

Programmer en seconde avec la TI-82-Stats

Fiche 1

1. Introduction :

1.1. Le programme officiel :

Le nouveau programme de mathématiques de la classe de seconde applicable à la rentrée 2009 comprend une section d'algorithmique.

Un algorithme est une suite d'instructions permettant de réaliser différentes actions. Les algorithmes sont partout : exemple donné par l'IREM de Lille, groupe AMECMI : *une recette de cuisine est un algorithme, en effet il s'agit de :*

- 1) Réunir les ingrédients.
- 2) Préparer.
- 3) Déguster.

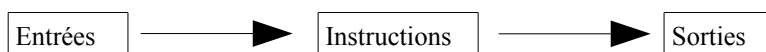
Bien sûr, nous nous intéressons aux algorithmes plus mathématiques ! Ces derniers sont partout.

En citant encore l'IREM de Lille, *sans algorithmes :*

- *La conception assistée par ordinateur n'existerait pas (on ne fabriquerait plus de nouveaux modèles d'avion, de voitures, de lunettes).*
- *On s'améliorerait plus les appareils électroniques, notamment les ordinateurs, les téléphones portables.*
- *On n'aurait jamais envoyé une fusée sur la lune (pas de calculs de trajectoires).*
- *On mourrait plus jeune (faute de traitement des images de scanners par exemple).*

1.2. Principe général :

On peut considérer un algorithme comme une machine avec le schéma suivant :



Bien sûr, il faudra procéder avec rigueur, sans ambiguïté, pour obtenir un résultat. On va donc employer un langage spécifique. Ils sont nombreux, avec chacun sa spécificité (techniques d'écritures en particulier), plus ou moins complexes. Avec les ordinateurs, nous avons opté pour le logiciel "Scratch", ici nous n'avons pas le choix, le langage sera celui de la calculatrice !

Pour faire plus rigoureux, un algorithme est une succession de manœuvres à accomplir, toujours dans le même ordre et de la même façon, manœuvres qui sont en nombre fini et qui s'appliquent à un nombre fini de données.

1.2. Exemples : donnés par l'IREM de Lille, groupe AMECMI

1.2.1. Cercle circonscrit à un triangle ABC

Les données sont : 3 points distincts non alignés

- L'algorithme :
- a) Tracer la médiatrice (D) de AB
 - b) Tracer la médiatrice (D') de AC
 - c) Appeler O le point d'intersection de (D) et (D')
 - d) Tracer le cercle (C) de centre O et de rayon OA

Le résultat est le cercle circonscrit au triangle ABC

1.2.2. Somme S des carrés des entiers de 1 à n

La donnée est n

- L'algorithme :
- a) Lire n
 - b) Poser $S = 0$
 - c) Pour i variant de 1 à n, faire $S = S + i^2$
 - d) Afficher S

Le résultat est le nombre S.

1.2.3. Crible d'Ératosthène

Le crible d'Ératosthène fournit la liste des nombres premiers de 1 à n, pour tout entier $n \geq 3$

La donnée est n

- L'algorithme :
- a) Lire n
 - b) Écrire la liste des nombres entiers de 2 à n.
 - c) Encercler 2, supprimer les autres multiples de 2 figurant dans la liste
 - d) Entourer d'un cercle le plus petit des nombres non encerclés restant, s'il y en a
 - e) Supprimer les autres multiples de ce nombre
 - f) Répéter l'étape d) jusqu'à ce qu'il n'y ait plus dans la liste restante de nombres non encerclés.
 - g) Afficher la liste restante

Le résultat est un tableau de nombres.

1.3. Bien débiter :

Il s'agit de prendre de "bonnes habitudes", dans les trois domaines : entrées, sorties et instructions, ce dernier étant le plus complexe. De façon générale, en ce qui concerne :

1.3.1. Les entrées :

Elles seront renouvelées chaque fois que l'on lancera le programme, d'où leur nom de variables.

Elles sont de trois types :

- Nombre
- Chaîne
- Liste

En début de programmation en seconde, nous n'utiliserons que le premier type.

Plus que jamais, il faut "étudier" le problème avant de commencer à chercher à le résoudre. Ici, il s'agit de déterminer les variables dont on a besoin.

