# Initiation au langage Basic GW-Basic Utilisé par PC-Basic

# Les premiers pas

Vous pouvez installer <u>PC-BASIC</u> pour programmer en Basic.

#### PC-BASIC-2.0.6

(Pour Windows 7 et supérieur)

#### Donnez à l'ordinateur votre première commande.

Tapez ceci exactement tel qu'il est ci-dessous :

#### PRINT "Allo je suis ton ordinateur!"

Vérifiez maintenant votre ligne. Avez-vous mis les guillemets là où nous les avons mis?

Si vous avez fait une erreur, pas de problème. Appuyez simplement sur la touche  $\leftarrow$  (Backspace) et le dernier caractère saisi disparaît.

Appuyez à nouveau dessus et l'avant-dernier disparaît (. .. et ainsi de suite et ainsi de suite ...).

Appuyez sur la touche ENTER et regardez. Votre ÉCRAN devrait ressembler à ceci :

OK PRINT "Allo je suis ton ordinateur!" Allo je ton ordinateur! OK

Votre ordinateur vous a juste obéi en imprimant le message que vous avez entré entre guillemets. Demandez-lui d'afficher un autre message.

Taper:

PRINT "2"

Appuyez sur ENTER. L'ordinateur vous obéit à nouveau et affiche votre message suivant : 2

Essayez-en une autre :

PRINT "2 + 2" ENTER

L'ordinateur vous obéit en imprimant : 2+2

Vous attendez sans doute bien plus qu'une mimique électronique... peut-être des réponses ! Donnez à votre ordinateur des chiffres sans les guillemets.

Taper:

PRINT 2 + 2 ENTER

Beaucoup mieux. Cette fois, l'ordinateur affiche la réponse :

4

Les guillemets ont évidemment un sens. Expérimentez avec eux un peu plus. Tapez chacune de ces lignes :

PRINT 5+4 ENTER PRINT "5+4" ENTER PRINT "5+4 Egale" 5+4 ENTER PRINT 6/2 "IS 6/2" ENTER PRINT "8/2" ENTER PRINT 8/2 ENTER

Des conclusions sur ce que font les citations?

Tout problème arithmétique est un jeu d'enfant pour l'ordinateur. Faites une longue division.

Taper:

PRINT "3862 Divisé par 13,2 est" 3862/13.2

Faites un problème de multiplication :

PRINT 1588 \* 23

Notez que le signe de multiplication de l'ordinateur est un astérisque (\*), plutôt que le signe que vous utilisez en mathématiques (X). L'ordinateur est si précis qu'il confondrait le signe de multiplication X avec le caractère alphabétique X.

Essayez quelques problèmes supplémentaires :

PRINT "15 \* 2 =" 15\*2 PRINT 18 \* 18 "Est le carre de 18" PRINT 33.3/22.82

Maintenant c'est votre tour. Écrivez deux lignes de commande qui affichent ces deux problèmes ainsi que leurs réponses :

157113.2= 85\*43=

Si vous utilisez les "bonnes" lignes de commande, voici ce que l'ordinateur affiche sur votre écran :

157 / 13.2 ''' 11.8838384 85 \* 43 = 4085 Prêt pour la réponse:

PRINT "157/13.2 =" 157113.2 PRINT "85 \* 43 =" 95\*113

# Il a ses règles...

À ce jour, l'ordinateur a probablement imprimé de petits messages amusants sur votre écran. Si ce n'est pas le cas, tapez cette ligne, en orthographiant délibérément mal le mot PRINT.

PRJNT "Allo"

L'ordinateur affiche :

Syntax error Ok

C'est la façon dont l'ordinateur dit,

"La commande 'PRJNT' n'est pas dans mon vocabulaire... Je n'ai aucune idée de ce que vous voulez que je fasse."

Chaque fois que vous obtenez l'erreur Syntax error, vous avez probablement fait une sorte d'erreur typographique.

L'ordinateur vous donne également des messages d'erreur lorsqu'il ne comprend pas ce que vous voulez qu'il fasse, mais il a l'impression que vous lui demandez de faire quelque chose d'illogique ou d'impossible.

Par exemple, essayez ceci :

PRINT 5/0

L'ordinateur affiche :

Division by zero Ok

Ce qui signifie "Ne me demandez pas de diviser par 0-c'est impossible !"

Si vous obtenez un message d'erreur que vous ne comprenez pas, allez à l'annexe. Nous y avons répertorié tous les messages d'erreur et ce qui les a probablement causés.

#### C'est aussi un show-off!

# La commande SCREEN et COLOR et PLAY

Pour définir les spécifications d'affichage.

# SCREEN [mode]

#### SCREEN 0

Mode texte uniquement Format texte  $40 \times 25$  ou  $80 \times 25$  avec une taille de boîte de caractères de  $8 \times 8$  ( $8 \times 14$  avec EGA) Affectation de 16 couleurs à l'un des 2 attributs Affectation de 16 couleurs à l'un des 16 attributs (avec EGA)

#### SCREEN 1

Graphiques à résolution moyenne de  $320 \times 200$  pixels Format texte  $40 \times 25$  avec une taille de boîte de caractères de  $8 \times 8$ Affectation de 16 couleurs à l'un des 4 attributs Prend en charge à la fois EGA et CGA 2 bits par pixel

#### SCREEN 2

Graphiques haute résolution  $640 \times 200$  pixels Format texte  $80 \times 25$  avec une taille de boîte de caractères de  $8 \times 8$ Affectation de 16 couleurs à l'un des 2 attributs Prend en charge à la fois EGA et CGA 1 bit par pixel

