

I.R.E.M. de LIMOGES Facultés des Sciences

*Journée Enseignement des Mathématiques  
en Limousin*

*Algorithmique et Calculatrice*

Jeudi 9 décembre 2010

Samuel ADABIA,

I.R.E.M. de Limoges

## Objectifs :

A - Présenter la notion d'algorithme :

- 1) Qu'est-ce qu'un algorithme?
- 2) Les éléments de base d'un algorithme
- 3) La forme générale d'un algorithme
- 4) Comment réussir l'écriture d'un algorithme?
- 5) Les différentes structures ou instructions de contrôle

B - Écrire un algorithme correspondant à un problème donné:

- en statistique (fréquences, vers la simulation, moyennes mobiles,...)
- en probabilité ( .... vers la simulation)

C - Traduire cet algorithme en langage de programmation machine: la calculatrice

D - Modifier un algorithme pour obtenir un résultat souhaité.

## AVANT TOUT DISCOURS UN COUP D'OEIL SUR LES OBJECTIFS OFFICIELS extrait Nouveau Programme de la Classe Seconde 2009

### Algorithmique (objectifs pour le lycée)

La démarche algorithmique est, depuis les origines, une composante essentielle de l'activité mathématique. Au collège, les élèves ont rencontré des algorithmes (algorithmes opératoires, algorithme des différences, algorithme d'Euclide, algorithmes de construction en géométrie). Ce qui est proposé dans le programme est une formalisation en langage naturel propre à donner lieu à traduction sur une calculatrice ou à l'aide d'un logiciel. Il s'agit de familiariser les élèves avec les grands principes d'organisation d'un algorithme : gestion des entrées-sorties, affectation d'une valeur et mise en forme d'un calcul, en opérant essentiellement sur des nombres entiers.

Dans le cadre de cette activité algorithmique, les élèves sont entraînés :

- à décrire certains algorithmes en langage naturel ou dans un langage symbolique ;
- à en réaliser quelques-uns à l'aide d'un tableur ou d'un petit programme réalisé sur une calculatrice ou avec un logiciel adapté ;
- à interpréter des algorithmes plus complexes.

Aucun langage, aucun logiciel n'est imposé.

L'algorithmique a une place naturelle dans tous les champs des mathématiques et les problèmes posés doivent être en relation avec les autres parties du programme (fonctions, géométrie, statistiques et probabilité, logique) mais aussi avec les autres disciplines ou la vie courante.

À l'occasion de l'écriture d'algorithmes et de petits programmes, il convient de donner aux élèves de bonnes habitudes de rigueur et de les entraîner aux pratiques systématiques de vérification et de contrôle.

#### Contenus

**Instructions élémentaires** (affectation, calcul, entrée, sortie).

Les élèves, dans le cadre d'une résolution de problèmes, doivent être capables :

- d'écrire une formule permettant un calcul ;
- d'écrire un programme calculant et donnant la valeur d'une fonction ; ainsi que les instructions d'entrées et sorties nécessaires au traitement.

**Boucle et itérateur, instruction conditionnelle**

Les élèves, dans le cadre d'une résolution de problèmes, doivent être capables :

- de programmer un calcul itératif, le nombre d'itérations étant donné ;
- de programmer une instruction conditionnelle, un calcul itératif, avec une fin de boucle conditionnelle.

## I Notion d'algorithme Définition

### Activité n°1

Dans une urne on dispose de boules vertes, jaunes, rouges. Soit l'expérience aléatoire qui consiste à tirer une boule de cette urne puis à ranger les boules suivant leur couleur.

### Activité n°2

Soit à construire à la règle non graduée et au compas, la médiatrice d'un segment donné. Dans ces deux activités nous avons mis en œuvre à chaque fois une série d'actions élémentaires qui vont nous conduire au résultat.

Nous dirons tout simplement qu'un **algorithme** est *une succession d'actions menées par étapes et qui permet la réalisation d'une tâche.*

**Exemple** Pour se prémunir contre la grippe A, H1N1, il est recommandé de se laver régulièrement les mains dans la journée. Pour se laver les mains, il faut

- un robinet ou un point d'eau,
- du savon,
- puis il faut ouvrir le robinet,
- se mouiller légèrement les mains,
- se savonner,
- se frotter les mains,
- se les rincer,
- fermer le robinet,
- s'essuyer les mains.

