

Chapitre 10

2^{nde}

Algorithmique

Dernière modification : 28 avril 2010

D. ZANCANARO



« Flower Chucker » BANKSY-POCHOIRISTE

Table des matières

I) Introduction	1
II) Qu'est-ce qu'un algorithme	2
II-1 Quelques exemples d'algorithmes	2
II-2 Définition	3
III) Instructions de base	3
IV) Boucle itérative	6
V) Instruction conditionnelle du type Si...alors...sinon...	9
VI) Boucle itérative et instruction conditionnelle	10
VII) index programmation	11
VII-1 Un logiciel pour ordinateur : Algobox	11
VII-1.1 Installation et utilisation	11
VII-1.2 Les procédures	11
VII-1.3 Tests	14
VII-1.4 Boucles	15
VII-2 Un logiciel pour ordinateur : Xcas	16
VII-2.1 Installation et utilisation	16
VII-2.2 Les procédures	16
VII-2.3 Tests	17
VII-2.4 Boucles	18
VII-3 TI	19
VII-3.1 Choix du mode « programme »	19
VII-3.2 Choix du nom du programme	19
VII-3.3 Ecriture des instructions	19
VII-4 Casio	19
VII-4.1 Choix du mode « programme »	19
VII-4.2 Choix du nom du programme	19
VII-4.3 Ecriture des instructions	19

Cours : Algorithmme

I) Introduction

Le mot algorithme vient du nom d'un mathématicien persan Al-Khuwarizmi qui a vécu au début du IX^e siècle. Dans un livre il exposa en effet le premier les méthodes de base de résolution pas à pas des équations. Cependant les algorithmes sont plus anciens, les mésopotamiens calculaient des valeurs approchées de racines carrées à l'aide d'algorithmes près de 18 siècle avant Jésus Christ. De nos jours la notion d'algorithme est lié à l'utilisation d'ordinateur, de calculatrice permettant de les appliquer, c'est pourquoi il faudra apprendre les syntaxes de divers langages de programmations.



II) Qu'est-ce qu'un algorithme

II-1 Quelques exemples d'algorithmes

Entrées : A et B deux points du plan
début
 | Tracer le cercle de centre A passant par B ;
 | Tracer le cercle de centre B passant par A ;
 | Nommer C et D les points d'intersection de ces deux cercles;
 | Construire le polygone $ADBC$;
fin

Algorithme 1 : Une construction géométrique

Entrées : A et B deux nombres réels
début
 | Calculer $A + B$ et remplacer A par cette valeur.;
 | Calculer $A - B$ et remplacer B par cette valeur.;
 | Afficher les nombres A et B ;
fin

Algorithme 2 : Un algorithme de calcul

Entrées : masse m totale
 beurre := $m/4$;
 sucre := $m/4$;
 farine := $m/4$;
 œuf := $m/4$;
début
 | Couper le beurre en petits morceaux et le mettre fondre doucement au bain-marie de préférence. Dès qu'il est fondu arrêter. Laisser refroidir mais attention : le beurre doit être encore liquide ! Il ne doit pas redevenir solide;
 | Mettre le four à préchauffer 160 (th 5);
 | Mettre les oeufs entiers avec le sucre dans un saladier. Battre longuement le mélange pour qu'il blanchisse et devienne bien mousseux;
 | Y ajouter le beurre fondu ET FROID;
 | Rajouter progressivement l'appareil la farine en l'incorporant bien. Cela doit donner une pâte élastique et un peu épaisse;
 | Verser la préparation dans un moule à manqué ou à cake bien beurré;
 | Laisser cuire environ une heure. Il faut surveiller le gâteau régulièrement.;
 | **si** *il semble brunir trop vite* **alors**
 | | il faut baisser un peu le four et mettre une feuille d'aluminium sur le dessus.
 | **fin**
 | Il faut que le dessus du gâteau soit blond fonc, mais pas trop coloré.;
 | **si** *lorsqu'une pique plantée en son milieu ressort sèche* **alors**
 | | le gâteau est cuit
 | **fin**
fin

Algorithme 3 : algorithme breton pur beurre

Remarque : Peut-être avez-vous entendu parler de l'algorithme d'Euclide qui permet de calculer le PGCD de nombres entiers a et b

II-2 Définition



Définition 1 :

Un algorithme est une suite finie d'instruction non ambiguë, à appliquer dans un ordre déterminé à des données.

Remarque : Il comprend trois phases que nous prendrons soin de distinguer :

1. Une phase d'initialisation : on entre des données
2. Une phase de traitement des données (suite de calcul)
3. Une phase de sortie des résultats (affichage des solutions du problème)



Exemple :

Le GPS utilise un algorithme pour déterminer le chemin menant d'un point à un autre. On entre d'abord le point de départ et le point d'arrivée, le GPS effectue une suite de calcul et affiche une ligne brisée qui représente le chemin que l'on doit prendre pour se rendre d'un point à un autre.

III) Instructions de base



Exemple :

On souhaite calculer le volume d'un cylindre de hauteur h et de rayon du disque de base r . Décrire un algorithme permettant de répondre au problème.

Solution :

1. Donnée : h et r .
2. Calcul : $\pi r^2 h$.
3. Affichage : on affiche le résultat du calcul précédent.

Pour que les instructions permettent de bâtir un programme dans des logiciels ou sur une calculatrice il faut donc faire apparaître les trois types d'instructions, ainsi on rédigea l'algorithme comme suit :

Entrées : h , r et v sont trois nombres réels
 Demander la valeur de h ;
 Demander la valeur de r ;
début
 | Calculer $\pi r^2 h$ et placer le résultat dans v ;
 | Afficher la valeur de v ;
fin

Algorithme 4 : Le volume d'un cylindre

Afficher l'image d'un nombre x par la fonction f définie par $f(x) = 3x^2 - 2x + 1$

L'algorithme est alors le suivant :

Entrées : x , y sont deux nombres réels
 Demander la valeur de x ;
début
 | Calculer $3x^2 - 2x + 1$ et mettre le résultat dans y ;
 | Afficher la valeur de y ;
fin

