

## 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 :**

$<, >, \leq, \geq, =, \neq$ .

**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).

# ALGOBOX : initiation à l'utilisation du logiciel et exemples d'algorithmes .

*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 ← 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 ← 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 ← 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
```

## 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 :*

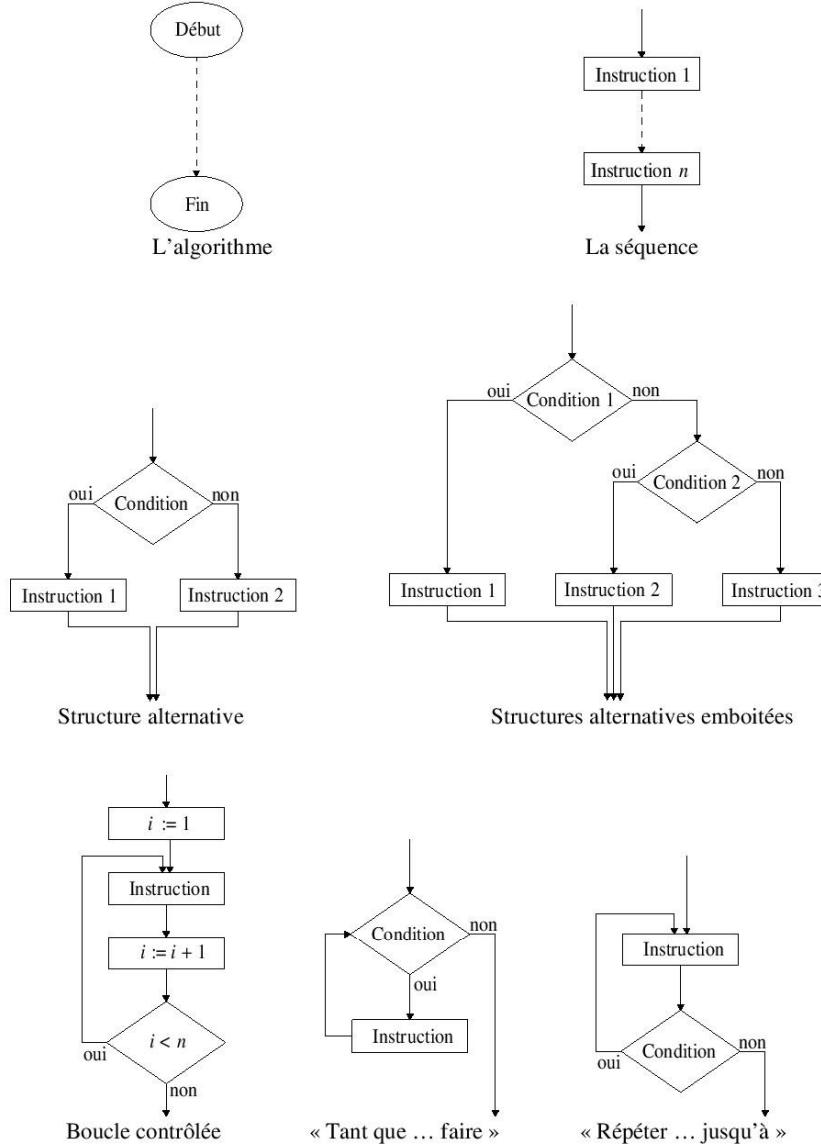
Objet : Soit la suite  $(u_n)$  définie par  $u_n = 2u_{n-1} - 4$  et  $u_0 = 5$ .

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 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
```

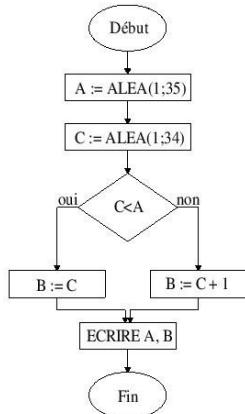
Enfin on peut aussi décider du nombre de tirages. Pour cela il suffit de modifier les trois dernières lignes de l'algorithme précédent par la séquence suivante :

```

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.



