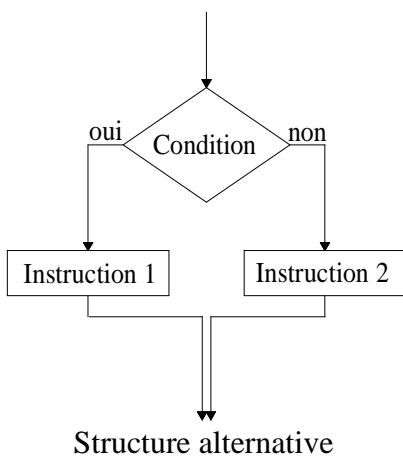
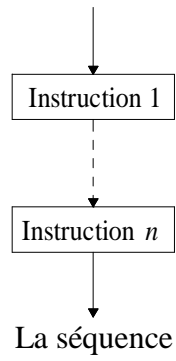
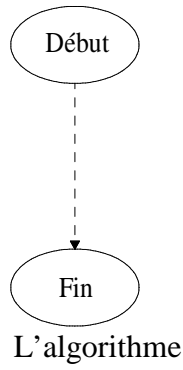
1. Calcul de u_k .

2. A partir de quel valeur de k , u_k est strictement supérieur à un réel m donné.

4. Procédure

C'est la décomposition d'un algorithme. Il faut définir les procédures au préalable.

III - Organigrammes



IV – Faire un algorithme

1. Tirage d'un nombre entier compris entre deux valeurs

On notera $\text{RANDOM}()$ la fonction qui génère un nombre pseudo-aléatoire compris en 0 et 1 et $E(x)$ la partie entière de la variable x . Le but est d'obtenir un nombre entier pseudo-aléatoire entre deux bornes choisies.

Tous les algorithmes qui suivent peuvent très facilement être transcrit avec AlgoBox

```
DEBUT
ECRIRE "Borne inférieure ?"
LIRE A
ECRIRE "Borne supérieure ?"
LIRE B
C:=E((B-A+1)*RANDOM()+A)
ECRIRE C
FIN
```

Si les bornes proposées ne sont pas entières ou si la borne supérieure est inférieure à la borne inférieure les résultats obtenus ne sont pas ceux attendus. On peut obliger l'utilisateur à respecter ces impératifs.

```
DEBUT
A:=0.1
TANT QUE E(A) ≠ A OU E(B) ≠ B OU
A>B
FAIRE
    ECRIRE "Borne inférieure A (nombre
entier) ?"
    LIRE A
    ECRIRE "Borne supérieure B (nombre
entier supérieur à A) ?"
    LIRE B
FIN TANT QUE
C:=E((B-A+1)*RANDOM()+A)
ECRIRE C
FIN
```

On peut aussi corriger automatiquement les données. Mettre les bornes dans l'ordre. Refuser le cas où la partie entière de la borne supérieure est strictement inférieure à la borne inférieure. Arrondir à l'entier directement supérieur la borne inférieure et à l'entier directement inférieur la borne supérieure.

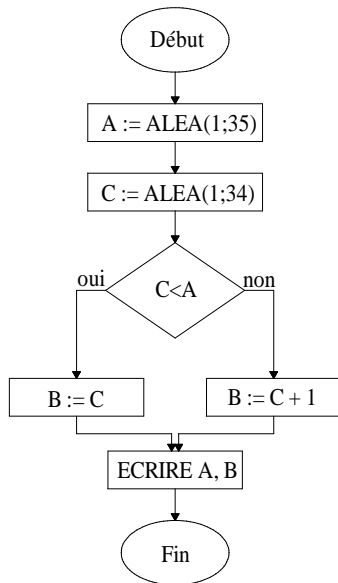
```
DEBUT
A:=0.1
B:=0.2
TANT QUE E(B)<A
FAIRE
    ECRIRE "Première borne ?"
    LIRE A
    ECRIRE "Deuxième borne ?"
    LIRE B
    SI A>B
        ALORS C:=A
        A:=B
        B:=C
    FIN SI
    SI E(B)<A
        ALORS ECRIRE "Il n'y a pas
d'entier entre ",A," et ",B
    FIN SI
FIN TANT QUE
SI E(A)<A
    ALORS A:=E(A)+1
FIN SI
B:=E(B)
C:=E((B-A+1)*RANDOM()+A)
ECRIRE C
FIN

Enfin on peut aussi décider du nombre de tirages. Pour cela il suffit de modifier les trois dernières lignes de l'algorithme précédent par la séquence suivante :
```

```
    ECRIRE "Nombre de tirages ?"
    LIRE D
    D:=E(D)
    SI D<1
        ALORS D:=1
    FIN SI
    SI D>10000
        ALORS D:=10000
    FIN SI
    POUR I DE 1 A D
    ITÉRER
        C:=E((B-A+1)*RANDOM()+A)
        ECRIRE C
    FIN D'ITÉRATION
FIN
```

2. Tirage sans remise de deux valeurs.

Désigner deux élèves au hasard dans une classe de 35 (tirer deux nombres distincts entre 1 et 35). On notera $ALEA(a ; b)$ la fonction qui génère un nombre pseudo-aléatoire compris entre a et b . Si cette fonction n'existe pas on peut la construire de la même manière qu'au paragraphe précédent.



```

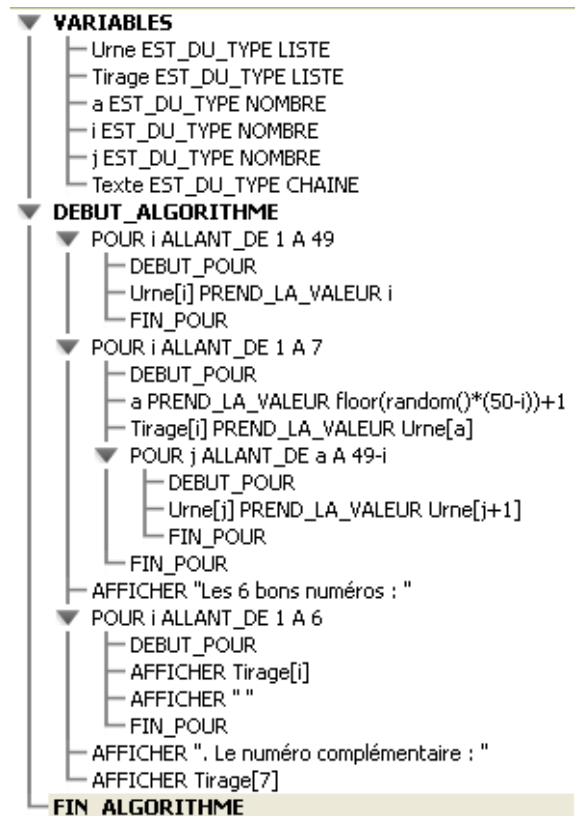
DEBUT
A := 1 + E(35*RANDOM())
B := 1 + E(34*RANDOM())
SI C < A
  ALORS B := C
  SINON B := C + 1
FIN SI
Ecrire A, " ; ", B
FIN
  
```

3. Tirage du Loto

Propose un tirage pseudo-aléatoire de six nombres, plus un, parmi 49 sans remise.

```

DEBUT
POUR i DE 1 A 49 ITERER
  urne(i) := i
FIN D'ITERER
POUR i DE 1 A 7 ITERER
  a := ALEA(1;50-i)
  tirage(i) := urne(a)
  POUR j DE a A 49-i ITERER
    urne(j) := urne(j+1)
  FIN D'ITERER
FIN D'ITERER
Ecrire "Les six bons numéros : "
POUR i DE 1 A 6 ITERER
  Ecrire tirage(i), " "
FIN D'ITERER
Ecrire "Numéro complémentaire : "
  ",tirage(7)
FIN
  
```



RÉSULTAT :

```

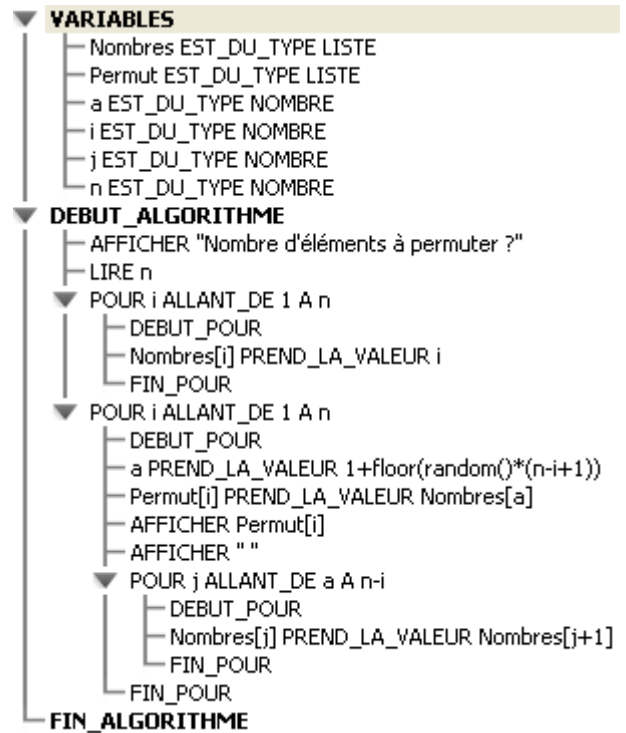
***Algorithme lancé***
Les 6 bons numéros : 29 28 40 27 11 48 . Le numéro complémentaire : 14
***Algorithme terminé***
  
```

4. Permutation de n éléments

Cet algorithme demande le nombre d'éléments de l'ensemble et propose une permutation pseudo-aléatoire.

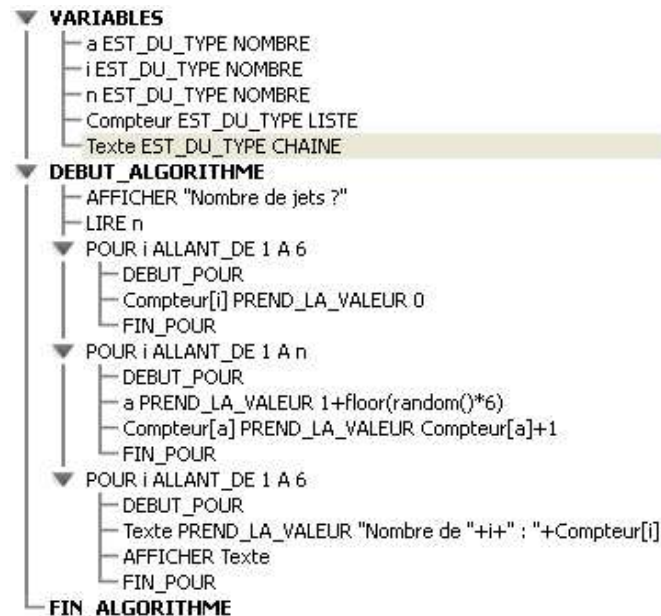
```

DEBUT
ECRIRE "Nombre d'éléments à permuter ?"
LIRE n
POUR i DE 1 A n ITERER
    nombres(i):=i
FIN D'ITERER
POUR i DE 1 A n ITERER
    a:=ALEA(1;n-i+1)
    permut(i):=nombres(a)
    ECRIRE permut(i)
    POUR j DE a A n-i
        nombres(j):=nombres(j+1)
    FIN D'ITERER
FIN D'ITERER
FIN
    
```

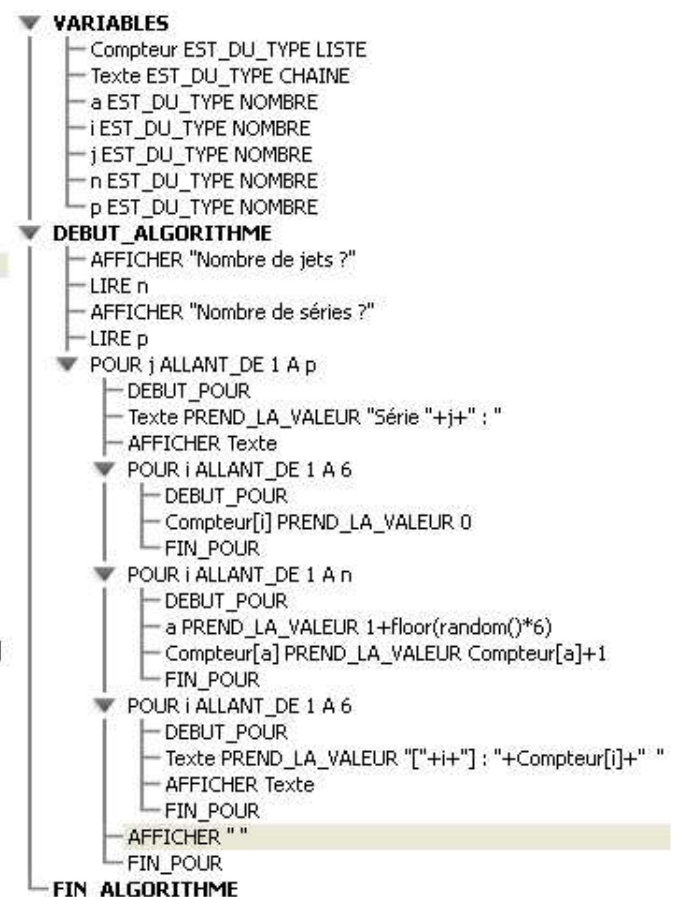


5. Lancers de dés

On utilise un dé à six faces. Ce programme demande le nombre de jets et totalise les résultats.