Algorithme 5 : Calcul d'une image par une fonction

Remarque : On aimerait bien faire tourner à l'aide de la calculatrice ou d'un logiciel ces deux derniers algorithmes, pour cela il faut connaître la syntaxe utilisée par la calculatrice ou le logiciel :

Syntaxe des instructions

1. *Calculatrice Casio* :
Saisir A se note ? $\rightarrow A$
 A prend la valeur 2 se note 2 $\rightarrow A$
Afficher A se note A ►
2. *Calculatrice TI* :
Saisir A se note Prompt A
 A prend la valeur 2 se note 2 $\rightarrow A$
Afficher A se note Disp A
3. *Algobox* :
Note : Avec algobox, les variables doivent être déclarées au début du programme. Saisir A se note Lire A
 A prend la valeur 2 se note A prend la valeur 2
Afficher A se note Afficher A
4. *Avec Xcas* :
Saisir A se note `input("A = ", A);`
 A prend la valeur 2 se note `A := 2`
Afficher A se note `print(A);`

Pour le premier algorithme :

Avec Algobox nous obtenons :

CODE DE L'ALGORITHME :

```

1  VARIABLES
2  h EST_DU_TYPE NOMBRE
3  r EST_DU_TYPE NOMBRE
4  v EST_DU_TYPE NOMBRE
5  DEBUT_ALGORITHME
6  LIRE h
7  LIRE r
8  v PREND_LA_VALEUR Math.PI*pow(r,2)*h
9  AFFICHER v
10 FIN_ALGORITHME

```

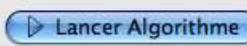
Généré par AlgoBox

Résultats


```

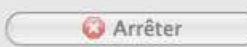
***Algorithme lancé***
9.424778
***Algorithme terminé***

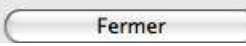
```

 Lancer Algorithme

Mode pas à pas

 Continuer

 Arrêter

 Fermer

Exercice 1 :

Donner un algorithme permettant d'échanger deux nombres A et B ^a

- a. On introduira une troisième variable C

Exercice 2 :

Que fait le programme suivant :

ALGOBOX : ELEVE_QUE_FAIT_LALGO

PRÉSENTATION DE L'ALGORITHME :

Que fait cet algorithme?

CODE DE L'ALGORITHME :

```

1  VARIABLES
2  a EST_DU_TYPE NOMBRE
3  b EST_DU_TYPE NOMBRE
4  m EST_DU_TYPE NOMBRE
5  DEBUT_ALGORITHME
6  LIRE a
7  LIRE b
8  SI (a<b) ALORS
9  DEBUT_SI
10  m PREND_LA_VALEUR b
11  FIN_SI
12  SINON
13  DEBUT_SINON
14  m PREND_LA_VALEUR a
15  FIN_SINON
16  AFFICHER m
17  FIN_ALGORITHME

```

Généré par AlgoBox

Exercice 3 :

Faire fonctionner l'algorithme suivant pour A valant 2, 3 et 4.

Entrées : A est un entier naturel

début

tant que $A \neq 1$ **faire**

si A est impair **alors**

A devient $3A + 1$ (i.e $A := 3A + 1$);

 Afficher A ;

fin

si A est pair **alors**

A devient $\frac{A}{2}$ (i.e $A := \frac{A}{2}$);

 Afficher A ;

fin

fin

fin

Algorithme 6 : boucle et test

Exercice 4 :

1. Convertir 12680 secondes en heures, minutes et secondes.
2. Ecrire un algorithme qui, pour un temps donné en secondes, le convertit en heures, minutes et secondes.

IV) Boucle itérative

Exemple :

Les parents de Léo versent 100-€ sur un livret à sa naissance, puis versent 20-€ chaque mois sur ce livret. On veut élaborer un algorithme donnant la somme sur ce livret au bout d'un certain nombre N de mois.

Pour cela, on initialise la somme S présente sur le livret à 100, puis on répète N fois la même opération, celle qui consiste à ajouter 20 à S : c'est un calcul itératif pour lequel le nombre d'itérations (i.e un calcul pour lequel le nombre d'itérations (i.e de répétitions) est connu, ici N).

```

Entrées :  $N$  est un entier naturel;
 $I$  est un nombre entier;
 $S$  est un nombre entier;
Saisir  $N$ ;
 $S := 100$ ;
début
  pour  $I$  de 1 jusqu'à  $N$  faire
    |  $S := S + 20$ 
  fin
  Afficher  $S$ 
fin

```

Algorithme 7 : boucle for

On peut résumer le fonctionnement de la boucle par le tableau suivant :

TABLE 1 – Fonctionnement de la boucle pour $N = 4$

I	S	Domaine de dérivabilité
Début		100
Fin de la première boucle	1	120
Fin de la deuxième boucle	2	140
Fin de la troisième boucle	3	160
Fin de la dernière boucle	4	180
Fin		180



Syntaxe des instructions

Pour I variant de I_0 à N
faire {instruction}

Fin pour

se traduit par :

1. *Calculatrice Casio* :
For $I_0 \rightarrow I$ to N
instructions
Next
2. *Calculatrice TI* :
For (I, I_0, N)
instructions
Next
3. *Algobox* :
POUR I ALLANT DE I_0 à N
DEBUT_POUR
instructions
FIN_POUR
4. *Avec Xcas* : Pour I de I_0 jusque N
faire instructions ;
fpour

Par conséquent, on obtient le programme suivant avec Algobox :

```

1  VARIABLES
2  N EST_DU_TYPE NOMBRE
3  I EST_DU_TYPE NOMBRE
4  S EST_DU_TYPE NOMBRE
5  DEBUT_ALGORITHME
6  LIRE N
7  S PREND_LA_VALEUR 100
8  POUR I ALLANT_DE 1 A N
9  DEBUT_POUR
10  S PREND_LA_VALEUR S+20
11  FIN_POUR
12  AFFICHER "Léo a donc sur son compte :"
13  AFFICHER S
14  FIN_ALGORITHME

```



Le programme sur Casio

:? $\rightarrow N$

: 100 $\rightarrow S$

: For 1 $\rightarrow I$ To N

: $S + 20 \rightarrow S$

: Next

: S ►

**Le programme sur TI**

```
: Input N  
: 100 → S  
: For(I, 1, N)  
: S + 20 → S  
: End  
: Disp S
```

```
Saisir(N);  
S:=100;  
pour I de 1 jusque N  
faire S:=S+20;  
fpour;  
afficher(S)
```

Programme 1 – Le programme sur XCAS

**Exercice 5** :

Ecrire un algorithme qui demande un nombre de départ et qui affiche les dix nombres suivants.

