

Quelques rappels

Quelques rappels

INTRODUCTION

L'ordinateur est un outil parfaitement incapable d'entreprendre quoi que ce soit tout seul. Dépourvu de tout sens d'initiative, il ne fait rien sans qu'on lui ait indiqué ce qu'il doit faire.

Vous devez donc le programmer.

Un programme est une suite d'instructions que l'ordinateur va suivre et exécuter pas à pas. Il est écrit dans un langage "compréhensible" par l'ordinateur, par exemple le BASIC.

Pour mieux comprendre ce que représente un programme, prenons comme exemple un mode d'emploi concernant une "recette grand-mère".

RECETTE : combien de cuillères de soupe un enfant doit-il manger quotidiennement pour grandir normalement ?

- demander l'âge de l'enfant (soit A son âge)
- demander la taille de l'enfant en cm
(soit T sa taille en cm)
- calculer $Q = A + T + 8$

Q est la quantité de cuillères de soupe à prendre quotidiennement

Exemple : Si l'enfant a 4 ans et mesure 120 centimètres, il lui faudra manger $4 + 120 + 8$, soit 132 cuillères de soupe.

Remarque :

(Il est déconseillé de mettre cette recette en pratique !)

Un mode d'emploi et un programme informatique sont parfaitement comparables ; en effet, dans les deux cas, une explication détaillée est nécessaire pour que le travail puisse s'effectuer correctement.

Il vous est certainement déjà arrivé au moins une fois d'avoir recours à un mode d'emploi pour effectuer une tâche. L'ordinateur travaille de la même manière : le programme lui explique dans les moindres détails ce qu'il doit faire.

La plupart des informations traitées se trouvent sous forme de variables. Une variable est une quantité susceptible de prendre diverses valeurs.

Ainsi dans la recette, Q la quantité de cuillères de soupe est une variable car elle varie en fonction de l'âge A et de la taille T qui sont également des variables ; tandis que dans la formule $Q = A + T + 8$, 8 est une constante.

Autre exemple :

Dans le calcul D (diamètre) = 2 fois R (rayon)
 D et R sont des variables
 2 est une constante (car cette valeur ne change jamais)

A chaque variable on donne un nom ; dans l'exemple précédent,
 nous avons appelé A l'âge de l'enfant, T sa taille et
 Q le nombre de cuillères de soupe.

Quels noms peut-on donner à une variable ?

Cela dépend de l'ordinateur qu'on utilise ;
 diverses règles existent :

- tout d'abord, dans la plupart des cas le nom donné à une variable doit obligatoirement commencer par une lettre de l'alphabet !

Dans ce cas :

A	}	est correct	3B	}	est incorrect
TVA			1 AGE		
Z3IB			-4BC		
AGE					

- la longueur du nom est souvent limitée à un certain nombre de caractères (parfois 2, seulement)

Dans ce cas :

AX	}	est correct	AGE	}	est incorrect
C3			A-3B		
XG			TVA		
CC			TAILLE		
B			A+TV		

REMARQUE :

Sur certains ordinateurs, vous pouvez donner à vos variables un nom de longueur quelconque, mais il ne tient compte que des 2 premiers caractères.

Dans ce cas :

FUSEE	}	sont des noms corrects
FUMEE		

mais il ne ferait aucune différence entre les 2, car elles commencent par les mêmes caractères FU, et utiliseront la même case mémoire pour FUSEE et FUMEE !

- Il est interdit de donner à un variable un nom réservé à l'ordinateur.

Exemple : une variable ne pourrait pas s'appeler INPUT ou PRINT ou END, ...
L'ordinateur risquerait de confondre le nom de la variable avec les instructions !

- Un nom de variable doit être en un seul mot, c'est-à-dire qu'il ne doit pas contenir d'espace. Sur certains ordinateurs on peut utiliser le signe _ (soulignement).

Dans ce cas :

VALEUR_AJOUTEE	}	est correct
VALEUR_AJOUTEE		
PRIX TOTAL		est incorrect

Dans le cas du TPC-8300, le nom d'une variable peut avoir n'importe quelle longueur mais on est quand même limité par la longueur maximum d'une ligne d'instruction qui est de 255 caractères. Seuls les deux premiers caractères permettent de distinguer deux variables commençant de la même façon.

REALISONS UN PETIT PROGRAMME

Nous allons prendre l'exemple que nous avons déjà étudié dans le 1er tome :

```
1 INPUT A
2 INPUT B
3 X=A+B
4 PRINT X
5 END
```

Rappelons qu'un programme n'est rien d'autre qu'une suite d'instructions détaillées (une explication théorique) permettant à l'ordinateur d'exécuter un travail.