Le programme suivant demande le nombre de jets par série et le nombre de séries. On peut obtenir, par exemple, 1000 séries de 1000 jets en quelques secondes.



6. Ecriture décimale illimitée périodique d'un rationnel. (Division à virgule)

Le but de cet algorithme est de déterminer la partie périodique de l'écriture décimale illimitée d'un nombre rationnel. C'est en fait la division à virgule poursuivie suffisamment loin pour déterminer cette période.

```

DEBUT
ECRIRE "Numérateur ?"
LIRE N
ECRIRE "Dénominateur ?"
LIRE D
R:=N
Q:=E(R/D)
X:=CONCATENER(Q;";")
POUR I DE 1 A D-1
ITERER
  R:=(R-Q*D)*10
  Q:=E(R/D)
  X:=CONCATENER(X;Q)
FIN D'ITERATION
X:=CONCATENER(N;"/";D;" = ";X)
ECRIRE X
FIN
  
```

L'algorithme précédent présente l'inconvénient de ne pas toujours donner le nombre de décimales nécessaires à la détermination de la période, dans le cas où le numérateur est strictement inférieur à dix fois le dénominateur. Pour y remédier il suffit d'ajouter, entre les lignes 8 et 9, la séquence suivante :

```

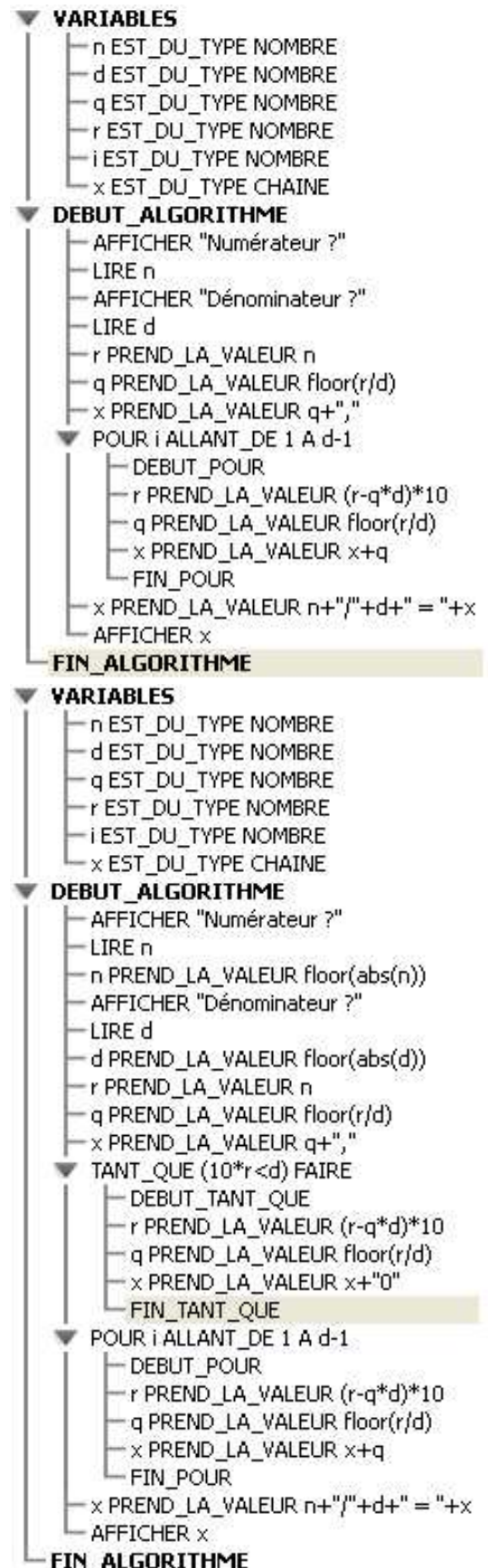
TANT QUE 10*N<D FAIRE
  R:=(R-Q*D)*10
  Q:=E(R/D)
  X:=CONCATENER(X;"0")
FIN TANT QUE
  
```

Cet algorithme donne $d - 1$ chiffres "significatifs" où d est le dénominateur. Ce n'est pas forcément la période mais la période comprend au plus $d - 1$ chiffres.

Exemples avec AlgoBox :

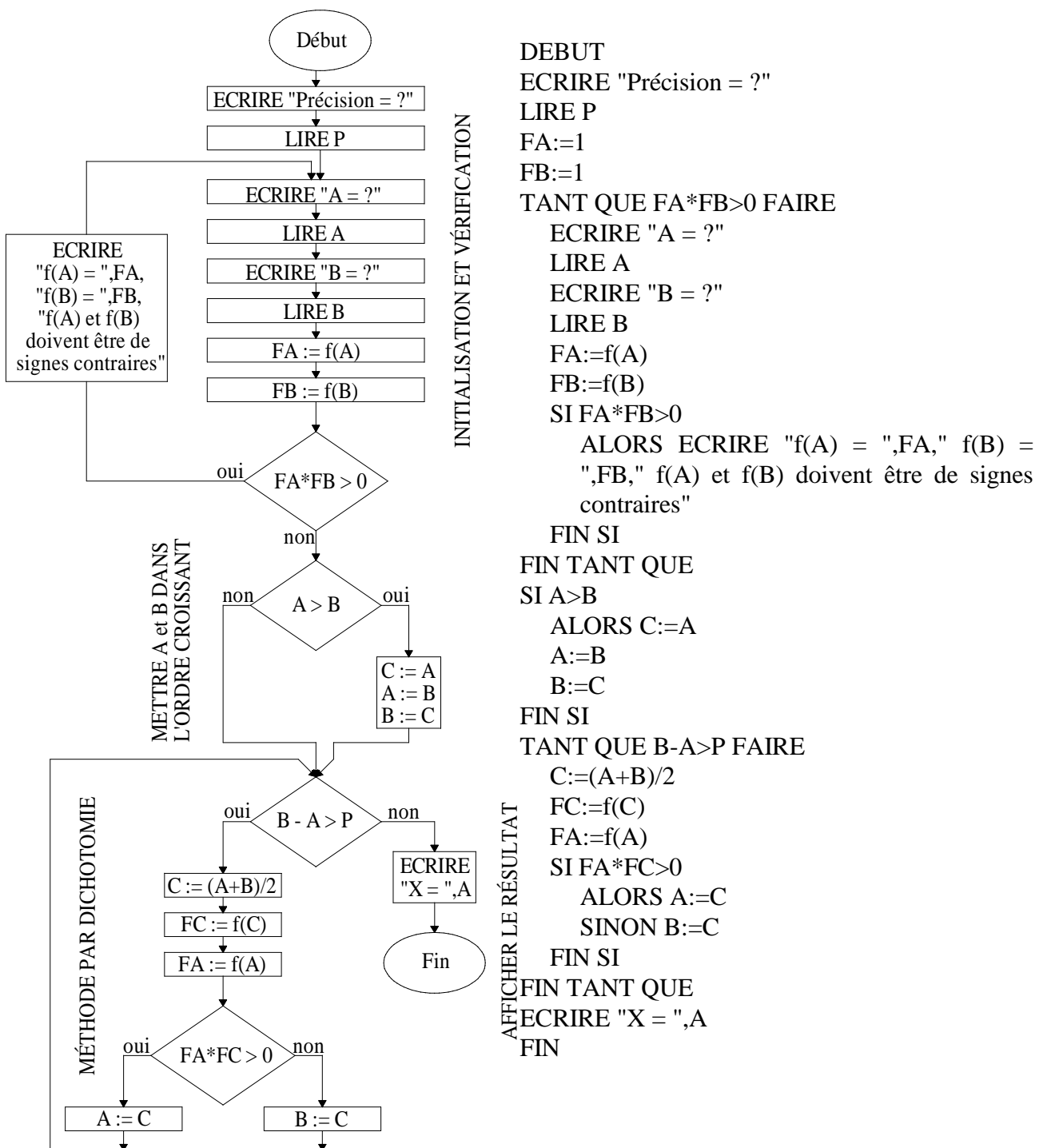
```

1/23 = 0,04347826086956521739130
5771/7 = 824,428571
5/43 = 0,116279069767441860465116279069767441860465
1/41 = 0,0243902439024390243902439024390243902439024390
  
```



7. Détermination des racines d'une équation polynomiale par dichotomie.

$2^{10} = 1024$ est voisin de 10^3 , on gagne 3 décimales toutes les dix opérations.



Il s'agit dans cet exemple de déterminer les zéros du polynôme définie sur \mathbb{R} par :

$$F_1(x) = x^6 + x^5 - 11x^4 - 10x^3 + 21x^2 + 9x - 3$$

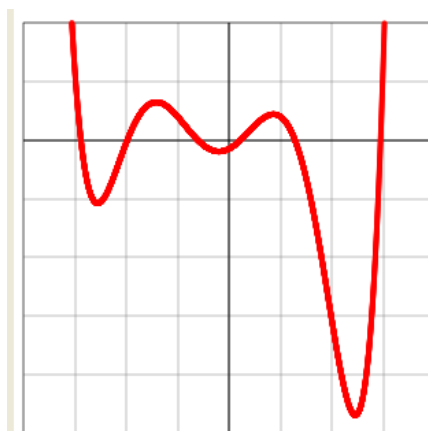
Ce polynôme s'annule pour six valeurs comprises entre -3 et 3. AlgoBox permet d'en déterminer des valeurs approchées avec une précision de 10^{-7} .

```

▼ VARIABLES
  A EST_DU_TYPE NOMBRE
  B EST_DU_TYPE NOMBRE
  C EST_DU_TYPE NOMBRE
  FA EST_DU_TYPE NOMBRE
  FB EST_DU_TYPE NOMBRE
  FC EST_DU_TYPE NOMBRE
  P EST_DU_TYPE NOMBRE
  I EST_DU_TYPE NOMBRE

▼ DEBUT_ALGORITHME
  POUR I ALLANT_DE 0 A 8000
    DEBUT_POUR
    TRACER_POINT (I/1000-4,F1(I/1000-4))
    FIN_POUR
  FA PREND_LA_VALEUR 1
  FB PREND_LA_VALEUR 1
  P PREND_LA_VALEUR 0.0000001
  TANT_QUE (FA*FB>0) FAIRE
    DEBUT_TANT_QUE
    AFFICHER "A = ?"
    LIRE A
    AFFICHER "B = ?"
    LIRE B
    FA PREND_LA_VALEUR F1(A)
    FB PREND_LA_VALEUR F1(B)
    SI (FA*FB>0) ALORS
      DEBUT_SI
      AFFICHER "f(A) = "
      AFFICHER FA
      AFFICHER " et f(B) = "
      AFFICHER FB
      AFFICHER "f(A) et f(B) doivent être de signes contraires"
      FIN_SI
    FIN_TANT_QUE
  SI (A>B) ALORS
    DEBUT_SI
    C PREND_LA_VALEUR A
    A PREND_LA_VALEUR B
    B PREND_LA_VALEUR C
    FIN_SI
  TANT_QUE (B-A>P) FAIRE
    DEBUT_TANT_QUE
    C PREND_LA_VALEUR (A+B)/2
    FC PREND_LA_VALEUR F1(C)
    FA PREND_LA_VALEUR F1(A)
    SI (FA*FC>0) ALORS
      DEBUT_SI
      A PREND_LA_VALEUR C
      FIN_SI
    SINON
      DEBUT_SINON
      B PREND_LA_VALEUR C
      FIN_SINON
    FIN_TANT_QUE
  AFFICHER "X ="
  AFFICHER A
  FIN_ALGORITHME

```



Xmin: -4 ; Xmax: 4 ; Ymin: -100 ; Ymax: 40 ; GradX: 1 ; GradY: 20

Opérations standards Utiliser une fonction numérique Dessiner dans un repère

Utiliser une fonction

Définir la fonction

F1(x)=

Le fichier Excel [Organigramme - Approximation par dichotomie.xls] montre le fonctionnement de cet algorithme à l'aide d'une animation.