**Exercice 6** :

Elaborer un algorithme permettant le calcul de la somme des entiers de 1 à n

**Exercice 7** :

Elaborer un algorithme permettant le calcul de la somme des inverses des n premiers entiers naturels non nuls.

**Exercice 8** :

Mathis verse sur un compte la somme de 1000€. Ce compte rapporte 5% d'intérêts par an.
Construire un algorithme permettant d'obtenir la somme sur le compte au bout de n années.

V) Instruction conditionnelle du type Si... alors... sinon...

Exemple :

Un magasin de photos propose le développement au tarif de 0,16€ l'unité; la tarif devient de 0,12€ l'unité pour une commande d'au moins 75 photos.

On veut élaborer un algorithme donnant le montant dépensé pour un nombre N de photos à développer.

Pour cela, on doit introduire une instruction conditionnelle dans l'algorithme :

- Si le nombre de photos N est strictement inférieur à 75, le montant est $N \times 0,16$
- Sinon le montant est $N \times 0,12$

C'est la structure alternative du type Si... alors... sinon... qui permet d'écrire cette condition

```

Entrées :  $N$  est un entier naturel;
 $S$  est un nombre entier;
Saisir  $N$  ;
début
  | si  $N < 75$  alors
  |   |  $S := N \times 0,16$ 
  | sinon
  |   |  $S := N \times 0,12$ 
  | fin
  | Afficher  $S$ 
fin

```

Algorithme 8 : Test : Si ... alors ... sinon ...



Syntaxe des instructions

Si condition C

alors instructions A

sinon instructions B

Fin Si

se traduit par :

1. *Calculatrice Casio* :
If condition C
Then instructions A
Else instructions B
Ifend
2. *Calculatrice TI* :
If condition C
Then instructions A
Else instructions B
End
3. *Algobox* :
SI condtion C ALORS
DEBUT_ SI
instructions A
FIN_ SI
SINON
DEBUT_ SINON
instructions B
FIN_ SINON
4. *Avec Xcas* : si condition C
alors instructions A
sinon instructions B
fsi

Avec algobox l'algorithme est le suivant :

```

1  VARIABLES
2  N EST_DU_TYPE NOMBRE
3  S EST_DU_TYPE NOMBRE
4  DEBUT_ALGORITHME
5  LIRE N
6  SI (N<75) ALORS
7  DEBUT_SI
8  S PREND_LA_VALEUR N*0.16
9  FIN_SI
10 SINON
11 DEBUT_SINON
12 S PREND_LA_VALEUR N*0.12
13 FIN_SINON
14 AFFICHER "Développer les photos coutent :"
15 AFFICHER S
16 AFFICHER "euros"
17 FIN_ALGORITHME

```

Exercice 9 :

Ecrire un algorithme demandant deux nombres distincts et donnant en sortie le plus grand des deux

Exercice 10 :

Ecrire un algorithme qui, à tout nombre, associe ce nombre s'il est positif et son opposé s'il est négatif

Exercice 11 :

Un commerce de reprographie facture 0,20€ les 20 premières photocopies et 0,10€ les suivantes .

1. Quel est le montant payé pour 15 photocopies ? pour 50 photocopies ?
2. Ecrire un algorithme permettant de calculer le montant payé quand le nombre de photocopies est donné. Programmer cet algorithme, puis vérifier les résultats de la question *a*.

VI) Boucle itérative et instruction conditionnelle

Exercice 12 :

Trouver tous les triangles rectangles (ayant pour longueur maximale l'entier N) dont les côtés sont des entiers consécutifs à l'aide d'un algorithme que vous programmerez

Exercice 13 :

Une balle lâchée d'une hauteur donnée rebondit chaque fois qu'elle touche le sol au $1/5^e$ de sa hauteur. On veut écrire un algorithme qui donne le nombre de rebonds de la balle avant que celle-ci soit à un millimètre du sol.

Exercice 14 :

Pour quelle valeur de n , la somme des n premiers entiers dépasse-t-elle 1000 ?