#### SCREEN 7

Graphiques à résolution moyenne de 320 × 200 pixels Format texte 40 × 25 avec une taille de boîte de caractères de 8 × 8 2, 4 ou 8 pages de mémoire avec respectivement 64K, 128K ou 256K de mémoire installées sur l'EGA Affectation de l'une des 16 couleurs à 16 attributs EGA requis 4 bits par pixel

#### **SCREEN 8**

Graphiques haute résolution 640 × 200 pixels Format texte 80 × 25 avec une taille de boîte de caractères de 8 × 8 1, 2 ou 4 pages de mémoire avec respectivement 64K, 128K ou 256K de mémoire installées sur l'EGA Affectation de l'une des 16 couleurs à 16 attributs EGA requis 4 bits par pixel

#### **SCREEN 9**

Graphiques à résolution améliorée de 640 × 350 pixels Format texte 80 × 25 avec une taille de boîte de caractères de 8 × 14 Affectation soit de 64 couleurs à 16 attributs (plus de 64K de mémoire EGA), soit de 16 couleurs à 4 attributs (64K de mémoire EGA) Deux pages d'affichage si 256 Ko de mémoire EGA sont installés EGA requis 2 bits par pixel (mémoire EGA 64K) 4 bits par pixel (mémoire EGA supérieure à 64K)

# COLOR [foreground][,[background]

#### Couleur

- 0 Noir
- 1 Bleu
- 2 Vert
- 3 Cyan 4 Rouge
- 4 Rouge
- 5 Magenta
- 6 Marron
- 7 Blanc
- 8 Gris
- 9 Bleu clair
- 10 Vert clair
- 11 Cyan clair
- 12 Rouge clair
- 13 Magenta clair
- 14 Jaune
- 15 Blanc haute intensité

# Exemple

# SCREEN 9

COLOR 1,3 (Caractères Bleu sur fond Cyan)

COLOR 7,1 (Caractères Blancs sur fond Bleu)

COLOR 7,0 (Caractères Blancs sur fond Noir)

# Son de l'ordinateur Off, Un, Deux •••

# SOUND freq, durée

freq est la fréquence souhaitée en Hertz (cycles par seconde). freq est une expression numérique comprise entre 37 et 32767.

durée est la durée souhaitée en impulsions d'horloge. Les tics d'horloge se produisent 18,2 fois par seconde. durée doit être une expression numérique comprise entre 0 et 65535.

Les valeurs inférieures à 0,022 produisent un son infini jusqu'à ce que la prochaine instruction SOUND ou PLAY soit exécutée.

Si la durée est zéro, toute instruction SOUND active est désactivée.

Si aucune instruction SOUND n'est en cours d'exécution, une durée de zéro n'a aucun effet.

Le son est exécuté en avant-plan ou en arrière-plan selon l'instruction PLAY.

Tapez ceci :

# SOUND 37,6

Si vous n'entendez rien, augmentez le volume et réessayez.

Ce que vous entendez correspond à 6 secondes de la tonalité la plus basse que l'ordinateur peut émettre.

Qu'en est-il du ton le plus élevé ?

Taper:

SOUND 255,50

Note DO (C)

SOUND 138.810,10

#### do(C), ré(D), mi(E), fa(F), sol(G), la(A), si(B)

C 130.810 C 523.250	D 146.830 D 587.330
E 164.810 E 659.260	F 174.610 F 698.460
G 196.000 G 783.990	A 220.000 A 880.000
B 246.940 B 987.770	C 261.630 C 1046.500
D 293.660 D 1174.700	E 329.630 E 1318.500
F 349.230 F 1396.900	G 392.000 G 1568.000
A 440.000 A 1760.000	B 493.880 B 1975.500

# PLAY

# PLAY "MBABCDABCDABCD"

# PLAY "L4 ABGD"

A-G [#,+,-] A-G sont des notes. # ou + après une note produit un dièse ; - produit un plat.

Toute note suivie de #, + ou - doit faire référence à une touche noire sur un piano.

L(n)

Définit la longueur de chaque note. L4 est une noire, L1 est une ronde, et ainsi de suite. n peut être compris entre 1 et 64.

La longueur peut également suivre la note pour modifier la longueur de cette note uniquement. A16 est équivalent à L16A.

MF Musique de premier plan. Les instructions PLAY et SOUND doivent être exécutées au premier plan. C'està-dire que chaque note ou son suivant n'est pas démarré tant que la note ou le son précédent n'est pas terminé. Il s'agit de la valeur par défaut initiale.

Fond de musique MB. Les instructions PLAY et SOUND doivent s'exécuter en arrière-plan. C'est-à-dire que chaque note ou son est placé dans un tampon permettant au programme BASIC de continuer son exécution pendant que la musique joue en arrière-plan. Jusqu'à 32 notes (ou silences) peuvent être jouées en arrière-plan en même temps.

MN Musique normale. Chaque note joue sept huitièmes du temps déterminé par L (longueur).

ML Music legato. Chaque note joue la période complète définie par L.

MS Musique saccadée. Chaque note joue les trois quarts du temps déterminé par L.

N(n) Joue la note n. n peut varier de 0 à 84. Dans les 7 octaves possibles, il y a 84 notes. n mis à 0 indique un silence.

O(n) Octave 0 définit l'octave actuelle. Il y a 7 octaves (0 à 6). La valeur par défaut est 4. Le do médian est au début de l'octave 3.

P(n) Pause. P peut varier de 1 à 64.

T(n) Tempo. T définit le nombre de L4 en une minute. n peut varier de 32 à 255. La valeur par défaut est 120.

. (période)

Une période après une note augmente le temps de jeu de la note de 3/2 fois la période déterminée par L (longueur de note) fois T (tempo). Plusieurs périodes peuvent apparaître après une note, et le temps de jeu est mis à l'échelle en conséquence.

