I) Les instructions

Une calculatrice est une machine qui exécute les ordres qu'on lui demande !

En langage informatique, un ordre s'appelle **une instruction**.

Quand on tape une instruction sur le clavier de la machine, on la termine par la touche Exe ou Enter, et la machine exécute l'instruction qu'on lui a demandé.

Dans toute la suite, cette touche sera remplacée par le symbole ↵

Voici un exemple d'instruction $2 + 3$ ↵ la machine exécute l'instruction et affiche 5

II) Les variables

L'intérieur d'une calculatrice est formé de deux parties.

Une mémoire appelée RAM (Random Access Memory) qui est composée de cases mémoire où l'utilisateur a accès (aléatoirement) et peut écrire des nombres.

Une unité centrale, qui exécute des calculs en utilisant les nombres stockés dans les cases mémoire. (l'unité centrale a besoin aussi de mémoire pour effectuer ses calculs. Elle s'appelle **la mémoire interne ou ROM** (Read Only Memory)
Comme son nom l'indique, l'utilisateur peut éventuellement lire le contenu de ses cases mémoire, mais en aucun cas ne peut écrire dedans)

Pour écrire un nombre (par exemple 3) dans la RAM

il suffit de donner un nom (par exemple A) à une case de mémoire.

(la calculatrice se charge de trouver un endroit libre dans sa RAM qu'elle nommera A)
et de donner la valeur 3 à cette case mémoire A .

(chaque fois que l'utilisateur demandera un calcul avec la lettre A , la calculatrice remplacera A par le contenu de la case mémoire correspondante (ici par 3))

A s'appelle une **variable (réelle)**. En langage informatique, donner la valeur 3 à A s'écrit A prend pour valeur 3 , en abrégé on écrit **A ppv 3**

Rem : Certains noms de variable ne peuvent pas être utilisés, car le système les utilise.

(Ans, Xmin, TblStart, Y1, abs , etc) . Utiliser A , B , ... , Z est sans risque.

Traduction en langage des calculatrices : A ppv 3 s'écrit $3 \rightarrow A$

(la flèche est obtenue avec la touche STO▶ STOre en anglais signifiant stocker
cette flèche est très parlante, elle indique qu'on met (ou stocke) 3 dans la variable A)

I) Définition d'un programme

Un **programme** est une **liste d'instructions** écrites dans un **langage de programmation exécutable** par une calculatrice programmable. (ou un ordinateur)

Une instruction est un ordre (ou une phrase impérative) qui effectue une action précise.
(Un programme s'appelle aussi un **logiciel**)

Une calculatrice programmable TI ne comprend qu'un seul langage de programmation.
(Un ordinateur peut comprendre plusieurs langages de programmation, certains étant très évolués et très proches de notre langage français)

Les langages de programmation sont généralement écrits en anglais.

Voici les traductions de quelques instructions des calculatrices TI

Entrer	Afficher	Si ... Alors ... Sinon	Fin	Etiquette	Aller à	Tant que	Pour
Input	Disp	If ... Then ... Else	End	Lbl	Goto	While	For

II) Comment écrire un programme ?

Pour écrire un programme, il suffit d'appuyer sur la touche **PRGM**
(abréviation de PRoGraM) , puis de sélectionner NEW (nouveau programme)

La calculatrice nous demande alors un nom pour notre programme.
(donnez un nom qui reflète ce que fait votre programme, par exemple EQUA2)

Sur l'écran apparaît alors PROGRAM : EQUA2

Il ne nous reste plus qu'à taper la succession d'instructions qui composent notre programme,
chaque instruction se terminant par ↵

Pour bien se repérer, la calculatrice écrira automatiquement : au début de chaque instruction.

Tout ce que l'on va écrire restera bien sur dans la calculatrice même si on l'éteint.

Si on veut alors voir ou modifier le programme déjà écrit, il suffira de faire **PRGM**
puis EDIT puis de sélectionner le nom du programme.
(**éditer** un programme, c'est voir la liste de toutes les instructions qui le composent)

III) Comment exécuter un programme ?

Lorsque notre programme sera écrit, il faudra bien demander à la calculatrice de l'**exécuter** !
Pour cela, il suffira de faire **PRGM** puis EXEC puis de sélectionner le nom du programme.
La calculatrice affiche alors prgmEQUA2 . Il suffira alors d'appuyer sur ↵ pour démarrer l'exécution du programme.

La calculatrice exécute dans l'ordre la 1^{ère} instruction, puis la 2^{ième} ... jusqu'à la dernière.

IV) Comment effacer un programme ?

Pour effacer le programme TOTO , touche **MEM** ,
puis Mem Mgtm/Del (ou Delete),
puis Prgm puis sélectionner le programme TOTO et ↵

➤ Pour écrire un programme, il faut d'abord l'écrire en français.
La liste des instructions écrites en français s'appelle un **algorithme**.

