Saisie des données et affichage

POUR TOUT TRAITEMENT L'ORDINATEUR DOIT POUVOIR LIRE* DES DONNEES, LES STOCKER EN MEMOIRE ET LES AFFICHER A L'ECRAN

* Lire des données se traduit en jargon informatique par le mot saisir des données (ceci dans le cas d'introduction manuelle d'informations par l'intermédiaire du clavier), d'où le mot saisie.

Ces instructions sont très liées. Effectivement, il n'est pas concevable de lire des données et de les traiter (nous le verrons ultérieurement) sans pouvoir à un moment quelconque du traitement suivre l'évolution de ces données en affichant un résultat, un message, etc...

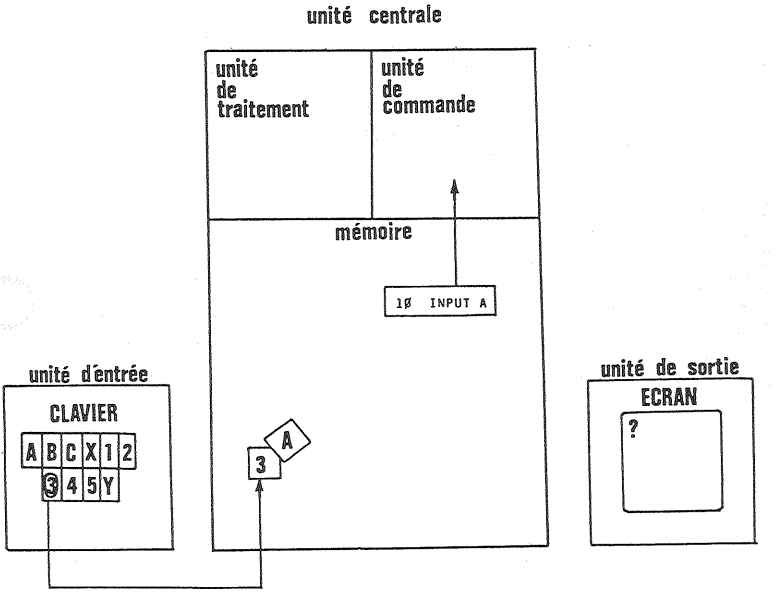
) Quand l'ordinateur rencontre une instruction INPUT il se met en attente.

Exemple : 1Ø INPUT A

Dans ce cas l'ordinateur vous demande un nombre et le range en mémoire avec l'étiquette A.

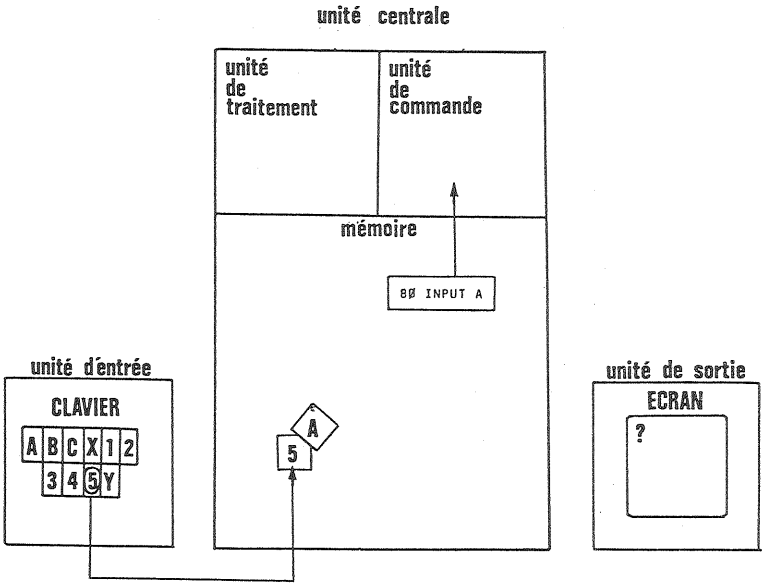
Ce dernier peut être entier ou décimal, positif, négatif ou nul, comme l'indique les exemples suivants :

Ø 13.5 -2Ø -2.6 12 -Ø.518 +2.52Ø2 -Ø.25 etc ...



Dans cet exemple la valeur 3 tapée au clavier sera rangée au fin fond de la mémoire dans une case nommée A. Dans cette variable A, il ne pourra y avoir qu'un seul nombre à la fois, mais sa valeur peut être modifiée à chaque instant.

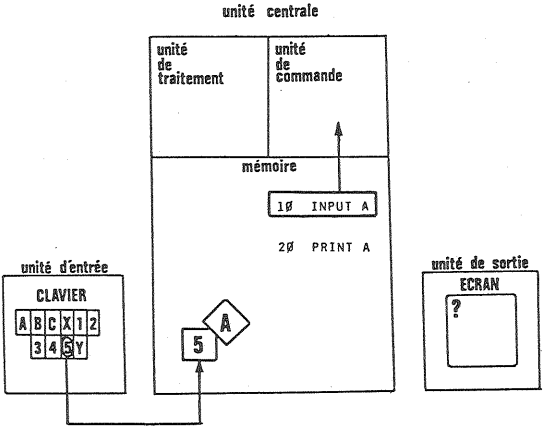
Si par la suite (instruction 8Ø, par exemple), l'ordinateur retombe sur une même instruction INPUT A, la nouvelle valeur de A écrasera inévitablement l'ancienne se trouvant déjà en mémoire.

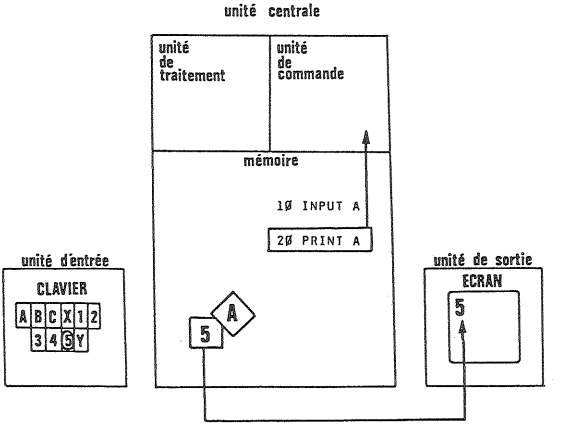


Règle : Toute valeur placée dans une variable déjà existante, écrase l'ancienne valeur.

L'instruction PRINT permet de rechercher au coeur même de la mémoire de l'ordinateur le contenu de n'importe quelle variable préalablement stockée et l'afficher à l'écran.

Exemple : 1Ø INPUT A
2Ø PRINT A





Mais attention : ce n'est pas parce que la variable A est affichée à l'écran qu'elle ne se trouve plus en mémoire. La valeur de cette variable A, ne sera perdue que si l'ordinateur rencontre une autre instruction INPUT A, ou d'une manière générale lorsque l'ordinateur rencontre une instruction qui place dans la variable A une autre valeur, ou bien sûr si l'on coupe le courant.

Règle : le contenu d'une variable ne change que si on place dans cette variable une autre valeur

Le programme que nous venons de voir :

```
1Ø INPUT A
2Ø PRINT A
```

ne fait qu'afficher le nombre qu'on lui a préalablement donné.

Autre exemple :

```
1Ø PRINT A
```

Pour ce programme (qui se limite à une instruction), 2 possibilités sont à envisager selon le type d'ordinateur :

- l'ordinateur affiche Ø car toutes les variables inconnues sont égales à zéro (ce qui est le cas du SANYO PHC-25).
- l'ordinateur indique qu'il y a erreur car, ne connaissant pas la variable en question (elle n'existe pas en mémoire), il ne peut l'afficher !

Mais l'instruction PRINT ne permet pas seulement d'afficher le contenu d'une case mémoire ; mais également des messages :

```
1Ø INPUT A
2Ø PRINT A
3Ø PRINT "A"
```

Dans cet exemple l'instruction 2Ø (PRINT A) affiche le contenu de la variable A (c'est-à-dire la valeur préalablement introduite) ; alors que l'instruction 3Ø (PRINT "A") affiche la lettre A.

En effet, les guillemets permettent d'afficher n'importe quels caractères sans que l'ordinateur ne se soucie de leur nature ou de leur orthographe.

Exemple :

```
1Ø PRINT "BONCHOUR"
2Ø PRINT "AGENT N. ZX-3BØØ3"
3Ø PRINT "OH REVOIR !"
```

Vous remarquerez au passage qu'à chaque instruction PRINT, l'affichage passe à la ligne suivante ! Il suffirait donc de rajouter au programme précédent 2 instructions :

```
   15 PRINT
et 25 PRINT
```

afin de laisser une ligne blanche après chaque message affiché.

Le fait de pouvoir afficher des messages nous permettra de rendre le programme plus clair pour celui qui va par la suite s'en servir :

Prenons un exemple concret :

Pour éditer une fiche de paye, l'ordinateur doit lire de nombreuses informations :

```
1Ø INPUT AGE
2Ø INPUT SB
3Ø INPUT CP
4Ø INPUT P
.
.
.
```

l'âge (AGE), le salaire de base (SB), le code postal (CP), le montant des primes (P), ...

Ce programme lit 4 variables AGE, SB, CP, P. Il est bien entendu que le traitement de la paye nécessite la lecture de beaucoup plus d'informations ; mais nous nous limitons volontairement à ces 4 variables pour simplifier l'exemple.

L'opérateur chargé par la suite de saisir de telles informations risquerait fort de s'y perdre et de saisir les données dans le désordre ne sachant quelle variable correspond à quel renseignement. En effet, pour chaque instruction INPUT, seul un point d'interrogation s'affiche sur l'écran !
Grâce à l'instruction PRINT il nous est possible d'éviter de telles ambiguités :

```
1Ø PRINT "AGE"
2Ø INPUT A
3Ø PRINT "SALAIRE DE BASE"
4Ø INPUT S
5Ø PRINT "CODE POSTAL"
6Ø INPUT CP
7Ø PRINT "PRIME"
8Ø INPUT P
```

Ce programme lit les mêmes quatre variables AGE, SB, CP,P mais avant chaque instruction INPUT un petit texte est affiché à l'écran ce qui permet à l'opérateur de savoir ce qu'il faut taper sur le clavier.

Dans les programmes donnés dans ce manuel, nous séparons les instructions, les numéros de lignes, les variables par des espaces, ceci afin d'obtenir une meilleure lisibilité des programmes.

Pour le BASIC du SANYO PHC-25, la séparation entre les termes n'est pas obligatoire mais conseillée.

Dans les quelques exemples suivants nous allons vous montrer de quelle manière on peut astucieusement combiner l'affichage de textes et de variables sur une même instruction PRINT

```
1Ø PRINT "QUEL AGE AS-TU"
2Ø INPUT A
3Ø PRINT "J'AI";A;"ANS"
```

Textes et variables sont autorisés derrière une même instruction PRINT à condition qu'ils soient séparés par un point virgule (sur certains ordinateurs une virgule)

```
1Ø PRINT "NB DE JOURS POUR MARS?"
2Ø INPUT J
3Ø PRINT "MARS A";J;"JOURS"
```

Dans cet exercice l'ordinateur vous indique de combien de jours se compose le mois de mars. Il est bien entendu que tout le monde sait que le mois de mars a 31 jours mais l'ordinateur lui ne le sait pas ; ainsi lorsque l'ordinateur exécute l'instruction 2Ø vous affichera bêtement MARS A 45 JOURS. Cela peut paraître assez inquiétant qu'une telle erreur puisse se produire ; mais rassurez-vous, il est possible de tester la variable tapée au clavier afin de corriger l'erreur ; mais il est encore trop tôt pour vous en parler.

```
1Ø PRINT "VOTRE POIDS EN KG"
2Ø INPUT P
3Ø PRINT "VOTRE TAILLE EN CM"
4Ø INPUT T
5Ø PRINT "VOUS PESEZ";P;"KG"
6Ø PRINT "VOUS MESUREZ";T;"CM"
```

```
1Ø  PRINT "DONNEZ UN NOMBRE"
2Ø  INPUT A
3Ø  PRINT "UN AUTRE NOMBRE"
4Ø  INPUT B
5Ø  PRINT "SVP ENCORE UN"
6Ø  INPUT C
7Ø  PRINT "MERCI"
8Ø  PRINT "LES NOMBRES SONT:";A;B;C
```

```
1Ø  PRINT "DONNEZ UN NOMBRE"
2Ø  INPUT X
3Ø  PRINT "VOUS M'AVEZ DONNE";X
```

```
1Ø  PRINT "DONNEZ VOTRE DATE DE NAISSANCE"
2Ø  PRINT "LE JOUR"
3Ø  INPUT J
4Ø  PRINT "LE MOIS"
5Ø  INPUT M
6Ø  PRINT "L'ANNEE"
7Ø  INPUT A
8Ø  PRINT "VOUS ETES NE LE";J;"-";M;"-";A
```

REMARQUE : si vous lui donnez le mois en lettre, il ne sera pas d'accord ; car il attend une variable numérique (M)

Complément sur les expressions numériques : chaque valeur positive est précédée d'un espace, chaque valeur négative est précédée d'un signe moins.
Pour améliorer la lisibilité des résultats affichés à l'écran, on prendra la précaution d'intercaler un espace entre chaque variable numérique.