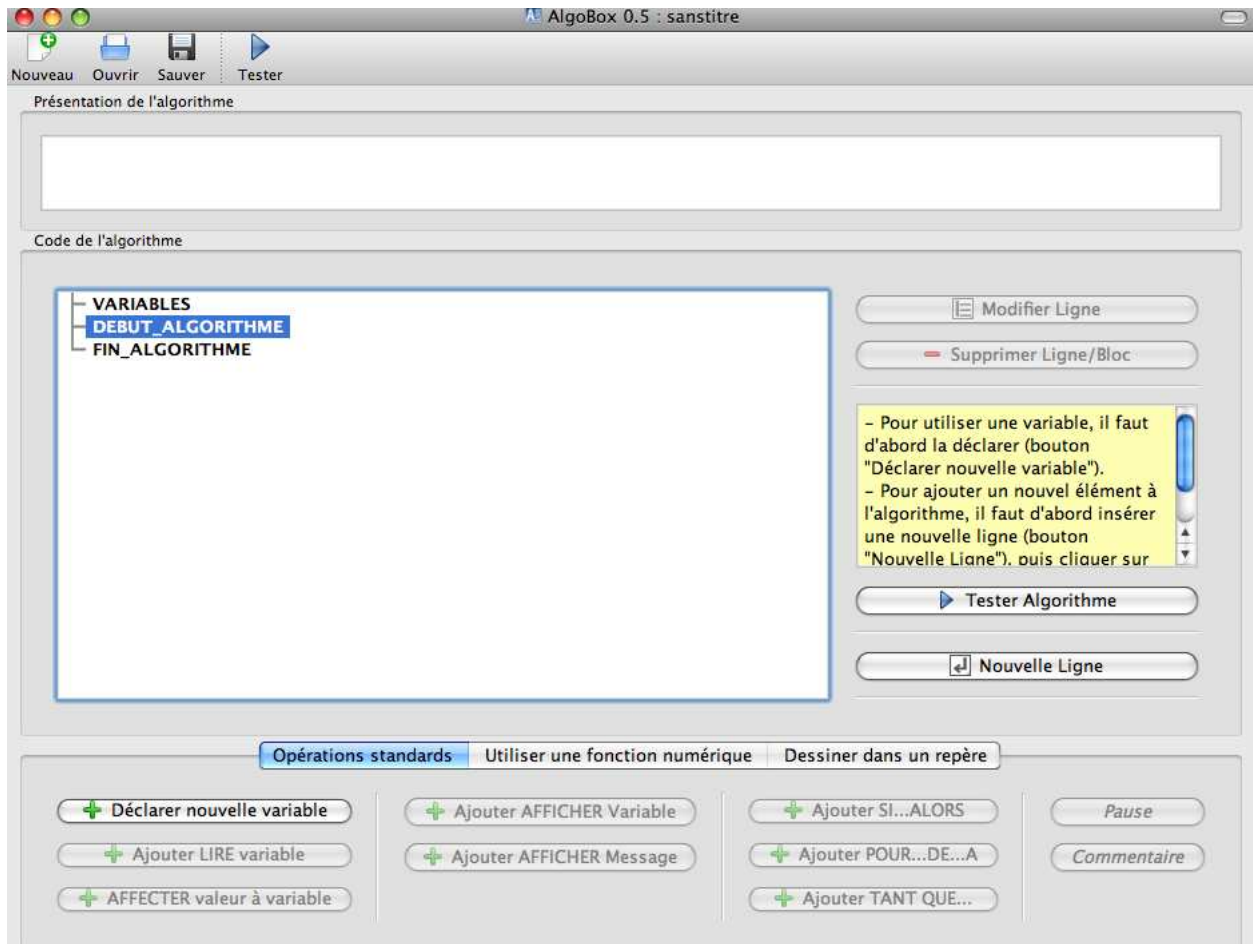
VII) L'index programmation

Pour écrire et utiliser un programme sur une calculatrice ou un logiciel, il faut connaître les instructions permettant l'accès au mode programmation et le fonctionnement de celui-ci.

VII-1 Un logiciel pour ordinateur : Algobox

VII-1.1 Installation et utilisation

Vous récupérez la version pour votre O.S. sur le Site Algobox. Vous suivez les instructions et voilà...

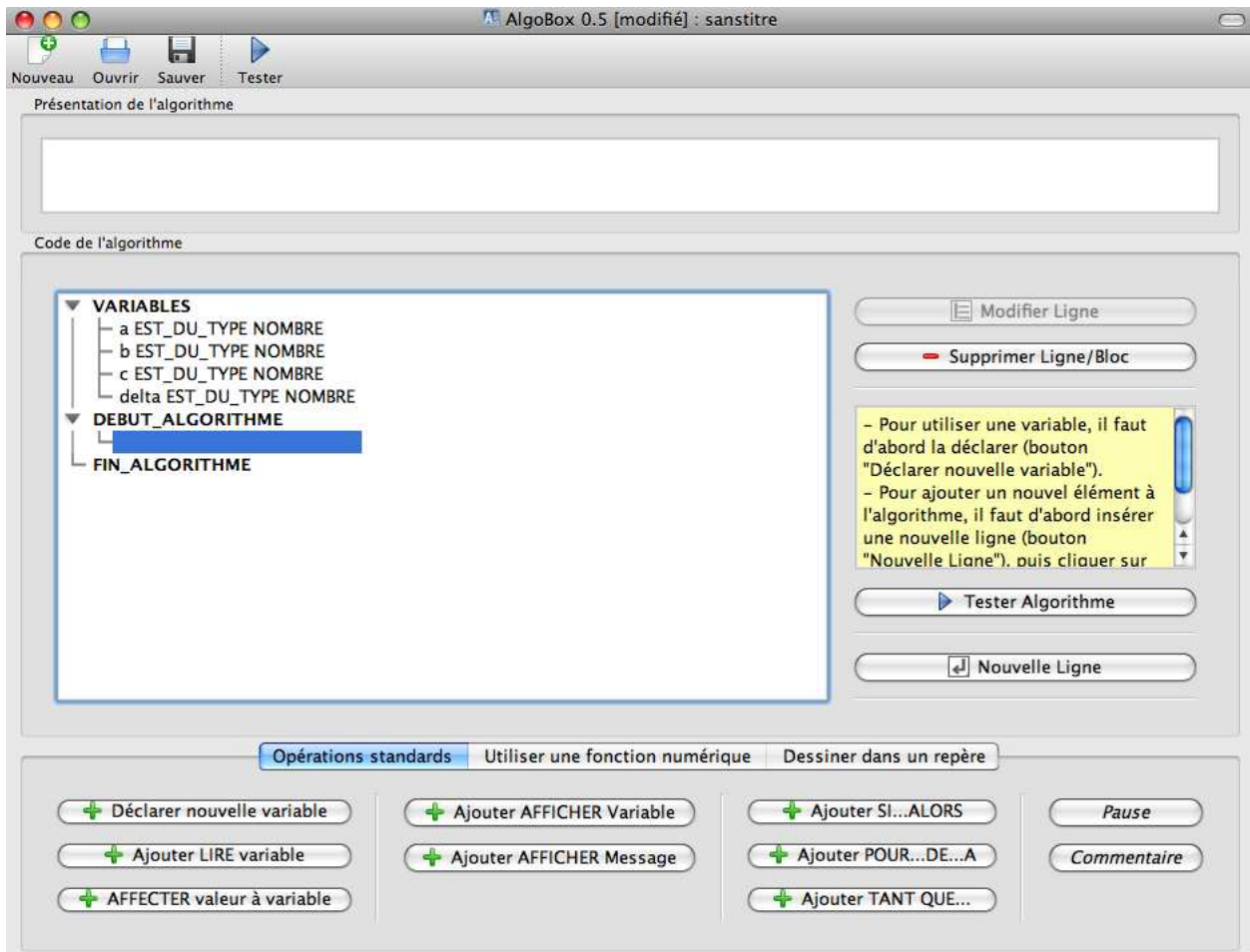


Algobox a été conçu dans la perspective d'enseigner les algorithmes au lycée, il conviendra idéalement ici pour tous les algorithmes que nous aurons besoin de traiter. Il est pourtant moins puissant que Xcas, le logiciel présenté dans la partie suivante. Il ne reste plus qu'à programmer...