```

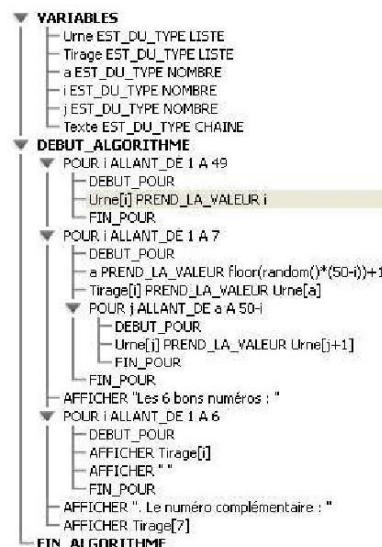
DEBUT
A := 1 + E(35*RANDOM())
B := 1 + E(34*RANDOM())
SI C < A
    ALORS B := C
    SINON B := C + 1
FIN SI
ECRIRE A, " ; ", B
FIN
    
```

## 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 50-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 :

```

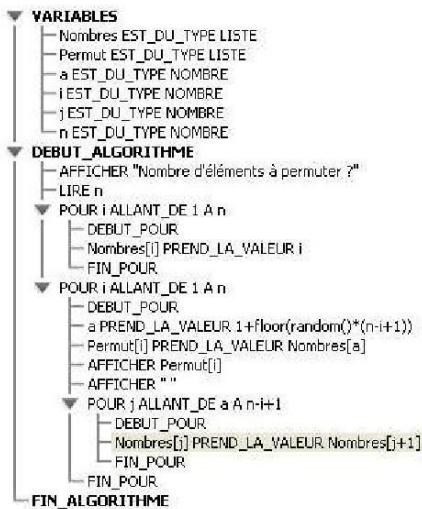
***Algorithme lancé***
Les 6 bons numéros : 29 28 40 27 11 48 . Le numéro complémentaire : 14
***Algorithme terminé***
    
```

#### 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+1
        nombres(j):=nombres(j+1)
    FIN D'ITERER
FIN D'ITERER
FIN
```



#### 5. Lancers de dés

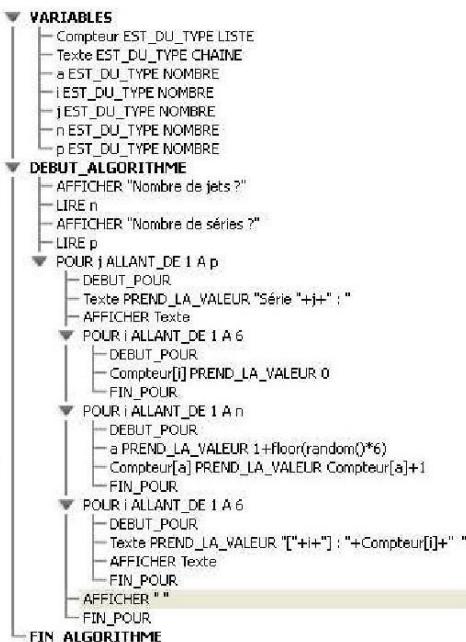
On utilise un dé à six faces. Ce programme demande le nombre de jets et totalise les résultats.

```

VARIABLES
- a EST_DU_TYPE NOMBRE
- i EST_DU_TYPE NOMBRE
- n EST_DU_TYPE NOMBRE
- Compteur EST_DU_TYPE LISTE
- Texte EST_DU_TYPE CHAINE

DEBUT_ALGORITHME
- AFFICHER "Nombre de jets ?"
- LIRE n
- POUR i ALLANT_DE 1 A n
    - DEBUT_POUR
    - Compteur[i] PREND_LA_VALEUR 0
    - FIN_POUR
- POUR i ALLANT_DE 1 A n
    - DEBUT_POUR
    - a PREND_LA_VALEUR 1+floor(random()*6)
    - Compteur[a] PREND_LA_VALEUR Compteur[a]+1
    - FIN_POUR
- POUR i ALLANT_DE 1 A 6
    - DEBUT_POUR
    - Texte PREND_LA_VALEUR "Nombre de "+i+": "+Compteur[i]
    - AFFICHER Texte
    - FIN_POUR
FIN_ALGORITHME
```

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

```

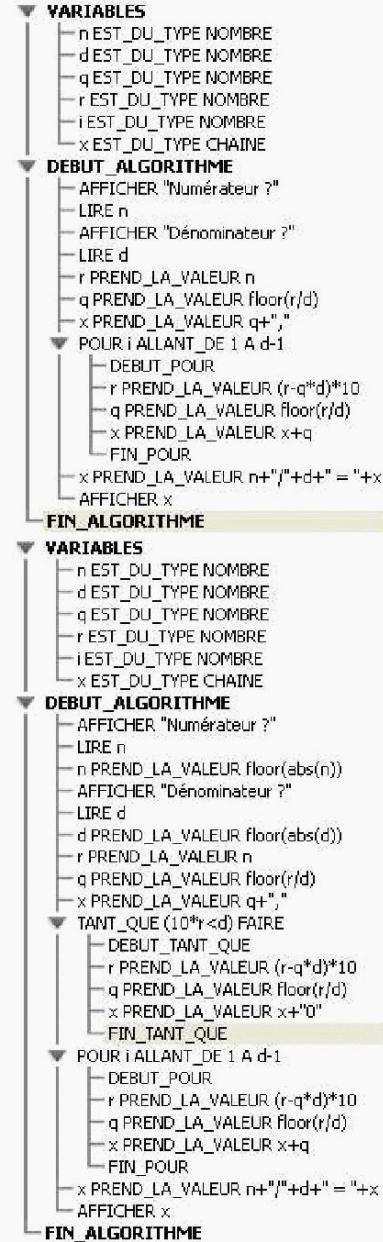
Cet algorithme donne  $d - 1$  chiffres "significatifs où  $d$  est le dénominateur. Ce n'est pas forcément la période mais la période comprend au plus  $d - 1$  chiffres.

Exemples avec AlgoBox :