Comme vous le voyez nous venons *de mettre en œuvre un algorithme* ( c'est à dire une succession d'actions simples par étapes) qui nous a conduit à nous laver les mains.

Un tel algorithme suscite des interrogations.

Avons-nous toujours à disposition

- 1)un robinet ou un point d'eau ? Dans le cas contraire que faut il faire ?
- 2)du savon? Dans le cas contraire que faut-il faire ?
- 3)un essuie-mains? Dans quel état est-il ? Dans le cas contraire que faut-il faire ?

**En conclusion**, pour exécuter un algorithme, il est nécessaire que certaines conditions soient réunies, vérifiées, validées. Dans le cas contraire que faut-il envisager ?

Peut-on commencer par se sécher les mains avant de mettre la main sous le robinet ouvert ? La réponse montre de toute évidence que dans un algorithme il s'agit d'une succession d'**actions ordonnées** exécutées par étapes.

En nous lavant les mains, quand sait-on qu'on sait bien se laver les mains ? qu'on les a lavées suffisamment ? Faut-il continuer ? Jusqu'à quand ? Combien de fois dois-je me laver les mains dans la journée ?

A cette dernière question on s'attend à la réponse : autant que nécessaire, après chaque activité.

Toutes ces questions nous amènent à faire des **tests**, à chercher à **boucler** ou à **terminer** ces actions.

## II Quelques Exemples d'algorithmes « célèbres »

1) L'algorithme d'EUCLIDE permet de d'obtenir le pgcd de deux entiers naturels. Une autre utilisation de cet algorithme a aussi permis d'écrire des fractions continues ou continuées (voir site de l'IREM, fractions continues)

2) L'algorithme de HÖRNER qui permet de factoriser, plus facilement, les polynômes de degré inférieur ou égal à 4.

3) L'algorithme du pivot de GAUSS pour ne citer que ceux là

## III Les éléments simples de base d'un algorithme sont :

- 1 Les variables
- 2 Entrée ou Initialisation des données
- 3 Traitement des données
- 4 Sortie

et la forme générale d'un algorithme est :

Début

### Déclaration des variables

ici on choisit le nom des lettres ( ce sont des variables ) qui vont représenter les données indispensables à la résolution du problème posé. On peut distinguer des données de type :

- 1) numérique
- 2) chaîne de caractères
- 3) logique ( vrai ou faux )
- 4) graphique

Les variables permettent de réserver de la place mémoire pour conserver des résultats intermédiaires, ou des résultats tout court.

### Entrée ou Initialisation des données

ici on récupère les données nécessaires à la résolution du problème.

Ici on peut attribuer à des variables leurs premières valeurs (on parle d'*initialisation*) ou on peut attribuer à des variables des résultats intermédiaires.(on parle d'*affectation*)

### Traitement des données

ici on met tout en œuvre pour résoudre le problème posé. C'est à dire que l'on donne une ou plusieurs instructions simples de sorte que l'ensemble permette la résolution du problème posé. Les variables interviennent ici pour récupérer des résultats

### Sortie

Ici on affiche le ou les résultats du problème posé.

Fin

Un algorithme s'écrit en langage naturel. Mais des logiciels spécialisés tels que ALGOBOX, LARP, permettent de faire tourner un algorithme. Dans cette activité nous avons choisi la calculatrice pour transcrire les algorithmes que nous allons écrire.

## **IV Vers la Programmation ou la transcription d'un algorithme**

### **Avec La TI 83 plus**

La touche PRGM permet de rentrer dans le mode programmation

EXEC EDIT NOUVEAU

Dans le menu «PRGM» de la **calculatrice** on fait «NOUVEAU» pour préciser que l'on crée un nouveau programme. On commence par lui donner un nom Après avoir validé le nom, on entre les instructions en se servant de :