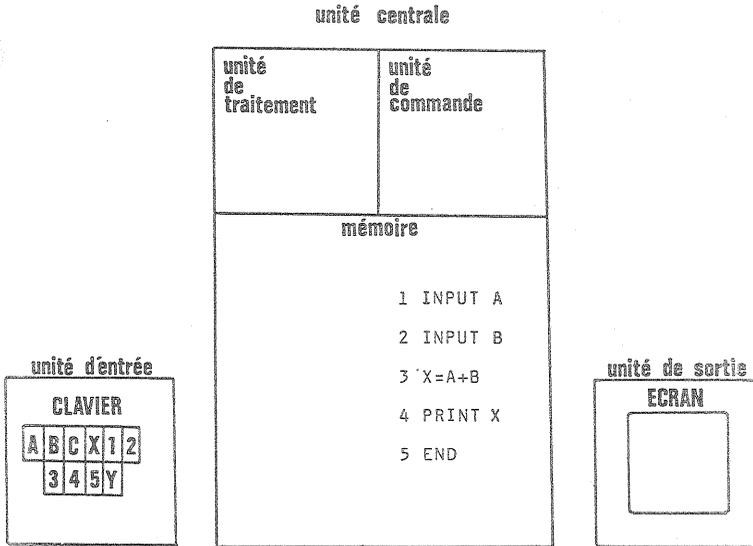
Notre programme se compose de 5 instructions :

2 instructions de lecture, 1 instruction de calcul, 1 instruction d'affichage et l'instruction END qui arrête le traitement.

Ce programme utilise 3 variables : 2 variables (A et B) qui sont lues dans les instructions 1 et 2 et 1 variable (X) qui est calculée dans l'instruction 3.

INTRODUCTION DU PROGRAMME
 =====

(le programme est placé dans la mémoire de l'ordinateur)

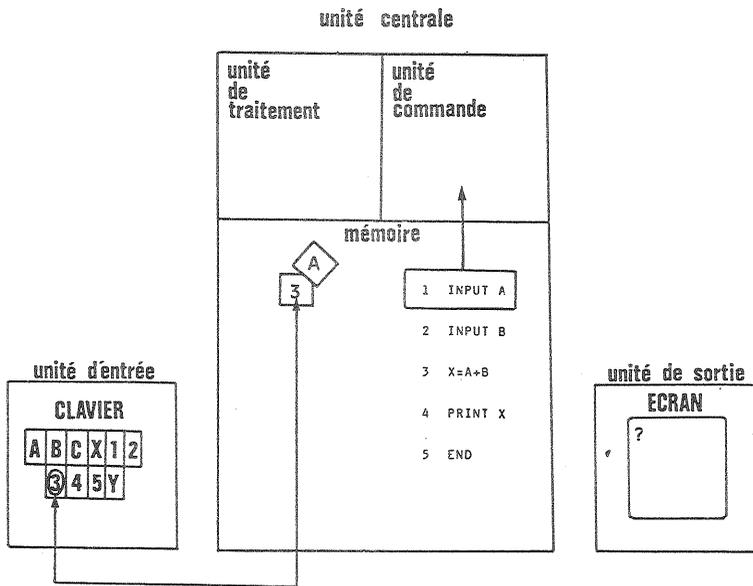


Grâce au clavier, on tape le programme caractère après caractère, instruction après instruction jusqu'à ce que le programme se trouve en mémoire.

EXECUTION DU PROGRAMME
 =====

(Après lui avoir donné l'ordre d'exécution, l'ordinateur exécute une instruction après l'autre du programme stocké dans sa mémoire).