Par exemple, A. fera jouer la note A une fois et demie le temps de jeu déterminé par L (longueur de la note) fois T (le tempo); deux points placés après A (A..) feront jouer la note à 9/4 fois sa valeur attribuée ; un A à trois points (A...) à 27/8, etc.

Des points peuvent également apparaître après un P (pause) et augmenter la durée de la pause comme décrit cidessus.

Xstring;

Exécute une sous-chaîne, où chaîne est une variable affectée à une chaîne de commandes PLAY.

En raison de la lenteur du taux d'interruption de l'horloge, certaines notes ne jouent pas à des tempos plus élevés ; par exemple, 1,64 à T255. Ces combinaisons note/tempo doivent être déterminées par l'expérimentation.

Votre ordinateur n'oublie jamais (... sauf si vous l'éteignez...)

>n Un symbole supérieur à précédant la note n joue la note dans l'octave immédiatement supérieure. <n Un symbole inférieur à précédant la note n joue la note dans l'octave inférieure suivante.

# A C P A A K5 Z H C 0 1 80.0 1</td

Une compétence qui rend votre ordinateur si puissant est sa "mémoire".

Faites-lui "souvenir" du nombre 13.

Taper:

A = 13

Maintenant "confondez" l'ordinateur en tapant ce que vous voulez.

Lorsque vous avez terminé, appuyez sur ENTER.

Voyez si l'ordinateur se souvient de ce que signifie A en tapant :

PRINT A

Votre ordinateur se souvient que A est 13 tant que vous le rester allumé ... ou jusqu'à ce que vous fassiez cela.

Taper:

A =17.2

Si vous lui demandez PRINT A maintenant, il affiche 17.2

Vous n'êtes pas obligé d'utiliser la lettre A. Vous pouvez utiliser n'importe quelle lettre de A à Z. En fait, vous pouvez utiliser n'importe quelle lettre de A à Z.

Taper:

B =15 C =20 BC = 25

Faites-lui afficher tous les nombres que vous lui avez demandé de retenir.

Taper:

PRINT A, B, C, BC

Si vous voulez que l'ordinateur mémorise une "chaîne" de lettres ou de chiffres, utilisez une lettre avec un signe de dollar (\$).

Taper:

A\$ = "ESSAYER DE" B\$ = "VOUS SOUVENIR" C\$ = "DE CECI, VOTRE" BC\$ = "GRAND ORDINATEUR"

Tapez ensuite :

PRINT A\$, B\$, C\$, BC\$

Les "types d'ordinateurs" ont un nom pour toutes les lettres que vous avez utilisées : "variables". Jusqu'à présent, vous avez utilisé ces variables :

LA MÉMOIRE DE VOTRE ORDINATEUR		
CHIFFRES	CARACTÈRES	
$A \rightarrow 17.2$ $B \rightarrow 15$ $C \rightarrow 20$ $BC \rightarrow 25$	A\$ → "ESSAYEZ DE" B\$ → "VOUS SOUVENIR" C\$ → "DE CECI, VOTRE" BC\$ → "GRAND ORDINATEUR"	

Vérifiez ponctuellement les variables ci-dessus pour voir si l'ordinateur se souvient de la bonne information. Par exemple, pour voir si BC contient toujours 25, tapez :

# PRINT BC

Considérez les variables comme de petites boîtes dans lesquelles vous pouvez stocker des informations. Un ensemble de boîtes est pour les chaînes ; l'autre ensemble est pour les nombres. Chaque boîte a une étiquette.

#### L'ordinateur est pointilleux sur ses règles

Pensez-vous que l'ordinateur accepte ces lignes ?

D = "G" Z = "CECI SONT DES DONNÉES DE CHAÎNE"

L'ordinateur répond aux deux lignes ci-dessus par ?

TYPE MISMATCH ERROR. Il vous dit que vous devez respecter ses règles.

Les règles "ignorées" par les lignes ci-dessus sont :

# RÈGLES SUR LES DONNÉES DE CHAINE

(1) Toutes les données entre guillemets sont STRING DATA.

(2) Vous ne pouvez affecter STRING DATA qu'aux variables AVEC UN SIGNE \$.

Pour que les lignes ci-dessus obéissent aux règles de l'ordinateur, utilisez un signe dollar avec le D et le Z.

Tapez :

D\$ = "G" Z\$ = "CECI SONT DES DONNÉES DE CHAÎNE"

L'ordinateur accepte maintenant ces lignes.

Que diriez-vous de cette ligne?

Pensez-vous que l'ordinateur l'accepte ?

D\$ = G

La ligne ci-dessus a ignoré ces règles :

# **RÈGLES SUR LES DONNÉES NUMÉRIQUES**

(1) Les nombres non entre guillemets sont des DONNÉES NUMÉRIQUES.

(2) Les données numériques ne peuvent être affectées qu'aux variables SANS SIGNE \$.

Tapez ceci, que l'ordinateur accepte :

D = Gz = 1

Vous l'avez maintenant ajouté à la mémoire de votre ordinateur.

LA MÉMOIRE DE VOTRE ORDINATEUR		
CHIFFRES	CARACTÈRES	
$D \rightarrow 6$ $Z \rightarrow 12$	D\$ → "6" Z\$ → "CECI SONT DES DONNÉES DE CHAÎNE"	

Maintenant, faites quelque chose d'intéressant avec ce que vous avez demandé à l'ordinateur de retenir.

Taper:

PRINT D \* 2

L'ordinateur imprime le produit de D fois 2.

Essayez cette ligne :

PRINT Z/D

L'ordinateur imprime le quotient de Z divisé par D.