- 1) L'instruction DISP. DISP" Bonjour" affichera Bonjour lors de l'exécution
- 2) Input permet de recueillir la valeur ou le caractère saisi au clavier.
- 3) Input" Entrer la valeur de n" , n permet d'afficher le texte puis de récupérer la valeur de n ou le caractère saisi au clavier.
- 4) Le OUTPUT permet de positionner à un endroit précis l'information.
- 5) 87 STO A ( signifie que l'on a mémorisé ou stocké 87 dans la variable A)
- 6)  $b*c$  STO A ( signifie que l'on a mémorisé ou stocké le produit de b et c dans la variable A)

## Avec une CASIO

Dans le menu «PRGM» de la **calculatrice CASIO**, on fait «NEW» pour préciser que l'on crée un nouveau programme. On commence par lui donner un nom. Après avoir validé le nom, on entre les instructions en se servant de :

1) La double apostrophe, par exemple, se trouve dans «OPTN» / «SYBL», permet d'afficher du texte. Par exemple "Bonjour"

2) Le «:» permet d'écrire deux instructions sur une même ligne.

Par exemple "Bonjour" : "Pierre"

3) Le « $\blacktriangleleft$ » équivaut à un retour à la ligne avec affichage du résultat.

4) L'utilisateur doit appuyer sur «EXE» pour obtenir l'affichage suivant.

5)  $87 \longrightarrow A$  (signifie que l'on a mémorisé ou stocké 87 dans la variable A)

6)  $b*c \Longrightarrow A$  (signifie que l'on a mémorisé ou stocké le produit de b et c dans la variable A)

### V Comment «réussir» l'écriture d'un algorithme?

On suppose régler le problème mathématique (si nécessaire ne pas hésiter à donner la formule)

1) Bien lire l'énoncé pour savoir ce que l'on attend de vous

2) Se poser la question et y répondre : de quoi ai-je besoin? (vous êtes en train de trouver les variables)

3) Y-a-t-il des résultats intermédiaires? Si oui comment les conserver?

4) Ne pas oublier de répondre à la question posée (vous êtes dans la partie SORTIE)

Il faut savoir qu'un algorithme se fait en deux temps :

Premier temps : Un Monsieur ou une Dame conçoit et écrit l'algorithme

Deuxième temps : Vous et moi qui n'étions pas à la conception de cet algorithme voudrions l'exécuter ne serait-ce que pour voir s'il tourne ou s'il fonctionne.

C'est pour cette raison que votre algorithme doit être simple, claire et compréhensible.

EXERCICES

**Exercice1:**

Pierre possède une parcelle formée d'une partie rectangulaire et sur la largeur d'une partie triangulaire ayant une aire fixe de  $15 \text{ m}^2$ .

– Écrire, **en identifiant clairement ses différentes parties**, un algorithme permettant de calculer et d'afficher l'aire totale de la parcelle de Pierre.

L'utilisateur doit fournir les données nécessaires.

– Écrire le programme calculatrice correspondant à l'algorithme ci-dessus.

– Vérifier le programme pour

a)  $L=15 \text{ m}$  et  $l= 7 \text{ m}$  on doit trouver  $A=120 \text{ m}^2$

b)  $L=23 \text{ m}$  et  $l= 19 \text{ m}$  on doit trouver  $A =452 \text{ m}^2$

**Exercice2 :**

Modifier le programme ci-dessus de sorte que l'utilisateur puisse fournir l'aire de la partie triangulaire. Vérifier

**Exercice4:** 1) Écrire un algorithme permettant de calculer puis d'afficher la somme des nombres pairs inférieurs ou égaux à un nombre fourni par l'utilisateur.

2) Écrire le programme calculatrice correspondant.

**Exercice5:** Modifier l'algorithme puis le programme ci-dessus pour connaître le nombre de nombres pairs utilisés pour calculer la somme.

**Exercice 6:**

1) Écrire un algorithme permettant de simuler un lancé d'un dé parfaitement équilibré. Afficher le résultat.

2) Écrire le programme calculatrice correspondant.

**Exercice7:**

1) Modifier l' algorithme puis le programme ci-dessus pour réaliser  $n$  lancers d'un dé,  $n$  est fourni par l'utilisateur; d'afficher la fréquence d'apparition de chaque chiffre.

2) Modifier l' algorithme puis le programme pour afficher la fréquence de nombres ainsi que la fréquence de nombres impairs.