1	Etape		34
2	Ordre dans le		4
3	n° du cycle		1
4			
5			
6			
7	VARIABLES		
8	Identificateur	Type	Valeur
9	P	Réelle	0,0000000000000001
10	A	Réelle	-3,0000000000000000
11	B	Réelle	-2,0000000000000000
12	C	Réelle	-2,5000000000000000
13	FA	Réelle	24,0000000000000000
14	FB	Réelle	24,0000000000000000
15	FC	Réelle	-21,2031250000000000
16	R	Texte	O
17			
18			
19			
20			
21			
22	DRAPEAUX		
23	Numéro	Description	Etat
24	1	FA*FB > 0	1
25	2	A > B	0
26	3	B - A > P	1
27	4	FA*FC > 0	0
28	5	R = "O"	1
29			
30			
31			
32			
33			
34			
35			
36			
37			
38			
39			
40	34		
41			
42			
43			
44			
45			
46			
47			
48			
49			
50			
51			
52			
53			
54			
55			
56			
57			
58			
59			
60			
61			

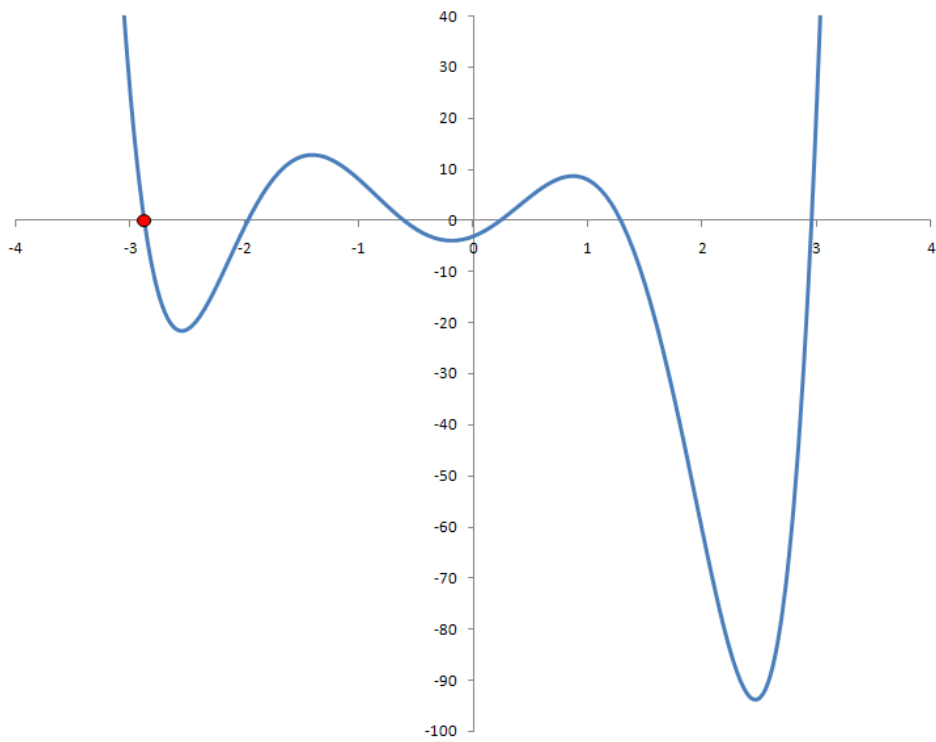
Flowchart:

```

    graph TD
      DEBUT --> ECRIRE_P["ECRIRE 'P = ?'"]
      ECRIRE_P --> LIRE_P
      LIRE_P --> ECRIRE_A["ECRIRE 'A = ?'"]
      ECRIRE_A --> LIRE_A
      LIRE_A --> ECRIRE_B["ECRIRE 'B = ?'"]
      ECRIRE_B --> LIRE_B
      LIRE_B --> FA["FA = f(A)"]
      FA --> FB["FB = f(B)"]
      FB --> FA_FB_0{"FA*FB > 0"}
      FA_FB_0 -- non --> LIRE_R["LIRE R"]
      LIRE_R --> ECRIRE["ECRIRE 'f(A) = ', FA, 'f(B) = ', FB, 'f(A) et f(B) doivent être de signes contraires 'Recommencer (O/N) ?'"]
      ECRIRE --> FA_FB_0
      FA_FB_0 -- oui --> A_B{"A > B"}
      A_B -- non --> C_A["C := A"]
      C_A --> A_B
      A_B -- oui --> A_B_assign["A = B"]
      A_B_assign --> B_C["B = C"]
      B_C --> B_A_P{"B - A > P"}
      B_A_P -- non --> ECRIRE_X["ECRIRE 'X = 'A'"]
      ECRIRE_X --> FIN
      B_A_P -- oui --> C_mid["C := (A+B)/2"]
      C_mid --> FC["FC = f(C)"]
      FC --> FA_mid["FA = f(A)"]
      FA_mid --> FA_FC_0{"FA*FC > 0"}
      FA_FC_0 -- non --> B_C_assign["B := C"]
      B_C_assign --> B_A_P
      FA_FC_0 -- oui --> A_C_assign["A := C"]
      A_C_assign --> FA_FC_0
  
```

Output Table:

	Affichage	Entrée	
1	P = ?	1E-15	2
4	A = ?	-4	5
7	B = ?	-3	6
13	f(A) = 1193		
	f(B) = 24		
	f(A) et f(B) doivent être de signes contraires		
	Recommencer (O/N) ?	O	14
17	A = ?	-2	18
20	B = ?	-3	21
331			



8. Distance de deux points, milieu d'un segment, équation d'une droite passant par deux points, médiatrice d'un segment.

A et B sont deux points de coordonnées respectives (x_A, y_A) et (x_B, y_B) dans le plan rapporté à un repère orthonormal (O, \vec{i}, \vec{j}) .

```

DEBUT
ECRIRE "Abscisse de A ?"
LIRE XA
ECRIRE "Ordonnée de A ?"
LIRE YA
ECRIRE "Abscisse de B ?"
LIRE XB
ECRIRE "Ordonnée de B ?"
LIRE YB
ECRIRE "A(",XA,";",YA,") et B(",XB,";",YB,")"
AB:=RACINE((XB-XA)^2+(YB-YA)^2)
ECRIRE "AB = ",AB
XI:=(XA+XB)/2
YI:=(YA+YB)/2
ECRIRE "I milieu de [AB], XI = ",XI," et YI = ",YI
SI XA=XB
  ALORS
  ECRIRE "(AB) : x = ",XA
  ECRIRE "Médiatrice de (AB) : y = ",YI
  SINON
  M1:=(YB-YA)/(XB-XA)
  P1:=(XB*YA-XA*YB)/(XB-XA)
  M2:=(XA-XB)/(YB-YA)
  P2:=(XB^2+YB^2-XA^2-YA^2)/2/(YB-YA)
  ECRIRE "(AB) : y = ",M1,"x + ",P1
  ECRIRE "Médiatrice de (AB) : y = ",M2,"x + ",P2
FIN SI
FIN
  
```

```

VARIABLES
- XA EST_DU_TYPE NOMBRE
- YA EST_DU_TYPE NOMBRE
- XB EST_DU_TYPE NOMBRE
- YB EST_DU_TYPE NOMBRE
- XI EST_DU_TYPE NOMBRE
- YI EST_DU_TYPE NOMBRE
- AB EST_DU_TYPE NOMBRE
- M1 EST_DU_TYPE NOMBRE
- P1 EST_DU_TYPE NOMBRE
- M2 EST_DU_TYPE NOMBRE
- P2 EST_DU_TYPE NOMBRE
- Texte EST_DU_TYPE CHAINE

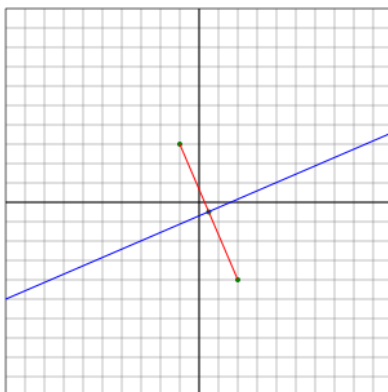
DEBUT_ALGORITHME
- AFFICHER "Abscisse de A ?"
- LIRE XA
- AFFICHER "Ordonnée de A ?"
- LIRE YA
- AFFICHER "Abscisse de B ?"
- LIRE XB
- AFFICHER "Ordonnée de B ?"
- LIRE YB
- Texte PREND_LA_VALEUR "A(",XA,";",YA,") et B(",XB,";",YB,")"
- AFFICHER Texte
- AB PREND_LA_VALEUR sqrt(pow(XB-XA,2)+pow(YB-YA,2))
- AFFICHER AB = "
- AFFICHER AB
- XI PREND_LA_VALEUR (XA+XB)/2
- YI PREND_LA_VALEUR (YA+YB)/2
- Texte PREND_LA_VALEUR "Coordonnées du milieu de [AB] : (" + XI + "; " + YI + ")"
- AFFICHER Texte

SI (XA==XB) ALORS
- DEBUT_SI
- Texte PREND_LA_VALEUR "Equation de la droite (AB) : x = " + XA
- AFFICHER Texte
- Texte PREND_LA_VALEUR "Equation de la médiatrice du segment [AB] : y = " + YI
- AFFICHER Texte
- FIN_SI
SINON
- DEBUT_SINON
- M1 PREND_LA_VALEUR (YB-YA)/(XB-XA)
- P1 PREND_LA_VALEUR (XB*YA-XA*YB)/(XB-XA)
- M2 PREND_LA_VALEUR (XA-XB)/(YB-YA)
- P2 PREND_LA_VALEUR (pow(XB,2)+pow(YB,2)-pow(XA,2)-pow(YA,2))/2/(YB-YA)
- Texte PREND_LA_VALEUR "Equation de la droite (AB) : y = " + M1 + "x + " + P1
- AFFICHER Texte
- Texte PREND_LA_VALEUR "Equation de la médiatrice du segment [AB] : y = " + M2 + "x + " + P2
- AFFICHER Texte
- FIN_SINON
- TRACER_POINT (XA,YA)
- TRACER_POINT (XB,YB)
- TRACER_POINT (XI,YI)
- TRACER_SEGMENT (XA,YA)->(XB,YB)
- TRACER_SEGMENT (-100,F1(-100))->(100,F1(100))
FIN_ALGORITHME
  
```

Toujours avec AlgoBox

```

Abscisse de A ?
Ordonnée de A ?
Abscisse de B ?
Ordonnée de B ?
A(-1;3) et B(2;-4)
AB = 7.61577311
Coordonnées du milieu de [AB] : (0.5;-0.5)
Equation de la droite (AB) : y = -2.33333333x + 0.66666667
Equation de la médiatrice du segment [AB] : y = 0.42857143x + -0.71428571
  
```



Xmin: -10 ; Xmax: 10 ; Ymin: -10 ; Ymax: 10 ; GradX: 1 ; GradY: 1

L'algorithme précédent traite mal le cas où la médiatrice est parallèle aux axes. La version ci-dessous corrige cet inconvénient.



9. Avec trois points.

Déterminer les longueurs des côtés, tester une condition d'alignement, déterminer les équations des droites, des médiatrices des côtés, déterminer les coordonnées du centre du cercle circonscrit ainsi que son rayon. Tracer le triangle, les 3 médiatrices et le cercle circonscrit. On peut imaginer encore bien d'autres choses (détermination de la nature du triangle, distance d'un point à la droite passant par les deux autres, détermination des médianes, des hauteurs, des bissectrices, enfin tout sur le triangle et plus encore). Avec AlgoBox :