```

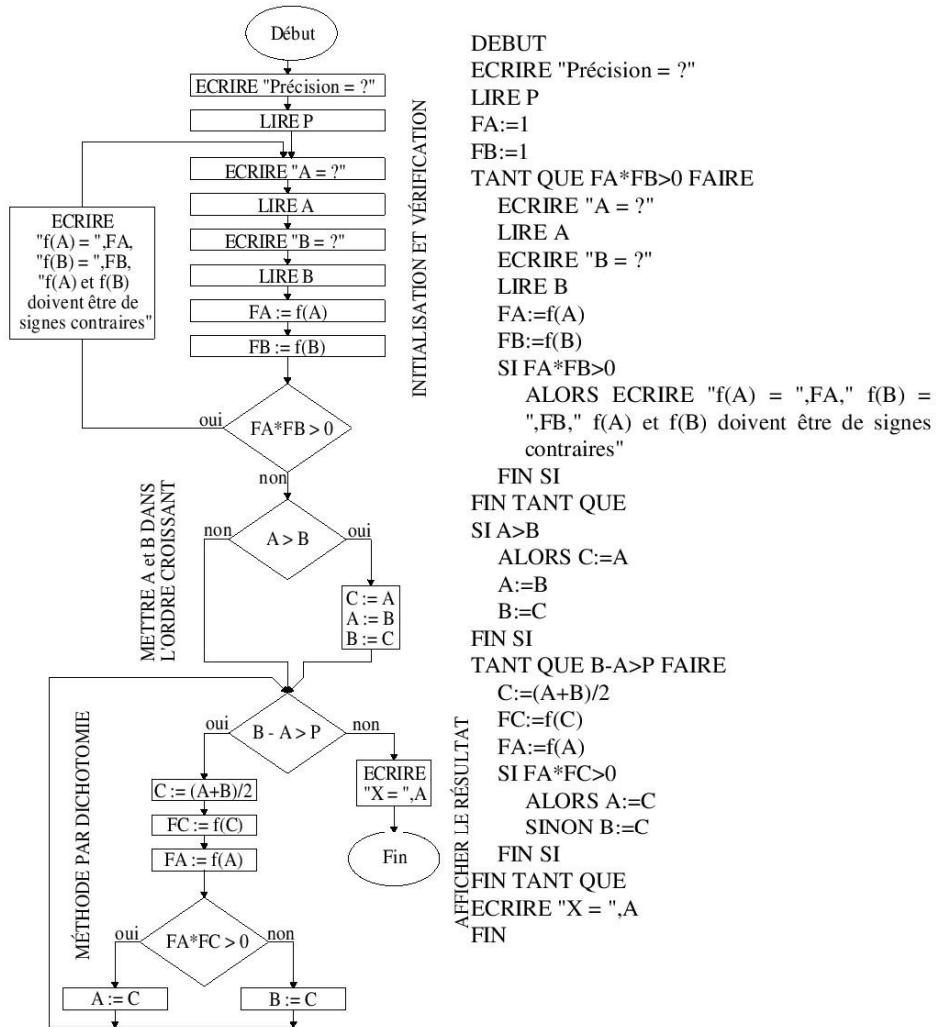
1/23 = 0,04347826086956521739130
5771/7 = 824,428571
5/43 = 0,116279069767441860465116279069767441860465
1/41 = 0,02439024390243902439024390243902439024390

```



**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 défini 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  $10^{-7}$ .

## ALGOBOX : initiation à l'utilisation du logiciel et exemples d'algorithmes .

**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 I
- FB PREND\_LA\_VALEUR I
- P PREND\_LA\_VALEUR 0.000001
- 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

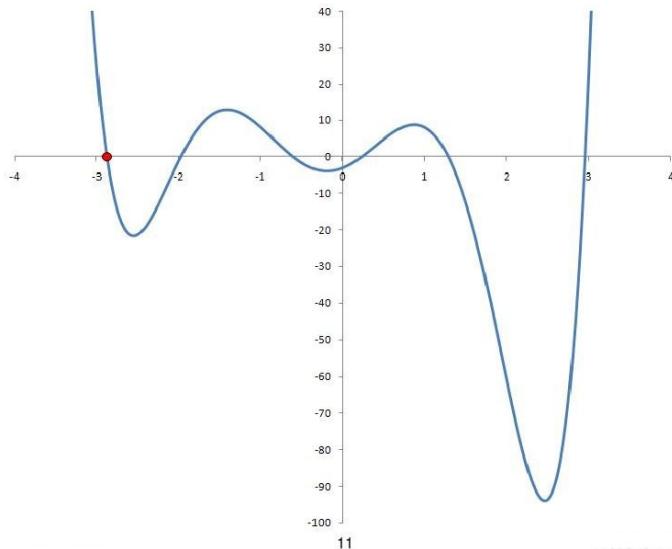
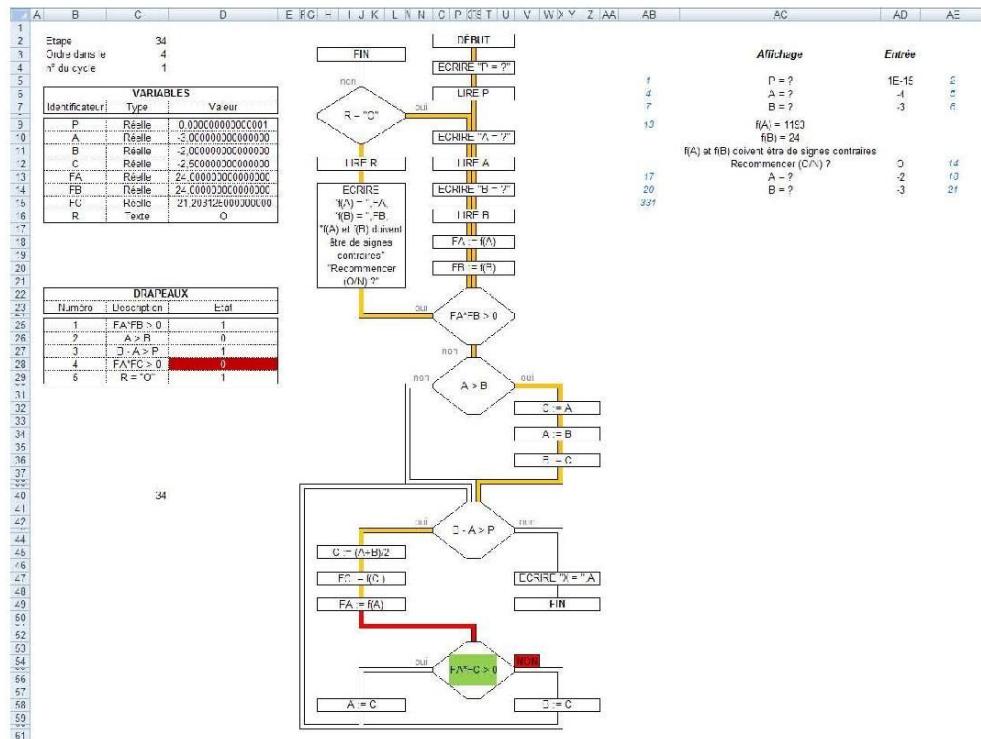
Utiliser une fonction

Définir la fonction

F1(x)=

## ALGOBOX : initiation à l'utilisation du logiciel et exemples d'algorithmes .

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



# ALGOBOX : initiation à l'utilisation du logiciel et exemples d'algorithmes .

## 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 P1END_LA_VALEUR,"A("+(X-XA)+";"+Y-YA+"); et B("+(XB-XA)+";"+YB-YA+");"
    -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 du segment [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_SEGMENT(XA,YA)>(>XB,YB)
    -TRACER_SEGMENT(-100,PI(-100))>(>120,PI(100))
FIN_ALGORITHME

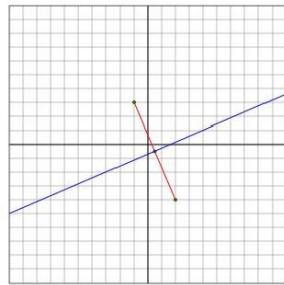
```

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 du segment [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