**Exercice8:**

1) Écrire un algorithme permettant de résoudre sur  $\mathbb{R}$  une équation de la forme  $ax^2+bx+c=0$ , où  $a$ ,  $b$  et  $c$  sont des coefficients réels,  $a$  étant différent de  $0$ .

2) Écrire le programme calculatrice correspondant.

**Exercice9 :**

- 1) Écrire un algorithme permettant de ranger par ordre croissant un n-uplets de nombres aléatoires. Le nombre n étant fourni par l'utilisateur. Afficher le n-uplets initial et le n-uplets rangé.
- 2) Écrire le programme calculatrice correspondant.

**Exercice 10:**

Modifier l' algorithme puis le programme ci-dessus pour n'afficher que le n-uplets composé uniquement de nombres pairs.

**Exercice 11 :**

Un magasin de vente en gros propose un même article à un prix affiché x sous les conditions suivantes :

Pour tout achat d'une quantité inférieure à 200 pièces, le marchand vous fait une réduction de 7,25 % du prix unitaire de l'article plus une réduction de 9,00 euros sur le prix total à payer.

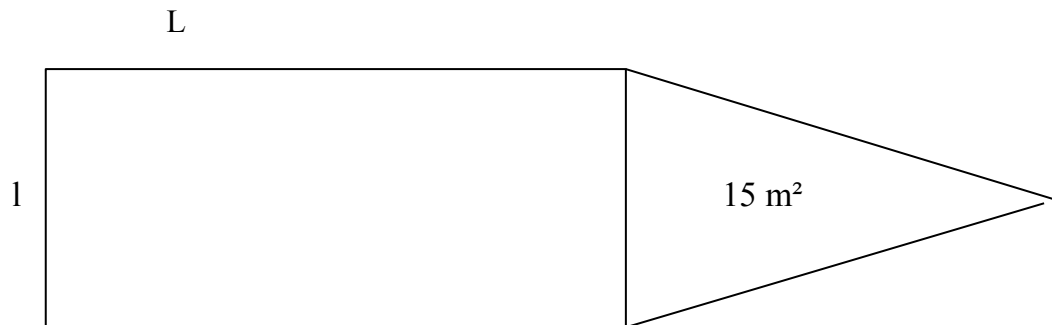
Pour tout achat d'une quantité supérieure à 200 pièces, il offre une réduction de 15% du prix unitaire et ne facture rien d'autres.

- 1) Écrire un algorithme permettant à l'utilisateur de connaître le prix à payer. Il doit fournir la quantité commandée.
- 2) Écrire le programme calculatrice correspondant.
- 3) Modifier l' algorithme puis le programme ci-dessus pour afficher le .



## Solution de l' Algorithme (Aire\_triangle est fixe)

### Situation



*Remarque :*  $L$  est la longueur et  $l$  la largeur de la partie rectangulaire.  
 $A$  est l'aire fixe de la partie triangulaire  
 $A1$  est l'aire de la partie rectangulaire  
 $A2$  est l'aire totale de la parcelle.

### **ENTREE ou INITIALISATION des données**

*A* Prend\_la\_valeur 15  
Saisir la longueur  
Saisir la largeur

### **TRAITEMENT des DONNEES**

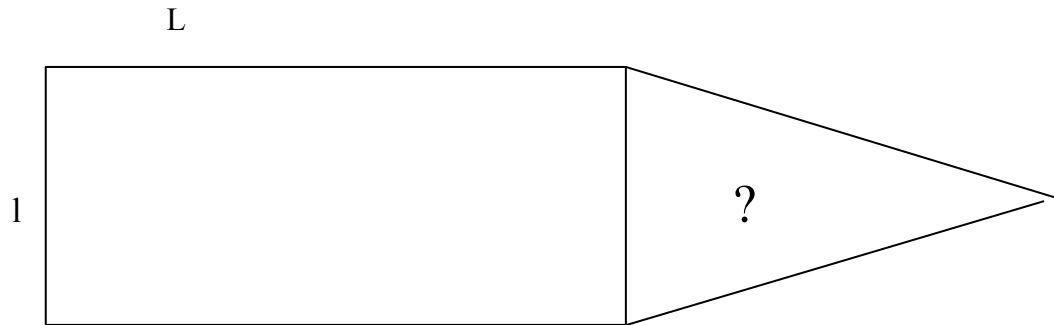
*A1* Prend\_la\_valeur  $L * l$   
*A2* Prend\_la\_valeur  $A + A1$