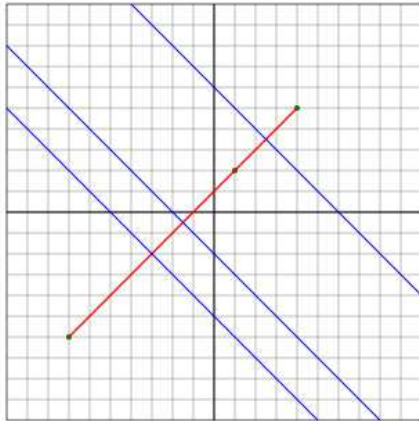
```

VARIABLES
- XA EST_DU_TYPE NOMBRE
- YA EST_DU_TYPE NOMBRE
- XB EST_DU_TYPE NOMBRE
- YB EST_DU_TYPE NOMBRE
- XC EST_DU_TYPE NOMBRE
- YC EST_DU_TYPE NOMBRE
- XI EST_DU_TYPE NOMBRE
- YI EST_DU_TYPE NOMBRE
- AB EST_DU_TYPE NOMBRE
- BC EST_DU_TYPE NOMBRE
- CA EST_DU_TYPE NOMBRE
- R EST_DU_TYPE NOMBRE
- M1 PREND_LA_VALEUR (YB-YA)/(XB-XA)
- P1 PREND_LA_VALEUR (XB*YA-XA*YB)/(XB-XA)
- M4 PREND_LA_VALEUR (XA-XB)/(YB-YA)
- P4 PREND_LA_VALEUR (pow(XB,2)+pow(YB,2)-pow(XA,2)-pow(YA,2))/2/(YB-YA)
- M2 PREND_LA_VALEUR (YC-YB)/(XC-XB)
- P2 PREND_LA_VALEUR (XC*YB-XB*YC)/(XC-XB)
- M5 PREND_LA_VALEUR (XB-XC)/(YC-YB)
- P5 PREND_LA_VALEUR (pow(XC,2)+pow(YC,2)-pow(XB,2)-pow(YB,2))/2/(YC-YB)
- M3 PREND_LA_VALEUR (YA-YC)/(XA-XC)
- P3 PREND_LA_VALEUR (XA*YC-XC*YA)/(XA-XC)
- M6 PREND_LA_VALEUR (XC-XA)/(YA-YC)
- P6 PREND_LA_VALEUR (pow(XA,2)+pow(YA,2)-pow(XC,2)-pow(YC,2))/2/(YA-YC)
- TRACER_POINT (XA,YA)
- TRACER_POINT (XB,YB)
- TRACER_POINT (XC,YC)
- TRACER_SEGMENT (XA,YA)->(XB,YB)
- TRACER_SEGMENT (XB,YB)->(XC,YC)
- TRACER_SEGMENT (XC,YC)->(XA,YA)
- TRACER_SEGMENT (-100,-100*M4+P4)->(100,100*M4+P4)
- TRACER_SEGMENT (-100,-100*M5+P5)->(100,100*M5+P5)
- TRACER_SEGMENT (-100,-100*M6+P6)->(100,100*M6+P6)
- SI (abs(AB+BC-CA)<0.000001 OU abs(BC+CA-AB)<0.000001 OU abs(CA+AB-BC)<0.000001) ALORS
  - DEBUT_SI
  - AFFICHER "A, B et C semblent alignés"
  - Texte PREND_LA_VALEUR "Les droites (AB), (BC) et (CA) semblent confondues elles ont pour équation :
  - AFFICHER Texte
  - Texte PREND_LA_VALEUR "La médiatrice du côté [AB] a pour équation : y = "+M4+"x + "+P4
  - AFFICHER Texte
  - Texte PREND_LA_VALEUR "La médiatrice du côté [BC] a pour équation : y = "+M5+"x + "+P5
  - AFFICHER Texte
  - Texte PREND_LA_VALEUR "La médiatrice du côté [CA] a pour équation : y = "+M6+"x + "+P6
  - AFFICHER Texte
  - AFFICHER "Ces 3 droites sont parallèles"
  - FIN_SI
  - SINON
  - DEBUT_SINON
  - AFFICHER "Les points A, B et C ne sont pas alignés"
  - Texte PREND_LA_VALEUR "La droite (AB) à pour équation : y = "+M1+"x + "+P1
  - AFFICHER Texte
  - Texte PREND_LA_VALEUR "La droite (BC) à pour équation : y = "+M2+"x + "+P2
  - AFFICHER Texte
  - Texte PREND_LA_VALEUR "La droite (CA) à pour équation : y = "+M3+"x + "+P3
  - AFFICHER Texte
  - Texte PREND_LA_VALEUR "La médiatrice du côté [AB] a pour équation : y = "+M4+"x + "+P4
  - AFFICHER Texte
  - Texte PREND_LA_VALEUR "La médiatrice du côté [BC] a pour équation : y = "+M5+"x + "+P5
  - AFFICHER Texte
  - Texte PREND_LA_VALEUR "La médiatrice du côté [CA] a pour équation : y = "+M6+"x + "+P6
  - AFFICHER Texte
  - XI PREND_LA_VALEUR (P5-P4)/(M4-M5)
  - YI PREND_LA_VALEUR M4*XI+P4
  - Texte PREND_LA_VALEUR "Les 3 médiatrices sont concourantes au point I de coordonnées (" +XI+" ; "
  - AFFICHER Texte
  - R PREND_LA_VALEUR sqrt(pow(XA-XI,2)+pow(YA-YI,2))
  - Texte PREND_LA_VALEUR "Ce point est le centre du cercle de rayon R = "+R+" circonscrit au trianl
  - AFFICHER Texte
  - Nb_de_pts PREND_LA_VALEUR 100
  - POUR I ALLANT_DE 1 A Nb_de_pts
  - DEBUT_POUR
  - A PREND_LA_VALEUR XI+R*cos(I*2*Math.PI/Nb_de_pts)
  - B PREND_LA_VALEUR YI+R*sin(I*2*Math.PI/Nb_de_pts)
  - C PREND_LA_VALEUR XI+R*cos((I+1)*2*Math.PI/Nb_de_pts)
  - D PREND_LA_VALEUR YI+R*sin((I+1)*2*Math.PI/Nb_de_pts)
  - TRACER_SEGMENT (A,B)->(C,D)
  - FIN_POUR
  - FIN_SINON
FIN_ALGORITHME

```

RÉSULTAT :

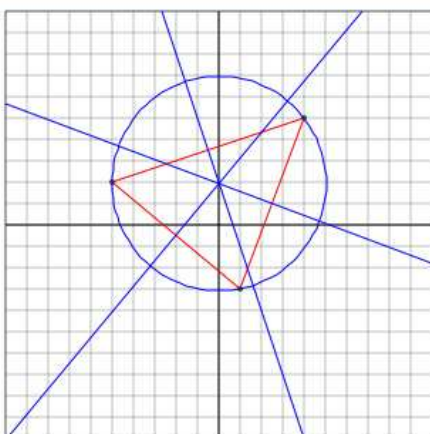
```
A(-7;-6), B(1;2), C(4;5)
AB = 11.3137085, BC = 4.24264069, CA = 15.55634919
A, B et C semblent alignés
Les droites (AB), (BC) et (CA) semblent confondues elles ont pour équation : y = 1x +
La médiatrice du côté [AB] a pour équation : y = -1x + -5
La médiatrice du côté [BC] a pour équation : y = -1x + 6
La médiatrice du côté [CA] a pour équation : y = -1x + -2
Ces 3 droites sont parallèles
```



Xmin: -10 ; Xmax: 10 ; Ymin: -10 ; Ymax: 10 ; GradX: 1 ; GradY: 1

RÉSULTAT :

```
A(-5;2), B(4;5), C(1;-3)
AB = 9.48683298, BC = 8.54400375, CA = 7.81024968
Les points A, B et C ne sont pas alignés
La droite (AB) à pour équation : y = 0.33333333x + 3.66666667
La droite (BC) à pour équation : y = 2.66666667x + -5.66666667
La droite (CA) à pour équation : y = -0.83333333x + -2.16666667
La médiatrice du côté [AB] a pour équation : y = -3x + 2
La médiatrice du côté [BC] a pour équation : y = -0.375x + 1.9375
La médiatrice du côté [CA] a pour équation : y = 1.2x + 1.9
Les 3 médiatrices sont concourantes au point I de coordonnées (0.02380952;1.92857144)
Ce point est le centre du cercle de rayon R = 5.02431728 circonscrit au triangle ABC
***Algorithme terminé***
```



Xmin: -10 ; Xmax: 10 ; Ymin: -10 ; Ymax: 10 ; GradX: 1 ; GradY: 1

Cet algorithme, relativement simple, ne traite pas correctement les cas où les médiatrices sont parallèles aux axes. L'algorithme suivant corrige cette insuffisance, mais il est beaucoup plus lourd. Il est conçu aussi différemment. Les points s'appellent A1, A2, A3 afin de pouvoir utiliser la variable d'itération pour les nommer.

```

▼ VARIABLES
  i EST_DU_TYPE NOMBRE
  j EST_DU_TYPE NOMBRE
  k EST_DU_TYPE NOMBRE
  n EST_DU_TYPE NOMBRE
  p EST_DU_TYPE NOMBRE
  x EST_DU_TYPE LISTE
  y EST_DU_TYPE LISTE
  a EST_DU_TYPE LISTE
  b EST_DU_TYPE LISTE
  c EST_DU_TYPE LISTE
  d EST_DU_TYPE LISTE
  Texte EST_DU_TYPE CHAINE
  m EST_DU_TYPE NOMBRE

▼ DEBUT_ALGORITHME
  ▼ POUR i ALLANT_DE 1 A 3
    DEBUT_POUR
    Texte PREND_LA_VALEUR "Quelle est l'abscisse du point A"+i+" ?"
    AFFICHER Texte
    LIRE x[i]
    Texte PREND_LA_VALEUR "Quelle est l'ordonnée du point A"+i+" ?"
    AFFICHER Texte
    LIRE y[i]
    FIN_POUR

  ▼ POUR i ALLANT_DE 1 A 3
    DEBUT_POUR
    Texte PREND_LA_VALEUR "A"+i+"("+x[i]+";"+y[i]+") "
    AFFICHER Texte
    FIN_POUR

  AFFICHER " "

  ▼ POUR i ALLANT_DE 1 A 3
    DEBUT_POUR
    j PREND_LA_VALEUR i%3+1
    k PREND_LA_VALEUR (i+1)%3+1
    d[i] PREND_LA_VALEUR sqrt(pow(x[j]-x[k],2)+pow(y[j]-y[k],2))
    x[i+3] PREND_LA_VALEUR (x[j]+x[k])/2
    y[i+3] PREND_LA_VALEUR (y[j]+y[k])/2
    a[i] PREND_LA_VALEUR y[k]-y[j]
    b[i] PREND_LA_VALEUR x[j]-x[k]
    c[i] PREND_LA_VALEUR x[k]*y[i]-x[j]*y[k]
    a[i+3] PREND_LA_VALEUR x[k]-x[j]
    b[i+3] PREND_LA_VALEUR y[k]-y[j]
    c[i+3] PREND_LA_VALEUR (pow(x[j],2)-pow(x[k],2)+pow(y[j],2)-pow(y[k],2))/2
    FIN_POUR

  ▼ POUR i ALLANT_DE 1 A 3
    DEBUT_POUR
    j PREND_LA_VALEUR i%3+1
    k PREND_LA_VALEUR (i+1)%3+1
    Texte PREND_LA_VALEUR "A"+i+"A"+j+"="+d[k]+" "
    AFFICHER Texte
    FIN_POUR

  AFFICHER " "

  ▼ POUR i ALLANT_DE 1 A 3
    DEBUT_POUR
    j PREND_LA_VALEUR i%3+1
    k PREND_LA_VALEUR (i+1)%3+1
    Texte PREND_LA_VALEUR "Le point l"+i+" milieu de [A"+j+"A"+k+"] a pour coordonnées ("+(x[j]+x[k])/2+";"+(y[j]+y[k])/2+")"
    AFFICHER Texte
    FIN_POUR

  ▼ SI (d[1]==0 OU d[2]==0 OU d[3]==0) ALORS
    DEBUT_SI
    AFFICHER "Au moins deux des points sont confondus"
    FIN_SI
  ▼ SINON
    DEBUT_SINON
    ▼ POUR i ALLANT_DE 1 A 3
      DEBUT_POUR
      TRACER_POINT (x[i],y[i])
      TRACER_SEGMENT (x[i],y[i])->(x[i%3+1],y[i%3+1])
      ▼ SI (b[i+3]==0) ALORS
        DEBUT_SI
        TRACER_SEGMENT (-c[i+3]/a[i+3],-100)->(-c[i+3]/a[i+3],100)
        FIN_SI
      ▼ SINON
        DEBUT_SINON
        TRACER_SEGMENT (-100,(100*a[i+3]-c[i+3])/b[i+3])->(100,(-100*a[i+3]-c[i+3])/b[i+3])
        FIN_SINON
      FIN_POUR
    ▼ SI (abs((x[2]-x[1])*y[3]-y[1])-(x[3]-x[1])*y[2]-y[1])<0.00000001) ALORS
      DEBUT_SI
      AFFICHER "Les points A1, A2, A3 semblent alignés"
      ▼ SI (a[1]==0) ALORS
        DEBUT_SI
        p PREND_LA_VALEUR -c[1]/b[1]
        Texte PREND_LA_VALEUR "Les droites (A1A2) = (A2A3) = (A3A1) ont pour équations y = "+p
        FIN_SI
      ▼ SINON
        DEBUT_SINON
        ▼ SI (b[1]==0) ALORS
          DEBUT_SI
          p PREND_LA_VALEUR -c[1]/a[1]
          Texte PREND_LA_VALEUR "Les droites (A1A2) = (A2A3) = (A3A1) ont pour équations x = "+p
          FIN_SI
        ▼ SINON
          DEBUT_SINON
          m PREND_LA_VALEUR -a[1]/b[1]
          p PREND_LA_VALEUR -c[1]/b[1]
          ▼ SI (p==0) ALORS
            DEBUT_SI
            Texte PREND_LA_VALEUR "Les droites (A1A2) = (A2A3) = (A3A1) ont pour équations y = "+m+"x"
            FIN_SI
          ▼ SINON
            DEBUT_SINON
            Texte PREND_LA_VALEUR "Les droites (A1A2) = (A2A3) = (A3A1) ont pour équations y = "+m+"x + "+p
            FIN_SINON
          FIN_SINON
        FIN_SINON
      FIN_SI
    AFFICHER Texte
  FIN_SI

```




La partie affichage peut être optimisée pour gagner quelques lignes. Cette version soigne particulièrement l'affichage évitant par exemple d'écrire $y = 3x + 0$ au lieu de $y = 3x$. Il utilise, pour les calculs, les équations de droites sous leur forme standard ($ax + by + c = 0$) pour éviter la dissociation des cas qu'introduit l'équation réduite ($y = mx + p$ ou $x = cste$). Il affiche cependant les résultats sous cette forme réduite.

10. Algorithme de Prabhakar

$u_0 \in \mathbb{N}$.

$$\left(u_k = \overline{a_n a_{n-1} \dots a_2 a_1 a_0} = \sum_{i=0}^n (a_i \times 10^i) \right) \Rightarrow \left(u_{k+1} = \sum_{i=0}^n a_i^2 \right).$$

Que constate-t-on ?

Quel que soit l'entier naturel duquel on part on aboutit soit au cycle

$4 \rightarrow 16 \rightarrow 37 \rightarrow 58 \rightarrow 89 \rightarrow 145 \rightarrow 42 \rightarrow 20 \rightarrow 4$ etc. soit à 1, soit à 0 (uniquement pour 0).

On peut étudier ce qu'il advient pour $p > 2$ en posant $u_{k+1} = \sum_{i=0}^n a_i^p$

Etude proposée avec *Excel*.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE							
1	Algorithme de Prabhakar																																					
2	Somme des carrés des chiffres														Somme des puissances 3 ièmes des chiffres																							
3		1	2	3	4	5	6	7	8	9	10	11	12	13	14		1	2	3	4	5	6	7	8	9	10	11	12	13	14								
4	164	374	428	942	856	1	6	4	3	7	4	4	2	8	9	4	2	8	5	164	374	428	942	856	1	6	4	3	7	4	4	2	8	9	4	2	8	5
5		292	2	9	2												2737	2	7	3	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
6		89	8	9													721	7	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
7		145	1	4	5												352	3	5	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
8		42	4	2													160	1	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
9		20	2	0													217	2	1	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10		4	4														352	3	5	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11		16	1	6													160	1	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12		37	3	7													217	2	1	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13		58	5	8													352	3	5	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14		89	8	9													160	1	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15		145	1	4	5												217	2	1	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16		42	4	2													352	3	5	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17		20	2	0													160	1	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18		4	4														217	2	1	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19		16	1	6													352	3	5	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

11. Recherche des extrema d'une fonction sur un intervalle

Recherche du maximum et de minimum d'une fonction par balayage.