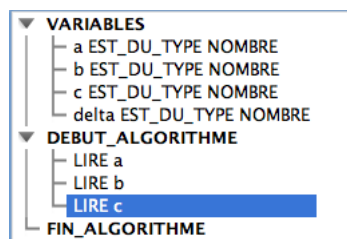
VII-1.2 Les procédures

Par exemple, nous voudrions créer une procédure qui prend trois nombres a , b et c en argument et renvoie $b^2 - 4ac$. Appelons-la au hasard... Δ :

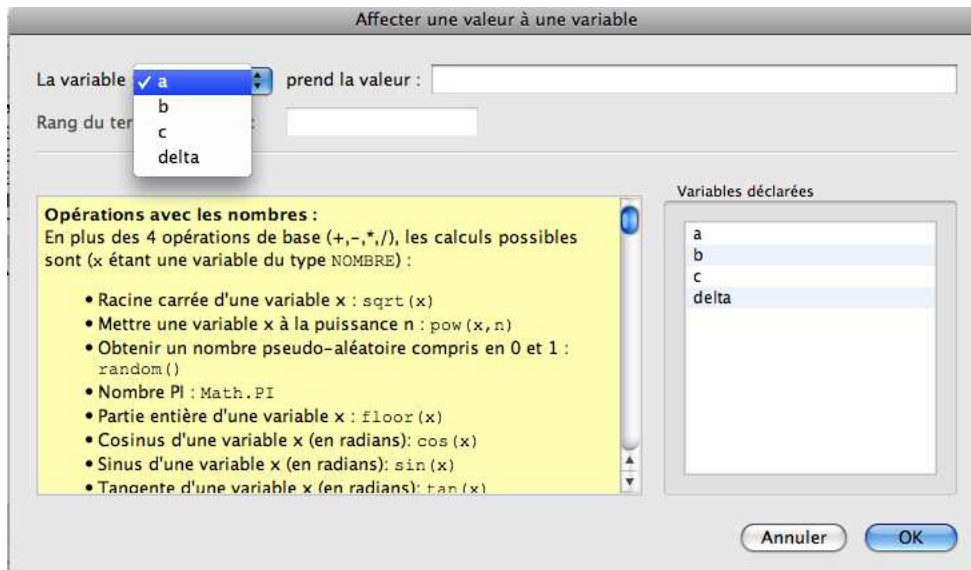
Dans algobox, nous prenons toujours soin de déclarer les variables avant de commencer la partie traitement des données, à l'aide du bouton déclarer une nouvelle variable :



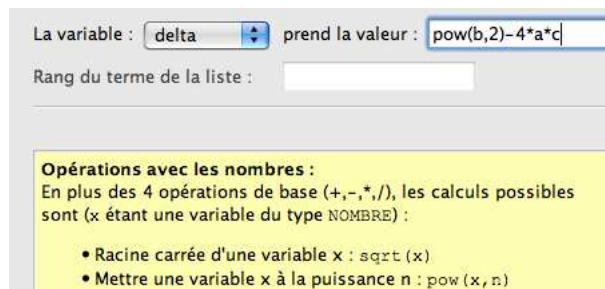
Avant toute chose, il faut demander à l'utilisateur de saisir les valeurs de a , b et c pour lesquelles il souhaite obtenir le calcul de Δ , pour cela on crée une nouvelle ligne puis on ajoute LIRE variable pour obtenir :



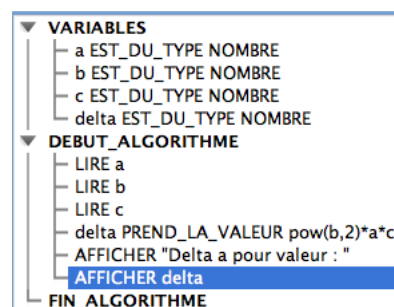
On souhaite désormais affecter à Δ la valeur $b^2 - 4ac$, pour cela on crée d'abord une nouvelle ligne puis on clique sur le bouton affecter valeur à variable et on obtient l'écran suivant :



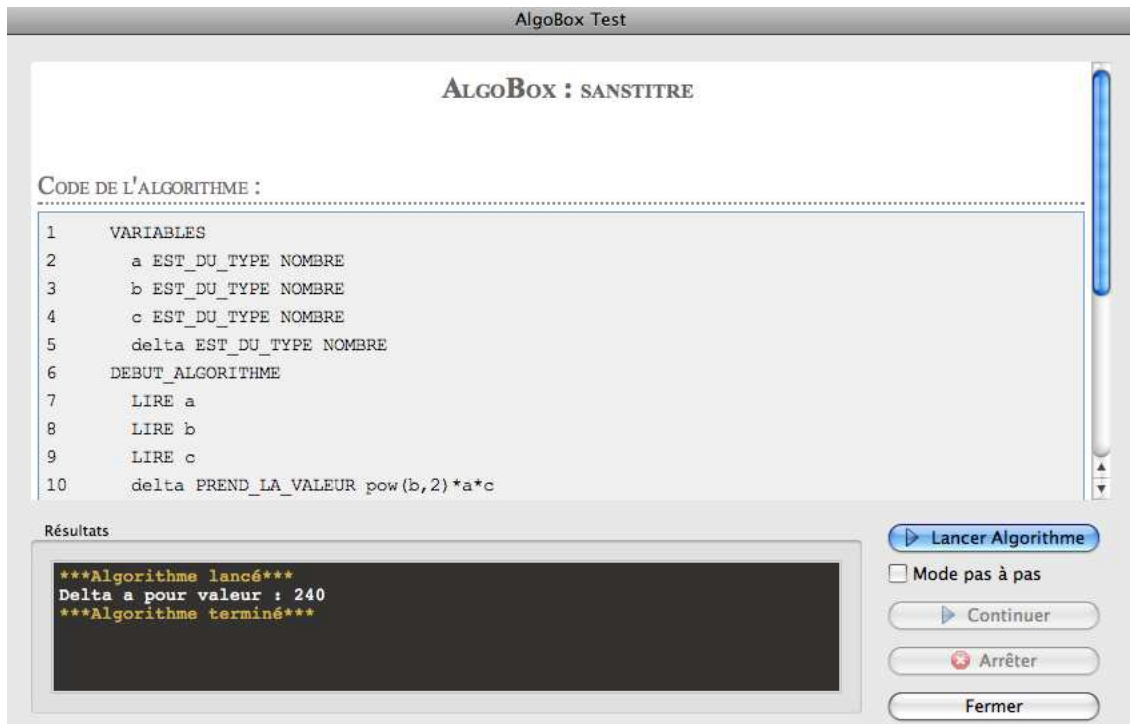
On choisit alors la variable Δ à laquelle on affecte la valeur $b^2 - 4ac$, en faisant bien attention de respecter la syntaxe (en jaune) des opérations de références utilisées par algobox :