Quand vous commencerez à programmer, veuillez bien noter (sur papier) les noms (éventuellement le type) des variables qui seront utilisées dans le programme.

1.3.2. Les sorties :

Le plus simple. Il peut s'agir d'un dessin, d'un son avec Scratch, mais avec la TI-82-Stats, il s'agira d'un ou plusieurs nombres, éventuellement d'un graphique ou d'un tableau de valeurs.

Chaque langage de programmation a sa syntaxe particulière, y compris pour les instructions de sortie : DIRE; AFFICHER; DISP; Dans la création (papier crayon) d'un programme, nous utiliserons celui qui nous paraît le plus intuitif : ECRIRE.

De même que pour les entrées, notez (sur papier) les noms des éléments qui seront cherchés par le programme (éventuellement la forme attendue).

1.3.3. Les instructions :

Domaine le plus complexe, le plus vaste. Nous l'aborderons petit à petit. Schématiquement, il y a, à notre niveau :

- Les instructions simples : FAIRE _____
- Les instructions conditionnelles¹ : SI _____ ALORS _____ SINON _____ .
- Les boucles : de plusieurs natures :

D'un nombre déterminé d'itérations² : FAIRE _____ POUR _____ ALLANT De _____ A _____
ou encore REPETE ____ FOIS

Ne pas confondre avec récursivité³.

Avec une fin conditionnelle : TANT QUE _____ FAIRE _____ ou FAIRE _____ JUSQU'A _____
ou encore REPETE INDEFINIMENT SI _____

1.3.4. L'affectation :

Les variables sont déclarées, sous leurs propres noms qui sont des identificateurs. A ce stade, nous n'avons fait que réserver un espace mémoire de l'ordinateur ou de la calculatrice. Au restaurant, sur le parking, nous aurions mis une affiche "RESERVE". Mais les places ne sont pas occupées ! Occuper ces places c'est donner une valeur aux espaces mémoires réservés. En programmation, on appelle cela **l'affectation**.

Exemples et exercices page suivante.

1 Elles peuvent s'imbriquer les unes dans les autres (fonctionnalité très utilisée dans les tableurs) ce qui donne beaucoup de "puissance" de calcul mais rend le programme plus complexe à créer ou à lire.

2 En mathématique, une itération désigne l'action de répéter un processus.

3 En informatique et en logique, une fonction ou un algorithme qui contient un appel à lui-même est dit récursif. Très utilisé pour les fractales.

Exemple 1

Déclaration de la variable	V
Affectation	$V \leftarrow 4$
Nouvelle affectation	$V \leftarrow V + 7$
Instruction de sortie	Écrire V
Que lit-on ?	

Exemple 2

Déclaration de la variable	a
Affectation	$a \leftarrow 3$
Affectation	$a \leftarrow 2a + 1$
Affectation	$a \leftarrow a^2$
Affectation	$a \leftarrow a - 4$
Instruction de sortie	Écrire a
Que lit-on ?	

Exemple 3 : Reprenons l'exemple 2 pour une étude générale :

Déclaration de la variable	a	
Affectation	$a \leftarrow x$	
Affectation	$a \leftarrow 2a + 1$	
Affectation	$a \leftarrow a^2$	
Affectation	$a \leftarrow a - 4$	
Instruction de sortie	Écrire a	Développer, réduire l'expression obtenue.
Que lit-on ?		

Exercice 1 : Programme \rightarrow Travail effectué

Application numérique :

Cas général :

Déclarations des variables	a ; b ; c	a ; b ; c	
Affectation	$a \leftarrow 3$	$a \leftarrow x$	
Affectation	$a \leftarrow a^2 - 2a + 1$	$a \leftarrow a^2 - 2a + 1$	
Affectation	$b \leftarrow 7$	$b \leftarrow y$	
Affectation	$b \leftarrow b^2 - 4b + 4$	$b \leftarrow b^2 - 4b + 4$	
Affectation	$c \leftarrow a + b - 9$	$c \leftarrow a + b - 9$	Si $c = 0$, que reconnaît-on ?
Instruction de sortie	Écrire c	Écrire c	
Que lit-on ?		Que lit-on ?	