Exemple : PRINT "VOUS ETES NE LE";J;" -";M;" -";A

Calculs

NOUS SAVONS DEJA COMMENT PROCEDER POUR SAISIR DES DONNEES, AFFICHER A L'ECRAN CES DONNEES OU DES MESSAGES, MAIS TOUT TRAITEMENT NECESSITE AUSSI DES CALCULS

Une instruction de calcul a toujours la forme suivante :

1Ø _____ = _____

variable dans laquelle est rangé le résultat

calcul à effectuer

Exemple : 1Ø X=A+B

Remarque : Cette instruction ne pourrait pas s'écrire :

1Ø A+B=X car la "variable résultat" doit toujours se trouver à gauche du signe =

Sachant que + est le signe de l'addition,
- est le signe de la soustraction,
* est le signe de la multiplication et
/ est le signe de la division :

```
1Ø INPUT X
2Ø X1=X+1
3Ø X2=X1+43
4Ø X3=X2+X-4.5
5Ø Y=X/2
6Ø A=X*3
7Ø Z=X+Y+A-X2
8Ø PRINT X3;Y;A;Z
```

Dans ce programme, si dans l'instruction 1Ø vous lui donnez le nombre 4Ø par exemple, il vous affichera :

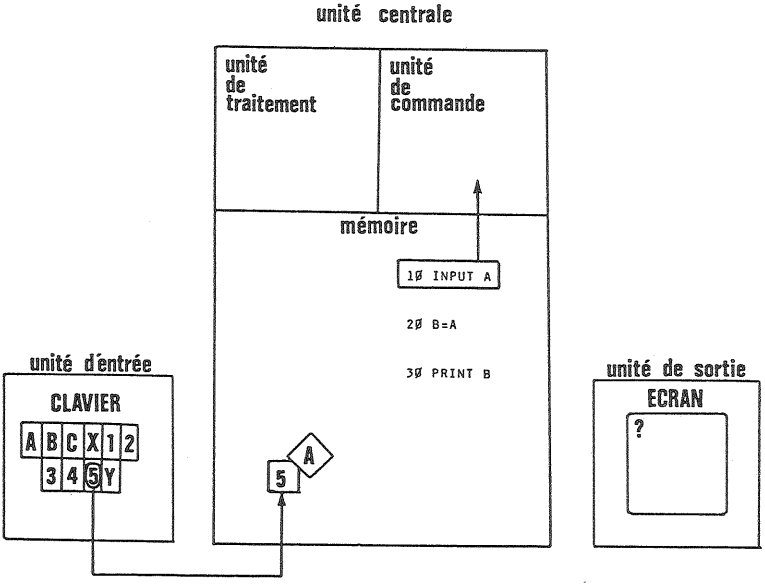
```
 119.5          2Ø          12Ø          96
```

EXERCICE

```
1Ø INPUT A
2Ø B=A
3Ø PRINT B
```

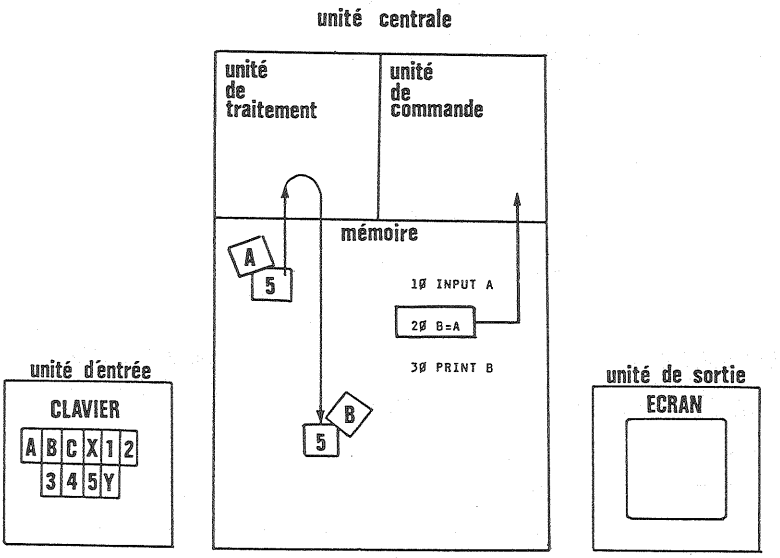
Dans ce programme on lit une variable A, on range son contenu dans une variable B qui est finalement affichée à l'écran :

PREMIERE ETAPE (1ère instruction)



La variable A est tapée au clavier, puis stockée en mémoire (5 par exemple).

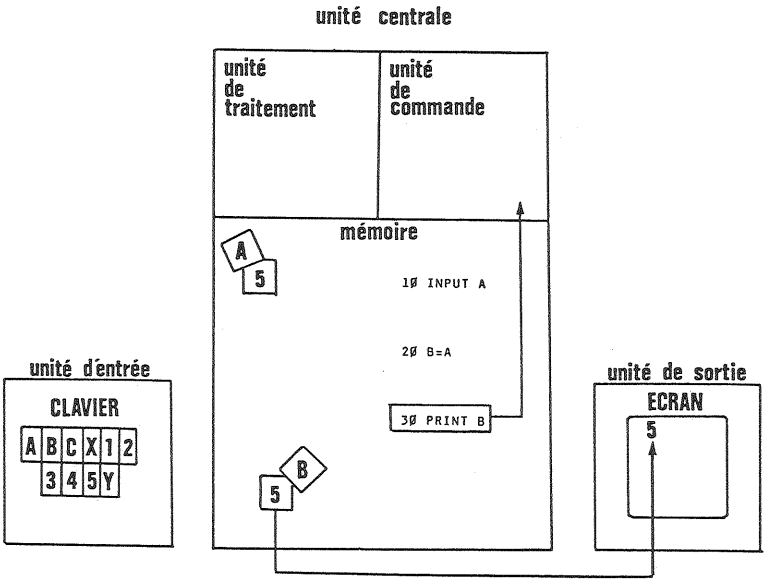
DEUXIEME ETAPE (2ème instruction)



L'ordinateur créé une nouvelle case mémoire (variable) qu'il appelle B et son contenu sera le même que celui de la variable A.

Remarque : la variable A n'est pas détruite !

TROISIEME ETAPE (3ème instruction)



Le contenu de la variable B est affiché à l'écran

Remarque :

les variables A et B ne sont pas détruites !

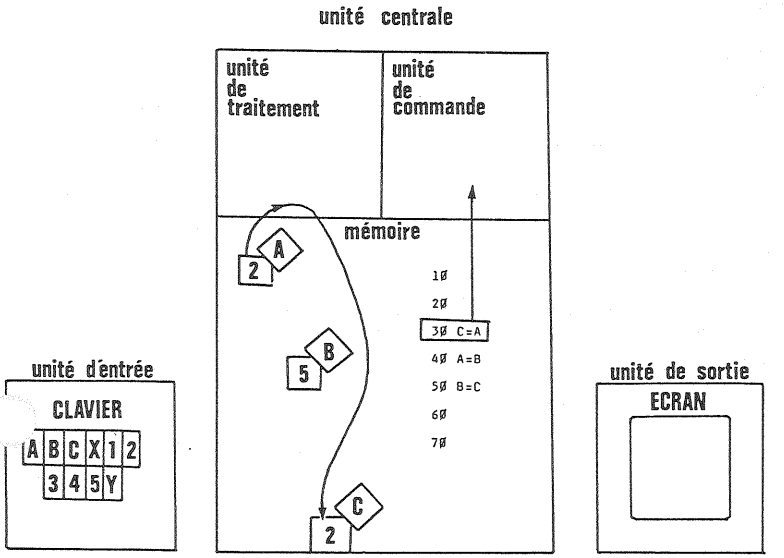
APPLICATION PRATIQUE DE L'EXERCICE PRECEDENT

```
1Ø INPUT A
2Ø INPUT B
3Ø C=A
4Ø A=B
5Ø B=C
6Ø PRINT A
7Ø PRINT B
```

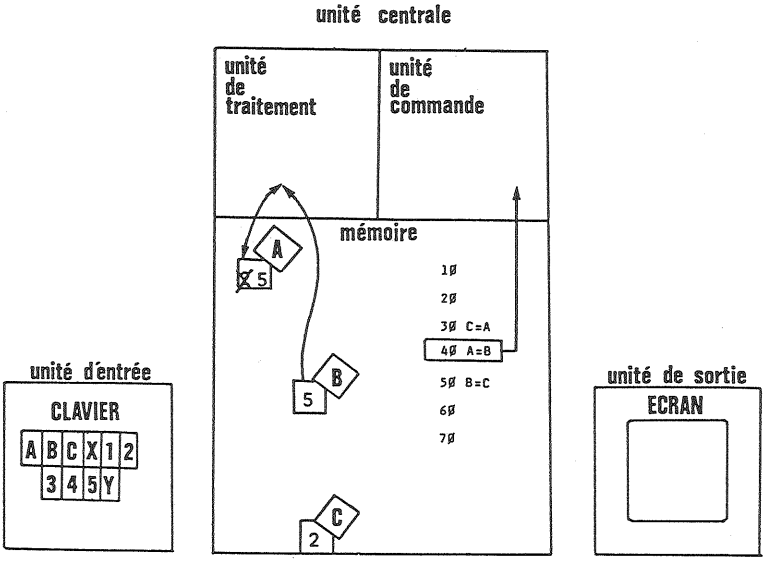
Les instructions 1Ø, 2Ø et 6Ø, 7Ø sont respectivement 2 instructions de lecture et 2 instructions d'affichage que nous avons déjà étudiées et qui n'ont plus de secret pour vous.

Ainsi dans les 3 schémas suivants seules les instructions 3Ø, 4Ø et 5Ø seront expliquées.

Admettons que les valeurs des 2 variables A et B lues dans les instructions 1Ø et 2Ø soient respectivement 2 et 5.

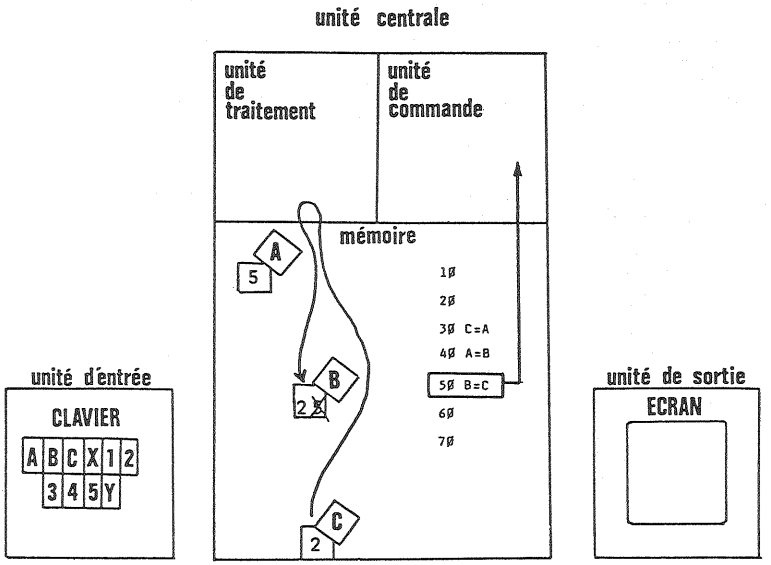


L'instruction 3Ø indique à l'ordinateur qu'il doit créer une case en mémoire appelée C et où il doit ranger le contenu de la variable A. Le contenu de A et de B ne change pas.



Puis le contenu de B est placé dans A.