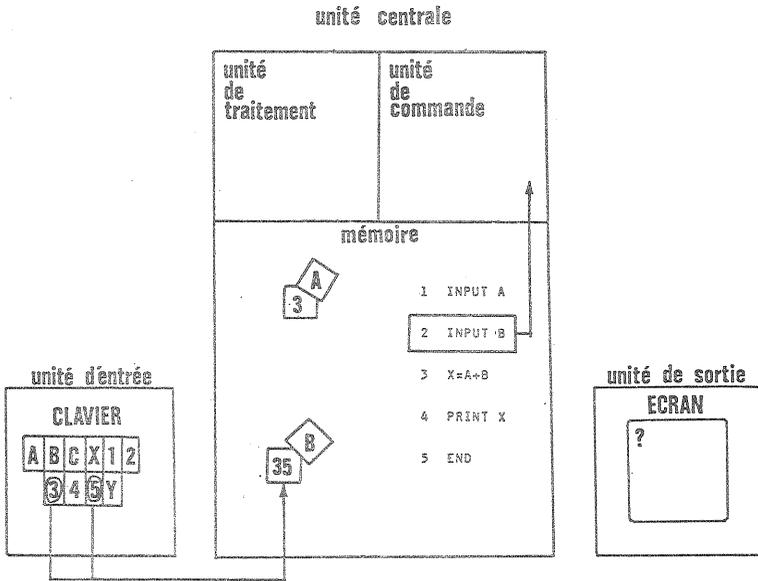
PREMIERE ETAPE (1ère instruction)



La première instruction est envoyée dans l'unité de commande. L'unité de commande a pour rôle d'analyser l'instruction et de l'exécuter ; c'est-à-dire d'envoyer dans les différents organes les impulsions électriques nécessaires :

- un ? s'affiche sur l'écran afin de vous signaler que l'ordinateur désire connaître la valeur de la variable A
- grâce au clavier, vous introduisez la valeur que vous désirez (le chiffre 3 par exemple)
- la valeur tapée au clavier sera envoyée dans la mémoire et stockée dans une case ; l'ordinateur "collera" sur cette case une étiquette qui porte le nom de la variable (A dans notre cas) afin de pouvoir retrouver cette variable par la suite

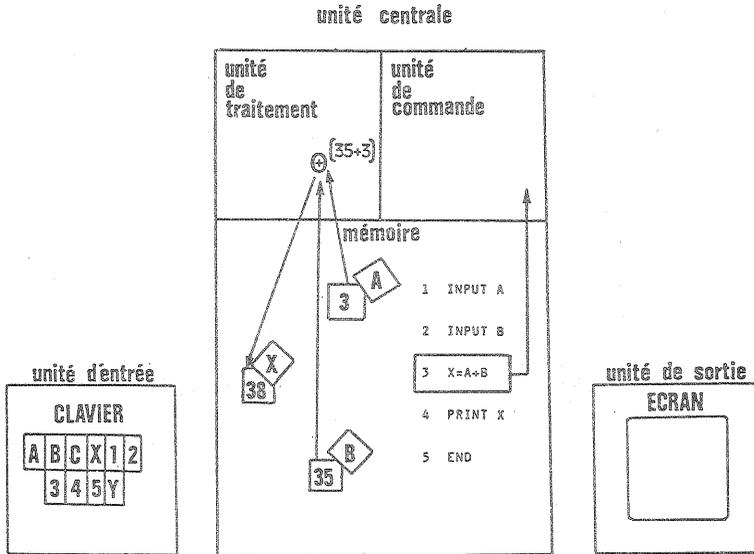
DEUXIEME ETAPE (2ème instruction)



Le processus est identique au précédent :

la valeur tapée au clavier (le nombre 35 par exemple)
est stockée en mémoire.

TROISIEME ETAPE (3ème instruction)



La troisième instruction indique (à droite du signe =) l'opération de calcul à effectuer et (à gauche du signe =) le nom qu'on a donné au résultat.

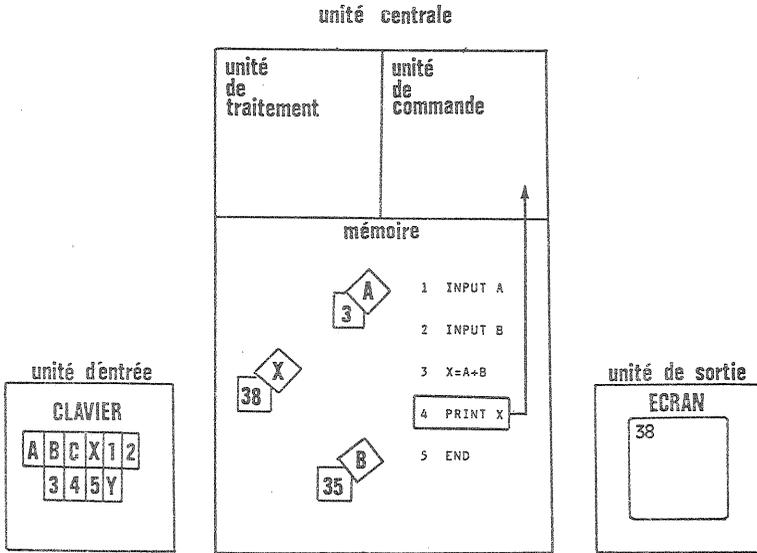
L'unité de commande recherche alors en mémoire la variable A (heureusement qu'une étiquette marquée A lui indique l'endroit où se trouve cette variable ; la mémoire est tellement grande !) et envoie son contenu dans l'unité de traitement.