Proposition à l'aide d'AlgoBox.

Définition de la fonction :

Opérations standards | Utiliser une fonction numérique | Dessiner dans un repère

Utiliser une fonction

Définir la fonction

F1(x)=

Définition de la fenêtre graphique :

Opérations standards | Utiliser une fonction numérique | Dessiner dans un repère

Utiliser un repère

Définir le repère

Xmin : Xmax :

Ymin : Ymax :

Graduations X : Graduations Y :

Ajouter code

Le programme :

VARIABLES

- a EST_DU_TYPE NOMBRE
- b EST_DU_TYPE NOMBRE
- n EST_DU_TYPE NOMBRE
- x EST_DU_TYPE NOMBRE
- i EST_DU_TYPE NOMBRE
- y EST_DU_TYPE NOMBRE
- xmax EST_DU_TYPE NOMBRE
- ymax EST_DU_TYPE NOMBRE
- xmin EST_DU_TYPE NOMBRE
- ymin EST_DU_TYPE NOMBRE
- Texte EST_DU_TYPE CHAINE

DEBUT_ALGORITHME

- AFFICHER "Borne inférieure ?"
 - LIRE a
 - AFFICHER "Borne supérieure ?"
 - LIRE b
 - AFFICHER "Nombre d'itérations ?"
 - LIRE n
 - x PREND_LA_VALEUR a
 - xmax PREND_LA_VALEUR a
 - ymax PREND_LA_VALEUR F1(a)
 - xmin PREND_LA_VALEUR a
 - ymin PREND_LA_VALEUR F1(a)
 - ▼ POUR i ALLANT_DE 1 A n
 - DEBUT_POUR
 - x PREND_LA_VALEUR $a+i*(b-a)/n$
 - ▼ SI (F1(x)>ymax) ALORS
 - DEBUT_SI
 - xmax PREND_LA_VALEUR x
 - ymax PREND_LA_VALEUR F1(x)
 - FIN_SI
 - ▼ SI (F1(x)<ymin) ALORS
 - DEBUT_SI
 - xmin PREND_LA_VALEUR x
 - ymin PREND_LA_VALEUR F1(x)
 - FIN_SI
 - FIN_POUR
 - Texte PREND_LA_VALEUR "Sur l'intervalle ["+a+";"+b+"],(calcul avec "+n+" itérations)"
 - AFFICHER Texte
 - Texte PREND_LA_VALEUR "le maximum de la fonction semble être égal à "+ymax+" pour x = "+xmax
 - AFFICHER Texte
 - Texte PREND_LA_VALEUR "le minimum de la fonction semble être égal à "+ymin+" pour x = "+xmin
 - AFFICHER Texte
 - n PREND_LA_VALEUR 1000
 - ▼ POUR i ALLANT_DE 0 A n
 - DEBUT_POUR
 - TRACER_POINT (a+i/n*(b-a),F1(a+i/n*(b-a)))
 - FIN_POUR
- ## FIN_ALGORITHME

Le résultat :

ALGOBOX : MINMAX1

TESTER L'ALGORITHME :

Lancer algorithme

(cliquer sur le bouton ci-dessus pour lancer ou relancer l'exécution de l'algorithme)

Remarque : si les messages "Algorithme lancé" et "Algorithme terminé" n'apparaissent pas au bout d'un moment dans la zone ci-dessous, c'est que l'algorithme contient une erreur.

RÉSULTAT :

```
***Algorithme lancé***  
Borne inférieure ?  
Borne supérieure ?  
Nombre d'itérations ?  
Sur l'intervalle [-2;1], (calcul avec 100000 itérations)  
le maximum de la fonction semble être égal à 2.41650197 pour x = -1.38743  
le minimum de la fonction semble être égal à -2.26835382 pour x = 0.72076  
  
***Algorithme terminé***
```

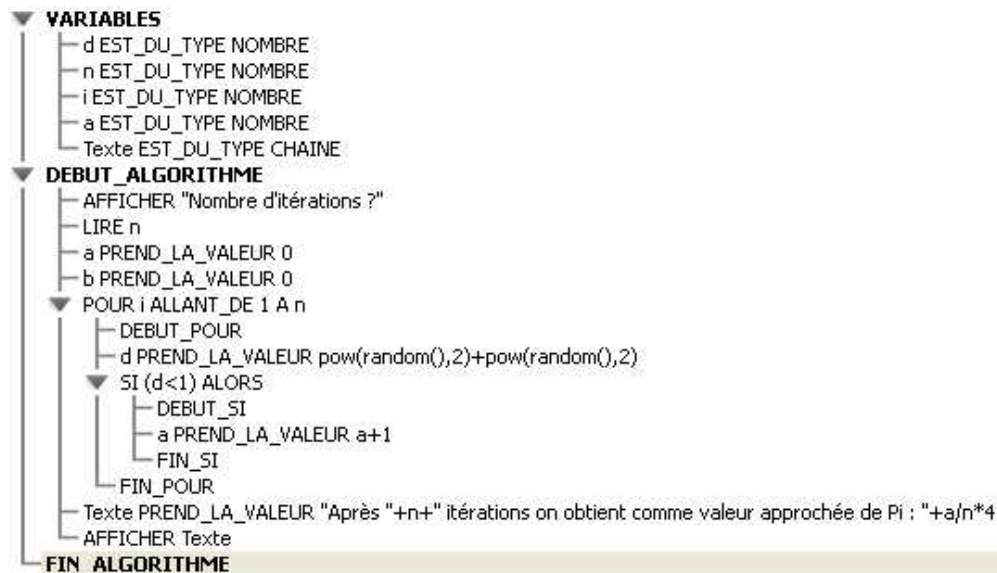


Xmin: -3 ; Xmax: 2 ; Ymin: -3 ; Ymax: 3 ; GradX: 1 ; GradY: 1

12. Méthode de Monte-Carlo

RÉSULTAT :

```
***Algorithme lancé***  
Nombre d'itérations ?  
Après 200000 itérations on obtient comme valeur approchée de Pi : 3.14222  
  
***Algorithme terminé***
```



13. Les aiguilles de Buffon

Voir fichier Excel : "Aiguilles de Buffon.xls"

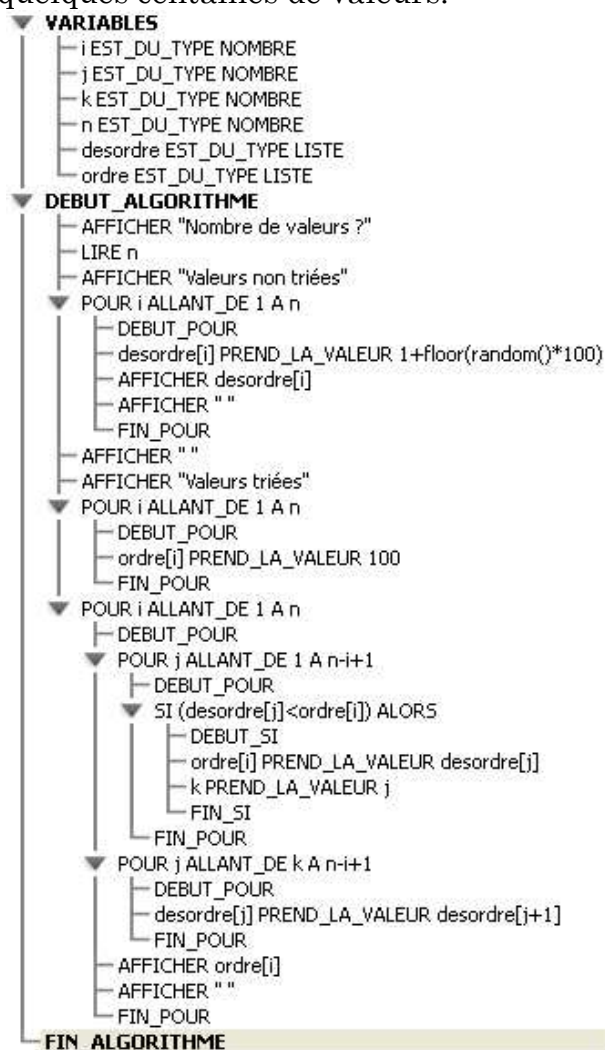
14. Tri par sélection

On cherche dans la liste la plus petite valeur. On la place au début et on recommence. Dans la version 0.4 d'Algobox le nombre d'affichages consécutifs est limité à 1000 caractères, au-delà, l'utilisateur doit confirmer sa volonté de continuer à afficher des données. L'algorithme fonctionne agréablement pour quelques centaines de valeurs.

```

DEBUT
ECRIRE "Nombre de valeurs ?"
LIRE n
POUR i DE 1 A n ITERER
  desordre(i):=ALEA(1;100)
FIN D'ITERER
POUR i DE 1 A n ITERER
  ordre(i):=100
FIN D'ITERER
POUR i DE 1 A n ITERER
  POUR j DE 1 A n-i+1 ITERER
    SI desordre(j)<ordre(i) ALORS
      ordre(i):=desordre(j)
      k:=j
    FIN SI
  FIN D'ITERER
  POUR j DE k A n-i+1 ITERER
    desordre(j):=desordre(j+1)
  FIN D'ITERER
ECRIRE ordre(i)
FIN D'ITERER
FIN

```

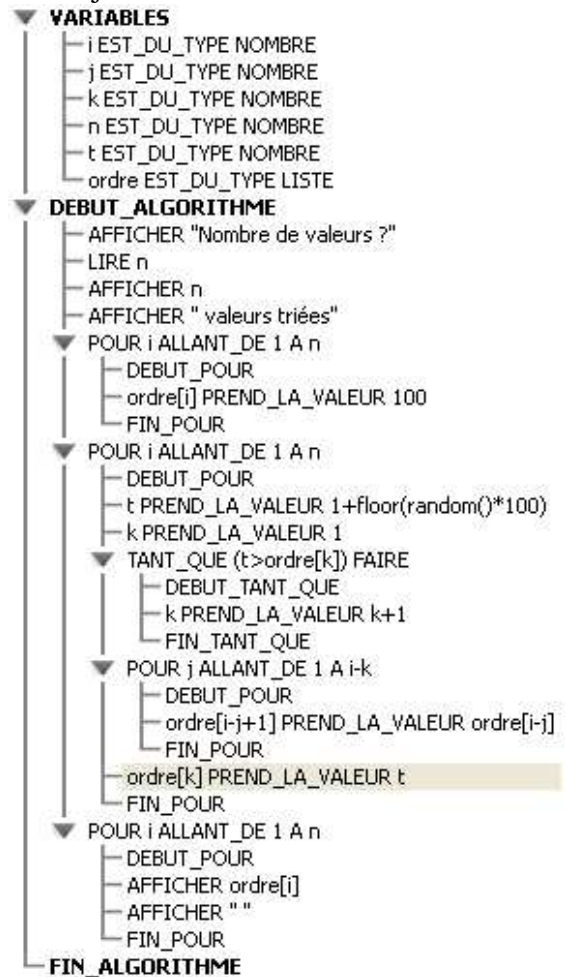


15. Tri par insertion

On prend une nouvelle valeur et on l'insère dans la liste déjà triée.

```

DEBUT
ECRIRE "Nombre de valeurs ?"
LIRE n
POUR i DE 1 A n ITERER
    ordre(i):=100
FIN D'ITERER
POUR i DE 1 A n ITERER
    t:=ALEA(1;100)
    k:=1
    TANT QUE t>ordre(k) FAIRE
        k:=k+1
    FIN TANT QUE
    POUR j DE 1 A i-k ITERER
        ordre(i-j+1):=ordre(i-j)
    FIN D'ITERER
    ordre(k):=t
FIN D'ITERER
POUR i DE 1 A n ITERER
    ECRIRE ordre(i)
FIN D'ITERER
FIN
    
```

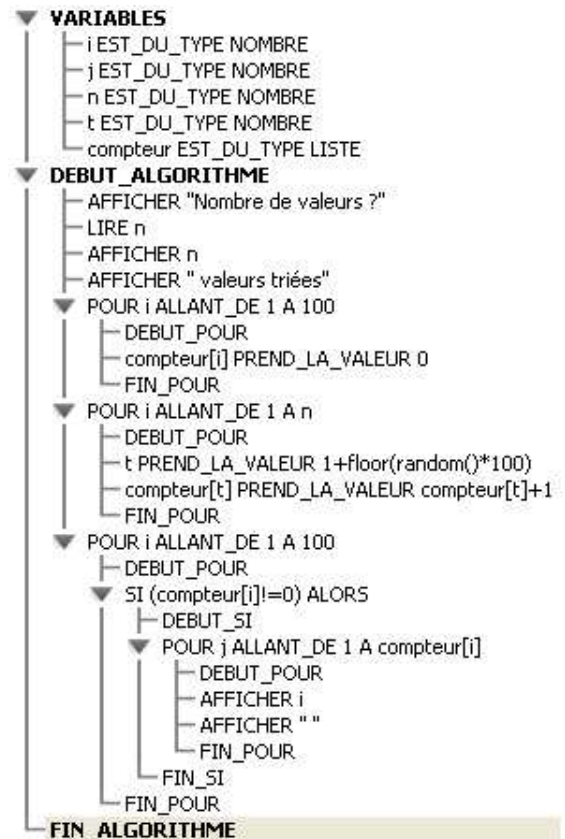


16. Tri par dénombrement

On compte le nombre de tirages pour chacune des valeurs et on restitue ces valeurs dans l'ordre.

```

DEBUT
ECRIRE "Nombre de valeurs ?"
LIRE n
POUR i DE 1 à 100 ITERER
    compteur(i):=0
FIN D'ITERER
POUR i DE 1 à n ITERER
    t:=ALEA(1;100)
    compteur(t):=compteur(t)+1
FIN D'ITERER
POUR i DE 1 à 100 ITERER
    SI compteur(i)<>0 ALORS
        POUR j DE 1 A compteur(i) ITERER
            ECRIRE i
        FIN D'ITERER
    FIN SI
FIN D'ITERER
FIN
    
```



19. Fractions

Disposez les volumes pour qu'on obtienne neuf fractions différentes.