L'ancien contenu de A est écrasé par le nouveau.
Le contenu de C et de B ne change pas.

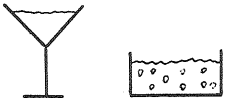


Enfin le contenu de C est placé dans B.
L'ancien contenu de B est écrasé par le nouveau.
Le contenu de A et de C ne change pas.

Nous étions partis avec la valeur 2 dans A et 5 dans B ...
et après l'instruction 5Ø, nous avons 2 dans B et 5 dans A !

Le petit programme que nous venons de voir ensemble échange le contenu des 2 variables A et B. Pour procéder à cet échange il faut inévitablement passer par une troisième variable ; effectivement, il serait impossible de permuter directement le contenu des variables A et B sans en écraser forcément l'une des deux.

Imaginez que par étourderie vous ayez mis du vin dans un verre à eau et de l'eau dans un verre à vin :



.... Pour échanger le contenu de ces 2 récipients, vous seriez obligés (comme dans le programme précédent) d'utiliser un 3ème récipient !

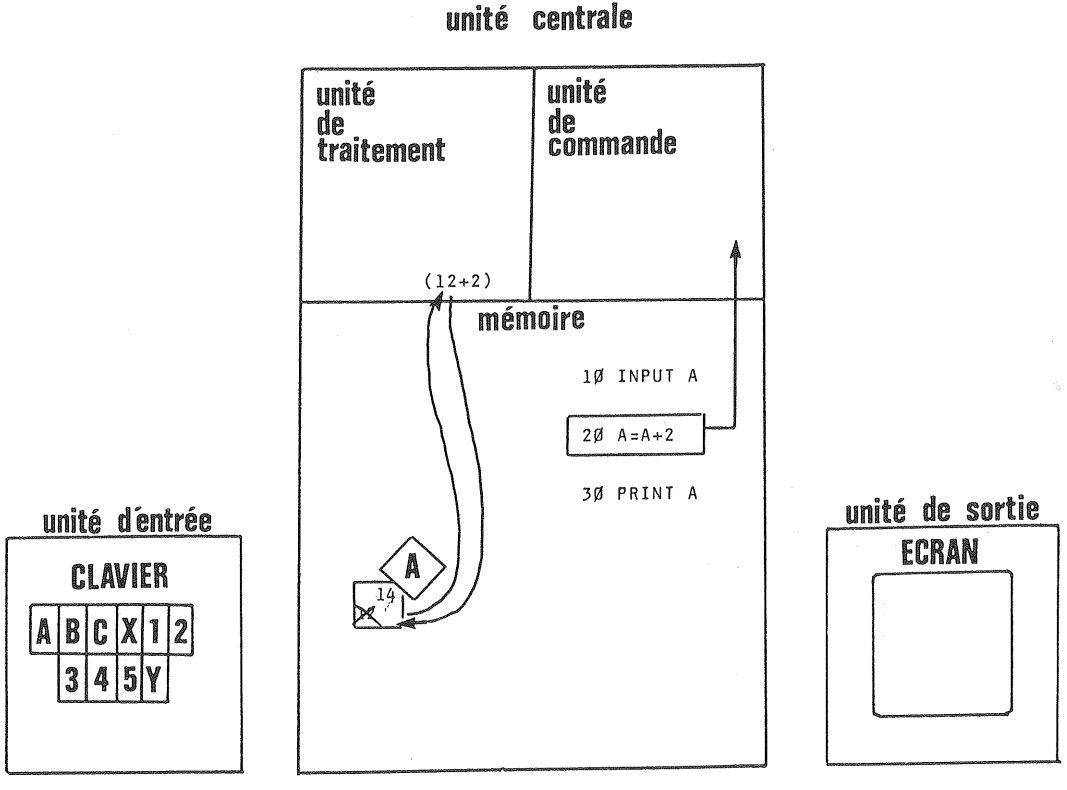
Petit exercice :

On lit un nombre A, on l'augmente de 2 et on affiche la nouvelle valeur.

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

Supposons qu'on ait entré la valeur 12 pour A.

L'instruction 2Ø écrase cette valeur 12 et la remplace par la valeur 12 + 2 soit 14.


unité centrale
unité de traitement
unité de commande
(12+2)
mémoire
1Ø INPUT A
2Ø A=A+2
3Ø PRINT A
A
14
unité d'entrée
CLAVIER
A B C X 1 2
3 4 5 Y
unité de sortie
ECRAN

Les 4 opérations arithmétiques sont les suivantes :

+	addition
-	soustraction
*	multiplication
/	division

QUELQUES MOTS SUR LES PARENTHESES

Les opérations de calcul ont certains niveaux de priorité.

La multiplication et la division ont une priorité sur l'addition et la soustraction.

Ex. : Soit le calcul suivant :

X=A+B/2

L'ordinateur fera d'abord la division de B par 2, puis l'addition de A avec le résultat de la division.

Ainsi pour les valeurs suivantes :

A=2 et B=4, on obtiendra X=4

Pour changer l'ordre dans lequel les opérations sont exécutées, il faut utiliser les parenthèses.

Les opérations entre parenthèses seront exécutées en premier ; à l'intérieur des parenthèses, l'ordre habituel des opérations est maintenu.

Remarque : Le nombre de parenthèses ouvrantes doit toujours être égal au nombre de parenthèses fermantes.

$A + \frac{B}{2}$ s'écrira en Basic A+(B/2) ou plus simplement A+B/2
puisque la division a une priorité supérieure à l'addition.
Mais $\frac{A+B}{2}$ s'écrira en Basic (A+B)/2 pour que l'ordinateur exécute l'addition avant la division !

QUELQUES EXEMPLES DE CALCUL :

calculs :	s'écrit en Basic :
$X=A+B+C+3$	X=A+B+C+3
$X=A-B-10.5$	X=A-B-10.5
$X=A+\frac{B}{2}$	X=A+B/2
$X=\frac{A}{B}+3$	X=A/B+3
$X=\frac{A+3}{B}$	X=(A+3)/B
$C=\frac{A+B}{2}$	C=(A+B)/2
$Y=A-\frac{B}{C}$	Y=A-B/C
$X1=\frac{A+B}{A-B}$	X1=(A+B)/(A-B)
X=2xA	X=2*A
X=(A+B)x(A-B)	X=(A+B)*(A-B)
$X=\frac{AxB}{C}$	X=(A*B)/C ou X=A*B/C qui donne le même résultat
$X=\frac{(A+B)xA}{A-B}$	X=((A+B)*A)/(A-B)
$X=\frac{A}{3}+\frac{B}{4}+\frac{A-B}{C}-7$	X=A/3+B/4+(A-B)/C-7
augmenter A de 1	A=A+1
transférer le contenu de la variable A dans la variable B	B=A (la variable A reste inchangée)
transférer le nombre 3.7 dans la variable X	X = 3.7
diminuer A de 2	A=A-2
multiplier B par 3	B=B*3
placer dans A, 3 fois la valeur de B (A=Bx3)	A=B*3

LE PETIT CONSEIL D'AMI ...

Utilisez les parenthèses si vous doutez... ; même si elles ne sont pas strictement nécessaires :

$A + \frac{B}{2}$ s'écrit A+B/2 mais pourrait également s'écrire A+(B/2)

$\frac{A * B}{C}$ s'écrit A*B/C mais pourrait également s'écrire (A*B)/C

AUTRES FONCTIONS DE CALCUL

Bien sûr, un ordinateur sait faire des opérations arithmétiques (+ - * /), comme toute calculatrice, mais il possède d'autres fonctions de calcul :

LA MISE A LA PUISSANCE

A à la puissance B (A^B) s'écrit en Basic A ↑ B
(sur certains ordinateurs on écrit A ^ B ou A ** B

Sur votre SANYO PHC-25, la mise à la puissance correspond au signe ^ . (touche située à droite du signe -).

Exemple : Calcul de la surface d'un cercle
soit RAY le rayon du cercle
DIAM le diamètre
SURF la surface

$$\begin{cases} RAY = DIAM/2 \\ SURF = 3.14 * RAY^2 \end{cases}$$

```
1Ø INPUT DIAM
2Ø RAY = DIAM/2
3Ø SURF = 3.14*(RAY ^ 2)
4Ø PRINT SURF
```

Explications :

Ligne 1Ø : L'ordinateur demande le diamètre du cercle.
Supposons qu'on introduise la valeur 10.
Dans ce cas, DIAM = 1Ø

Ligne 2Ø : L'ordinateur divise le diamètre par 2 et met le résultat dans RAY
Dans ce cas, RAY = 5

Ligne 3Ø : L'ordinateur calcule d'abord les opérations entre parenthèses (RAY ^ 2) donne 25
Puis il multiplie ce résultat par 3.14 ce qui donne 78.5, valeur stockée dans SURF
SURF = 78.5

Ligne 4Ø : L'ordinateur affiche le résultat soit 78.5

REMARQUE : L'instruction 2Ø et 3Ø pourrait s'écrire en 1 instruction : SURF=3.14*((DIAM/2) ^ 2)

LA RACINE CARREE

$\sqrt{A}$ s'écrit en Basic SQR(A)

Exemple : calcul du diamètre d'un cercle en connaissant la surface

```
1Ø INPUT SURF
2Ø DIAM=(SQR(SURF/3.14))*2
3Ø PRINT DIAM
```

Explication :

Ligne 1Ø : L'ordinateur demande la surface du cercle.
Donnons : 78.5
Dans ce cas SURF=78.5

Ligne 2Ø : L'ordinateur calcule d'abord (SURF/3.14) ce qui donne 25, dont il extrait la racine carrée (SQR(25)) donne 5, ce résultat multiplié par 2 donne 1Ø.
Cette valeur finale est stockée dans DIAM

Ligne 3Ø : Affiche la valeur de DIAM

LA VALEUR ABSOLUE

$|A|$ (valeur absolue de A) s'écrit en Basic ABS(A)

Exemple :

```
1Ø INPUT A
2Ø X=ABS(A)
3Ø PRINT X
```

^ est la valeur absolue de A ; c'est-à-dire A sans son signe ;

si A = - 3.47 alors X = 3.47
A = 3.47 alors X = 3.47

LE SIGNE

La fonction SGN nous permet de connaître le signe d'un nombre.

Exemple :

```
1Ø INPUT A
2Ø X=SGN(A)
3Ø PRINT X
```

Si A > Ø, SGN(A) est égal à 1
Si A=Ø, SGN(A) est égal à Ø
Si A < Ø, SGN(A) est égal à -1

LA PARTIE ENTIERE

Cette opération arrondit à la partie entière inférieure.

Exemple :

```
1Ø INPUT A
2Ø X=INT(A)
3Ø PRINT X
```

X est la partie entière de A c'est-à-dire A après avoir éliminé les décimales.

```
Si A vaut 25, X vaut 25
Si A vaut 25.202025, X vaut 25
Si A vaut 25.902025, X vaut 25
attention : si A vaut - 3.9, X vaut -4
```

Comment arrondir au centime ?

```
1Ø PRINT "DONNEZ UN NOMBRE"
2Ø INPUT A
3Ø X=A*1ØØ
4Ø Y=INT(X)
5Ø Z=Y/1ØØ
6Ø PRINT Z
```

```
Si vous donnez 15.3871     il affichera 15.38
Si vous donnez 17.328      il affichera 17.32
Si vous donnez  1.187431   il affichera  1.18
```

Explication :

- Prenons par exemple A qui vaut 15.3871
- Dans l'instruction 3Ø on multiplie A par 100 et on place le résultat dans X ; nous aurons alors X qui vaut 1538.71
- Dans l'instruction 4Ø on prend la partie entière de X et on place le résultat dans Y ; nous aurons alors Y qui vaut 1538
- Dans l'instruction 5Ø on divise Y par 100 et on place le résultat dans Z ; nous aurons alors Z qui vaut 15.38

... c'est pas plus difficile que çà !

les instructions 3Ø, 4Ø et 5Ø peuvent s'écrire sur
l seule ligne :

```
1Ø PRINT "DONNEZ UN NOMBRE"
2Ø INPUT A
3Ø X=A*1ØØ:Y=INT(X):Z=Y/1ØØ
4Ø PRINT Z
```

Mais ce programme peut s'écrire d'une autre manière, en utilisant moins de lignes programme et moins d'instructions.