De même le contenu de la variable B est également envoyé dans l'unité de traitement.

L'unité de traitement effectue alors l'opération arithmétique précisée dans l'instruction (une addition dans notre cas) 3 + 35 fait 38

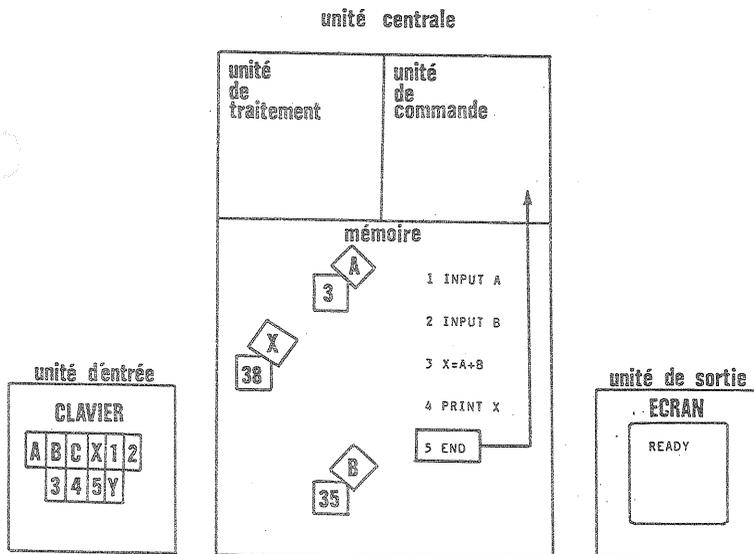
Le résultat est stocké en mémoire (également repéré par une petite étiquette : X)

QUATRIEME ETAPE (4ème instruction)



L'ordinateur recherche en mémoire la variable X et affiche son contenu sur l'écran.

CINQUIEME ETAPE (5eme instruction)



L'instruction END indique à l'ordinateur que le traitement est fini ; il affiche alors sur l'écran un message (READY par exemple qui veut dire PRET. Prêt pour faire autre chose...)

REMARQUES :
=====

- 1) Il est possible de mettre 2 ou plusieurs instructions sur une même ligne, en séparant ces instructions par le signe : (double point)

Le programme 1 INPUT A
 2 INPUT B
 3 X=A+B
 4 PRINT X
 5 END

pourrait s'écrire : 1 INPUT A:INPUT B
 2 X=A+B
 3 PRINT X
 4 END

ou encore : 1 INPUT A
 2 INPUT B:X=A+B:PRINT X
 3 END

par exemple

Cette méthode peut être intéressante dans certains cas (nous le verrons ultérieurement) mais présente également des inconvénients. Ne pas l'utiliser pour le moment.

- 2) Il est préférable de ne pas numéroter les instructions : 1, 2, 3, 4, ..., mais de les numéroter de dix en dix, par exemple. Cela permettra éventuellement d'intercaler de nouvelles instructions en cas d'oubli et vous évitera ainsi de retaper tout le programme.

Il est bien entendu qu'à force d'intercaler de nouvelles instructions, on sera bloqué tôt ou tard ; mais si autant de modifications sont nécessaires, c'est que le programme de base ne doit pas être très au point.

Notre programme devient donc :

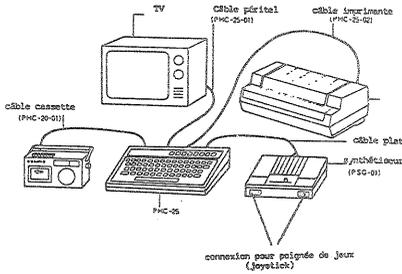
10 INPUT A
20 INPUT B
30 X=A+B
40 PRINT X
50 END

Remarque : le zéro est barré (\emptyset) pour ne pas le confondre
avec la lettre O

De la théorie à la pratique

DEBALLER LE MATERIEL

PRESENTATION DES DIFFERENTES UNITES =====



le micro-ordinateur PHC-25 se compose d'une unité d'entrée (un clavier), d'une unité de sortie (poste TV ou moniteur) et d'une unité centrale qui se trouve sous le clavier.