Exercice 2 : Travail à effectuer \rightarrow Programme

Dans l'exemple 3, vous avez reconnu une fonction. Construire alors les algorithmes qui donnent les fonctions suivantes.

Remarque : le nombre de lignes des tableaux est indicatif, elles ne sont peut-être pas toutes nécessaires (ou bien il peut en manquer !). A vous de voir.

$f_1(x) = (2x-3)^2 - 7$		

$f_2(x) = 2x^2 - 4x + 1$		

$f_3(x) = \sqrt{x-3} - 1$		



$f_4(x) = \frac{2x-1}{x+3} + 9$		

1.3.5. Avec la calculatrice :

Voici ce que donne la programmation avec une TI-82-Stats :

Travail à faire faire : calculer $A = 2x^2 - 4x + 1$ pour des valeurs de x saisies à la calculatrice. Voir f_2

Appuyer sur la touche  Apparaît : 

Curseur sur "NEW" et "Entrée" :  

La calculatrice demande un nom (8 lettres max) en "ALPHA"

Pour le saisir, verrouiller la touche ALPHA.



Le nom étant saisi, avec « Entrée » on a l'écran ci-dessous et

l'invite à saisir la première instruction après les deux points :



Appuyer de nouveau sur la touche "PRGM" : CTL pour les instructions, I/O pour les entrées /sorties et EXEC pour l'exécution. Pour cet exercice nous n'avons besoin que d'une entrée et d'une sortie :

```
PROGRAM:FICHE01
:Input "VARIABLE
",X
:Disp 2*X^2-4*X+
1
:
```

Nom du programme, en ALPHA

Commentaire, en ALPHA, entre deux "

Nom de la variable, ce qui équivaut à une déclaration

Le calcul

Instruction d'affichage

```
CTL  EXEC
 Input
2:Prompt
3:Disp
```

A la fin de la saisie, touche 2^{de} et "Quitte" enregistre automatiquement le programme. Appuyer sur la touche "PRGM" et EXEC cette fois, puis son numéro et le programme démarre.

Une suggestion : dans la mesure où cela n'alourdit pas trop le programme, **insérer le maximum de commentaires dans la liste des instructions** (chaque langage le permet). Cette "astuce" améliore grandement la lisibilité du programme et permet de mieux savoir où l'on est dans son déroulement.

A vos calculatrices !

2. Au travail :

3.1. Distance de deux points dans le plan :

Connaissant les coordonnées de deux points du plan, voici un programme qui donne la distance entre ces deux points. Le nommer DIST2, le saisir et l'exécuter sur votre calculatrice.

```
PROGRAM:EX01
:Input "ABS A ",
X
:Input "ABS B ",
Y
:Input "ORD A ",
Z
:Input "ORD B ",
T
:Disp √((X-Y)²+(Z-T)²)
:
```

Aide à la saisie :

Input et Disp sont des E / S

Nous avons donné des commentaires sur les variables, en mode ALPHA, avec des espaces pour la clarté du texte (mode ALPHA et touche 0). Ne pas oublier les " eux aussi en mode ALPHA touche +.

Attention aux parenthèses dans la formule !

Exécuter plusieurs fois ce programme avec des données différentes.

Vérifier au brouillon avec un repère !

Prolongement : Créer un programme nommé DIST3 qui donne la distance de 2 points de l'espace.

Remarques :

Pour placer plus d'une instruction sur la même ligne, séparez-les par un double point (touche ALPHA au-dessus du point).

Pour insérer une nouvelle ligne de commande, placer le curseur à l'endroit souhaité, puis fonction 2^{de} "Insérer" et "Entrée".

Pour effacer une ligne de commande, tapez "Annul" puis "suppr" pour effacer le double point.

3.2. Distances de sécurité :

Par sécurité, un véhicule doit respecter une distance minimale avec le véhicule qui le précède, afin d'avoir le temps de freiner avant une collision. Ce temps correspond à celui de la perception puis de la réaction du conducteur, ainsi que des possibilités de freinage du véhicule.

Ce temps est fonction de la vitesse du véhicule. Des études statistiques ont montré que cette distance peut être calculée par la formule : $D = 8 + 0,2 v + 0,003 v^2$, où v est en kmh^{-1} et D en mètres.