Pour cela, on utilise les parenthèses :

Arrondir A à 2 décimales :

```
1Ø INPUT A
2Ø X=(INT(A*1ØØ))/1ØØ
3Ø PRINT X
```

Ligne 1Ø : L'ordinateur demande un chiffre
Donnons : A=34.377841

Ligne 2Ø : X=(INT(A*1ØØ))/1ØØ
L'ordinateur calcule d'abord A*100, ce qui donne 3437.7841
Puis, calcule la valeur entière de ce résultat.
On obtient 3437, qu'il divise par 100, soit 34.37
valeur qui correspond bien à l'arrondi à 2 décimales du chiffre 34.377841

Pour obtenir l'arrondi à 2 décimales d'un chiffre, il suffit donc :

- de multiplier ce chiffre par 1ØØ
- en prendre la partie entière
- diviser le résultat par 1ØØ

Pour arrondir à 3 chiffres, il faudrait multiplier puis diviser par 1ØØØ, à 4 chiffres par 1ØØØØ, etc...

Comment arrondir A à 2 décimales au plus près .

Si A = 141.13321, X vaut 141.13
Si A = 141.13921, X vaut 141.14 par exemple

```
1Ø INPUT A
2Ø X=(INT((A+Ø.ØØ5)*1ØØ))/1ØØ
3Ø PRINT X
```

Pour arrondir à 2 décimales au plus près, il suffit avant de multiplier par 1ØØ, d'ajouter Ø.ØØ5 au nombre.

En effet, si le nombre = 1 4 1 . 1 3 3 2 1
et on y ajoute Ø.ØØ5 + Ø . Ø Ø 5

1 4 1 . 1 3 8 2 1 inchangé

si le nombre = 1 4 1 . 1 3 8 2 1
et on y ajoute Ø.ØØ5 + Ø . Ø Ø 5

1 4 1 . 1 4 3 2 1 augmente de 1

D'une manière générale si le nombre A vaut

xxx.xxxxx
↑

Si le chiffre qui se trouve en 3ème position après la virgule vaut Ø, 1, 2, 3, ou 4, l'addition à Ø.ØØ5 ne change rien sur la 2ème position après la virgule.

Si le chiffre qui se trouve en 3ème position après la virgule vaut 5, 6, 7, 8, ou 9, l'addition à Ø.ØØ5 a pour conséquence d'augmenter de 1 le chiffre qui se trouve en 2ème position après la virgule.

AUTRES PETITS PROGRAMMES ...

```
1Ø PRINT "PRIX DU KILO DE BANANES"
2Ø INPUT PB
3Ø PRINT "COMBIEN DE BANANES DANS UN KILO"
4Ø INPUT NB
5Ø X=PB/NB
6Ø P= INT(X)
7Ø PRINT "PRIX D'UNE BANANE";P
```

Le prix d'une banane est arrondi au Frs. inférieur

```
1Ø PRINT "DONNEZ UN NOMBRE DECIMAL"
2Ø INPUT ND
3Ø X=INT(ND)
4Ø Y=X+1
5Ø PRINT "LE NOMBRE";ND;"EST COMPRIS ENTRE";X;"ET";Y
```

Si vous donnez le nombre 147.27
l'ordinateur vous répondra :
LE NOMBRE 147.27 EST COMPRIS ENTRE 147 ET 148

```
1Ø PRINT "DONNEZ LES DIMENSIONS DU RECTANGLE"
2Ø PRINT "LONGUEUR"
3Ø INPUT LO
4Ø PRINT "LARGEUR"
5Ø INPUT LA
6Ø P=2*(LO+LA)
7Ø S=LO*LA
8Ø PRINT "LE PERIMETRE      EST"; P
9Ø PRINT "LA SURFACE EST";S
```

LES FONCTIONS TRIGONOMETRIQUES

COS : Cosinus
SIN : Sinus
TAN : Tangente
LOG : logarithme
EXP : exponentielle

```
1Ø INPUT A
2Ø X=COS(A)
3Ø PRINT X
```

Ce programme calcule le cosinus de A ;
mais A est en radians !
Si l'on veut donner A en degrés,
il faudra écrire l'instruction 2Ø :
2Ø X=COS(A*Ø.Ø174533)

Une remarque sur les chiffres :

Un nombre trop grand peut s'écrire sous la forme :
nombre + exposant

1.80258 E + 6 qui veut dire $1.80258 * 10^6$

Exemple :
```
1Ø INPUT A
2Ø X=EXP(A)
3Ø PRINT X
```

Si vous donnez pour A, un nombre inférieur ou égal à 2Ø,
l'affichage de X se fait normalement.

Si vous donnez pour A, un nombre supérieur à 2Ø, X sera affiché
sous la forme ... E ...

Réciproquement, un nombre très petit peut également s'écrire
par exemple :

1.80258 E - 25 qui veut dire

$$\frac{1.80258}{(10^{25})}$$

CONCLUSION :

Ce qu'il faut retenir pour toute instruction de calcul :

- à droite du signe = se trouve la description du traitement
- à gauche du signe = se trouve le nom de la variable qui contiendra le résultat du traitement

Fonctions mathématiques exprimées en BASIC :

Les fonctions mathématiques ne faisant pas partie du BASIC du SANYO peuvent être calculées comme suit :

FONCTION	EQUIVALENT BASIC
SECANTE	SEC(X)=1/COS(X)
COSECANTE	CSC(X)=1/SIN(X)
COTANGENTE	COT(X)=1/TAN(X)
HYPERBOLIQUE SINUS	SINH(X)=(EXP(X)-EXP(-X))/2
HYPERBOLIQUE COSINUS	COSH(X)=(EXP(X)-EXP(-X))/2
HYPERBOLIQUE TANGENTE	TANH(X)=EXP-(-X)/EXP(X)+EXP(-X))*2+1
HYPERBOLIQUE SECANTE	SECH(X)=2/(EXP(X)+EXP(-X)
HYPERBOLIQUE COSECANTE	CSCH(X)=2/(EXP(X)-EXP(-X)
HYPERBOLIQUE COTANGENTE	COTH(X)=EXP(-X)/(EXP(X)-EXP(-X)*2+1
INVERSE HYPERBOLIQUE SINUS	ARCSINH(X)=LOG(X+SQR(X*X+1)
INVERSE HYPERBOLIQUE COSINUS	ARCCOSH(X)=LOG(X+SQR(X*X-1)
INVERSE HYPERBOLIQUE TANGENTE	ARCTANH(X)=LOG((1+X)/(1-X))/2
INVERSE HYPERBOLIQUE SECANTE	ARCSECH(X)=LOG((SQR(-X*X+1)+1/X)
INVERSE HYPERBOLIQUE COSECANTE	ARCCSCH(X)=LOG((SGN(X)*SQR(X*X+1)/X
INVERSE HYPERBOLIQUE COTANGENTE	ARCCOTH(X)=LOG((X+1)/(X-1))/2

Tests

MAIS DANS TOUT TRAITEMENT, IL EST EGALEMENT NECESSAIRE DE POUVOIR TESTER (COMPARER) LES VARIABLES

Une instruction de test s'écrit toujours :

IF (si) condition THEN (alors) instruction

- Si la condition est vérifiée, alors l'ordinateur exécute l'instruction indiquée, puis continue l'exécution du programme
- Si la condition n'est pas vérifiée, l'ordinateur passe directement à la suite du programme.

Exemple : reprenons l'exercice vu au début de ce chapitre :

```
1Ø PRINT "NB DE JOURS POUR MARS"
2Ø INPUT J
3Ø PRINT "MARS A";J;"JOURS"
```

L'ordinateur pourrait nous afficher n'importe quelle bêtise ; par exemple : MARS A 45 JOURS

Le programme pourrait alors s'écrire :

```
1Ø PRINT "NB DE JOURS POUR MARS"
2Ø INPUT J
3Ø IF J=31 THEN PRINT "OUI MARS A 31 JOURS"
4Ø IF J≠31 THEN PRINT "NON MARS N'A PAS";J;"JOURS"
```

≠ veut dire différent ; si ce signe n'existe pas sur le clavier, on le remplace par les 2 signes < > (inférieur-supérieur) :

```
4Ø IF J <> 31 THEN ...
```

Sur le SANYO PHC-25, le signe < > est donc utilisé pour symboliser le terme différent.

Les différents opérateurs utilisés pour écrire les conditions :

=	égal
≠	différent (si ce signe n'existe pas on le remplace par les 2 signes <>)
<	inférieur (plus petit)
>	supérieur (plus grand)
≤	inférieur ou égal (si ce signe n'existe pas on le remplace par les 2 signes < =)
≥	supérieur ou égal (si ce signe n'existe pas on le remplace par les 2 signes > =)
OR	qui veut dire OU } utilisé pour combiner plusieurs conditions
AND	qui veut dire ET }
NOT	fonction NON (veut dire "PAS")

Lire un nombre et afficher le message

- "ZERO" si ce nombre est nul
- "Positif" si ce nombre est supérieur à zéro
- "Négatif" si ce nombre est inférieur à zéro

```
1Ø INPUT A
2Ø IF A=Ø THEN PRINT "ZERO"
3Ø IF A>Ø THEN PRINT "POSITIF"
4Ø IF A<Ø THEN PRINT "NEGATIF"
```

On pourrait compléter ce programme de la manière suivante :

```
1Ø PRINT "DONNEZ MOI UN NOMBRE"
2Ø INPUT A
3Ø PRINT "CE NOMBRE EST:"
4Ø IF A=0 THEN PRINT "NUL"
5Ø IF A>Ó THEN PRINT "NEGATIF"
6Ø IF A<Ó THEN PRINT "POSITIF"
7Ø X=INT(A)
8Ø IF X=A THEN PRINT "ENTIER"
```

Les instructions 7Ø et 8Ø se comprennent en remarquant que si la partie entière d'un nombre est égale au nombre même, c'est que ce nombre est sans virgule décimale, donc il est entier !

Autre exemple :

- NIGHT CLUB -

INTERDIT AUX MINEURS

L'ordinateur demande votre âge,

Si vous avez moins de 18 ans, vous n'avez pas le droit d'entrer dans le Night Club

Si vous êtes majeurs, l'ordinateur vous souhaite la bienvenue

```
1Ø PRINT "VOTRE AGE"
2Ø INPUT A
3Ø IF A>18 THEN PRINT "BIENVENUE"
4Ø IF A<18 THEN PRINT "INTERDIT AUX MINEURS"
```

Et si on tape 18 lors de l'exécution de l'instruction 2Ø ?

L'ordinateur n'affiche rien !

Nous voyons que nous avons oublié un cas : A=18

Il va donc falloir modifier notre programme

en rajoutant par exemple la ligne 25 :

```
25 IF A=18 THEN PRINT "TOUT JUSTE!"
```

ou en modifiant l'instruction 3Ø

```
3Ø IF A≥18 THEN PRINT "BIENVENUE"
```

Dans le domaine de l'informatique, il est important de ne pas oublier de cas !

L'ordinateur ne prenant aucune initiative.