Cela fonctionnerait-il?

PRINT D\$ \* 2

Est-ce que vous l'avez essayé?

Cela oblige l'ordinateur à imprimer le même ? TYPE MISMATCH ERROR.

Il ne peut pas multiplier les données de chaîne.

Barrez ci-dessous les commandes que l'ordinateur rejette :

#### EXERCICE AVEC VARIABLES

F = 22.9999988

M = "19.2"

DZ\$ = "RAPPELER VOUS DE CA POUR MOI"

M\$ = 15

Z = F + F

Achevé?

Voici les commandes que l'ordinateur accepte.

F = 22.9999988

DZ\$ = "RAPPELER VOUS DE CA POUR MOI"

Z = F + F

# **RÈGLES SUR LES VARIABLES**

Vous pouvez utiliser deux caractères quelconques de A à Z pour une variable. Le premier caractère doit être une lettre de A à Z ; cependant, la seconde peut être soit un chiffre, soit une lettrer.

Si vous souhaitez lui attribuer des données de chaîne, mettez un signe dollar après. Sinon, il ne peut contenir que des données numériques.

#### Regarde comme c'est facile?



# Tapez NEW

Cela efface tout ce qui peut se trouver dans la "mémoire" de l'ordinateur. Arrête l'exécution d'un programme, supprime le programme en mémoire, exécute CLEAR et RESTORE et rend le contrôle à l'utilisateur.

Tapez maintenant cette ligne. Assurez-vous de taper le numéro 10 en premier, c'est important.

# 10 PRINT "ALLO, JE SUIS TON ORDINATEUR"

Vous avez appuyé sur ENTER ? Il ne s'est rien passé, n'est-ce pas ?

Rien que vous puissiez voir, c'est-à-dire. Vous venez de taper votre premier programme.

Taper:

RUN et appuyez sur ENTER

L'ordinateur exécute docilement votre programme.

Tapez RUN encore et encore au contenu de votre cœur. L'ordinateur exécute votre programme à tout moment, autant de fois que vous le souhaitez.

Puisque cela fonctionne si bien, ajoutez une autre ligne au programme.

Taper:

20 PRINT "QUEL EST VOTRE NOM?"

Tapez maintenant :

LIST

Votre ordinateur est docilement liste votre programme en entier. Votre écran devrait ressembler exactement à ceci :

10 PRINT "ALLO, JE SUIS TON ORDINATEUR" 20 PRINT "QUEL EST VOTRE NOM?"

Que pensez-vous qu'il se passera lorsque vous l'exécuterez ? Essayez-le. Taper: RUN

L'ordinateur affiche :

ALLO, JE SUIS TON ORDINATEUR QUEL EST VOTRE NOM?

Répondez à la question de l'ordinateur, puis appuyez sur (ENTER.... Quoi ?

Il y a encore une erreur ? Syntax error

Lorsque vous tapez simplement votre nom, l'ordinateur ne comprend pas ce que vous voulez dire. En fait, l'ordinateur ne peut rien comprendre à moins que vous ne lui parliez à sa façon.

Utilisez un mot que l'ordinateur comprend : INPUT.

Tapez cette ligne :

30 INPUT A\$

Cela indique à l'ordinateur de s'arrêter et d'attendre que vous tapiez quelque chose, qu'il appelle A\$. Ajoutez une ligne de plus au programme :

40 PRINT "Allo," A\$

Maintenant, listez à nouveau le programme pour voir si le vôtre ressemble au mien. Taper: LIST

Votre programme devrait ressembler à ceci :

10 PRINT "ALLO, JE SUIS TON ORDINATEUR" 20 PRINT "QUEL EST VOTRE NOM?" 30 INPUT A\$ 40 PRINT "Allo ," A\$

Pouvez-vous deviner ce qui se passera lorsque vous l'exécuterez ? Essayez-le : RUN

Cela a bien fonctionné, n'est-ce pas?

C'est probablement ce qui s'est passé lorsque vous avez exécuté le programme (selon ce que vous avez tapé comme nom) :

ALLO, JE SUIS TON ORDINATEUR QUEL EST VOTRE NOM? ? Moi et Toi Allo,Moi et Toi Ok

EXÉCUTEZ à nouveau le programme en utilisant des noms différents ...

(L'ordinateur ne se soucie pas de comment vous vous appelez.)

Voici ce que la ligne 30 a fait à la mémoire de votre ordinateur chaque fois que vous avez exécuté le programme (en supposant que vous lui ayez donné les mêmes noms que nous) :



Il existe un moyen plus simple d'exécuter votre programme encore et encore sans avoir à taper la commande RUN.

Tapez cette ligne :

50 GOTO 10

Maintenant, lancez-le. Le programme s'exécute encore et encore sans s'arrêter. GOTO dit à l'ordinateur de revenir en arrière à la ligne 10 :

<b>≥</b> 1Ø	PRINT "
20	PRINT "
30	INPUT A
40	PRINT "I
50	GOTO 10

Votre programme s'exécute désormais en permanence. Chaque fois qu'il arrive à la ligne 50, il revient à la ligne 10. Nous appelons cela une boucle "loop".

La seule façon d'arrêter cette boucle sans fin est d'appuyer sur la touche BREAK ou CTRL + C pour PC-Basic.

#### Mettez toute l'attention sur votre nom

Changez la ligne 50 pour donner à votre nom l'attention qu'il mérite. Comment modifier une ligne de programme ?

Tapez-le simplement à nouveau, en utilisant le même numéro de ligne.