On peut lui raccorder une unité de cassette, une imprimante et un synthétiseur musical avec 2 joysticks.

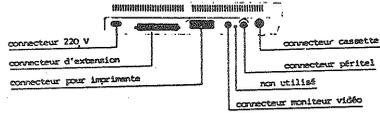
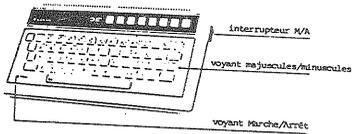
PRECAUTIONS D'UTILISATION =====

- Assurez-vous que vous utilisez bien un magnétophone SANYO TRC 1550 ou équivalent
- La vitesse de défilement doit être constante ; il est donc recommandé d'utiliser un branchement sur secteur 220 V (par l'intermédiaire de l'adaptateur/chargeur 6 volts) ; l'inconvénient de l'utilisation de piles sèches est que la vitesse varie avec l'état des piles
- Aux moindres symptômes anormaux, débranchez votre matériel et contactez votre revendeur
- Eteignez toujours votre matériel lorsque vous manipulez (branchement ou débranchement) les différents câbles de liaison



Evitez les chocs, les endroits exposés aux rayons du soleil.
Evitez de verser du liquide sur l'appareil.
Ne jamais démonter votre micro-ordinateur.

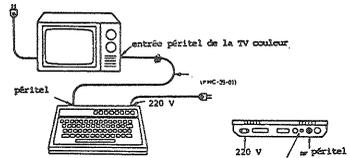
BRANCHEMENT DU MATERIEL



clavier (vue avant)

clavier (vue arrière)

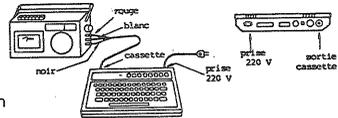
- . Connectez le poste TV ou le moniteur
- . Connectez le câble d'alimentation 220 V sur l'unité centrale
- . Branchez également vos différents périphériques



1) Connection du magnétophone

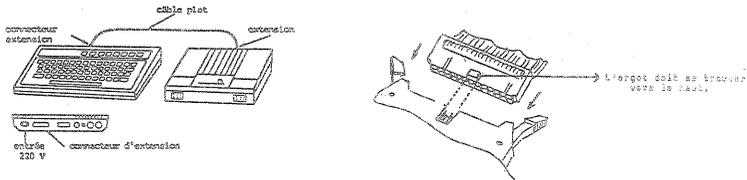
Branchement du magnétophone :

- Vérifiez que votre câble de connection est bien branché :
 - . borne rouge sur la position MIC (microphone)
 - . borne blanche sur la position EAR (écouteur)
 - . borne noire sur la position REM (commande à distance), sauf dans le cas où votre magnétophone ne possède pas de commande à distance



2) Connection du synthétiseur musical

Il existe un synthétiseur musical (PSG 10) qui peut être connecté au bus d'extension prévu à l'arrière de l'appareil.



3) Connection du joystick

(manche à balai)

On peut connecter jusqu'à 2 joysticks.
On connecte le câble du joystick sur le synthétiseur.

4) Connection de votre imprimante

A connecter sur le connecteur imprimante prévu à l'arrière de l'appareil.

MISE EN MARCHÉ

- . branchez votre téléviseur ou le moniteur au 220 V
- . branchez l'unité centrale au 220 V
- . allumez votre téléviseur ou moniteur
- . allumez votre micro-ordinateur (interrupteur à droite du clavier) ; la lumière POWER doit s'allumer ainsi que la lumière du synthétiseur s'il est branché

le message `Screen ?` doit s'afficher

"Screen" veut dire "écran" ; vous devez taper 1 ou 2 ; nous verrons ultérieurement ce que signifie ce code

- . tapez le chiffre 2 et apparaîtra alors sur l'écran le message suivant :

```
PHC-25 SANYO BASIC V.1.3
User Memory 8121 Bytes
Ready
```

Ce qui veut dire :

```
Je me présente :
PHC-25 de SANYO, on peut me parler en BASIC Version 1.3
mémoire disponible : 8121 octets
prêt pour vous servir
```

Remarque : si vous aviez tapé 1 au lieu de 2 à la question "SCREEN ?" il vous aurait affiché : 14265 Bytes