Il ne nous reste plus qu'à traduire cet **algorithme** (ou cet **organigramme**) en un **programme**, dans le **langage de programmation exécutable** par notre calculatrice TI.

➤ Quand on écrit un programme, pour trouver la liste des instructions, touche PRGM

I) L'instruction d'affectation → (ppv)

Nous l'avons déjà vu. Ex 3 → A cad A ppv 3
On dit qu'on affecte la valeur 3 à la variable A

II) L'instruction d'affichage **Disp** (Display veut dire Afficher)

Cette instruction se trouve dans le menu I/O (Input / Output) ou E/S (Entrée / Sortie)

☞ Ecrire et exécuter les programmes suivants

```
: 3 → A
: Disp 2A
: Disp 3A
```

Disp 2A La calculatrice affiche 2 fois le contenu de la variable A (6)
Disp 3A La calculatrice affiche 3 fois le contenu de la variable A (9)

```
: 3 → A
: Disp "2A"
: Disp "3A"
```

Disp "texte" La calculatrice affiche ce qui est écrit dans les guillemets
(essayer avec Disp " IL FAIT BEAU ")

```
: 3 → A
: Disp IL FAIT BEAU
```

La calculatrice affiche un message d'erreur car elle ne connaît pas la variable IL FAIT BEAU

```
: 3 → A
: Disp "2A =", 2A
: Disp "3A =", 3A
```

Cette instruction est très intéressante car la calculatrice affiche
2A = 6 puis 3A = 9

```
: 3 → A
: 2A
: 3A
```

On voit sur cet exemple qu'il est nécessaire dans un programme d'utiliser l'instruction Disp pour afficher les calculs intermédiaires sinon la calculatrice n'affiche que le dernier calcul demandé (sur certains modèles de TI, la machine n'affiche même rien du tout)

III) L'instruction Pause

Cette instruction se trouve dans le menu CTL (en anglais ConTroL)

A l'exécution, la calculatrice travaille vite ! Quand elle a fini, elle écrit Done (fait)
Si on veut l'arrêter pendant l'exécution d'un programme (par exemple pour lire un calcul intermédiaire), il suffit de rajouter l'instruction **Pause** dans le programme. A l'exécution, pour finir la pause de la calculatrice, et lui demander de continuer, on appuie sur ↵

IV) L'instruction d'entrée Input (Entrer)

Cette instruction se trouve dans le menu I/O (ou E/S)

☞ Ecrire et exécuter les programmes suivants

<pre>: Input A : Disp "2A"</pre>

A l'exécution, un ? apparaît. La calculatrice attend qu'on lui entre un nombre. Entrer un nombre suivi de ↵ La machine affiche 2A

<pre>: Input A : Disp 2A</pre>

A l'exécution, un ? apparaît. Entrer un nombre.
La machine affiche le double du nombre entré.

<pre>: Input "B =", A : Disp "2B =", 2A</pre>

A l'exécution, ? est remplacé par B =
Entrer un nombre. (par exemple 5) La machine affiche 2B = 10
Le nombre entré est stocké dans la variable A, bien qu'on le nomme B

V) L'instruction d'effacement d'écran ClrHome (ou EffEcr)

ClrHome ou Effacer l'écran. Cette instruction se trouve dans le menu I/O (ou E/S)
Cette instruction peut être utilisée par exemple au début d'un programme pour effacer l'écran.

VI) L'instruction d'arrêt Stop

Cette instruction se trouve dans le menu CTL
L'instruction **Stop** arrête définitivement l'exécution du programme.
(Il n'est pas nécessaire de la rajouter en fin de programme)

VII) L'instruction conditionnelle (ou test) If ... Then ... End

Elle est composée de trois mots **If ... Then ... End** (Si ... Alors ... Fin)

Cette instruction se trouve dans le menu CTL

La syntaxe est la suivante

```

: If condition
: Then
: instruction
: etc
: instruction
: End

```

} La calculatrice exécutera ces instructions
} seulement si la condition est vraie.
} (Si la condition est fausse, elle ne les exécutera pas)

On peut rajouter un **Else** (Sinon) à l'instruction conditionnelle..

```

: If condition
: Then
: instruction
: etc
: instruction
: Else
: instruction
: etc
: instruction
: End

```

} La calculatrice exécutera ces instructions
} seulement si la condition est vraie.
} (Si la condition est fausse, elle ne les exécutera pas)

} La calculatrice exécutera ces instructions
} seulement si la condition est fausse.
} (Si la condition est vraie, elle ne les exécutera pas)

Les variables Y

On les utilise pour travailler avec des fonctions.

☞ Appuyer sur la touche Y= et écrire la formule $Y_1 = 2X - 1$

On peut alors utiliser dans un programme la variable Y1

X est une variable de la RAM, cad qu'on peut lire et modifier son contenu.