Après validation il ne reste plus qu'à afficher la valeur de Δ en se laissant guider par le logiciel on obtient : (ne pas oublier de créer une nouvelle ligne avant toute nouvelle opération !)



Il ne reste plus qu'à tester l'algorithme en calculant Δ pour, par exemple $a = 3$, $b = 4$ et $c = 5$, on obtient :



VII-1.3 Tests

Et si l'on souhaite programmer la valeur absolue d'un nombre x avec algoebox, on tape :

```

1  VARIABLES
2  x EST_DU_TYPE NOMBRE
3  resultat EST_DU_TYPE NOMBRE
4  DEBUT_ALGORITHME
5  LIRE x
6  SI (x>0) ALORS
7  DEBUT_SI
8  resultat PREND_LA_VALEUR x
9  FIN_SI
10 SINON
11 DEBUT_SINON
12 resultat PREND_LA_VALEUR -x
13 FIN_SINON
14 AFFICHER "La valeur absolue de "
15 AFFICHER x
16 AFFICHER " est :"
17 AFFICHER resultat
18 FIN_ALGORITHME

```

Noter, l'existence d'un mode pas à pas, pratique pour comprendre les erreurs de programmation éventuellement commises..., après test de l'algorithme on obtient par exemple pour $x = -7$:

ALGOBox : SANSTITRE

CODE DE L'ALGORITHME :

```

1  VARIABLES
2  x EST_DU_TYPE NOMBRE
3  resultat EST_DU_TYPE NOMBRE
4  DEBUT_ALGORITHME
5  LIRE x
6  SI (x>0) ALORS
7  DEBUT_SI
8  resultat PREND_LA_VALEUR x
9  FIN_SI
10 SINON

```

Résultats

```

***Algorithme lancé***
La valeur absolue de -7 est :7
***Algorithme terminé***

```

Mode pas à pas

VII-1.4 Boucles

Calculons par exemple la somme des n premiers entiers :

```

1  VARIABLES
2  n EST_DU_TYPE NOMBRE
3  S EST_DU_TYPE NOMBRE
4  i EST_DU_TYPE NOMBRE
5  DEBUT_ALGORITHME
6  LIRE n
7  POUR i ALLANT_DE 1 A n
8  DEBUT_POUR
9  S PREND_LA_VALEUR S+i
10 FIN_POUR
11 AFFICHER "La somme des "
12 AFFICHER n
13 AFFICHER " premiers entiers est : "
14 AFFICHER S
15 FIN_ALGORITHME

```

En testant l'algorithme pour $n = 100$ on obtient :

CODE DE L'ALGORITHME :

```

1  VARIABLES
2  n EST_DU_TYPE NOMBRE
3  S EST_DU_TYPE NOMBRE
4  i EST_DU_TYPE NOMBRE
5  DEBUT_ALGORITHME
6  LIRE n
7  POUR i ALLANT_DE 1 A n
8  DEBUT_POUR
9  S PREND_LA_VALEUR S+i
10 FIN_POUR

```

Résultats

```

***Algorithme lancé***
La somme des 100 premiers entiers est : 5050
***Algorithme terminé***

```

Mode pas à pas

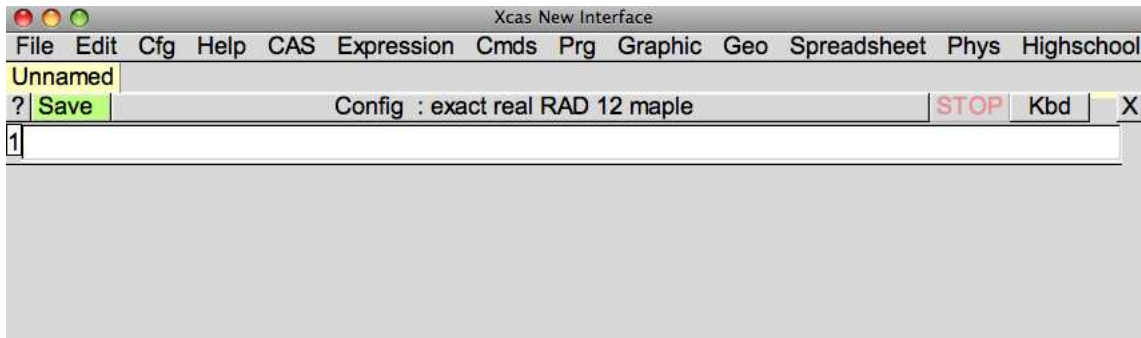
Remarque : On raconte qu' à 7 ou 10 ans, Karl Gauss, mathématiciens de génie, a trouvé la manière de calculer la somme des nombres de 1 à 100 très rapidement, à la grande surprise de son professeur.

VII-2 Un logiciel pour ordinateur : Xcas

VII-2.1 Installation et utilisation

Vous récupérez la version pour votre O.S. sur le Site [XCAS](#).