### **SORTIE**

*Afficher* " L'aire totale est : "  
*Afficher*  $A2$

## Solution de l'Algorithme (Aire\_triangle est variable)

### Situation



*Remarque :*  $L$  est la longueur et  $l$  la largeur de la partie rectangulaire.  
 $A$  est l'aire fixe de la partie triangulaire  
 $A1$  est l'aire de la partie rectangulaire  
 $A2$  est l'aire totale de la parcelle.

### **ENTREE ou INITIALISATION des données**

**Saisir  $A$**   
Saisir la longueur  
Saisir la largeur

### **TRAITEMENT des DONNEES**

$A1$  Prend la valeur  $L * l$   
 $A2$  Prend la valeur  $A + A1$

### **SORTIE**

Afficher " L'aire totale est : "  
Afficher  $A2$

## Solution de l'Algorithme

Début\_algorithme

N est\_un\_nombre // N est le nombre fourni par l'utilisateur

I est\_un\_nombre // I est le nombre pair «immédiat »

S est\_un\_nombre //S contiendra la somme de tous les nombres pairs inférieurs ou égaux à n

Saisir n

S prend\_la\_valeur 0

Tant que N > 1 Faire

    I prend\_la\_valeur N

Si I est pair Alors

        S prend\_la\_valeur S+I

Finsi

    N prend\_la\_valeur N-1

Fin Tant que

Afficher La somme des nombres pairs inférieurs ou égaux à n est :

Afficher S

Fin\_algorithme

Remarque: Sans la partie déclarative notre algorithme est plus difficile à comprendre du premier coup.

### Avec La TI 83 plus

La touche PRGM permet de rentrer dans le mode programmation

EXEC EDIT NOUVEAU

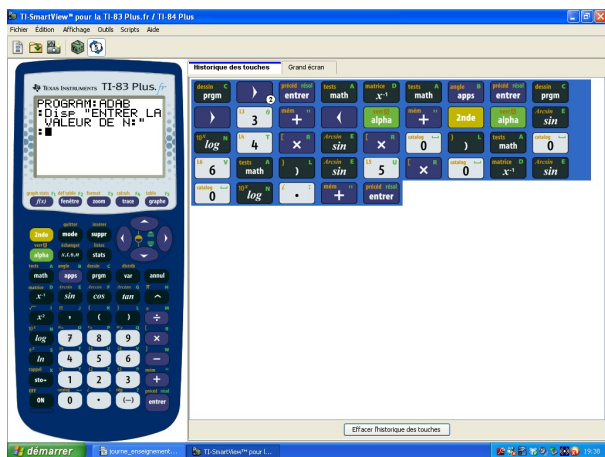
La conversation avec la calculatrice se fait par :

- 7) L'instruction Disp. Ainsi DISP" Bonjour" affichera Bonjour lors de l'exécution
- 8) Input qui permet de recueillir la valeur ou le caractère saisi au clavier.
- 9) Input" Entrer la valeur de n" , n permet d'afficher le texte puis de récupérer la valeur ou le caractère saisi au clavier.
- 10)Le OUTPUT permet de positionner à un endroit précis l'information.

Programme correspondant à l'algorithme de l'exercice 1

Les Touches

Version Simple



```
PROGRAM:ADAB
:DISP "ENTRER LA
VALEUR DE N:"
:INPUT N
:0→S
:While N>1
:N→I
:If P9cd(N,2)=2
```

Programme exécuté pour n=49

Suite et fin du programme



Avec une CASIO

La double apostrophe, par exemple, se trouve dans «OPTN» / «SYBL».

Le « : » permet d'écrire deux instructions sur une même ligne.

Le « ▲ » équivaut à un retour à la ligne avec affichage du résultat. L'opérateur doit appuyer sur « EXE » pour obtenir l'affichage suivant.