Autre exemple :

L'ordinateur doit lire 2 nombres ; les additionner et afficher le résultat si celui-ci est supérieur ou égal à 1Ø ; si le résultat est inférieur à 1Ø, il doit afficher "FAINEANT" ; on ne dérange pas un ordinateur pour si peu de chose !

```
1Ø INPUT A
2Ø INPUT B
3Ø X=A+B
4Ø IF X≥1Ø THEN PRINT X
5Ø IF X<1Ø THEN PRINT "FAINEANT"
```

Un exercice du même type :

L'ordinateur affiche le plus grand des deux nombres :
s'ils sont égaux, on affiche "EGAUX"

```
10 PRINT "DONNEZ DEUX NOMBRES"
20 INPUT A
30 INPUT B
40 IF A=B THEN PRINT "NOMBRES EGAUX"
50 IF A>B THEN PRINT "LE PLUS GRAND EST";A
60 IF B>A THEN PRINT "LE PLUS GRAND EST";B
```

```
10 PRINT "VOTRE AGE"
20 INPUT A
30 IF A<8 OR A>99 THEN PRINT "MON OEIL!"
40 PRINT "BONJOUR"
```

L'instruction 30 se lit :

```
Si A<8 ou A>99 alors afficher "MON OEIL"
 ↓       ↓          ↓       ↓
IF       OR        THEN    PRINT
```

```
10 PRINT "DONNEZ LES DIMENSIONS D'UN RECTANGLE
20 PRINT "LONGUEUR"
30 INPUT LO
40 PRINT "LARGEUR"
50 INPUT LA
60 IF LA=LO THEN PRINT"MENTEUR, C'EST UN CARRE"
70 IF LO>3*LA THEN PRINT "IL EST LONG!"
80 IF LO<1.5*LA AND LA<>LO THEN PRINT "C'EST PRESQUE UN CARRE"
90 IF LO<LA THEN PRINT "VOUS ETES SUR QUE CE N'EST PAS UN CERCLE?"
```

Autre exemple :

Voyage autorisé aux personnes majeures ayant moins de 90 ans et disposant au moins de 500 francs.

```
1Ø PRINT "VOTRE AGE"
2Ø INPUT A
3Ø PRINT "SOMME DISPONIBLE"
4Ø INPUT S
5Ø IF A≥18 AND A≤90 AND S≥500 THEN PRINT "BON VOYAGE:"
6Ø IF A<18 OR A>90 OR S<500 THEN PRINT "NON AUTORISE"
```

Nous avons utilisé ici des tests dits composés, en appliquant les fonctions AND et OR

COMPLEMENT

Dans une instruction de test, il est possible d'indiquer plusieurs instructions :

Structure : IF condition(s) THEN instructions
(si) (alors)

les différentes instructions sont séparées par des doubles points

Le principe reste le même :

Si la condition est vraie, alors et seulement dans ce cas, l'ordinateur exécute les instructions indiquées après le THEN, puis continue l'exécution du programme.

Exemple :

```
1Ø INPUT A
2Ø INPUT B
3Ø IF A+B>=1Ø THEN X=A+B:PRINT X
4Ø IF A+B<1Ø THEN PRINT "FAINEANT"
```

UN AUTRE EXEMPLE DU TEST POUR LES MATHEUX

Je vous propose de programmer l'exercice suivant :

```
1Ø INPUT A
2Ø X=1/A
3Ø PRINT X
```

Testez votre programme

Pour A=2, l'ordinateur affiche 0.5
Pour A=3, l'ordinateur affiche 0.333333...
Pour A=5, l'ordinateur affiche 0.2

etc...

Mais en tapant zéro, l'écran renverra t-il une bonne réponse au problème ?

L'ordinateur nous lance un gros mot !

... car une division par zéro est impossible !

Il suffirait de tester si la valeur de A est nulle ou non !

Le programme Basic sera :

```
1Ø INPUT A
2Ø IF A=Ø THEN PRINT "IMPOSSIBLE"
3Ø IF A≠Ø THEN X=1/A:PRINT X
```

Remarque :

Certains ordinateurs se bloquent lorsqu'ils doivent faire une division par zéro, d'autres affichent un message en anglais ; en tout cas, ils s'arrêtent tous de travailler.

On dit en jargon informatique que le programme se plante.

L'instruction IF possède sur de plus en plus d'ordinateurs le complément ELSE (qui veutdire SINON).
La fonction IF THEN ELSE permet l'écriture de programmes structurés tout en améliorant leur lisibilité.

IF	condition(s)	THEN	instruction(s)	ELSE	instruction(s)
Si		alors		sinon	

Votre ordinateur SANYO PHC-25 possède l'instruction ELSE, profitez-en !

Exemple :

```
1Ø INPUT A
2Ø IFA=Ø THEN PRINT "NUL" ELSE PRINT "NON NUL"
```

Si A=0 alors l'ordinateur exécute la ou les instruction(s) après THEN puis il passe à la ligne d'instruction suivante

Si A≠0 l'ordinateur exécute la ou les instruction(s) après ELSE puis il passe à la ligne suivante.

Instruction de fin de traitement

INSTRUCTION END

- Lorsque l'ordinateur rencontre une instruction END, il s'arrête :

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

Dans ce cas, l'ordinateur s'arrête sur l'instruction 3Ø, et les instructions 4Ø et 5Ø ne sont pas exécutées.

- L'instruction END en fin de programme n'est pas obligatoire, lorsque l'ordinateur ne rencontre plus d'instruction, il s'arrête

```
Exemple : 1Ø INPUT A
          2Ø INPUT B
          3Ø X=A+B
          4Ø PRINT X
```

L'ordinateur s'arrêtera après l'instruction 4Ø.

- Pourtant l'instruction END est nécessaire dans certains programmes. Notre programme "Night Club" vu précédemment pourrait alors s'écrire.

```
Par exemple :   1Ø PRINT "VOTRE AGE"
                2Ø INPUT A
                3Ø IF A < 18 THEN PRINT "INTERDIT AUX MINEURS":END
                4Ø PRINT "BIENVENUE"
```

Si on avait omis l'instruction END en ligne 3Ø, l'ordinateur afficherait systématiquement le message BIENVENUE ; ce qui n'a évidemment aucun sens.

Nous avons ici supprimé le test à la ligne 4Ø ; sachant que lorsque la condition n'est pas vérifiée l'ordinateur passe automatiquement à la ligne suivante.

Instruction de branchement

INSTRUCTION GOTO (ALLER À ...)

Jusqu'à présent, nous avons vu que les instructions sont exécutées par l'ordinateur dans l'ordre de leur numérotation ; mais on peut programmer un saut à une ligne d'instruction grâce à l'instruction GOTO.

GOTO 60 par exemple veut dire aller à 6Ø
(qui veut dire saut dans tous les cas)

1) Saut inconditionnel

Exemple : 1Ø PRINT "BONJOUR"
2Ø GOTO 1Ø

Tant qu'on n'arrêtera pas ce programme (en éteignant le courant ou d'une autre manière), ce programme affichera le mot "Bonjour" sans jamais se lasser !!!

On dit en jargon informatique que le programme boucle.

Pour arrêter l'exécution du programme, pressez simultanément sur les touches [CTRL] et [C]

Autre exemple :

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

Le programme nous demandera indéfiniment 2 valeurs et nous affichera leur somme.

2) Saut conditionnel

(qui veut dire saut dans le cas où une condition est vérifiée)

Il peut arriver que l'on veuille sauter à une ligne d'instruction que dans certaines conditions.

Exemple : on demande à l'opérateur de saisir un nombre, l'ordinateur doit afficher l'inverse du nombre.

Si l'opérateur introduit un nombre nul, (ce qui provoquerait une erreur), l'ordinateur lui demande d'introduire un autre nombre.

PROGRAMME :

```
1Ø PRINT "NOMBRE"
2Ø INPUT A
3Ø IF A=Ø THEN GOTO 2Ø
4Ø X=1/A
5Ø PRINT X
```

On voit qu'on peut donc contrôler certaines valeurs introduites au clavier et reboucler sur la lecture du nombre s'il y a erreur de saisie.

Remarque :

Le numéro d'instruction derrière GOTO doit exister dans le programme, sans quoi l'ordinateur ne sait pas où aller.

```
1Ø PRINT "NOMBRE"
2Ø INPUT A
3Ø IF A=Ø THEN GOTO 15
4Ø X=1/A
5Ø PRINT X
```

Ce programme ne peut pas marcher, car la ligne 15 n'existe pas !

Voyons quelques exercices pour bien comprendre l'utilité de cette instruction :

```
1Ø PRINT "VOTRE JOUR DE NAISSANCE"
2Ø INPUT J
3Ø IF J < 1 OR J > 31 THEN PRINT "ERREUR" : GOTO 1Ø
   .
   .
   .
```

Dans l'instruction 3Ø, on indique que si J est inférieur à 1 ou supérieur à 31 (ce qui est impossible), l'ordinateur doit afficher le message ERREUR puis revenir à l'instruction 1Ø, pour redemander le jour !

Compter jusqu'à 5 :

```
1Ø X=1
2Ø PRINT X
3Ø X=X+1
4Ø IF X≤5 THEN GOTO 2Ø
```

Ligne 1Ø : on met X à la valeur initiale 1
(cette phase est encore appelée initialisation de X)

Ligne 2Ø : on affiche X

Ligne 3Ø : on augmente X de 1, on dit encore qu'on incrémente X de 1
(cette phase est encore appelée incrémentation)

Ligne 4Ø : le programme saute à l'instruction 2Ø tant que X est plus petit ou égal à 5 (afin d'afficher les 5 premiers nombres).
Le test de la ligne 4Ø est encore appelé test de fin car on teste si on a fini ou pas.

Cet exercice est, ce qu'on appelle une boucle : une boucle est une partie de programme que l'ordinateur exécute plusieurs fois.

Dans l'exercice, on a exécuté 5 fois les instructions X=X+1 et PRINT X.

Dans une boucle, on trouve toujours :

- un compteur : c'est une variable qui augmente de 1 par exemple à chaque tour, pour qu'on sache toujours où on en est (dans notre cas, il s'agit de X)

- une initialisation du compteur en début de programme, pour commencer à une valeur connue

- un test sur cette variable qui permet de faire plusieurs tours dans le programme.
  On teste la fin de boucle.

```
1Ø X=1
2Ø PRINT "ATTENTION JE COMMENCE"
3Ø PRINT "C'EST MON TOUR";X
4Ø X=X+1
5Ø IF X ≤ 1Ø THEN GOTO 3Ø
6Ø PRINT "J'AI LE VERTIGE"
7Ø END
```

```
1Ø GOTO 1Ø
```

Ce programme ne s'arrêtera jamais !
On dit que l'instruction boucle sur elle-même.

COMPTE A REBOURS

```
1Ø X=1Ø
2Ø PRINT X
3Ø X=X-1
4Ø IF X>Ø THEN GOTO 2Ø
5Ø PRINT "FEU"
```

Les 10 premiers nombres pairs

```
1Ø X=2
2Ø PRINT X
3Ø X=X+2
4Ø IF X≤2Ø THEN GOTO 2Ø
```

Si l'on veut avoir les 1Ø premiers nombres impairs, il suffit de remplacer l'instruction 1Ø par :

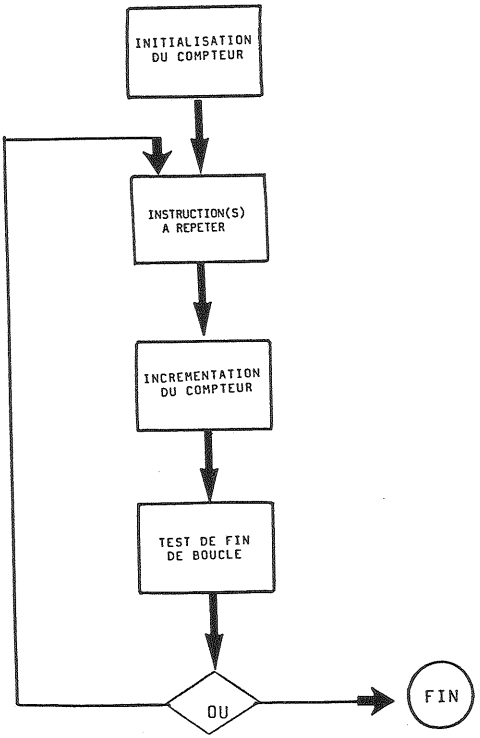
```
1Ø X=1
```

Une punition vite écrite !

```
1Ø I=1
2Ø PRINT "JE NE COPIERAI PLUS"
3Ø I=I+1
4Ø IF I ≤ 1Ø THEN GOTO 2Ø
5Ø END
```

Ligne 1Ø : initialisation du compteur I à 1
Ligne 2Ø : instruction que l'on répétera 1Ø fois
Ligne 3Ø : incrémentation du compteur
Ligne 4Ø : test de fin

Une boucle comprend donc toujours les étapes suivantes :



Les outils de programmation

1. ORGANIGRAMME

Un organigramme permet de visualiser d'une façon schématique et logique le déroulement d'un programme, c'est un outil d'aide INDISPENSABLE pour le programmeur : l'écriture de l'organigramme constitue une étape importante avant l'écriture du programme.

De plus, supposez que vous deviez traduire un programme COBOL en BASIC et que vous ne connaissiez que le BASIC ?

Si vous disposez de l'organigramme du programme COBOL (un organigramme est écrit en français), il suffira de le transcrire en BASIC sans avoir à apprendre le COBOL !