Taper:

50 GOTO 40

Voici à quoi ressemble le programme maintenant :

10 PRINT "F 20 PRINT "F 30 INPUT A1 C40 PRINT "F 50 GOTO 40 Tapez RUN et regardez ce que fait cette boucle. Lorsque vous en avez assez vu, appuyez sur la touche Break (CTRL + C).

Il y a un grand changement que vous pouvez faire simplement en ajoutant une virgule ou un point-virgule. Essayez d'abord la virgule. Tapez à nouveau la ligne 40, mais avec une virgule à la fin :

## 40 PRINT A\$,

Exécutez le programme. La virgule semble tout imprimer sur deux colonnes. Appuyez sur Break (CTRL + C) et essayez le point-virgule.

Taper:

# 40 PRINT A\$;

et exécutez (RUN) ... Vous ne pourrez probablement pas savoir ce que fait le programme tant que vous n'aurez pas appuyé sur Break (CTRL + C). Vous voyez comment le point-virgule rassemble tout ?

# **RÈGLES SUR LA PONCTUATION IMPRIMÉ**

C'est ce que la ponctuation à la fin d'une ligne PRINT fait faire à l'ordinateur :

1. Une virgule fait passer l'ordinateur à la colonne suivante. Utilisez-le pour imprimer en colonnes.

2. Un point-virgule fait que l'ordinateur reste là où il est. Utilisez-le pour « entasser » ce que vous imprimez ensemble.

3. Aucune ponctuation fait passer l'ordinateur à la ligne suivante. Utilisez-le imprimés en rangées.

#### Démonstration couleur/son

Envie de jouer un peu plus avec la couleur et le son ?

Tout d'abord, effacez la mémoire.

Rappelez-vous comment? (NEW)

Entrez ensuite ce programme :

10 PRINT "POUR ME FAIRE CHANGER DE TON" 20 PRINT "SAISISSEZ UN NOMBRE DE 1 A 255" 30 INPUT T 40 SOUND T,50 50 GOTO 10

Exécutez le programme pour obtenir un échantillon des tonalités de l'ordinateur.

BUG; Si vous obtenez une erreur Syntax error lorsque vous exécutez ce programme, c'est que vous entrez un nombre en dehors des valeurs acceptées. Cette erreur, comme toutes les erreurs, empêchera l'ordinateur d'exécuter le programme.

#### Syntaxe de SOUND

#### SOUND freq, durée

freq est la fréquence souhaitée en Hertz (cycles par seconde). freq est une expression numérique comprise entre 37 et 32767.

durée est la durée souhaitée en impulsions d'horloge. Les tics d'horloge se produisent 18,2 fois par seconde. durée doit être une expression numérique comprise entre 0 et 65535.

Les valeurs inférieures à 0,022 produisent un son infini jusqu'à ce que la prochaine instruction SOUND ou PLAY soit exécutée.

Si la durée est zéro, toute instruction SOUND active est désactivée. Si aucune instruction SOUND n'est en cours d'exécution, une durée de zéro n'a aucun effet.

Le son est exécuté en avant-plan ou en arrière-plan selon l'instruction PLAY.

Que se passe-t-il si vous modifiez la ligne 40 pour

#### 40 SOUND 50,T

ASTUCE : revenez au chapitre Les premiers pas où nous parlons du SON.

Vous connaissez la réponse?

Si vous faites le changement ci-dessus, l'ordinateur émet la même tonalité à chaque fois, mais pour une durée différente, selon le nombre que vous utilisez.

#### FAITES VOTRE PROPRE PROGRAMME

Appuyez d'abord sur BREAK (CTRL+C), puis effacez ce programme en tapant NEW. Voyez maintenant si vous pouvez écrire un programme, similaire à celui ci-dessus, pour que l'ordinateur affiche une certaine couleur. N'oubliez pas qu'il y a 9 couleurs, de 0 à 8.

ASTUCE : la ligne 40 pourrait être ; 40 CLS(T)

Voici notre programme :

5 SCREEN 9:COLOR 7,0:CLS 10 PRINT "POUR ME FAIRE CHANGER DE COULEUR" 20 PRINT "TAPEZ UN CHIFFRE ENTRE 1 ET 8" 30 INPUT T 40 COLOR T,0:CLS 50 GOTO 10

Ajouter de la politesse au programme.

Appuyer sur la touche BREAK (CTRL+C) est une façon bâclée d'arrêter le programme de s'exécuter.

Pourquoi ne pas demander poliment à l'ordinateur si vous êtes prêt à terminer le programme ?

Remplacez la ligne 50 du programme ci-dessus par :

# 50 PRINT "VOUS VOULEZ VOIR UNE AUTRE COULEUR?"

Ajoutez ensuite ces lignes :

60 INPUT R\$ 70 IF R\$ = "OUI" THEN 20

Exécutez le programme.

Tapez OUI et le programme continue de fonctionner.

Tapez autre chose et le programme se termine.

Voici à quoi ressemble le programme maintenant :

5 SCREEN 9:COLOR 7,0:CLS 10 PRINT "POUR ME FAIRE CHANGER DE COULEUR" 20 PRINT "TAPEZ UN CHIFFRE ENTRE 1 ET 8" 30 INPUT T 40 COLOR T,0:CLS 50 PRINT "VOUS VOULEZ VOIR UNE AUTRE COULEUR?" 60 INPUT R\$ 70 IF R\$ = "OUI" THEN 20

Voici ce que font les nouvelles lignes :

la ligne 50 imprime une question.

la ligne 60 dit à l'ordinateur de s'arrêter et d'attendre une réponse : R\$.

la ligne 70 indique à l'ordinateur de revenir à la ligne 20 SI (et seulement si) votre réponse (R\$) est "oui".

Sinon, le programme se termine, puisqu'il a plus de lignes.