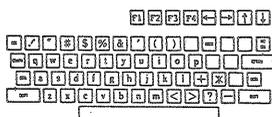
- . vous pouvez faire le réglage de luminosité et de contraste sur votre poste de télé ou sur le moniteur
- . pour éteindre votre micro-ordinateur, respectez la règle suivante :
 - éteindre d'abord le poste de télé ou moniteur, puis le micro-ordinateur, attendre au moins 3 secondes avant de rallumer votre micro-ordinateur

PRESENTATION DU CLAVIER

Le clavier est de type QWERTY, car les premières lettres sont Q W E R T Y, en opposition avec le clavier de type français (AZERTY)

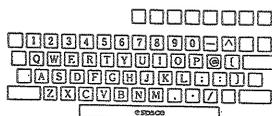
- Appuyez sur la touche A, le caractère a apparaît à l'écran ; on peut donc afficher toutes les lettres de l'alphabet en minuscule et les signes au bas des touches

Voir schéma ci-dessous :



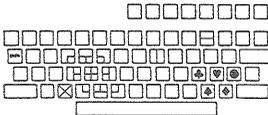
- Appuyez sur la touche SHIFT (une des deux) et tout en laissant le doigt sur cette touche, appuyez sur la touche A ; le caractère A apparaît à l'écran ; on peut donc afficher toutes les lettres de l'alphabet en majuscule et les chiffres et les signes au haut des touches

Voir schéma ci-dessous :



- la longue touche centrale au bas du clavier appelée "barre d'espacement" ou "espace" ou parfois "blanc" ou "space" permet comme vous l'avez sûrement deviné de taper des espaces

- On appelle curseur, le carré de lumière qui clignote sur l'écran pour vous indiquer votre position
- Lorsque vous appuyez sur le touche **GRAPH** et en même temps sur une touche figurant sur le schéma ci-dessous, vous obtiendrez un caractère graphique :



- En laissant votre doigt sur une touche quelconque plus que 1 seconde, le caractère appuyé se répétera automatiquement sur l'écran
- Tapez la lettre O et le chiffre zéro ; vous remarquerez que le chiffre 0 est barré (Ø) pour ne pas le confondre avec la lettre O

- Afin d'éviter d'appuyer constamment sur la touche **SHIFT** (si l'on veut taper un texte en majuscule) il suffit d'appuyer sur la touche **LOCK** (le voyant rouge s'allume) ; vous êtes alors en mode MAJUSCULE ; pour se remettre en mode minuscule, appuyez à nouveau sur la touche **LOCK** (le voyant s'éteint)

- les 4 touches



vous permettent de vous promener sur l'écran (déplacement du curseur)

entraînez-vous un peu à manipuler le clavier avant de passer à la suite

MANIPULATIONS

Attention !

Avant d'aborder ce chapitre, éteignez votre micro-ordinateur afin d'effacer la mémoire ; en effet, vous auriez pu enregistrer involontairement en mémoire des ordres qui perturberaient les expériences que nous allons aborder dans le présent chapitre

- rallumez votre micro-ordinateur, répondez 1 ou 2 à la question "SCREEN ? " ; et nous voilà prêt pour entamer un dialogue sérieux avec notre micro-ordinateur.

Avant de dialoguer avec notre ordinateur, souhaitons lui le bonjour.

Nous allons d'abord faire une faute de frappe.

Tapez le mot BOMJOUR en appuyant successivement sur les

touches B O M J O U R

Pour corriger cette erreur de frappe, tapez sur la touche

INS
DEL

(en haut à droite) ; le curseur (petit carré qui clignote) reviendra en arrière et effacera un caractère après l'autre.

Retapez alors B O N J O U R correctement

Que répond la machine ?

Rien pour l'instant, car il faut lui signaler que nous avons terminé de taper le message.

Pour cela, il faut presser sur la touche

RETURN

Que répond-il maintenant ?

Un gros mot !

Syntax error

Cet ordinateur n'a pas l'air très poli !

Essayons de le faire calculer :

Tapez : C A L C U L E Z = 2 ± 3

Pour laisser un blanc entre le Z et le 2, utilisez la touche espace (la grande barre horizontale du bas du clavier) ; quand le message est tapé, appuyez sur la touche

RETURN

L'ordinateur répond encore un gros mot !

Il n'a pas l'air d'être très intelligent !

Le message syntax error veut dire erreur de syntaxe.

En effet, l'ordinateur ne comprend pas nos différents messages !