Créer un algorithme (dans le tableau ci-dessous) qui nous donne cette distance en fonction de la vitesse.

Appeler le professeur pour valider ou invalider votre organigramme

Puis créer le programme (nommé DISTSEC1).

Appeler le professeur pour valider ou invalider votre programme

Une fois validé, recopier les instructions de ce programme dans le tableau ci-dessous.

Organigramme

Programme TI-82-Stats

Prolongement : Lorsque la chaussée est mouillée, il faut augmenter cette distance de 40%. Créer un nouveau programme nommé DISTSEC2 qui tiennent compte de cette situation.

3.3. Pythagore (et oui, encore lui !)

3.3.1. Le théorème :

On veut créer un programme qui calcule la longueur de l'hypoténuse connaissant la longueur des côtés de l'angle droit. Commencer (toujours !) par un organigramme papier.

Appeler le professeur pour valider ou invalider votre organigramme

Puis créer le programme (nommé PYTATH).

Appeler le professeur pour valider ou invalider votre programme

Une fois validé, recopier les instructions dans le tableau ci-dessous.

Organigramme

Programme TI-82-Stats

3.3.2. Le théorème bis :

Créer un programme nommé PYTABIS qui demande la longueur de l'hypoténuse, la longueur d'un des côtés de l'angle droit et qui affiche la longueur de l'autre côté de l'angle droit.

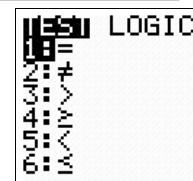
Ce programme doit tester si la longueur de l'hypoténuse est bien supérieure à celle du côté de l'angle droit, votre première instruction conditionnelle. La syntaxe est : SI ____ ALORS ____ SINON ____ .

Pour votre première instruction conditionnelle, nous vous en donnons la structure (le squelette)

Organigramme
Variables :
Si
Test :
Si test oui
Sortie :
Si test non
Sortie
FinSI

Programme TI-82-Stats

Remarque : le(s) symbole(s) $>$, $<$, ... se trouve(nt) dans le menu "Test", fonction 2^{de} touche "math" :



Prolongement : Ces deux programmes peuvent être combinés avec celui de l'activité 3.1.

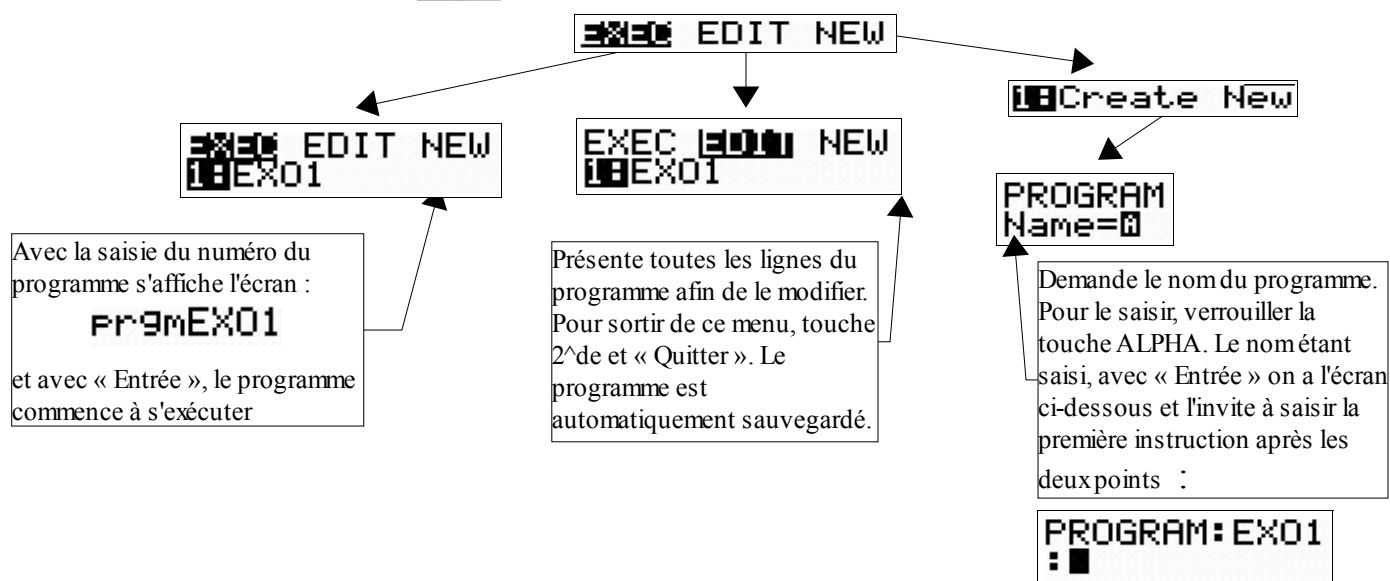
Attention, il y a un piège (mathématique). Lequel ?