Quand j'étais petit garçon, on me donna neuf volumes de l'Histoire d'Angleterre par Hume, en me promettant en abondance des fusils, des poneys, et des tas d'autres choses, à condition que j'étudie ces volumes. Je dois admettre que ce que j'ignore de l'Histoire d'Angleterre ferait plus que doubler la taille d'une bibliothèque ordinaire, mais, par contre, j'ai découvert d'intéressants casse-tête basés sur ces volumes de poids. J'ai découvert, par exemple, qu'en plaçant les volumes sur deux étagères comme le montre le dessin, la fraction $6\ 729/13\ 458$ est exactement égale à $1/2$. Est-il possible, en utilisant les neuf volumes de trouver d'autres combinaisons qui forment des fractions égales à $1/3$, $1/4$, $1/5$, $1/6$, $1/7$, $1/8$ et $1/9$?



Les casse-tête mathématiques de Sam Loyd. Martin GARDNER. Dunod. (cité par P. Terracher)

1/2		1/3		1/4		1/5		1/6		1/7		1/8		1/9	
nb. sol.: 12		nb. sol.: 2		nb. sol.: 4		nb. sol.: 12		nb. sol.: 3		nb. sol.: 7		nb. sol.: 46		nb. sol.: 3	
N	D	N	D	N	D	N	D	N	D	N	D	N	D	N	D
6 729	13 458	5 823	17 469	3 942	15 768	2 697	13 485	2 943	17 658	2 394	16 758	3 187	25 496	6 381	57 429
6 792	13 584	5 832	17 496	4 392	17 568	2 769	13 845	4 653	27 918	2 637	18 459	4 589	36 712	6 471	58 239
6 927	13 854			5 796	23 184	2 937	14 685	5 697	34 182	4 527	31 689	4 591	36 728	8 361	75 249
7 269	14 538			7 956	31 824	2 967	14 835			5 274	36 918	4 689	37 512		
7 293	14 586					2 973	14 865			5 418	37 926	4 691	37 528		
7 329	14 658					3 297	16 485			5 976	41 832	4 769	38 152		
7 692	15 384					3 729	18 645			7 614	53 298	5 237	41 896		
7 923	15 846					6 297	31 485					5 371	42 968		
7 932	15 864					7 629	38 145					5 789	46 312		
9 267	18 534					9 237	46 185					5 791	46 328		
9 273	18 546					9 627	48 135					5 839	46 712		
9 327	18 654					9 723	48 615					5 892	47 136		
												5 916	47 328		
												5 921	47 368		
												6 479	51 832		
												6 741	53 928		
												6 789	54 312		
												6 791	54 328		
												6 839	54 712		
												7 123	56 984		
												7 312	58 496		
												7 364	58 912		
												7 416	59 328		
												7 421	59 368		
												7 894	63 152		
												7 941	63 528		
												8 174	65 392		
												8 179	65 432		
												8 394	67 152		
												8 419	67 352		
												8 439	67 512		
												8 932	71 456		
												8 942	71 536		
												8 953	71 624		
												8 954	71 632		
												9 156	73 248		
												9 158	73 264		
												9 182	73 456		
												9 316	74 528		
												9 321	74 568		
												9 352	74 816		
												9 416	75 328		
												9 421	75 368		
												9 523	76 184		
												9 531	76 248		
												9 541	76 328		

Le tableau ci-dessus donne les solutions de manière exhaustive.

Il existe bien des manières de chercher les solutions à l'aide d'un algorithme mais une manière originale consiste à essayer des permutations de façon aléatoires avec un test d'arrêt.

Ce qui revient à écrire n'importe quoi et compter sur la chance ... et ça marche (le plus souvent) !



RÉSULTAT :

```

***Algorithme lancé***
Pour la fraction 1/n valeur de n (entre 1 et 9) ?
5832/17496 = 1/3

***Algorithme terminé***

```

V - Exemples d'algorithmes utilisant les instructions de chaînes de caractères.

20. Coder et décoder en "Jules César"

Il s'agit du codage le plus simple, celui par décalage des caractères.



Le programme demande si l'on veut coder ou décoder.
 Si l'on veut coder il demande la valeur du décalage choisi et il code.
 Si l'on veut décoder il demande si l'on connaît la valeur du décalage.
 Dans l'affirmative il decode le texte sinon il donne les 25 versions possibles et il n'y a plus qu'à choisir. En général une seule a un sens.
 On ne doit utiliser que les caractères minuscules non accentués et l'espace.

Exemple de codage d'un texte en code 15:

RÉSULTAT :

```
***Algorithme lancé***
Voulez-vous décoder(1) ou coder (2) ?
Ecrire le texte :
Entrez le numéro du code :
Texte : il faut rendre a cesar ce qui est a cesar
Texte traité : xa upji gtcsgt p rthpg rt fjx thi p rthpg
***Algorithme terminé***
```

Exemple de décodage d'un texte dont on ignore le décalage :

```
***Algorithme lancé***
Voulez-vous décoder(1) ou coder (2) ?
Voulez-vous décoder(1) ou coder (2) ?
Connaissez-vous le code (o/n) ?
Ecrire le texte :
Code 0 : kl c rj kifg vtirjv tvjri tv gfik jrclk
Code 1 : lm d sk ljgh wujskw uwksj uw hgjl ksclml
Code 2 : mn e tl mkhi xvktlx vxltk vx ihkm ltemm
Code 3 : no f um nlij ywlumy wymul wy jiln mufon
Code 4 : op g vn omjk zxmvmz xznmv xz kjmo nvgpo
Code 5 : pq h wo pnkl aynwoa yaown ya lknp owhqp
Code 6 : qr i xp qolm bzoxpb zbpzo zb mloq pxirq
Code 7 : rs j yq rpmn capyqc acqyp ac nmpr qyjsr
Code 8 : st k zr sqno dbqzrd bdrzq bd onqs rzkts
Code 9 : tu l as trop ecrase cesar ce port salut
Code 10 : uv m bt uspq fdsbtf dftbs df qpsu tbmvu
Code 11 : vw n cu vtqr getcug eguct eg rqtv ucnwv
Code 12 : wx o dv wurs hfudvh fhvdu fh sruw vdoxw
Code 13 : xy p ew xvst igvewi giwev gi tsvx wepyx
Code 14 : yz q fx ywtu jhwfxj hjxvw hj utwy xfqzy
Code 15 : za r gy zxuv kixgyk ikygx ik vuxz ygraz
Code 16 : ab s hz ayvw ljyhzl jlzhy jl wvya zhsba
Code 17 : bc t ia bzwx mkziam kmaiz km xwzb aitcb
Code 18 : cd u jb caxy nlajbn lnbja ln yxac bjudc
Code 19 : de v kc dbyz ombkco mockb mo zybd ckved
Code 20 : ef w ld ecza pncldp npdlc np azce dlwfe
Code 21 : fg x me fdab qodmeq oqemd oq badf emxgf
Code 22 : gh y nf gebc rpenfr prfne pr cbeg fnyhg
Code 23 : hi z og hfcd sqfogs qsgof qs dcfh gozih
Code 24 : ij a ph igde trgpht rthpg rt edgi hpaji
Code 25 : jk b qi jhef ushqui suiqh su fehj iqbkj

***Algorithme terminé***
```

Le texte original est en code 0 est la seule traduction qui est un sens est celle en code 9. On obtient une fameuse phrase dont la propriété (c'est un palindrome qui se conserve donc par décalage) mériterait à elle seule un algorithme.

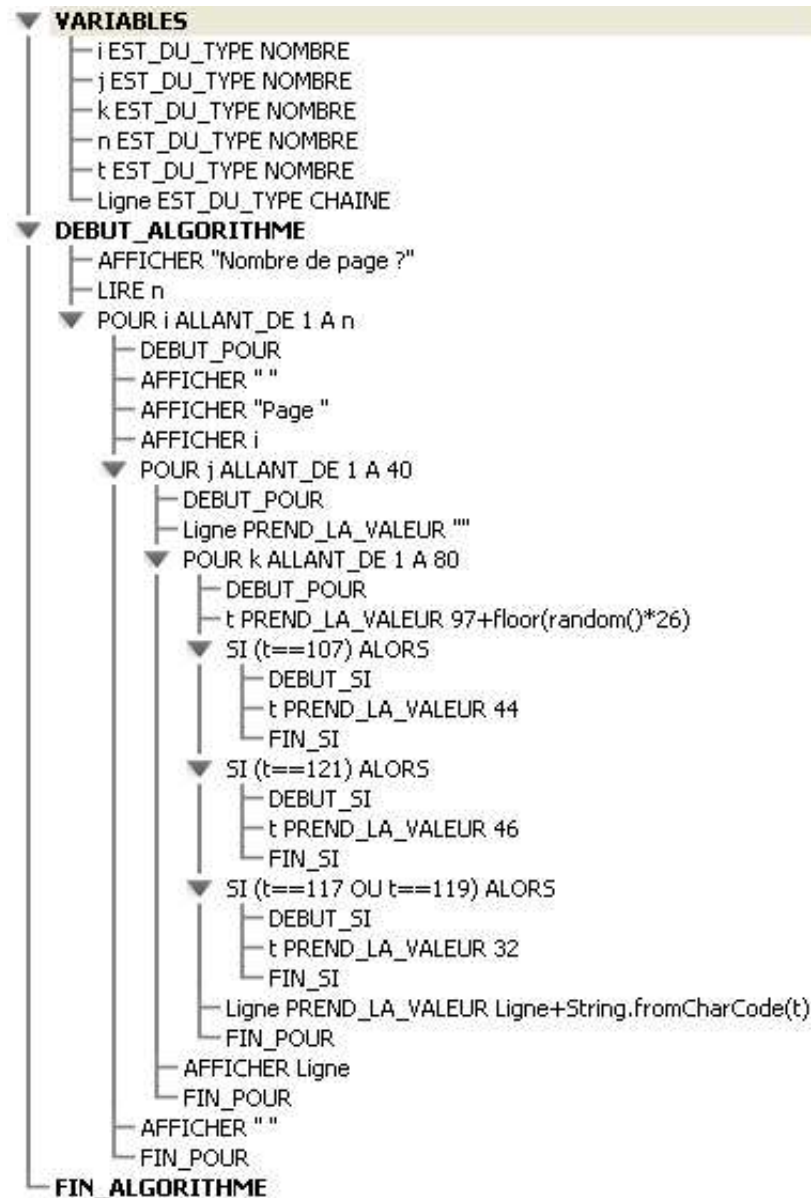
21. La bibliothèque de Babel

Illustration de la nouvelle de Jorge Luis Borgès (voir page suivante).

"...chaque livre a quatre cent dix pages ; chaque page, quarante lignes, et chaque ligne, environ quatre-vingts caractères noirs."

"Le manuscrit original du présent manuscrit ne contient ni chiffres ni majuscules. La ponctuation a été limitée à la virgule et au point. Ces deux signes, l'espace et les vingt-deux lettres de l'alphabet sont vingt-cinq symboles suffisants énumérés par l'inconnu. (Note de l'éditeur)"

Pour respecter le texte de Borgès à la lettre, on a remplacé le *k* (code ascii 107) par la virgule (44), le *y* (121) par le point (46), le *u* (117) et le *w* (119) par l'espace (32). L'algorithme ci-dessous génère des pages aléatoires d'un volume aléatoire.



Remarque : un volume de la bibliothèque de Babel comporte $80 \times 40 \times 410 = 1\,312\,000$ caractères.

Il y a 25 caractères possibles donc $25^{1\,312\,000} \approx 2 \times 10^{1\,834\,097}$ volumes différents. En comptant 1 dm^3 par livre, les livres seuls occuperaient $2 \times 10^{1\,834\,085} \text{ km}^3$. En considérant l'univers comme une boule de 15 milliards d'année-lumières de rayon on obtient le nombre dérisoire de $1,2 \times 10^{70} \text{ km}^3$.