Il va falloir lui parler dans le seul langage qu'il connaît :
le BASIC

Nous allons écrire le petit programme d'addition entre 2 nombres
vu dans le chapitre précédent

```
1Ø INPUT A
2Ø INPUT B
3Ø X=A+B
4Ø PRINT X
5Ø END
```

Tapez le programme, mais introduisons une erreur volontaire
dans la première ligne (IMPUT au lieu de INPUT)

```
1Ø IMPUT A      RETURN
2Ø INPUT B      RETURN      N'oubliez pas de taper
3Ø X=A+B        RETURN      RETURN      à la fin de
4Ø PRINT X      RETURN      chaque instruction
5Ø END          RETURN
```

Pour exécuter le programme, tapez

l'ordre RUN RETURN

Encore un gros mot !

Ah oui, c'est normal, nous avons fait une erreur de syntaxe
à la ligne 1Ø.

Pour la corriger, il suffit de retaper la ligne et de la
valider par RETURN

Vous retapez donc : 1Ø INPUT A RETURN

Tapez LIST RETURN pour voir si la correction a bien été
effectuée ; l'ordre LIST, affiche sur l'écran le programme
stocké en mémoire ;
pour exécuter à nouveau le programme, tapez

RUN RETURN

Un point d'interrogation s'affiche sur l'écran.
L'ordinateur demande une information ; il attend qu'on lui
introduise la valeur de A.

Donnons la valeur 5 par exemple

Tapez 5 puis RETURN évidemment

Un autre point d'interrogation s'affiche pour la demande de la valeur B.

Donnons la valeur 8 par exemple ; sans oublier d'appuyer sur

L'ordinateur exécute alors (très rapidement) l'instruction
 $3\emptyset \quad X=A+B$
 puis affiche la valeur de X à l'instruction $4\emptyset$

Résultat correct n'est-ce-pas ?

L'instruction $5\emptyset$ END dit à l'ordinateur qu'il a terminé son travail.

Il affiche alors Ready qui veut dire PRET :
 il est à nouveau prêt pour travailler

```

    RUN
    ? 5
    ? 8
    13
    Ready
    
```

Recommencez plusieurs fois à exécuter le programme.

Faites RUN

. donnez lui des valeurs négatives : (exemple : - 15)
 le signe - se trouve au-dessus du P

. donnez lui également des nombres décimaux :
 attention, pour les nombres décimaux, on n'utilise pas la virgule mais le point (exemple : 5.742)

Après avoir essayé plusieurs fois, et vérifié les résultats des calculs, nous allons supprimer l'instruction : $5\emptyset$ END

Pour cela, il suffit de taper $5\emptyset$ puis

Tapez LIST , et vous découvrirez que l'instruction $5\emptyset$ a bien disparu

Exécutez à nouveau le programme en tapant RUN

Donnez à nouveau 2 valeurs

Que remarquez-vous ?

Le programme marche exactement de la même façon.
 En effet, l'instruction END n'est pas obligatoire à la fin du programme. Mais nous verrons cette instruction plus en détail dans la suite de ce cours.

Comment faudrait-il faire pour que notre programme fasse maintenant la somme de 3 nombres ?

1) Il faudrait rajouter une ligne :

25 INPUT C par exemple

Heureusement que nous avons numéroté le programme de 10 en 10 afin de pouvoir intercaler des lignes !

2) Il faudrait modifier la ligne 30 :

30 X=A+B+C

Pour rajouter la ligne 25, facile ! Vous tapez

25 INPUT C

Pour modifier la ligne 30, nous pourrions retaper la ligne bien sûr, mais nous allons utiliser les touches de déplacement du curseur.

Il suffit de "monter sur la ligne 30 avec et

d'avancer à la fin de la ligne avec puis tapez

(rajouter) + C ;

enfin avec la touche permet de redescendre

Faites LIST pour voir si le programme est comme suit :

```
10 INPUT A
20 INPUT B
25 INPUT C
30 X=A+B+C
40 PRINT X
```

Exécutez le programme !

RUN

Donnez lui 3 valeurs pour vérifier si ce nouveau programme fonctionne comme prévu.

Pour l'instant notre programme se trouve dans la mémoire de l'ordinateur.

Tapez CLS

L'écran s'efface mais le programme est toujours là. Pour en être convaincu,

tapez LIST

Si maintenant vous éteignez votre ordinateur ou s'il se produit une panne de courant, le programme sera perdu, il faudra alors retaper le programme.

Dans notre cas, cela ne serait pas trop grave car il ne fait que 5 lignes, mais imaginez que vous perdiez un programme qui a 50 lignes ou même davantage !