--

3. Les commandes de la TI-82-Stats :

3.1. Premières commandes :

A l'appui de la touche  apparaît sur l'écran :



Remarques :

Pour verrouiller la touche ALPHA lors de la saisie de textes, touche 2^{de} et ALPHA :



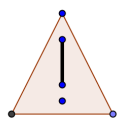
Un nom de programme commence par une lettre et comporte huit caractères.

Un programme ne s'arrête qu'à la fin de ses instructions. Attention aux boucles sans arrêt ! Éteindre alors la calculatrice !

Pour supprimer tout un programme, comme ce dernier est dans la mémoire de la calculatrice, les touches "suppr" ou "annul" sont sans effet. Il faut donc entrer dans la mémoire, touche 2^{de} "mém"




ou "annul" sont sans effet. Il faut donc entrer dans la mémoire, touche 2^{de} "mém"



Puis 2 DELETE et enfin 7 Prgm suivi de son numéro. **MANOEUVRE EXTREMEMENT DANGEREUSE !** Bien suivre les instructions et se concentrer sur ses choix pour ne perdre définitivement que ce que l'on veut !

3.2. Instructions CTL (pour contrôle) :

Le programme étant nommé, un autre appui sur la touche  donne accès à ce menu.

Les instructions ne sont pas à saisir en mode ALPHA mais par le menu de la machine.

1	IF		
2	THEN	SI ____ ALORS ____ SINON ____	Teste une condition
3	ELSE		
4	FOR(POUR ____	Crée une boucle incrémentielle
5	WHILE	TANT QUE ____	Crée une boucle conditionnelle
6	REPEAT	REPETE	Crée une boucle conditionnelle
7	END	FIN	Fin de boucle If-Then-Else
8	PAUSE	PAUSE	Suspend l'exécution du programme
9	Lbl		Définit une étiquette
0	GOTO	ALLER A	Aller à une étiquette
A	IS<	(variable,valeur)	Incréméte et omet si variable plus grande que valeur
B	DS<	(variable,valeur)	Incréméte et omet si variable plus petite que valeur
C	MENU		Définit un menu et contrôle les branchements
D	PRGM		Appelle et exécute un autre programme (récursivité)

E	RETURN	RETOUR	revient d'un sous-programme ou sous-routine ⁴ .
F	STOP	ARRET	Met fin à l'exécution demandée
G	EFF VAR		Suppression de variables
H	GraphStyle(Choisi le style du graphe

Seule une partie de ces instructions sera employée pour débiter en programmation en classe de seconde.

Remarques :

Dans un programme, la touche 2^{de} suivie de "Insérer" crée une nouvelle ligne pour entrer de nouvelles instructions.

3.3. Instructions E/S (pour Entrées - Sorties) :

1	Input	Entrer une valeur ou utiliser le curseur libre pour parcourir le graphe affiché
2	Prompt	Demande l'introduction des valeurs des variables
3	Disp	Affiche un texte, une valeur ou l'écran initial si Disp n'est suivi de rien
4	AffGraph	Affiche le graphe courant
5	AffTable	Affiche une table
6	Output(Affiche un texte à l'emplacement spécifié
7	codeTouche	Détecte la frappe d'une touche au clavier
8	EffEcr	Efface l'affichage
9	EffTable	Efface la table courante
0	CaptVar	Imprime l'écran
A	Capt(Obtient une variable d'une autre TI-82-Stats
B	Envoi(Envoie une variable à une autre TI-82-Stats

4 En informatique, une routine est une séquence d'instructions réalisant une fonction particulière.