REGLES :

. Un organigramme commence toujours par (début)

... et se termine par une (fin)

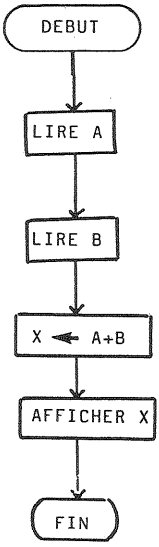
.. Les instructions (ou parfois des groupes d'instructions) sont représentées par des rectangles.
A l'intérieur du rectangle, on indique en français ce que fait l'instruction (ou le groupe d'instructions).

Exemples :

organigramme	devient en	BASIC
Lire A	→	INPUT A
X ← A+B	→	X=A+B
Afficher X	→	PRINT X
Afficher "BONJOUR"	→	PRINT "BONJOUR"

... Des flèches indiquent la séquence des instructions, c'est-à-dire l'ordre dans lequel devront s'exécuter les instructions.

Exemple :



L'ordinateur devra d'abord lire A et B, puis calculer X, puis afficher X.

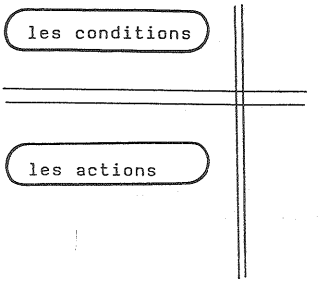
Il est conseillé de toujours faire un organigramme avant de commencer à programmer.

La mise au point des programmes et la compréhension en sera facilitée.

2. TABLES DE DECISIONS

Les tables de décision constituent un second mode de représentation d'un traitement ; on utilise les tables de décision plutôt que les organigrammes dans le cas où un programme comporte beaucoup de tests.

Une table de décision se présente sous forme d'un tableau :



Prenons un exemple :

Calcul d'un nouvel impôt d'un montant I fonction du revenu annuel (R), de l'âge du contribuable (A) et du nombre d'enfants (E)

Le texte dit :

". si le revenu annuel est inférieur ou égal à FRS. 50.000,-, aucun impôt n'est à payer

. si le revenu annuel est supérieur ou égal à FRS. 150.000,-, le montant de l'impôt est de 10% du revenu annuel ; cet impôt sera à payer mensuellement

. si le revenu est compris entre FRS. 50.000,- et 100.000,- le tableau ci-dessous sera à appliquer :

- si l'âge du contribuable est supérieur ou égal à 40 ans :

pas d'enfant à charge	: 7 %
1 enfant à charge	: 6 %
2 enfants à charge	: 5 %
3 enfants à charge	: 4 %
Plus que 3 enfants à charge	: 3 %

- si l'âge du contribuable est inférieur à 40 ans :

pas d'enfant à charge	: 5 %
1 enfant à charge	: 4 %
2 enfants à charge	: 3 %
3 enfants à charge	: 2 %
plus que 3 enfants à charge	: 1 %

Dans tous ces cas, l'impôt est à payer annuellement

. si le revenu est supérieur ou égal à FRS. 100.000,- et inférieur à FRS. 150.000,-, le tableau ci-dessous sera à appliquer :

pas d'enfant à charge	: 9 %
1 enfant à charge	: 8 %
2 enfants à charge	: 7 %
3 enfants à charge	: 6 %
plus que 3 enfants à charge	: 5 %

Dans ce cas, l'impôt est à payer trimestriellement

la table de décision donne :

REVENU	R ⩽ 50000	X																
	R > 50000 ET R ⩽ 100000			X	X	X	X	X	X	X	X	X	X					
	R > 100000 ET R < 150000													X	X	X	X	X
	R ⩾ 150000		X															
AGE	A ⩾ 40			X	X	X	X	X										
	A < 40								X	X	X	X	X					
ENFANTS	E = 0			X					X					X				
	E = 1				X					X					X			
	E = 2					X					X					X		
	E = 3						X					X					X	
	E > 3							X					X					X
MONTANT	1Ø %		X															
	9 %													X				
	8 %														X			
	7 %			X												X		
	6 %				X												X	
	5 %					X			X									X
	4 %						X			X								
	3 %							X			X							
	2 %											X						
	1 %												X					
	Ø %	X																
PAIEMENT	annuel			X	X	X	X	X	X	X	X	X	X					
	trimestriel													X	X	X	X	X
	mensuel		X															

Dans le cas où le problème à étudier est complexe, il est parfois préférable de réaliser une table de décision, avant de dessiner l'organigramme ; en effet, la table de décision permet de contrôler facilement si des cas particuliers n'ont pas été oubliés.

3. COMMENTER UN PROGRAMME ...

L'organigramme et les tables de décision sont des outils d'analyse et se font avant d'écrire le programme ; mais il est bon de mettre des remarques (commentaires) dans le programme afin d'en clarifier la lecture et la compréhension.

L'instruction REM (qui vient de REMARK en anglais) permet d'ajouter du texte, dans le programme sans en modifier l'exécution.

Exemple :

```
1Ø REM CALCUL DE LA SOMME DE 2 NOMBRES
2Ø INPUT A
3Ø INPUT B
4Ø X=A+B
5Ø PRINT X
6Ø END
```

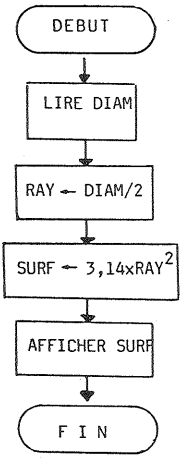
ou encore

```
1Ø REM SOMME DE 2 NOMBRES
2Ø REM LECTURE DES NOMBRES
3Ø INPUT A
4Ø INPUT B
5Ø REM CALCUL-ADDITION
6Ø X=A+B
7Ø REM AFFICHAGE DU RESULTAT
8Ø PRINT X
9Ø END
```

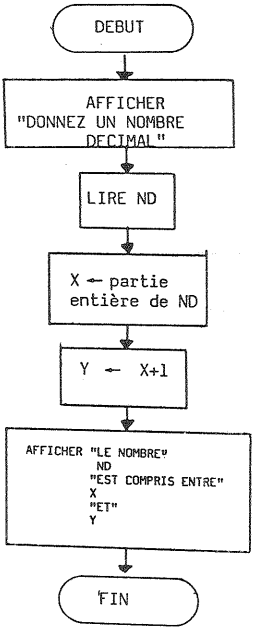
4. EXERCICES D'ORGANIGRAMMES

Nous allons à présent reprendre des programmes du chapitre précédent et représenter leurs organigrammes.

Calcul de la surface d'un cercle :

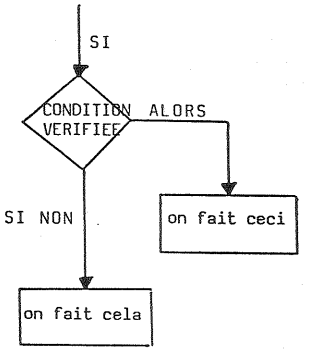


Calcul de l'intervalle d'un nombre décimal

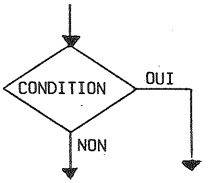


Mais comment allons-nous représenter des tests et des branchements ?

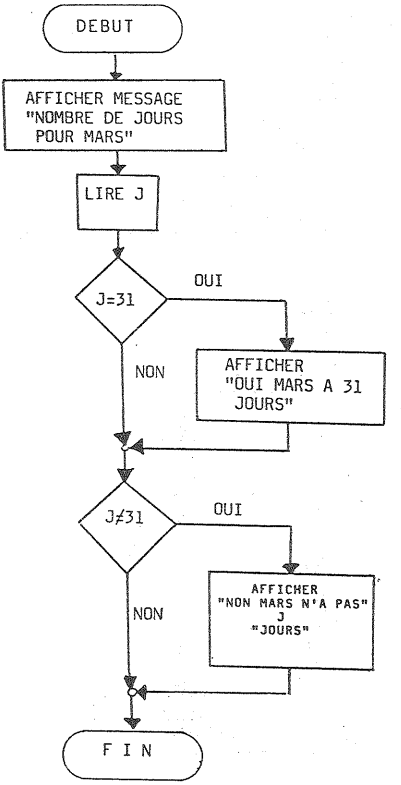
Un test se symbolise par un losange dans lequel on indique la condition :



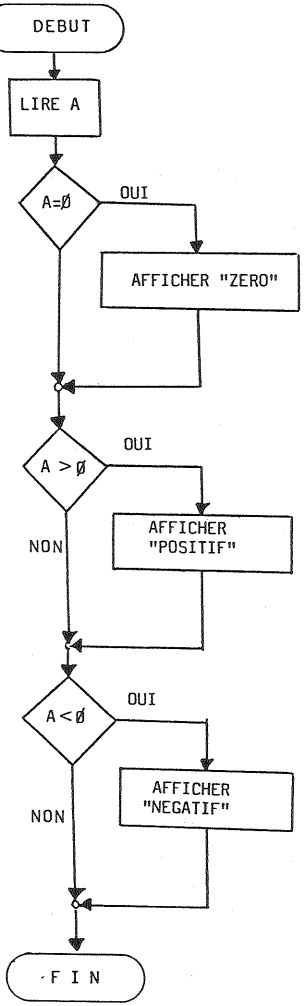
ou plus simplement



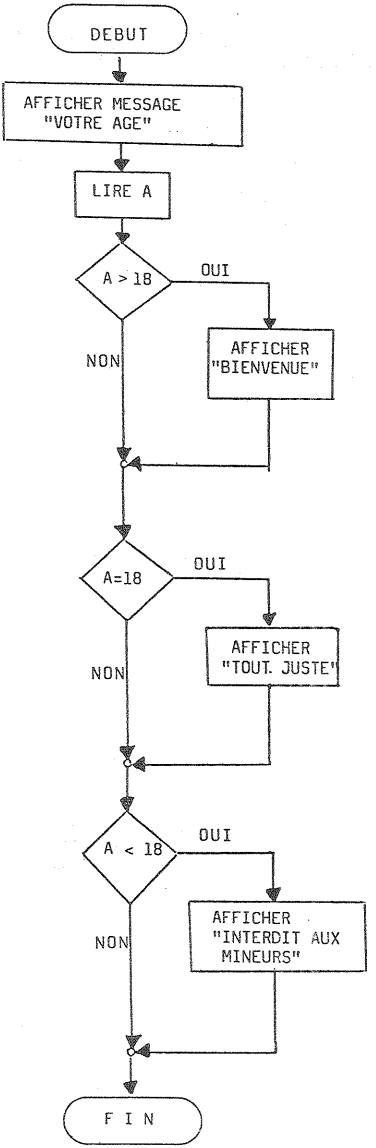
Reprenons le programme qui demandait le nombre de jours dans le mois de mars :



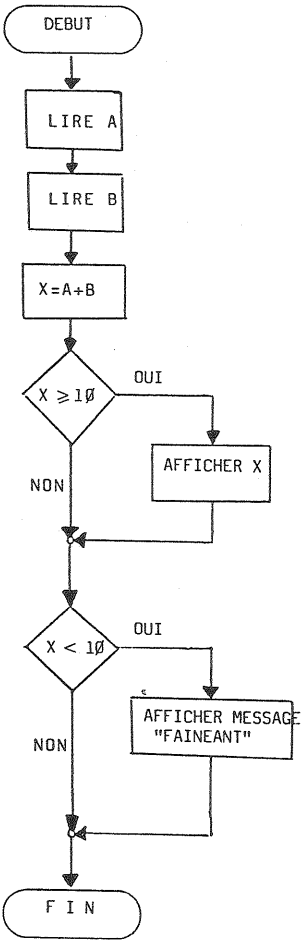
Lire un nombre et afficher s'il est positif, négatif ou nul



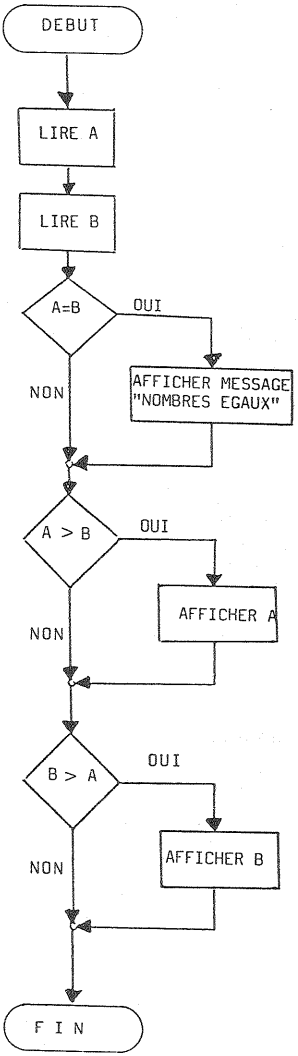
NIGHT CLUB


DEBUT
AFFICHER MESSAGE "VOTRE AGE"
LIRE A
A > 18
OUI
NON
AFFICHER "BIENVENUE"
A=18
OUI
NON
AFFICHER "TOUT. JUSTE"
A < 18
OUI
NON
AFFICHER "INTERDIT AUX MINEURS"
F I N