Si on écrit $2 \rightarrow X \downarrow$ la variable X aura pour valeur 2

Si on écrit `Disp X` la calculatrice affiche le contenu de la variable X (cad 2)

Y1 est une variable de la ROM, cad qu'on peut lire (mais pas modifier) son contenu.

Si on écrit $2 \rightarrow Y_1$, la calculatrice répond ERROR.

Si on écrit `Disp Y1` $\downarrow$, elle affiche l'image du contenu de la variable X par *f* (cad 3). Chaque fois que le contenu de la variable X est modifié, la calculatrice modifie immédiatement le contenu de la variable Y1. Essayons pour voir ...

Mais on peut écrire $Y_1 \rightarrow A \downarrow$ (cela revient à recopier le contenu de Y1 dans A)

➔ Pour écrire Y1 touche Y-Vars (ou touche Vars puis Y-Vars)
 (On peut utiliser un grand nombre de variables Y (Y1, Y2, etc))

Une **boucle** est une instruction ou un groupe d'instructions que la calculatrice va, en exécutant le programme, répéter un certain nombre de fois.

1) Une boucle peut être construite avec les instructions Lbl et Goto

A l'exécution d'un programme, la calculatrice exécute toutes les instructions dans l'ordre, de la première à la dernière

Mais, il existe une instruction qui retourne en arrière.
Cette instruction, c'est **Goto** (Aller à)
Elle s'utilise avec **Lbl** (LaBeL veut dire Etiquette)

Voici ce qu'il faut écrire pour demander d'entrer un nombre non nul (pour faire recommencer si jamais on entre le nombre 0)

```
: Lbl E
: Input "a =", A
: If A = 0
: Then
: Goto E
: End
```

2) Une boucle peut aussi être construite avec l'instruction While

```
: While condition
: instruction
: etc
: instruction
: End
```

La calculatrice exécutera ces instructions tant que la condition est vraie.
(Traduction **Tant que condition Faire instructions**)

3) Une boucle peut aussi être construite avec l'instruction For

```
: For ( variable , départ , arrivée , incrément )
: instruction
: etc
: instruction
: End
```

La *variable* est initialisée à *départ*, et est incrémentée de la valeur de *incrément* à chaque passage dans la boucle. La calculatrice exécutera ces instructions tant que la valeur de la variable ne dépasse pas *arrivée*
(Traduction **Pour variable = départ jusqu'à arrivée par pas de incrément Faire instructions**)

Résumé Les programmes suivants sont équivalents

```
: 3 → A
: Lbl B
: If A ≤ 7
: Then
: instructions
: A + 2 → A
: Goto B
: End
```

```
: 3 → A
: While A ≤ 7
: instructions
: A + 2 → A
: End
```

```
: For ( A , 3 , 7 , 2 )
: instructions
: End
```

➤ Pour insérer une instruction entre deux instructions déjà écrites, se placer au début de la deuxième instruction et taper INS (Shift DEL) suivi de ↵
INS veut dire INSert (insérer) et DEL veut dire DELete (effacer)

➤ Une **commande** est une suite d'instructions séparées par : et terminée par ↵

Au lieu d'écrire : Input "a=", A
: Disp "2a =", 2A on peut écrire une commande : Input "a =", A : Disp "2a =", 2A

Cela a un avantage. Le programme écrit est plus court.

Cela a un inconvénient. Le programme devient difficile à lire si tout est collé.

➤ On ne peut pas modifier le nom d'un programme.

➤ Nous avons appris dans ces feuilles à **programmer une calculatrice**.

Lorsque nous demandons à une calculatrice de tracer une courbe ou de nous montrer un tableau de valeurs, elle ne fait qu'exécuter des programmes déjà écrits par le fabriquant

➤ Comment tester si un nombre est entier ?

Un nombre entier est égal à sa partie entière. La partie entière d'un nombre p est notée $\text{Int}(p)$

(essayons dans le mode de calcul normal $\text{Int}(2,35)$ ↵ la calculatrice répond 2 etc)

Pour trouver Int , touche MATHS sous-menu NUM

➤ Les booléens (et, ou, non)

Juste après Input "p =", P on peut

tester si p est un nombre strictement négatif **ou** un nombre non entier, et dans ce cas afficher " p doit être un entier naturel voyons ... " et faire recommencer la saisie de p

Cela s'écrit **IF** $P < 0$ **or** $\text{Int}(P) \neq P$

(on pourrait aussi tester la condition contraire, cad si p est un nombre positif **et** un entier et dans ce cas aller à la suite du programme, sinon afficher " p doit être un entier naturel " et faire recommencer la saisie de p

Cela s'écrit **IF** $P \geq 0$ **and** $\text{Int}(P) = P$)

or (ou), **and** (et) sont des opérateurs booléens. Touche TEST puis LOGIC