## ALGOBOX : initiation à l'utilisation du logiciel et exemples d'algorithmes .

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

```
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 du segment [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
    -TRACER_SEGMENT (-1000,YI)->(1000,YI)
    -FIN_SI
  -SINON
    -DEBUT_SINON
    -SI (YA==YB) ALORS
      -DEBUT_SI
      -Texte PREND_LA_VALEUR "Equation de la droite (AB) : y = " +YA
      -AFFICHER Texte
      -Texte PREND_LA_VALEUR "Equation de la médiatrice du segment [AB] : x = " +XI
      -AFFICHER Texte
      -TRACER_SEGMENT (XI,-1000)->(XI,1000)
      -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
      -TRACER_SEGMENT (-100,F1(-100))->(100,F1(100))
      -FIN_SINON
    -FIN_SINON
    -TRACER_POINT (XA,YA)
    -TRACER_POINT (XB,YB)
    -TRACER_POINT (XI,YI)
    -TRACER_SEGMENT (XA,YA)->(XB,YB)
  -FIN_ALGORITHME
```

# ALGOBOX : initiation à l'utilisation du logiciel et exemples d'algorithmes .

## 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 EST_DU_TYPE NOMBRE
P1 EST_DU_TYPE NOMBRE
M2 EST_DU_TYPE NOMBRE
P2 EST_DU_TYPE NOMBRE
M3 EST_DU_TYPE NOMBRE
P3 EST_DU_TYPE NOMBRE
Texte EST_DU_TYPE CHAINE
M4 EST_DU_TYPE NOMBRE
P4 EST_DU_TYPE NOMBRE
M5 EST_DU_TYPE NOMBRE
P5 EST_DU_TYPE NOMBRE
M6 EST_DU_TYPE NOMBRE
P6 EST_DU_TYPE NOMBRE
I EST_DU_TYPE NOMBRE
Nb_de_pts EST_DU_TYPE NOMBRE
A EST_DU_TYPE NOMBRE
B EST_DU_TYPE NOMBRE
C EST_DU_TYPE NOMBRE
D EST_DU_TYPE NOMBRE

DEBUT_ALGORITHME
AFFICHER "Abscisse du point A ?"
LIRE XA
AFFICHER "Ordonnée du point A"
LIRE YA
AFFICHER "Abscisse du point B ?"
LIRE XB
AFFICHER "Ordonnée du point B"
LIRE YB
AFFICHER "Abscisse du point C ?"
LIRE XC
AFFICHER "Ordonnée du point C"
LIRE YC

Texte PREND_LA_VALEUR "A(" + XA + "," + YA + "), B(" + XB + "," + YB + "), C(" + XC + "," + YC + ")"
AFFICHER Texte
AB PREND_LA_VALEUR sqrt(pow(XB-XA,2)+pow(YB-YA,2))
BC PREND_LA_VALEUR sqrt(pow(XC-XB,2)+pow(YC-YB,2))
CA PREND_LA_VALEUR sqrt(pow(XA-XC,2)+pow(YA-YC,2))
Texte PREND_LA_VALEUR "AB = " + AB + ", BC = " + BC + ", CA = " + CA
AFFICHER Texte
M1 PREND_LA_VALEUR (YB-YA)/(XB-XA)
P1 PREND_LA_VALEUR (XB-YA-XA*YB)/(XB-XA)
M4 PREND_LA_VALEUR (XA-XB)/(YA-YB)
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-XB-YB*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-XC-YA*YC)/(XA-XC)
M6 PREND_LA_VALEUR (YC-XA)/(YA-YC)
P6 PREND_LA_VALEUR (pow(XA,2)+pow(YA,2)-pow(XC,2)-pow(YC,2))/2/(YA-YC)
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)

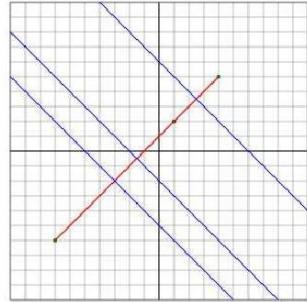
SI (abs(AB+BC-CA) < 0,000001 OU abs(BC+CA-AB) < 0,000001 OU abs(CA+AB-BC) < 0,000001) ALORS
    DEBUT_SI
        AFFICHER "A, B et C semblent alignés"
        Texte PREND_LA_VALEUR "Les droites (AB), (BC) et (CA) semblent confondues elles ont pour équation : "
        AFFICHER Texte
        Texte PREND_LA_VALEUR "La médiatrice du côté [AB] a pour équation : y = " + M1 + "x + " + P1
        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) a pour équation : y = " + M1 + "x + " + P1
            AFFICHER Texte
            Texte PREND_LA_VALEUR "La droite (BC) a pour équation : y = " + M2 + "x + " + P2
            AFFICHER Texte
            Texte PREND_LA_VALEUR "La droite (CA) a 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*X1+P4
            Texte PREND_LA_VALEUR "Les 3 médiatrices sont concourantes au point I de coordonnées (" + XI + "," + YI + ")"
            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 triangle ABC"
            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*pi/Math.PI/Nb_de_pts)
        B PREND_LA_VALEUR YI+R*sin(I*2*pi/Math.PI/Nb_de_pts)
        C PREND_LA_VALEUR XI+R*cos((I+1)*2*pi/Math.PI/Nb_de_pts)
        D PREND_LA_VALEUR YI+R*sin((I+1)*2*pi/Math.PI/Nb_de_pts)
        TRACER_SEGMENT(A,B) > (C,D)
    FIN_POUR
FIN_SINON
FIN_ALGORITHME

```

## ALGOBOX : initiation à l'utilisation du logiciel et exemples d'algorithmes .

RÉSULTAT :