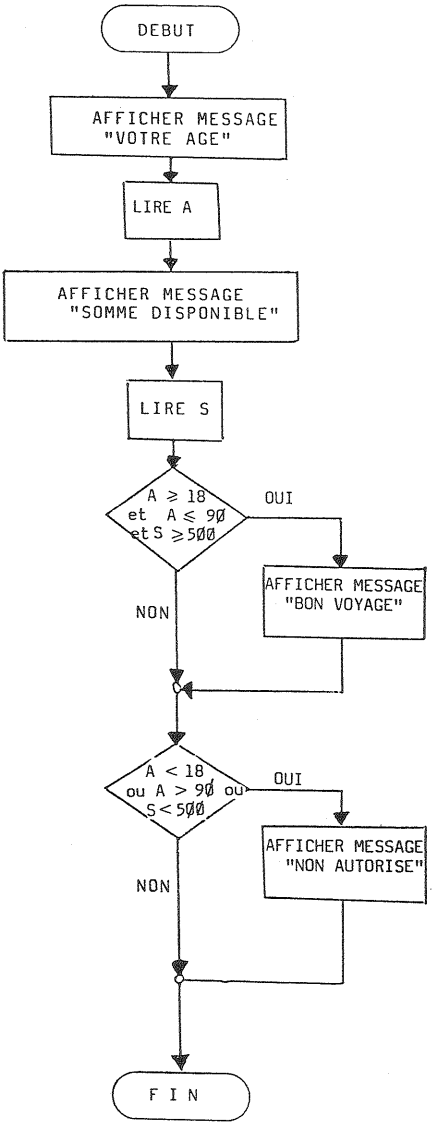
Lire 2 nombres.
Afficher le résultat si la somme des 2 nombres est supérieure ou égale à 1Ø sinon afficher "Fainéant"



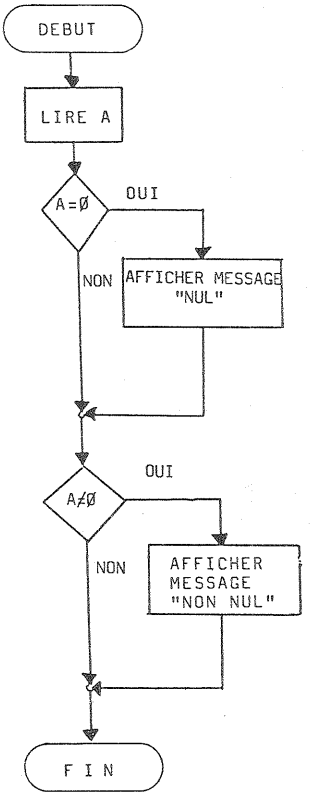
Afficher le plus grand de 2 nombres



Voyage autorisé aux personnes majeures ayant moins de 90 ans et disposant au moins de FRS. 500,-

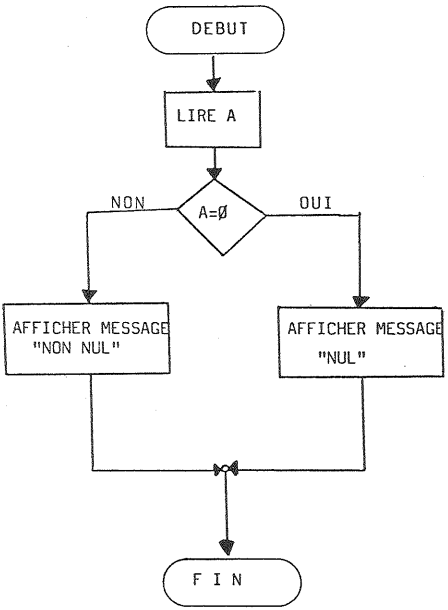


Lire un nombre ; afficher NUL si le nombre = Ø
afficher NON NUL si le nombre ≠ Ø



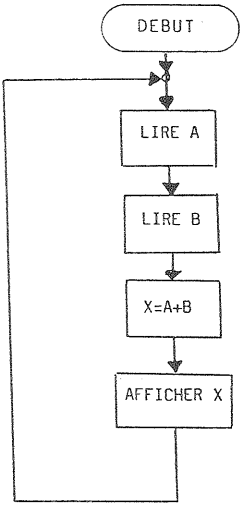
Ce programme s'écrit :

```
1Ø INPUT A
2Ø IF A=Ø THEN PRINT "NUL"
3Ø IF A≠Ø THEN PRINT "NON NUL"
4Ø END
```



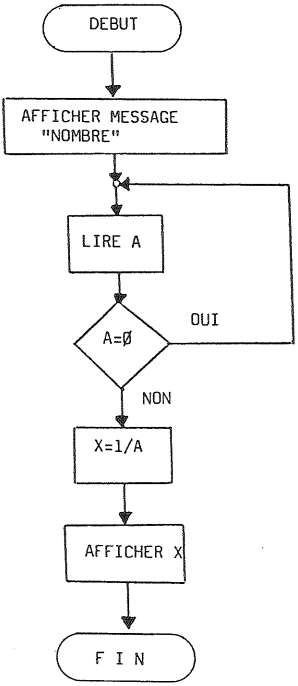
Dans ce cas, vous disposez de l'instruction ELSE (sinon)
et le programme s'écrit :

```
1Ø INPUT A
2Ø IF A≠Ø THEN PRINT "NUL" ELSE PRINT "NON NUL"
3Ø END
```

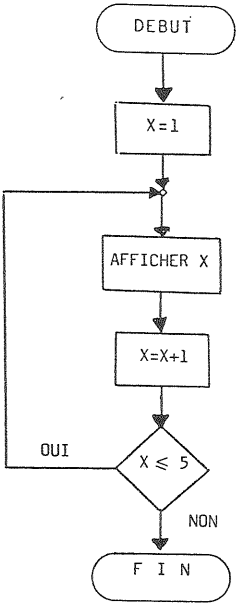


Le programme correspondant à cet organigramme, nous demandera indéfiniment 2 nombres et nous affichera leur somme !

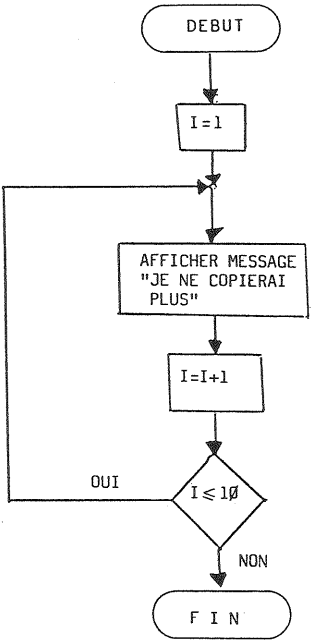
Lire un nombre.
Si ce nombre est nul, redemander le nombre ;
sinon afficher l'inverse du nombre



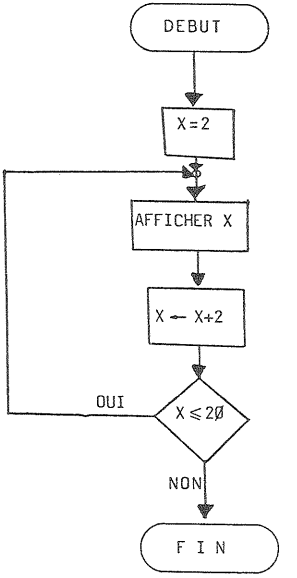
Compter jusqu'à 5 :



La punition vite écrite !



Les 10 premiers nombres pairs !



Nous verrons encore d'autres organigrammes tout au long du manuel.

Exercices

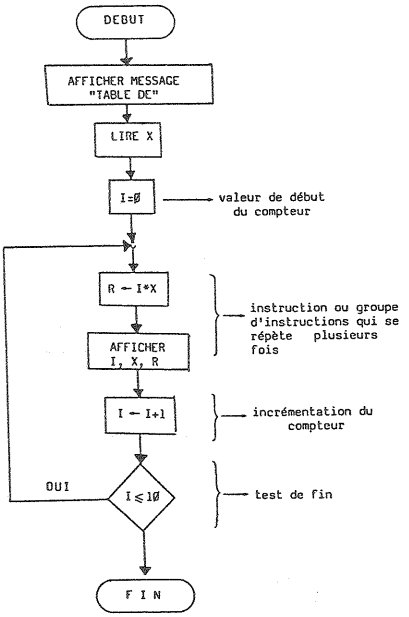
1) TABLE DE MULTIPLICATION

L'ordinateur vous demande un chiffre et affiche la table de multiplication.

Si le chiffre = 3, l'ordinateur doit afficher :

```
Ø*3=Ø
1*3=3
2*3=6
3*3=9
4*3=12
 :
 :
1Ø*3=3Ø
I*X=R
```

Organigramme



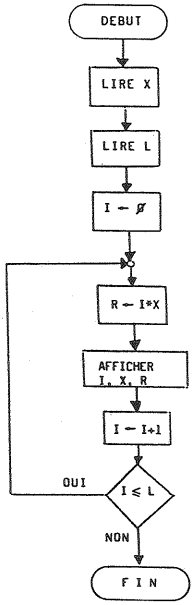
PROGRAMME BASIC :

```
1Ø PRINT "Table de"
2Ø INPUT X
3Ø I=Ø
4Ø R=I*X
5Ø PRINT I;"*";X;"=";R
6Ø I=I+1
7Ø IF I≤ 1Ø THEN GOTO 4Ø
8Ø END
```

AUTRE EXERCICE :

Afficher la table de X sur une longueur L

Organigramme :



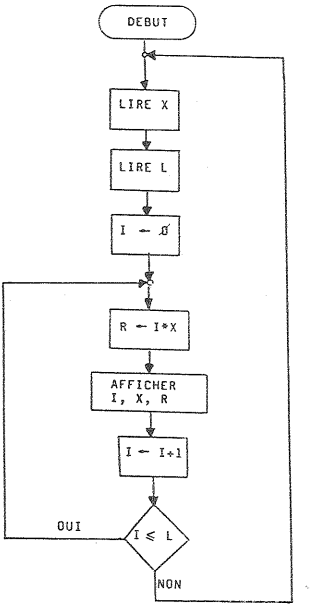
Programme BASIC :

```
10 PRINT "TABLE DE"
20 INPUT X
30 PRINT "LONGUEUR"
40 INPUT L
50 I=0
60 R=I*X
70 PRINT I;"*";X;"=";R
80 I=I+1
90 IF I ≤ L THEN GOTO 60
100 END
```

Il s'agit là encore une fois d'une opération répétitive et on utilise une boucle !

AUTRE EXERCICE :

Afficher la table de multiplication de X sur une longueur L, puis afficher une autre table de multiplication ... et ainsi de suite.




```
1Ø PRINT "Table de"
2Ø INPUT X
3Ø PRINT "LONGUEUR"
4Ø INPUT L
5Ø I=Ø
6Ø R=I*X
7Ø PRINT I;"X";X;"=";R
8Ø I=I+1
9Ø IF  I ≤ L THEN GOTO 6Ø
1ØØ GOTO 1Ø
```

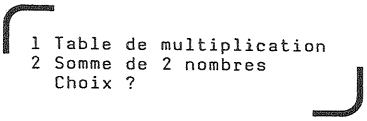
Ce programme vous redemandera sans arrêt une autre table de multiplication, il n'en aura jamais assez de calculer.