Vous avez couvert beaucoup de terrain dans cette section. J'espère que nous ne faisons qu'aiguiser votre appétit pour plus. Ne vous inquiétez pas si vous ne le comprenez pas encore parfaitement. Profitez simplement de votre ordinateur.

Ne vous inquiétez pas de IF ... THEN pour le moment. Nous y consacrerons dans une autre section plus loin.

#### **Compter le battement**



Dans ce chapitre, vous expérimenterez des effets sonores d'ordinateur. Avant de faire cela, vous devez apprendre à l'ordinateur à compter.

Taper:

10 FOR X = 1 TO 10 20 PRINT "X = " X 30 NEXT X 40 PRINT "J'AI FINI DE COMPTER"

Exécutez le programme. (Rappelez-vous taper NEW avant de taper un nouveau programme.)

Exécutez le programme plusieurs fois.

A chaque fois, remplacez la ligne 10 par l'une de ces lignes :

10 FOR X := 1 TO 100 10 FOR X := 5 TO 15 10 FOR X := **-2** TO 2 10 FOR X := 20 TO 24

Voyez-vous ce que FOR et NEXT font faire à l'ordinateur ?

Ils le font compter.

Regardez le dernier programme que nous vous avons suggéré d'essayer :

10 FOR X := 20 TO 24 20 PRINT "X = " X 30 NEXT X 40 PRINT "J'AI FINI DE COMPTER"

La ligne 10 indique à l'ordinateur que le premier nombre doit être 20 et le dernier doit être 24. Il utilise X pour étiqueter tous ces nombres.

La ligne 30 indique à l'ordinateur de continuer à revenir à la ligne 10 pour le nombre suivant - **le NEXT X** - jusqu'à ce qu'il atteigne le dernier nombre (nombre 24).

Regardez la ligne 20. Puisque la ligne 20 est entre les lignes FOR et NEXT, l'ordinateur doit imprimer la valeur de X chaque fois qu'il compte :

X = 21 X = 21 X = 22 X = 23X = 24

Ajoutez une autre ligne entre FOR et NEXT :

15 PRINT" ... CALCUL ..."

et exécutez le programme. À chaque décompte, votre ordinateur exécute toutes les lignes que vous choisissez d'insérer entre FOR et NEXT.

Programme, Faite le vous même 4-1 Écrivez un programme qui oblige l'ordinateur à imprimer votre nom 10 fois.

CONSEIL : Le programme doit compter jusqu'à 10.

Programme, Faite le vous même 4-2 Écrivez un programme pour afficher le table de multiplication de 9 (9\*1 à 9\*10).

ASTUCE : PRINT 9\*X est une ligne de programme parfaitement légitime.

Programme, Faite le vous même 4-3 Écrivez un programme qui affiche les tables de multiplication de 9\*1 à 9\*25.

ASTUCE : En ajoutant une virgule dans la ligne PRINT, vous pouvez afficher tous les problèmes et résultats sur votre écran en même temps.

Achevé?

Voici nos programmes :

Program 4-1 10 FOR X = 1 TO 10 20 PRINT "THOMAS" 30 NEXT X Program 4-2 10 FOR X= 1 TO 10 20 PRINT "9 \* " X "=" 9\*X 30 NEXT X Program 4-3 10 FOR X = 1 TO 25 20 PRINT "9 \* " X "=" 9\*X,

30 NEXT X

# **Compter par deux**

Maintenant, faites en sorte que l'ordinateur compte un peu différemment.

Effacez votre programme en tapant **NEW** puis tapez le programme original, en utilisant une nouvelle ligne 10 :

10 FOR X = 2 TO 10 STEP 2 20 PRINT "X = " X 30 NEXT X 40 PRINT "J'AI FINI DE COMPTER"

Exécutez le programme.

Voyez-vous ce que fait le STEP 2 ?

Il fait compter l'ordinateur par 2.

La ligne 10 indique à l'ordinateur que :

Le premier X est 2 Le dernier X est 10

... ET ÉTAPE 2 ... Tous les X entre 2 et 10 sont séparés par deux. .. c'est-à-dire 2, 4, 6, 8 et 10.

(L'ÉTAPE 2 indique à l'ordinateur d'ajouter deux pour obtenir chaque NEXT X.)

Pour que l'ordinateur compte par 3, séparez tous les X de trois.

Essayez ceci pour la ligne 10 :

10 FOR X=3 TO 10 STEP 3

Exécutez le programme.

Cela affiche sur votre écran :

X = 3X = 6X = 9

Il passe le dernier X (nombre 10) car 9 + 3 = 12.

Essayez quelques autres lignes FOR ... STEP afin que vous puissiez voir plus clairement comment cela fonctionne :

10 FOR X = 5 TO 50 STEP 5 10 FOR X = 10 TO 1 STEP-1 10 FOR X= 1 TO 20 STEP 4

# **Compter les sons**

Maintenant que vous avez appris à l'ordinateur à compter, vous pouvez ajouter du son.

Effacez votre ancien programme et tapez ceci :

10 FOR X = 37 TO 255 20 PRINT "TONE" X 30 SOUND X,1 40 NEXT X

Note:

SOUND freq, durée

freq accepte les valeurs de 37 à 32767 et durée les valeurs de 0 à 65535

Ce programme fait compter l'ordinateur de 1 à 255 (par 1).

Chaque fois qu'il compte un nouveau nombre, il fait ce que les lignes 20 et 30 lui disent de faire :

Ligne 20- Il affiche X, le compte actuel. Line 30- Il génère le ton de X.

Par exemple:

La première fois que l'ordinateur arrive à FOR, à la ligne 10, il rend X égal à 1.