Bien qu'énorme la bibliothèque est finie. Le narrateur semble penser le contraire. Borgès est trop fin mathématicien pour faire une telle erreur. Alors par quel prodige ? Il faut lire le dernier paragraphe.

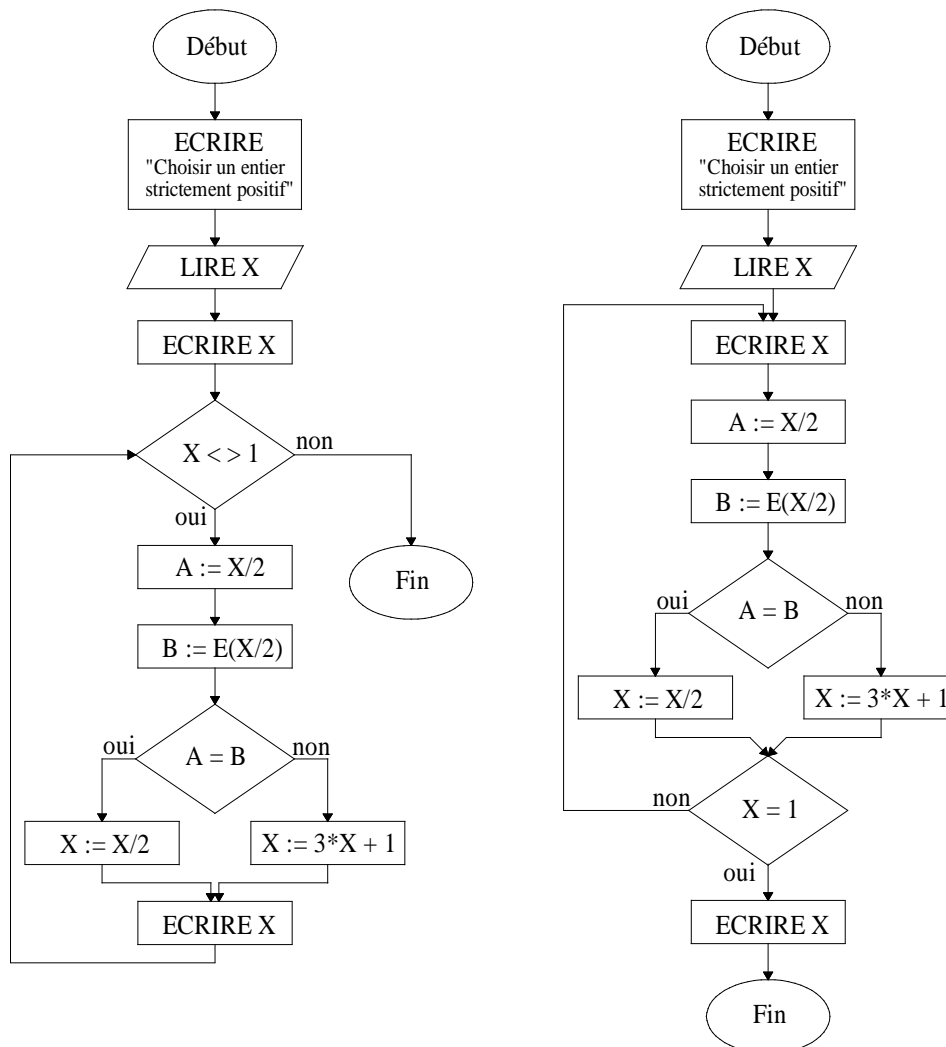
VI – Lire un algorithme

22. Conjecture de Syracuse

Variations sur « TANT QUE ... FAIRE » et « RÉPÉTER ... JUSQU'À ».

```
DÉBUT
ÉCRIRE "Choisir un entier strictement positif"
LIRE X
ÉCRIRE X
TANT QUE X <> 1
FAIRE
  A := X/2
  B := E(X/2)
  SI A = B
    ALORS X := X/2
    SINON X := 3*X + 1
  FIN SI
  ÉCRIRE X
FIN TANT QUE
FIN
```

```
DÉBUT
ÉCRIRE "Choisir un entier strictement positif"
LIRE X
RÉPÉTER
  ÉCRIRE X
  A := X/2
  B := E(X/2)
  SI A = B
    ALORS X := X/2
    SINON X := 3*X + 1
  FIN SI
FIN DE RÉPÉTER
JUSQU'À X = 1
ÉCRIRE X
FIN
```



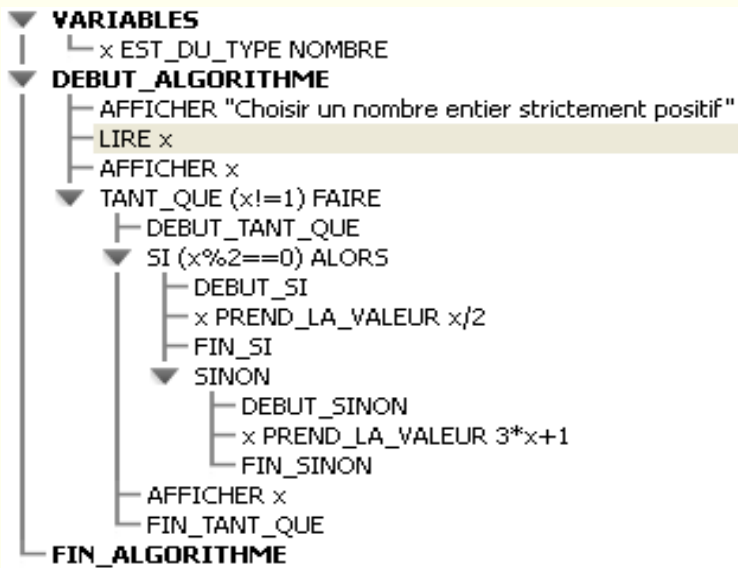
Cet algorithme semble se terminer après un nombre fini d'opérations. C'est la conjecture de Syracuse ou conjecture de Collatz ou conjecture d'Ulam ou conjecture tchèque.

En dépit de la simplicité de son énoncé, cette conjecture n'a pas encore été démontrée.

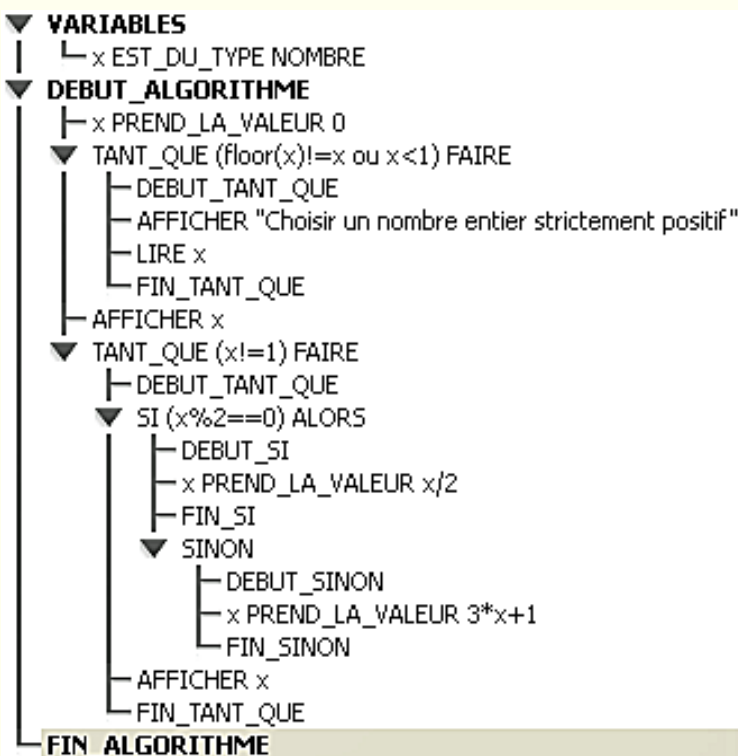
Paul Erdős, mathématicien hongrois, dit à son propos que « les mathématiques ne sont pas encore prêtes pour de tels problèmes » !

Avec AlgoBox :

Plus court en utilisant le reste de la division euclidienne ($x\%y$ donne le reste de la division euclidienne de x par y).



Plus sûr en testant l'entrée.

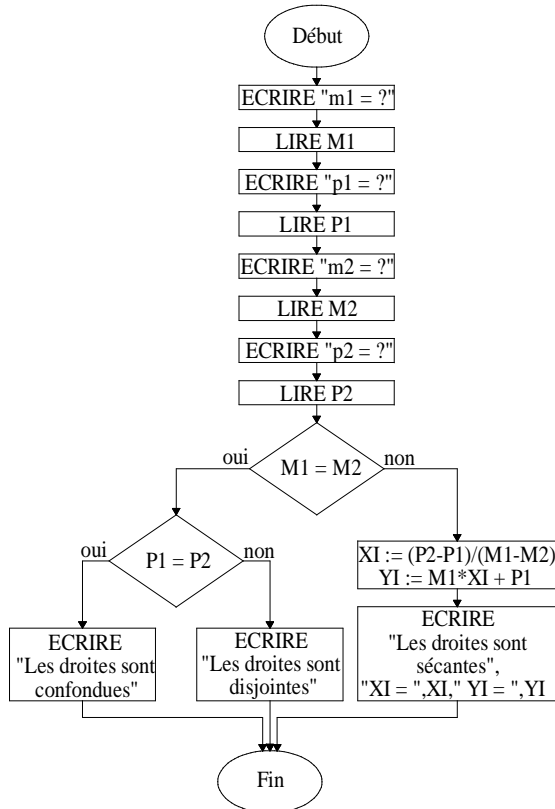


Sur ce type d'algorithme (Syracuse, Prabhakar, ...) l'emploi d'un tableur est particulièrement intéressant. Ci-dessous un exemple à l'aide d'Excel 2007.

- Conjecture de Syracuse (Essai).xlsx (peut être utilisé avec Excel 2003) ;
- Conjecture de Syracuse (Calculs).xlsx (seulement avec Excel 2007) ;
- Conjecture de Syracuse (Table).xlsx (seulement avec Excel 2007).

VII – Exemples de programmes dans différents langages

23. Intersection de deux droites



```

ECRIRE "m1 = ?"
LIRE M1
ECRIRE "p1 = ?"
LIRE P1
ECRIRE "m2 = ?"
LIRE M2
ECRIRE "p2 = ?"
LIRE P2
SI M1 = M2
  ALORS SI P1 = P2
    ALORS ECRIRE "Les droites sont confondues"
  SINON ECRIRE "Les droites sont disjointes"
FIN SI
SINON  $XI := (P2 - P1)/(M1 - M2)$ 
   $YI := M1 * XI + P1$ 
  ECRIRE "Les droites sont sécantes", "XI = ", XI, " YI = ", YI
FIN SI
FIN
  
```

Le fichier Excel "Organigramme – Intersection de deux droites.xls" montre le fonctionnement de cet algorithme à l'aide d'une animation.

Programme pour Casio Graph 35

Syntaxe : **If condition : Then instruction1 : ... Else instruction2 : ... If End : instruction3 : ...**

Programme en deux versions. La seconde plus sophistiquée dans sa présentation.

=====INTERDTE=====

```

"M1 = "?→A ↵
"P1 = "?→B ↵
"M2 = "?→C ↵
"P2 = "?→D ↵
If A=C ↵
Then If B=D ↵
Then "LES DROITES SON
T  CONFONDUES" ↵
Else "LES DROITES SON
T  DISJOINTES" ↵
IfEnd ↵
Else "LES DROITES SON
T  SECANTES" ↵
(D-B)/(A-C)→X ↵
A×X+B→Y ↵
"XI = ":X ↵
"YI = ":Y
  
```

=====INTERDTE=====

```

ClrText ↵
"M1 = "?→A ↵
"P1 = "?→B ↵
"M2 = "?→C ↵
"P2 = "?→D ↵
ClrText ↵
Locate 1,1,"M1 = " ↵
Locate 6,1,A ↵
Locate 1,2,"P1 = " ↵
Locate 6,2,B ↵
Locate 1,3,"M2 = " ↵
Locate 6,3,C ↵
Locate 1,4,"P2 = " ↵
Locate 6,4,D ↵
If A=C ↵
Then If B=D ↵
Then Locate 1,5,"DROITES CONFONDUES" ↵
Else Locate 1,5,"DROITES DISJOINTES" ↵
IfEnd ↵
Else Locate 1,5,"DROITES SECANTES" ↵
(D-B)/(A-C)→X ↵
A×X+B→Y ↵
Locate 1,6,"XI = ":Locate 6,6,X ↵
Locate 1,7,"YI = ":Locate 6,7,Y
  
```

Programme pour TI-82 Stats.fr

```

PROGRAM:INTERDTE      :Disp "LES DROIT      :Disp "LES DROIT
:EffEcr              ES SONT"          ES SONT"
:Input "M1=",A        :Disp "CONFONDUE      :Disp "SECANTES
:Input "P1=",B        S"                    EN I"
:Input "M2=",C        :Else                    : (D-B) / (A-C)→X
:Input "P1=",D        :Disp "LES DROIT      :A*X+B→Y
:EffEcr              ES SONT"          :Disp "XI="
:If A=C               :Disp "DISJOINTE     :Output(3,4,X)
:Then                 S"              :Disp "YI="
:If B=D               :End                    :Output(4,4,Y)
:Then                 :Else                    :End