2) MENU

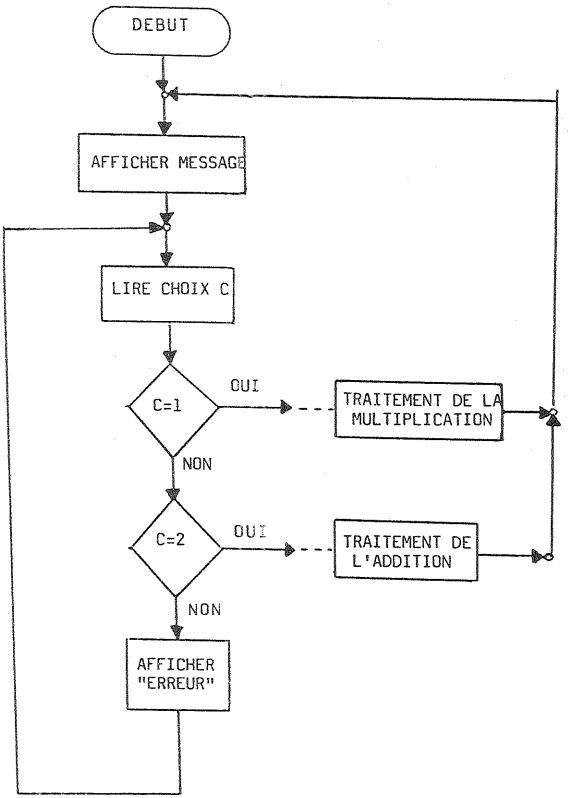
L'ordinateur vous propose de choisir entre 2 possibilités de travail

1 la table de multiplication
2 la somme de 2 nombres

A l'écran s'affiche :



et à vous de rentrer la valeur 1 ou 2 suivant ce que vous voulez lui faire faire.


DEBUT
AFFICHER MESSAGE
LIRE CHOIX C
C=1
OUI
TRAITEMENT DE LA MULTIPLICATION
NON
C=2
OUI
TRAITEMENT DE L'ADDITION
NON
AFFICHER "ERREUR"

```
1Ø PRINT "1=TABLE DE MULTIPLICATION"
2Ø PRINT "2=SOMME DE 2 NOMBRES"
3Ø PRINT "CHOIX "
4Ø INPUT C
5Ø IF C=1 THEN GOTO 1ØØ
6Ø IF C=2 THEN GOTO 2ØØ
7Ø PRINT "ERREUR"
8Ø GOTO 3Ø
```

Saisie du choix

```
1ØØ PRINT "TABLE DE MULTIPLICATION"
11Ø PRINT "TABLE DE"
12Ø INPUT X
13Ø PRINT "LONGUEUR"
14Ø INPUT L
15Ø I=Ø
16Ø R=I*X
17Ø PRINT I;"*";X;"=";R
18Ø I=I+1
19Ø IF I<=L THEN GOTO 16Ø
195 GOTO 1Ø
```

Table de multiplication

```
2ØØ PRINT "SOMME DE 2 NOMBRES"
21Ø INPUT A
22Ø INPUT B
23Ø X=A+B
24Ø PRINT X
25Ø GOTO 1Ø
```

Somme de 2 nombres

3) POUR LES ANGOISSES DE LA BALANCE !

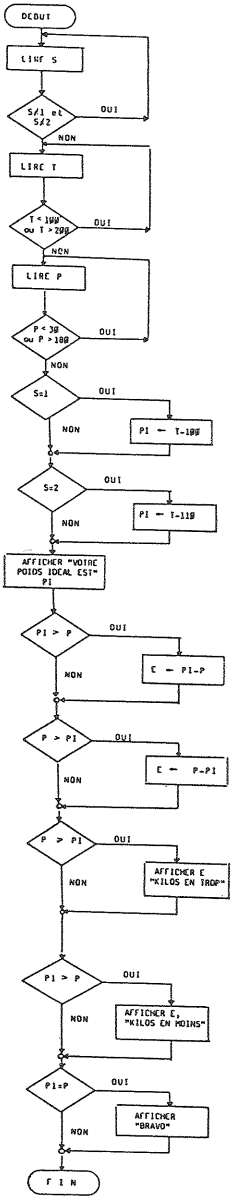
Calcul de votre poids idéal

Soit S votre sexe (1=homme, 2=femme)
Soit T votre taille en cm
Soit P votre poids en kg
Soit PI le poids idéal et E l'écart que l'ordinateur doit calculer

Programme :

```
10 INPUT S
20 IF S≠1 AND S≠2 THEN GOTO 10
30 INPUT T
40 IF T< 100 OR T > 200 THEN GOTO 30
50 INPUT P
60 IF P< 30 OR P > 180 THEN GOTO 50
70 IF S=1 THEN PI=T-100
80 IF S=2 THEN PI=T-110
90 PRINT "VOTRE POIDS IDEAL EST";PI
100 IF PI> P THEN E=PI-P
110 IF P > PI THEN E=P-PI
120 IF P > PI THEN PRINT E;"KG EN TROP !"
130 IF PI>P THEN PRINT E;"KG EN MOINS !"
140 IF PI=P THEN PRINT "BRAVO !"
150 END
```

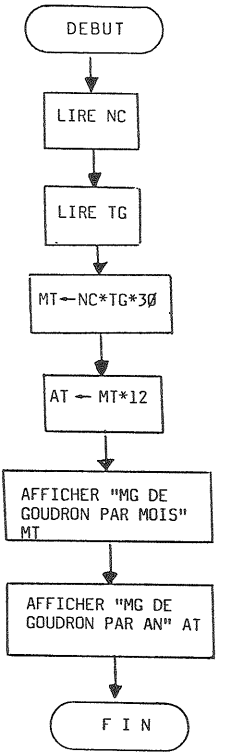
Organigramme



4) DE QUOI GOUDRONNER L'AUTOROUTE !

L'ordinateur demande le nombre de cigarettes que vous fumez par jour (NC), leur teneur en goudron (TG), et vous affiche combien vous absorbez de mg de goudron par mois et par an.

Organigramme :



MT = quantité de goudron totale par mois (en mg)
AT = quantité de goudron totale par an (en mg)

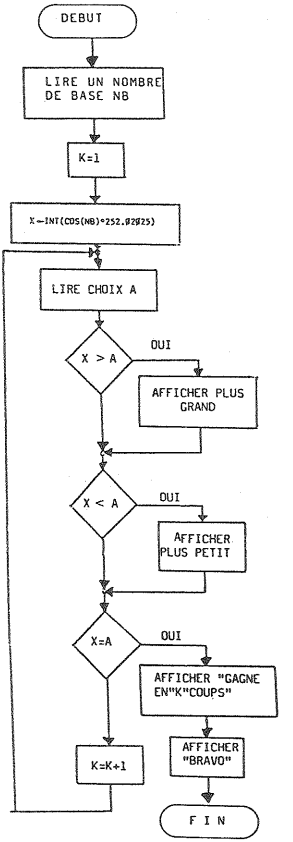
Programme :

```
1Ø PRINT "NB DE CIGARETTES/JOUR"
2Ø INPUT NC
3Ø PRINT "TENEUR(enMG)"
4Ø INPUT TG
5Ø MT=NC*TG*3Ø
6Ø AT=MT*12
7Ø PRINT"GOUDRON ABSORBE/MOIS";MT
8Ø PRINT "GOUDRON ABSORBE/AN";AT
9Ø END
```

5) UN JEU POUR LES PETITS ET LES GRANDS ...

Vous devez deviner le nombre que l'ordinateur a imaginé !

Organigramme :



Programme :

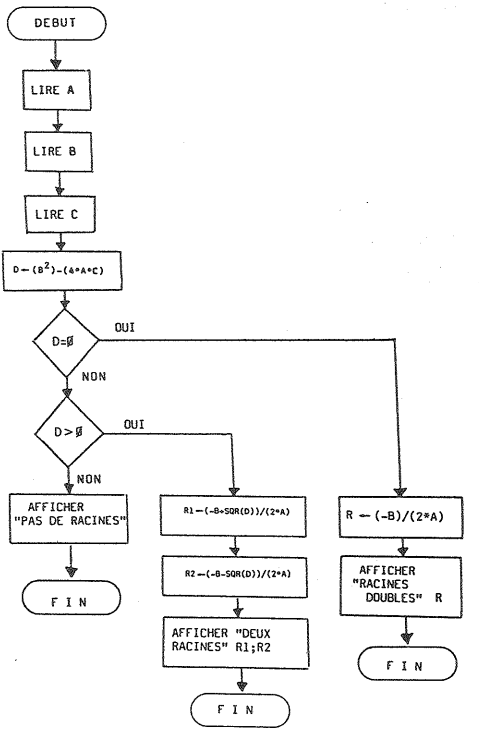
```
1Ø PRINT "NOMBRE DE BASE"
15 INPUT NB
2Ø K=1
3Ø X=INT(COS(NB)*252.02025)
4Ø PRINT "VOTRE CHOIX"
5Ø INPUT A
6Ø IF X>A THEN PRINT "PLUS GRAND"
7Ø IF X<A THEN PRINT "PLUS PETIT"
8Ø IF X=A THEN GOTO 11Ø
9Ø K=K+1
1ØØ GOTO 4Ø
11Ø PRINT "VOUS AVEZ GAGNE";
12Ø PRINT "EN";K;"COUPS"
13Ø PRINT "BRAVO"
14Ø END
```

Attention : Le nombre peut être positif ou négatif !

6) POUR LES MATHEUX !

Résolution d'une équation du 2ème degré du type $Ax^2 + Bx + C$

Organigramme :



Programme :

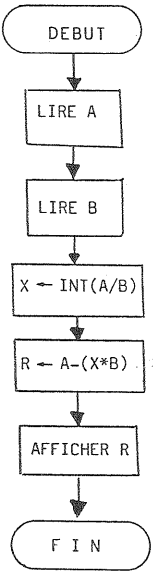
```
 1Ø INPUT A
 2Ø INPUT B
 3Ø INPUT C
 4Ø D=(B^2)-(4*A*C)
 5Ø IF D=Ø THEN GOTO 9Ø
 6Ø IF D>Ø THEN GOTO 12Ø
 7Ø PRINT "PAS DE RACINES"
 8Ø END
 9Ø R=(-B)/(2*A)
1ØØ PRINT "RACINES DOUBLES";R
11Ø END
12Ø R1=(-B+SQR(D))/(2*A)
13Ø R2=(-B-SQR(D))/(2*A)
14Ø PRINT "DEUX RACINES";R1;R2
15Ø END
```

7) LE RESTE

On donne A et B, 2 nombres entiers et l'ordinateur doit calculer le reste (R) de la division de A par B.

Exemple. Si A vaut 8 et B vaut 3 ⟶ R=2
A vaut 15Ø et B vaut 5Ø ⟶ R=Ø
A vaut 7 et B vaut 2 ⟶ R=1

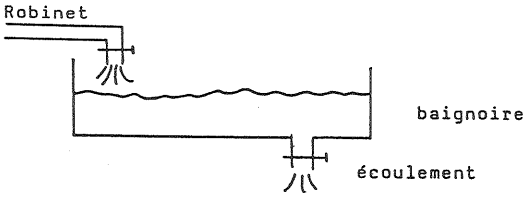
Dans le BASIC, il n'y a pas d'instructions de calcul d'un reste il faut donc se débrouiller avec les instructions existantes.




```
1Ø INPUT A
2Ø INPUT B
3Ø X=INT(A/B)
4Ø R=A-(X*B)
5Ø PRINT R
6Ø END
```

8) BAIGNOIRE

Etudiez ce programme :



DR représente le débit du robinet en $m^{3/h}$
DE représente le débit de l'écoulement en $m^{3:h}$
V représente le volume de la baignoire en m^3

.... Ce programme vous indiquera dans combien de temps vous pourrez prendre votre bain

```
 5 REM BAIGNOIRE
1Ø PRINT "DEBIT DU ROBINET EN M3/H"
15 INPUT DR
2Ø PRINT "DEBIT DE L'ECOULEMENT EN M3/H"
25 INPUT DE
3Ø PRINT "VOLUME DE LA BAIGNOIRE EN M3/H"
35 INPUT V
4Ø IF DR ≠ 0 THEN GOTO 5Ø
43 PRINT "PAS D'EAU,PAS DE BAIN"
47 END
5Ø IF DR>DE THEN GOTO 7Ø
6Ø PRINT "LA BAIGNOIRE RESTERA VIDE"
65 END
7Ø D=DR-DE
8Ø TH=V/D
9Ø TM=INT(TH*60)
1ØØ PRINT "DEBIT EFFECTIF DE REMPLISSAGE:"
1Ø5 PRINT D;" M3/H"
11Ø PRINT "DUREE DE REMPLISSAGE:"
12Ø PRINT TM;" MINUTES"
13Ø END
```