Ensuite, il va à la ligne 20 et affiche 1, la valeur de X. Ensuite, la ligne 30 a le son de tonalité #1. Ensuite, il retourne à la ligne 10 et rend X égal à 2 Etc.

Que pensez-vous que le calculateur fera si vous apportez cette modification à la ligne 10 :

10 FOR X = 255 TO 1 STEP -1

#### EXERCICE DE PROGRAMMATION

À l'aide de STEP, changez la ligne 10 pour que l'ordinateur émette des tonalités de:

(1) Le bas de la gamme vers le haut, fredonnant chaque dixième note.

(2) Le haut de la gamme vers le bas, fredonnant chaque dixième de note.

(3) Le milieu de la gamme vers le haut, fredonnant chaque cinquième note.

Prêt pour les réponses ?

10 FOR X = 1 TO 255 STEP 10 10 FOR X = 255 TO 1 STEP -10 10 FOR X = 128 TO 255 STEP 5 Voyez maintenant si vous pouvez écrire un programme qui fait bourdonner l'ordinateur :

(1) du bas de sa plage vers le haut, puis

(2) du haut de gamme jusqu'au bas

10 FOR X = 37 TO 255 20 SOUND X,1 30 NEXT X 40 FOR X = 255 TO 37 STEP -1 S0 SOUND X,1 60 NEXT X

Mais peut-il chanter ?

Oui. Dans la section <u>spectacle et son</u>. vous apprendrez à composer vos chansons préférées.



#### **Regarder l'horloge**

Vous êtes maintenant prêt à montrer à votre ordinateur comment lire l'heure.

Taper:

10 FOR Z= 1 TO 460\*2 20 NEXT Z 30 PRINT " JE COMPTE JUSQU'A 920"

Exécutez le programme. Soyez patient et attendez quelques secondes. Deux secondes, pour être précis. Il faut deux secondes à votre ordinateur pour compter jusqu'à 920.

Les lignes 10 et 20 définissent une pause de la minuterie dans votre programme. En faisant compter jusqu'à 920 à l'ordinateur, vous maintenez l'ordinateur occupé pendant deux secondes.

Comme vous pouvez le constater, il s'agit d'un travail préparatoire pour un chronomètre.

Effacez le programme et tapez :

10 PRINT "COMBIEN DE SECONDES?" 20 INPUT S 30 FOR Z = 1 TO 460\*S 40 NEXT Z 50 PRINT S " SECONDES SONT PASSÉES ! ! !"

Exécutez le. Saisissez le nombre de secondes que vous souhaitez chronométrer sur votre chronomètre.

PROGRAMME faites le vous même 1

Cela aiderait si le chronomètre pouvait sonner une sorte d'alarme. Ajoutez des lignes à la fin du programme pour lui donner une alarme.

Voici le programme que nous avons écrit :

10 PRINT "COMBIEN DE SECONDES?" 20 INPUT S 30 FOR Z = 1 TO 460\*S 40 NEXT Z 50 PRINT S " SECONDES SONT PASSÉES ! ! !" 60 FOR T = 120 TO 180 70 SOUND T,1 80 NEXT T 90 FOR T = 150 TO 140 STEP -1 100 SOUND T,1 110 NEXT T 120 GOTO 50

Remarquez la ligne GOTO à la fin du programme. Le message continue de s'afficher et l'alarme continue de sonner jusqu'à ce que vous appuyiez sur BREAK ou CTRL+C.

#### Compter dans le temps

Avant d'en faire plus avec l'horloge, demandez à l'ordinateur de compter dans le temps. Ce concept vous deviendra précieux sous peu. (mon précieux comme dirait Gollum ...)

Tapez ce nouveau programme :

10 FOR X = 1 TO 3 20 PRINT "X = " X 30 FOR Y = 1 TO 2 40 PRINT, "Y=" Y 50 NEXT Y 60 NEXT X

Remarquer la virgule à la ligne 40. Essayez sans la virgule. La virgule fait afficher "Y= " Y sur la colonne suivante.

X	= 1			
		Y	=	1
. 4		Y	=	2
X	= 2			
		Y	=	1
		Y	=	2
Χ :	= 3			
		Y	=	1
		Y	=	2

Appelez cela un compte dans un compte ou une boucle dans une boucle----ce que vous préférez.

Les programmeurs appellent cela une "boucle imbriquée". Voici ce que fait le programme :

- I. Il compte X de 1 à 3. A chaque fois il compte X :
  - A. Il affiche la valeur de X
  - B. Il compte Y de 1 à 2. A chaque fois il compte Y : 1. Il affiche la valeur de Y

Chaque fois que vous mettez une boucle à l'intérieur d'une autre boucle, vous devez fermer la boucle intérieure avant de fermer la boucle extérieure :

Cor	rect	Incorrect	
10	FOR X = 1 TO 3	3 / 10	FOR X = 1 TO 3
120	FOR Y = 1 TO 2	2 320	FOR Y = 1 TO 2
30	NEXT Y	<b>X</b> 30	NEXT X
40	NEXT X	40	NEXT Y

#### Faire une horloge

Avec ces outils, vous pouvez faire faire beaucoup plus à l'ordinateur.

Tapez ceci :

10 FOR S = 0 TO 59 20 PRINT S 30 SOUND 150,2 40 FOR T = 1 TO 390 50 NEXT T 60 NEXT S 70 PRINT"1 MINUTE EST PASSE"



Exécutez le programme. C'est ce que ça fait:

I. Il compte les secondes de 0 à 59. A chaque fois il compte une seconde :

A. Il affiche la second.

B. Il émet une tonalité.

C. Il s'arrête assez longtemps pour qu'une seconde passe.

II. Lorsqu'il a fini de compter toutes les secondes de 0 à 59, il affiche un message indiquant qu'une minute s'est écoulée.