```
A(-7;-6), B(1;2), C(4;5)
AB = 11.3127085, 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
```

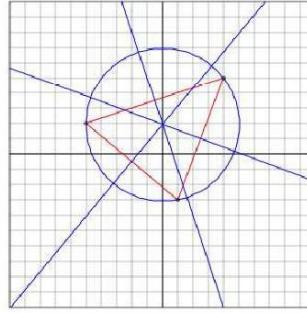


Xmin -10 ; Xmax 10 ; Ymin -10 ; Ymax 10 ; GradX 1 ; GradY 1

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) a pour équation : y = 0.33333333x + 3.66666667
La droite (BC) a pour équation : y = 2.66666667x + -5.66666667
La droite (CA) a 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
```

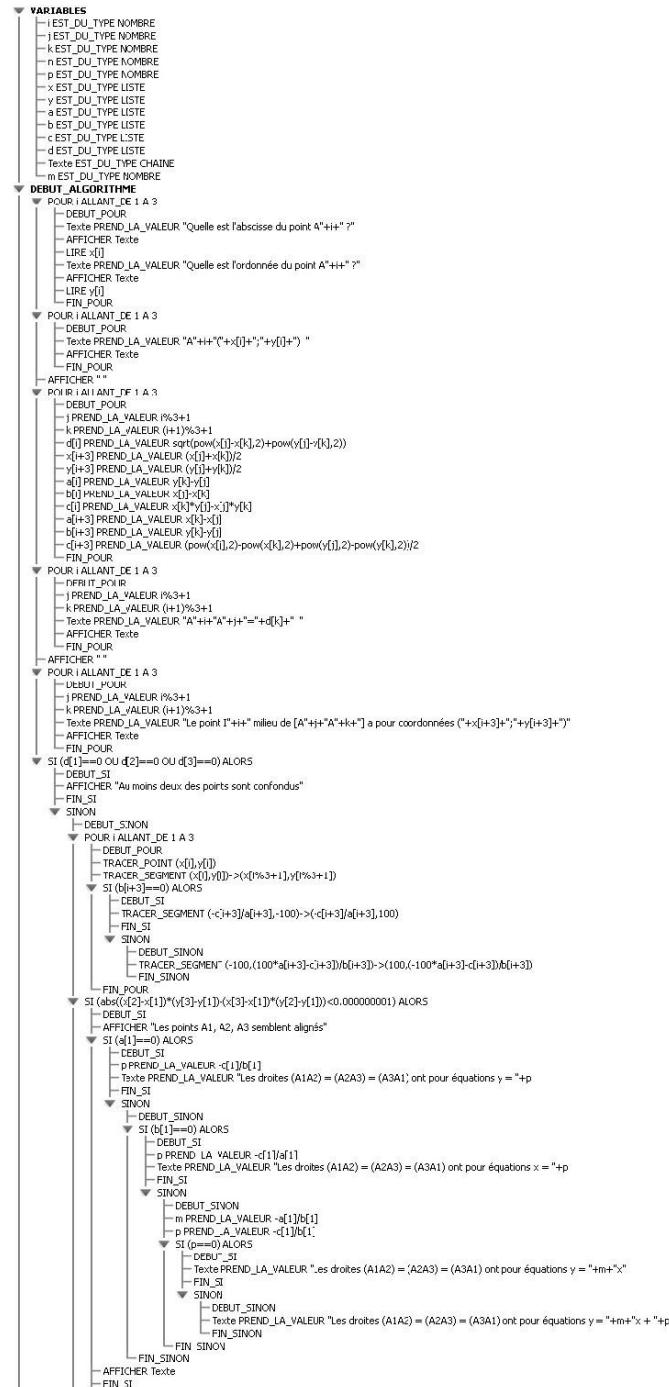
\*\*\*Algorithme terminé\*\*\*



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.