Par mesure de sécurité, nous allons enregistrer notre programme sur une cassette.

Vérifiez que votre unité de cassettes est bien branchée.

Mettez une cassette dans l'unité.

Rembobinez la cassette, (touche REW).

Si ça ne marche pas, il faut sortir la prise noire, (côté magnétophone) rembobiner puis remettre la prise noire.

L'unité de cassettes est prête.

L'ordre d'enregistrement est CSAVE
C comme cassette
SAVE qui veut dire sauver, enregistrer

La syntaxe est la suivante :

CSAVE"nom du programme"

Tapez par exemple :
CSAVE"ADDITION"

ADDITION représente le nom du programme

Pressez sur les touches REC et PLAY du magnétophone.

Puis appuyez sur la touche

La cassette tourne un petit moment, puis l'ordinateur affiche Ready pour dire qu'il a fini et qu'il est prêt à recevoir un autre ordre.

Pour vérifier que le programme est bien enregistré sur la cassette, nous allons utiliser l'instruction CLOAD ?

Rembobinez la cassette (sortir la prise noire du magnétophone, presser sur REW puis remettre la prise noire).

Tapez : CLOAD?"ADDITION"

Puis appuyez sur la touche PLAY du magnétophone

Pressez

L'ordinateur effectue une comparaison entre le programme en mémoire et celui qui est sur la cassette.

Si l'écran affiche le message READY, l'enregistrement s'est bien fait, vous pouvez continuer les manipulations.

Si l'écran affiche le message BAD, il y a eu une erreur, il faut recommencer l'enregistrement.

Laissons la cassette un moment de côté et tapons NEW

Que se passe-t-il ?

Rien apparemment !

En tout cas, l'ordinateur a compris notre message car il n'affiche pas SYNTAX ERROR !

Tapez RUN

Il ne se passe rien

Tapez LIST

Toujours rien

Hé bien, notre programme a tout simplement disparu.

L'ordre NEW efface toute la mémoire.

NEW a le même effet qu'une coupure de courant.

Heureusement que nous avons sauvegardé notre programme sur cassette.

Nous allons le rechercher sur la cassette et le mettre en mémoire.

Rembobinez la cassette. Si ça ne marche pas, il faut sortir la prise noire (côté magnétophone) rembobiner puis remettre la prise noire.

Pour charger un programme de la cassette vers la mémoire, on utilise l'instruction CLOAD

C comme cassette

LOAD qui veut dire charger

La syntaxe utilisée est CLOAD"nom du programme"

Tapez CLOAD"ADDITION" et pressez sur la touche du magnétophone.

L'ordinateur affiche FOUND : ADDITI
READY

Remarques : * pour le nom du programme, l'ordinateur ne prend en compte que 6 caractères, c'est pourquoi il affiche FOUND : ADDITI

Faites LIST

RUN

Notre programme est à nouveau en mémoire.

* Attention : un programme enregistré avec un nom en minuscules, respectivement en majuscules doit être relu avec le même nom en minuscules, respectivement en majuscules.

COMPLEMENTS

=====

- 1) On peut mettre plusieurs instructions sur une ligne en les séparant par des doubles points (:)

Le programme d'addition aurait pu s'écrire :

```
1Ø INPUT A:INPUT B:INPUT C
2Ø X=A+B+C:PRINT X
```

- 2) Comment éditer le programme sur l'imprimante ?

Tapez LLIST pour sortir la liste du programme sur papier

- 3) Effacez la mémoire (NEW) et tapez le programme suivant :

```
1Ø PRINT "BONJOUR"
2Ø GOTO 1Ø
```

Tapez RUN

Le programme affiche BONJOUR sans jamais s'arrêter.

En effet, l'instruction 2Ø renvoie le programme chaque fois à l'instruction 1Ø. GOTO veut dire aller à (nous le verrons ultérieurement).

Pour arrêter l'exécution du programme, il faut presser sur la touche et

L'ordinateur s'arrête et affiche
BREAK IN 1Ø par exemple

ce qui signifie "arrêté à l'instruction 1Ø"

On peut donc ainsi arrêter tout programme pendant son exécution.

4) Longueur d'une ligne d'instructions

Le nombre maximum de caractères possibles sur une ligne d'instruction est de 128 ; lors de l'introduction d'une ligne d'instruction, comportant un grand nombre de caractères, ne tapez RETURN qu'à la fin de l'instruction, même si cette instruction prend plus de 1 ligne sur l'écran