```

Programme pour AlgoBox



Avec Excel sans VBA

	B	C	D	E	F	G	H	I
2	Intersection de deux droites							
3	$m_1 = -4$			Les deux droites sont sécantes au point I de coordonnées :				
4	$p_1 = 2$			$x = -1$		$y = 6$		
5	$m_2 = 5$							
6	$p_2 = 11$							

Contenus des cellules

E3 : =SI(C3=C5;"";SI(C4=C6;"Les deux droites sont confondues";"Les deux droites sont disjointes");"Les deux droites sont sécantes au point I de coordonnées :")

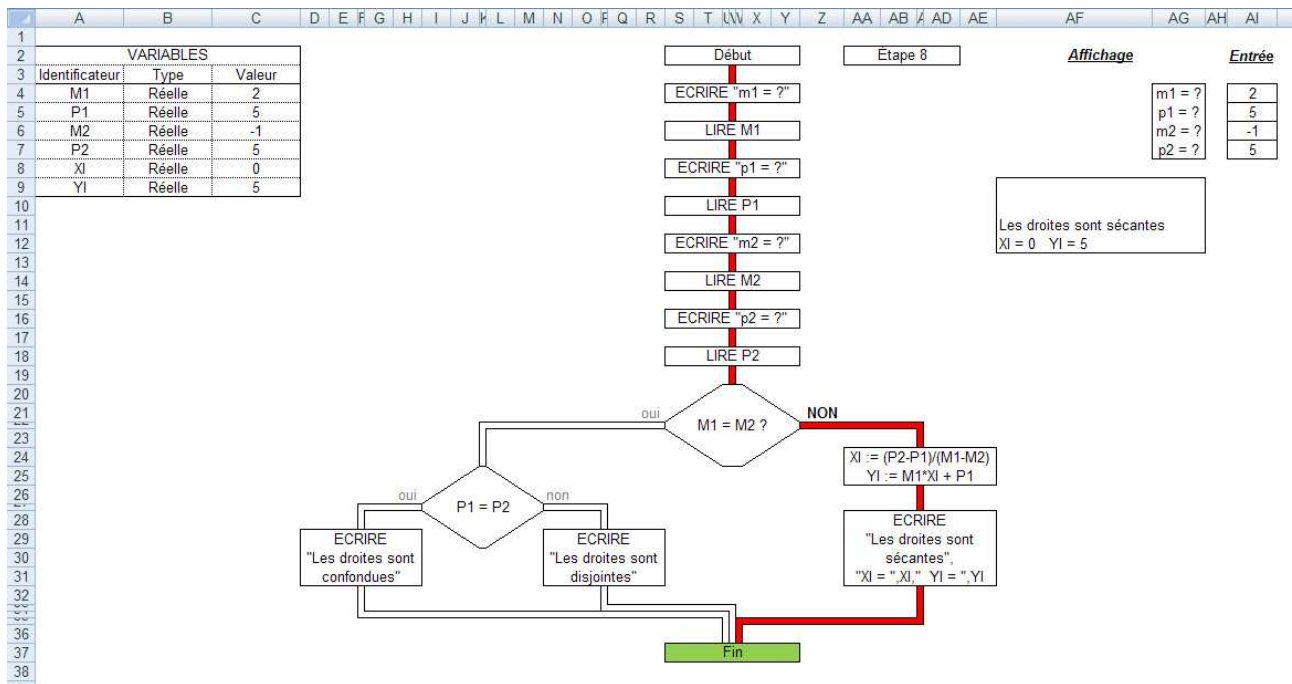
E4 : =SI(C3=C5;"";"x = ")

F4 : =SI(C3=C5;"";(C6-C4)/(C3-C5))

G4 : =SI(C3=C5;"";"y = ")

H4 : =SI(C3=C5;"";C3*F4+C4)

Le fichier Excel "Organigramme – Intersection de deux droites.xls" montre le fonctionnement de cet algorithme à l'aide d'une animation.



VIII – En guise de premier algorithme avec AlgoBox

24. Image d'un réel par une fonction, tableau de valeurs, tracé

Calculer l'image d'un nombre réel par une fonction, faire un tableau de valeurs, tracer la courbe représentative.

Fonction1

Algorithme élémentaire pour le calcul de l'image d'un réel par la fonction définie sur \mathbb{R} par :

$$f(x) = x^2 - 2x + 1.$$



Fonction 2

Même algorithme que *Fonction 1* mais en utilisant le module fonction d'AlgoBox.

The screenshot shows the AlgoBox interface with three tabs: "Opérations standards", "Utiliser une fonction numérique", and "Dessiner dans un repère". The "Utiliser une fonction numérique" tab is active, and the checkbox "Utiliser une fonction" is checked. Below this, the "Définir la fonction" section contains a text box with the function definition: $F1(x) = \text{pow}(x, 2) - 2 * x + 1$. Below the text box is a flowchart with the following steps:

- VARIABLES**
 - x EST_DU_TYPE NOMBRE
 - y EST_DU_TYPE NOMBRE
- DEBUT_ALGORITHME**
 - AFFICHER "Valeur de x ?"
 - LIRE x
 - y PREND_LA_VALEUR F1(x)
 - AFFICHER "f(x) = "
 - AFFICHER y
- FIN_ALGORITHME**

Fonction 3

Même algorithme que *Fonction 2* mais avec un affichage plus sophistiqué (Utilisation de la variable "texte" de type "CHAINE").

The flowchart for 'Fonction 3' includes the following steps:

- VARIABLES**
 - x EST_DU_TYPE NOMBRE
 - y EST_DU_TYPE NOMBRE
 - texte EST_DU_TYPE CHAINE
- DEBUT_ALGORITHME**
 - AFFICHER "Valeur de x ?"
 - LIRE x
 - y PREND_LA_VALEUR F1(x)
 - texte PREND_LA_VALEUR "f("+x+") = "+y
 - AFFICHER texte
- FIN_ALGORITHME**

Fonction 4

Détermine une série d'images. Utilise la boucle POUR...DE...A.

The flowchart for 'Fonction 4' includes the following steps:

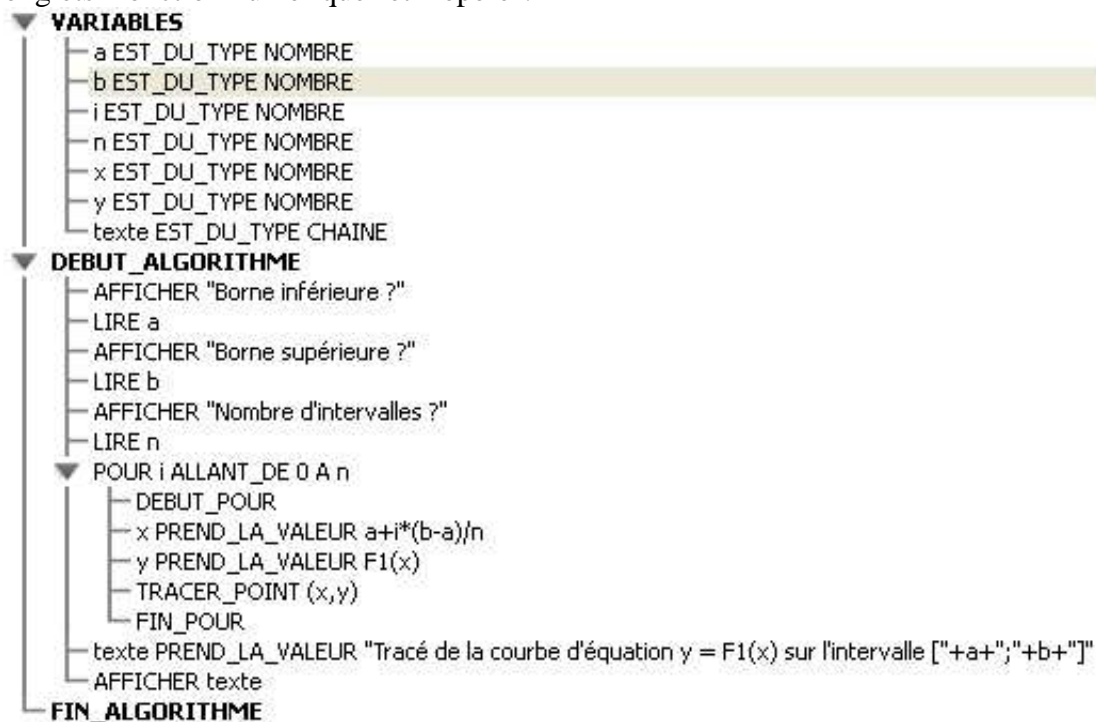
- VARIABLES**
 - x EST_DU_TYPE NOMBRE
 - y EST_DU_TYPE NOMBRE
 - i EST_DU_TYPE NOMBRE
 - n EST_DU_TYPE NOMBRE
 - p EST_DU_TYPE NOMBRE
 - texte EST_DU_TYPE CHAINE
- DEBUT_ALGORITHME**
 - AFFICHER "Première valeur ?"
 - LIRE x
 - AFFICHER "Nombre de valeur ?"
 - LIRE n
 - AFFICHER "Pas ?"
 - LIRE p
 - POUR i ALLANT_DE 1 A n**
 - DEBUT_POUR
 - y PREND_LA_VALEUR F1(x)
 - texte PREND_LA_VALEUR "x = "+x+" y = "+y
 - AFFICHER texte
 - x PREND_LA_VALEUR x+p
 - FIN_POUR
- FIN_ALGORITHME**

RÉSULTAT :

```
x = -3   y = 16
x = -2   y = 9
x = -1   y = 4
x = 0    y = 1
x = 1    y = 0
x = 2    y = 1
x = 3    y = 4
x = 4    y = 9
```

Fonction 5

Dessine la représentation graphique d'une fonction à l'aide d'une série de points. Utilise les onglets "fonction numérique" et "repère".



Opérations standards Utiliser une fonction numérique Dessiner dans un repère

Utiliser une fonction

Définir la fonction

F1(x)=

Opérations standards Utiliser une fonction numérique Dessiner dans un repère

Utiliser un repère

Définir le repère

Xmin : Xmax :

Ymin : Ymax :

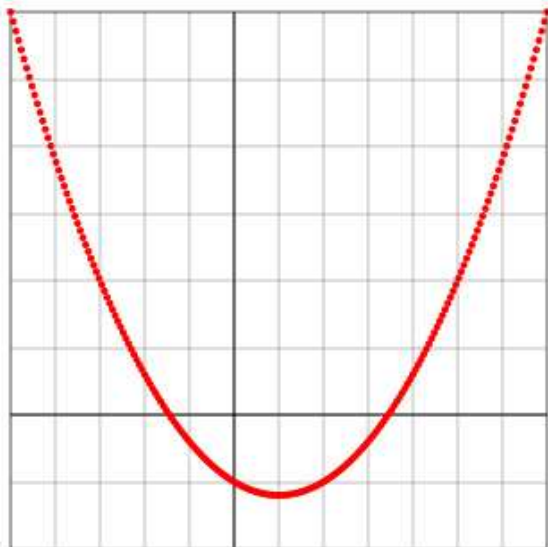
Graduations X : Graduations Y :

Ajouter code

Tracé à l'aide de 200 points (on peut en définir 200 000 !).

RÉSULTAT :

```
***Algorithme lancé***  
Borne inférieure ?  
Borne supérieure ?  
Nombre d'intervalles ?  
Tracé de la courbe d'équation  $y = F1(x)$  sur l'intervalle  $[-5;7]$   
  
***Algorithme terminé***
```



Xmin: -5 ; Xmax: 7 ; Ymin: -10 ; Ymax: 30 ; GradX: 1 ; GradY: 5

Annexes

Fichier Word 1997-2003 + MathType 6 :

Exemples d'algorithmes pour la Seconde.doc (*ce fichier*)

Fichier Excel 1997-2003

Aiguilles de Buffon.xls
Algorithme de Kaprekat.xls
Algorithme de Prabhakar.xls
Ecriture décimale périodique illimitée.xls
Organigramme – Approximation par dichotomie.xls
Organigramme – Intersection de deux droites.xls

Fichier Excel 2007

Algorithme de tri.xlsx
Conjecture de Syracuse (Essai).xlsx
Conjecture de Syracuse (Calculs).xlsx
Conjecture de Syracuse (Table).xlsx

Fichier AlgoBox 0.4 (logiciel libre) (<http://www.xmlmath.net/algobox/>)

01-Alea1.alg	11-MinMax1.alg
01-Alea2.alg	12-Monte-Carlo1.alg
01-Alea3.alg	15-Triparinsertion1.alg
01-Alea4.alg	14-Triparselection1.alg
02-Combin1.alg	16-Tripardénombrément1.alg
03-Loto1.alg	17-Babylone1.alg
04-Permutation1.alg	19-MartinGardner1.alg
05-Dés1.alg	20-JulesCesar1.alg
05-Dés2.alg	21-BibliothèquedeBabel1.alg
06-Division1.alg	22-Syracuse1.alg
06-Division2.alg	22-Syracuse2.alg
07-Dichotomie1.alg	23-InterDroites1.alg
07-Dichotomie2.alg	24-Fonction1.alg
08-Deuxpoints1.alg	24-Fonction2.alg
08-Deuxpoints2.alg	24-Fonction3.alg
09-Troispoints1.alg	24-Fonction4.alg
09-Troispoints2.alg	24-Fonction5.alg

Fichier AutoSketch 2.1

Organigr.skd