# ALGOBOX : initiation à l'utilisation du logiciel et exemples d'algorithmes .



# ALGOBOX : initiation à l'utilisation du logiciel et exemples d'algorithmes .

```

    ▼ SINON
        └─ DEBUT_SI
            └─ AFFICHER "Les points A1, A2, A3 ne sont pas alignés"
        ▼ POUR i ALLANT_DE 1 A 3
            └─ DEBUT_POUR
                └─ i PREND_LA_VALEUR i%3+1
                └─ k PREND_LA_VALEUR (i+1)%3+1
            ▼ SI (a[i]=b[i]) ALORS
                └─ DEBUT_SI
                    └─ p PREND_LA_VALEUR <0>[b[0]]
                    └─ Texte PREND_LA_VALEUR "La droite (A"+i+"A"+k+) a pour équation : y = "+p
                └─ FIN_SI
            ▼ SINON
                └─ DEBUT_SI
                    └─ SI (b[i]==0) ALORS
                        └─ DEBUT_SI
                            └─ p PREND_LA_VALEUR <i>[0]/b[0]
                            └─ Texte PREND_LA_VALEUR "à droite (A"+i+"A"+k+) a pour équation : x = "+p
                        └─ FIN_SI
                    ▼ SINON
                        └─ DEBUT_SI
                            └─ m PREND_LA_VALEUR <i>[0]/b[0]
                            └─ p PREND_LA_VALEUR <i>[0]/b[0]
                        ▼ SI (p==0) ALORS
                            └─ DEBUT_SI
                                └─ Texte PREND_LA_VALEUR "La droite (A"-i+"A"+k+) a pour équation : y = "+m+"x"
                            └─ FIN_SI
                        ▼ SINON
                            └─ DEBUT_SI
                                └─ Texte PREND_LA_VALEUR "La droite (A"+i+"A"+k+) a pour équation : y = "+m+"x + "+p
                            └─ FIN_SI
                        └─ FIN_SI
                    └─ FIN_SI
                └─ FIN_SI
            └─ FIN_SI
        └─ FIN_POUR
        └─ AFFICHER Texte
        └─ FIN_POUR
    ▼ POUR i ALLANT_DE 4 A 6
        └─ DEBUT_POUR
            └─ j PREND_LA_VALEUR (i+1)%3+1
            └─ k PREND_LA_VALEUR (i+1)%3+1
        ▼ SI (c[i]==0) ALORS
            └─ DEBUT_SI
                └─ p PREND_LA_VALEUR <i>[0]/b[i]
                └─ Texte PREND_LA_VALEUR "La médiatrice du segment [A"+i+"A"+k+] a pour équation : y = "+p
            └─ FIN_SI
        ▼ SINON
            └─ DEBUT_SI
                └─ SI (b[i]==0) ALORS
                    └─ DEBUT_SI
                        └─ p PREND_LA_VALEUR <i>[0]/b[0]
                        └─ Texte PREND_LA_VALEUR "La médiatrice du segment [A"-i+"A"+k+] a pour équation : x = "+p
                    └─ FIN_SI
                ▼ SINON
                    └─ DEBUT_SI
                        └─ m PREND_LA_VALEUR <i>[0]/b[i]
                        └─ p PREND_LA_VALEUR <i>[0]/b[0]
                    ▼ SI (p==0) ALORS
                        └─ DEBUT_SI
                            └─ Texte PREND_LA_VALEUR "La médiatrice du segment [A"+i+"A"+k+] a pour équation : y = "+m+"x"
                        └─ FIN_SI
                    ▼ SINON
                        └─ DEBUT_SI
                            └─ Texte PREND_LA_VALEUR "La médiatrice du segment [A"+i+"A"+k+] a pour équation : y = "+m+"x + "+p
                        └─ FIN_SI
                    └─ FIN_SI
                └─ FIN_SI
            └─ FIN_SI
        └─ FIN_SI
    ▼ FIN_ALGORITHME

```

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  $x = \text{cste}$ ). Il affiche cependant les résultats sous cette forme réduite.

### 10. Algorithme de Prabhakar

$u_0 \in \mathbb{N}$ .

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

Que constate-t-on ?