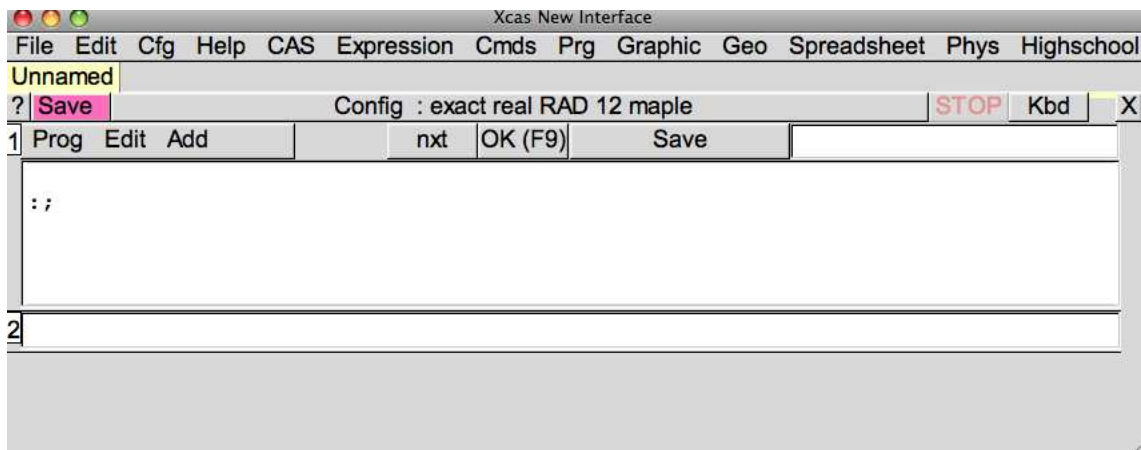
Vous suivez les instructions et voilà...



Il y a beaucoup à dire sur XCAS mais nous nous contenterons d'explorer les outils de programmation.

Nous allons donc ouvrir une fenêtre de programmation en tapant `[Alt] + [P]`.

Une fenêtre de programmation apparaît :

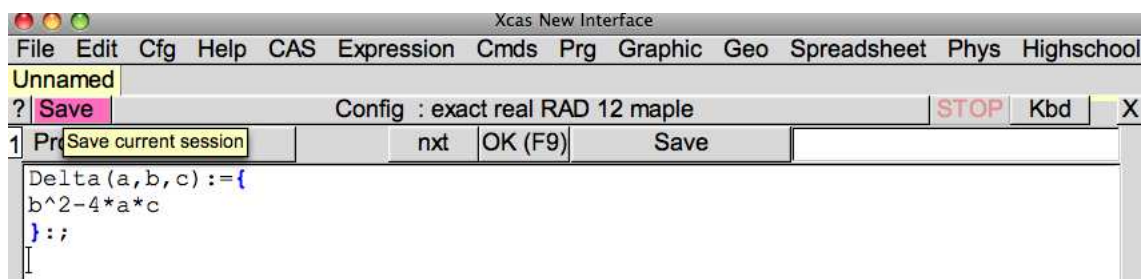


Il ne reste plus qu'à programmer...

VII-2.2 Les procédures

XCAS est écrit en C++ et a donc une syntaxe proche de ce langage...mais en bien plus simple.

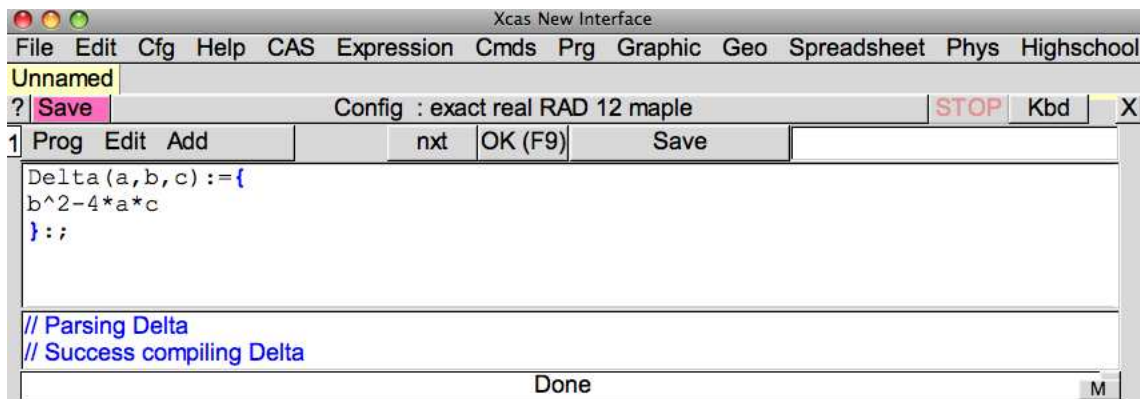
Par exemple, nous voudrions créer une procédure qui prend trois nombres a , b et c en argument et renvoie $b^2 - 4ac$. Appelons-la au hasard...Delta :



Par la suite, nous représenterons ce script par :

```
Delta(a,b,c) := {
b^2-4*a*c
};;
```

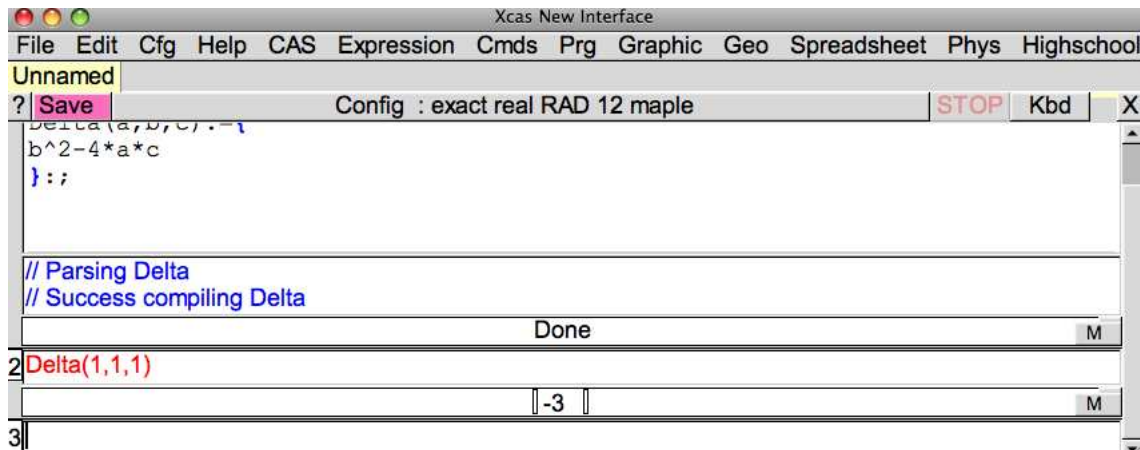
Il suffit ensuite de cliquer sur OK ou de taper sur **F9**