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

4 → 16 → 37 → 58 → 89 → 145 → 42 → 20 → 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																																
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
6		89	8	9														721	7	2	1	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
8		42	4	2														160	1	6	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
10		4	4															352	3	5	2	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
12		37	3	7														217	2	1	7	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
14		89	8	9														160	1	6	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
16		42	4	2														352	3	5	2	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
18		4	4															217	2	1	7	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

### 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
<input checked="" type="checkbox"/> Utiliser une fonction Définir la fonction <code>F1(x)= pow(x,3)+pow(x,2)*3^x-1</code>		

Définition de la fenêtre graphique :

Opérations standards	Utiliser une fonction numérique	Dessiner dans un repère
<input checked="" type="checkbox"/> Utiliser un repère Définir le repère Xmin : -3    Xmax : 2 Ymin : -3    Ymax : 3 Graduations X : 1    Graduations Y : 1 <div style="float: right;"> <a href="#">Ajouter TRACER POINT</a>  <a href="#">Ajouter TRACER SEGMENT</a> </div>		

Le programme :



## ALGOBOX : initiation à l'utilisation du logiciel et exemples d'algorithmes .

Le résultat :

**ALGOBOX : MINMAX1**

TESTER L'ALGORITHME

**Lancer algorithme**

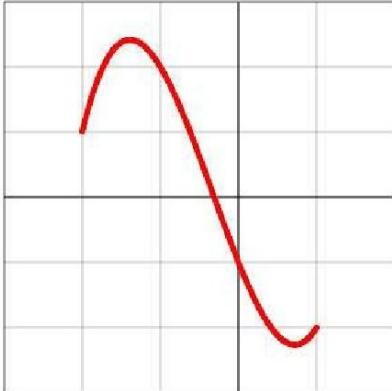
(cliquer sur le bouton ci-dessus pour lancer ou relancer l'exécution de l'algorithme)

**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

```
***Algorithmme 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

***Algorithmme terminé***
```



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

### 12. Méthode de Monte-Carlo

RÉSULTAT

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

***Algorithmme terminé***
```

```

▼ VARIABLES
  |- d EST_DU_TYPE NOMBRE
  |- n EST_DU_TYPE NOMBRE
  |- iEST_DU_TYPE NOMBRE
  |- a EST_DU_TYPE NOMBRE
  |- Texte EST_DU_TYPE CHAINE

▼ DEBUT_ALGORITHME
  |- AFFICHER "Nombre d'itérations ?"
  |- LIRE n
  |- a PREND_LA_VALEUR 0
  |- b PREND_LA_VALEUR 0
  ▼ POUR i ALLANT_DE 1 A n
    |- DEBUT_POUR
    |- d PREND_LA_VALEUR pow(random(),2)+pow(random(),2)
    ▼ SI (d<1) ALORS
      |- DEBUT_SI
      |- a PREND_LA_VALEUR a+1
      |- FIN_SI
    |- FIN_POUR
    |- Texte PREND_LA_VALEUR "Après "+n+" itérations on obtient comme valeur approchée de Pi : "+a/n*4
    |- AFFICHER Texte
  |- FIN_ALGORITHME

```

### 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.

Performance : 5000 valeurs triées en 1 min 55 s. [Pour un ordinateur de référence.]

```

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

```

```

▼ VARIABLES
  |- i EST_DU_TYPE NOMBRE
  |- j EST_DU_TYPE NOMBRE
  |- k EST_DU_TYPE NOMBRE
  |- n EST_DU_TYPE NOMBRE
  |- desordre EST_DU_TYPE LISTE
  |- ordre EST_DU_TYPE LISTE

▼ DEBUT_ALGORITHME
  |- AFFICHER "Nombre de valeurs ?"
  |- LIRE n
  |- AFFICHER "valeurs non triées"
  ▼ POUR i ALLANT_DE 1 A n
    |- DEBUT_POUR
    |- desordre[i] PREND_LA_VALEUR 1+floor(random()*100)
    |- AFFICHER desordre[i]
    |- AFFICHER ""
    |- FIN_POUR
    |- AFFICHER ""
    |- AFFICHER "valeurs triées"
  ▼ POUR i ALLANT_DE 1 A n
    |- DEBUT_POUR
    |- ordre[i] PREND_LA_VALEUR 100
    |- FIN_POUR
  ▼ POUR i ALLANT_DE 1 A n
    |- DEBUT_POUR
    ▼ POUR j ALLANT_DE 1 A n-i+1
      |- DEBUT_POUR
      |- SI (desordre[j]<ordre[i]) ALORS
        |- DEBUT_SI
        |- ordre[i] PREND_LA_VALEUR desordre[j]
        |- k PREND_LA_VALEUR j
        |- FIN_SI
      |- FIN_POUR
    |- POUR j ALLANT_DE k A n-i+1
      |- DEBUT_POUR
      |- desordre[j] PREND_LA_VALEUR desordre[j+1]
      |- FIN_POUR
    |- AFFICHER ordre[i]
    |- AFFICHER ""
    |- FIN_POUR
  |- FIN_ALGORITHME

```