Pour évaluer cette procédure en un certain triplet, nous entrons dans une cellule de calcul :

```
Delta(1,1,1)
```

ce qui apparaît sous la forme :



VII-2.3 Tests

Avec XCAS, on peut programmer en français ou en anglais.

```
val_abs(x):={
si x>0 alors x sinon -x fsi;
};;
```

ou

```
val_abs(x):={
if(x>0)then{x}else{-x}
};;
```

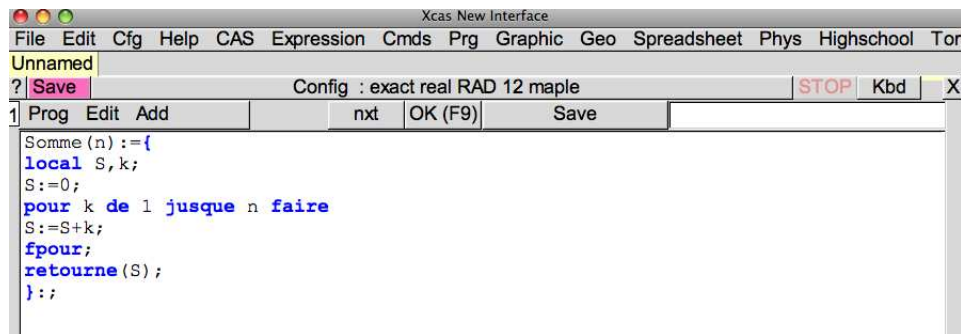
ou même

```
val_abs(x):={
ifte(x>0,x,-x)
};;
```

VII-2.4 Boucles

Le bilinguisme est ici aussi de rigueur. Calculons par exemple la somme des premiers entiers :

```
Somme(n) := {
  local S, k;
  S := 0;
  pour k de 1 jusque n faire
    S := S + k;
  fpour;
  retourne(S);
};;
```



ou

```
Somme(n) := {
  local S, k;
  S := 0;
  for(k := 1; k <= n; k := k + 1) {
    S := S + k;
  }
  return(S);
};;
```

ou

```
Somme(n) := {
  local S, k;
  S := 0;
  k := 1;
  tantque k <= n faire
    S := S + k;
    k := k + 1;
  ftantque;
  retourne(S);
};;
```

ou

```
Somme(n) := {
  local S, k;
  S := 0;
  k := 1;
  while(k <= n) {
    S := S + k;
    k := k + 1;
  };
  return(S);
};;
```

VII-3 TI

VII-3.1 Choix du mode « programme »

Appuyer sur la touche **prgm**

Une liste de programmes apparaît à l'écran. Un menu de fonctions demande de choisir entre : **EXEC**, **EDIT** et **NOUV**

- Pour créer un nouveau programme, on choisit **NOUV** puis on appuie sur la touche **entre**
- Pour modifier un ancien programme, on choisit **EDIT** puis on appuie sur la touche **entre**
- Pour lancer un programme, on choisit **EXEC** puis on sélectionne le programme désiré avant de valider à l'aide de la touche **entre**

VII-3.2 Choix du nom du programme

L'affichage indique : Nom=

On écrit alors un nom de programme, formé d'au plus huit caractères et l'on valide en appuyant sur la touche **entre**

VII-3.3 Ecriture des instructions

Après chaque ligne entrée, on appuie sur **entre**. Le programme passe à la ligne suivante et : s'affiche. Pour avoir accès aux instructions spécifiques de programmation, on utilise l'instruction **prgm**.

Le menu : **CTL**, **E/S**, **EXEC** s'affiche :

1. le menu **CTL** fournit les instructions : If, Then, Else, For(, While, Repeat, End, Pause, ...
2. le menu **E/S** donne accès à : Input, prompt, Disp.

Appuyer sur **2nde mode** pour quitter l'éditeur.

VII-4 Casio

VII-4.1 Choix du mode « programme »

Amener le curseur sur l'icône **PRGM**, puis appuyer sur **EXE**. Une liste de programmes apparaît à l'écran. Un menu de fonctions demande de choisir entre **EXE**, **EDIT**, **NEW**, **DEL**, **DEL A**, **▶**

- Pour créer un nouveau programme, on choisit **NOUV**
- Pour modifier un ancien programme, on choisit **EDIT** obtenue à l'aide de la touche **F2**
- Pour lancer un programme, on choisit **EXE** puis on sélectionne le programme désiré avant de valider à l'aide de la touche **F1**

VII-4.2 Choix du nom du programme

L'affichage indique : Program Name

[.....]

On écrit alors un nom de programme, formé d'au plus huit caractères et l'on valide en appuyant sur la touche **EXE**

VII-4.3 Ecriture des instructions

Après chaque ligne entrée, on appuie sur **EXE**. Une flèche s'affiche et le programme passe à la ligne suivante. Pour avoir accès aux instructions spécifiques de programmation, on utilise les touches **SHIF** **VAR**.

Le menu : **COM**, **CTL**, **JUMP**, **?**, **◀**, **▶** s'affiche. En sélectionnant **▶** (**F6**), on accède au menu : **CLR**, **DISP**, **REL** et **I/O** :

le menu **COM** fournit les instructions : If, Then, Else, Iend, For, To, Step, NExt, While, Wend, Do, LpW
Appuyer sur **SHIF EXIT** pour quitter l'éditeur.