Il existe un moyen d'améliorer l'apparence de ce programme. Ajoutez cette ligne pour effacer l'écran :

#### 15 CLS

Exécutez maintenant le programme. Cette fois, l'ordinateur passe par ces étapes :

I. Il compte les secondes de 0 à 59 (lignes 10 et 60). Chaque lime compte une seconde :

- A. Il efface l'écran (ligne 15).
- B. II affiche la seconde (ligne 20).
- C. Il émet une tonalité (ligne 30).
- D. Il fait une pause suffisamment longue pour qu'une seconde passe (lignes 40 et 50).

II. Lorsqu'il a fini de compter toutes les secondes de 0 à 59, il affiche un message indiquant qu'une minute s'est écoulée (ligne 70).

En utilisant cela comme base, il est facile de crée une horloge à part entière :

10 FOR H = 0 TO 23 20 FOR M = 0 TO 59 30 FOR S = 0 TO 59 40 CLS 50 PRINT H": "M": "S 60 SOUND 150,2 70 FOR T = 1 TO 375 80 NEXT T 90 NEXT S 100 NEXT M 110 NEXT H Voici un aperçu de ce que fait l'ordinateur dans ce programme :

I. Il compte les heures de 0 10 23 (ligne 10). A chaque fois il compte une nouvelle heure :

A. Il compte les minutes de 0 à 59 (ligne 20). A chaque fois cela compte une nouvelle minute :

- 1. Il compte les secondes de 0 à 59 (lignes 30 et 90). A chaque fois il compte une nouvelle seconde : :
  - a. II efface l'écran (ligne 40).
  - b. Il affiche l'heure, la minute et la seconde (ligne 50).
  - c. Ca sonne un seul (ligne 60).
  - d. Il s'arrête assez longtemps pour qu'une seconde passe (Unes 70 et 80).
- 2. Lorsqu'il a fini de compter les 59 secondes, il revient à la ligne 20 pour la minute suivante (ligne 100).
- B. Quand il a fini de compter toutes les 59 minutes, il recule de 10 lignes pour la prochaine heure (ligne 110).

II. Lorsqu'il a fini de compter toutes les heures (0-23), le programme se termine.

En ajoutant cette ligne, 120 GOTO 10, l'horloge fonctionnera perpétuellement.

Vous avez du mal avec ce programme ? Ignorez-le pour l'instant.

Cela vous semblera facile plus tard.

PROGRAMME faites le vous même 2

Entre les lignes 90 et 100, vous pouvez ajouter des tonalités qui retentiront à chaque minute. Écrivez un programme qui fait cela.

PROGRAMME faites le vous même 3

Écrivez un programme qui fait que votre ordinateur affiche chacune de ses différentes couleurs pendant 1 seconde chacune.

#### PROGRAMME 2

Lignes ajoutées au programme de l'horloge :

92 FOR T=200 TO 210 STEP 5 94 SOUND T, 1 95 NEXT T 97 FOR T=210 TO 200 STEP -5 98 SOUND T, 1 99 NEXT T

#### PROGRAMME 3

10 FOR C = 0 TO 14 20 COLOR 1,C 30 FOR X = 1 TO 460 40 NEXT X 50 NEXT C 60 COLOR 7,0

Décision, Décision ...



Voici une décision facile pour l'ordinateur :

Si vous tapez "rouge" ... alors rendez l'écran rouge ... ou Si vous tapez "bleu" ... alors rendez l'écran bleu

Assez facile ?

Ensuite, demandez à l'ordinateur de le faire.

Tapez ce programme :

10 SCREEN 9 20 COLOR 7,0 30 CLS 40 PRINT "VOULEZ-VOUS L'ECRAN ROUGE OU BLEU?" 50 INPUT C\$ 60 IF C\$ = "ROUGE" THEN 100 70 IF C\$ = "BLEU" THEN 200 100 COLOR 7,4 110 END 200 COLOR 7,1

Exécutez le programme plusieurs fois. Essayez à la fois "ROUGE" et "BLEU" comme réponses.

Voici ce que fait le programme :

Ligne 10 configure l'écran en mode 9

Graphiques à résolution améliorée de  $640 \times 350$  pixels Format texte  $80 \times 25$  avec une taille de boîte de caractères de  $8 \times 14$ 16 couleurs Ligne 20 configure l'écran avec un fond noir et une écriture blanche

Ligne 30 efface l'écran

Si vous répondez "ROUGE" ... alors

- 1. la ligne 60 envoie l'ordinateur à la ligne 100.
- 2. la ligne 100 rend votre écran rouge.
- 3. la ligne 110 termine le programme. (Si l'ordinateur atteint la ligne 110, il n'atteint jamais la ligne 200.)

... D'autre part

Si vous répondez "BLEU" ... alors

- 1. La ligne 70 envoie l'ordinateur à la ligne 200.
- 2. La ligne 200 rend votre écran bleu.
- 3. Comme la ligne 200 est la dernière ligne du programme, le programme s'arrête là.

Que se passe-t-il si vous répondez par autre chose que "ROUGE" ou "BLEU" ?

Exécutez à nouveau le programme. Cette fois, répondez "VERT".

Cela rend l'écran rouge. Est-ce que tu sais pourquoi?

CONSEIL : Si la condition n'est pas vraie, l'ordinateur ignore la partie THEN de la ligne et passe à la ligne de programme suivante.

# **EXERCICE DE PROGRAMMATION**

Il existe un moyen pour que ce programme rejette toute réponse autre que "ROUGE" ou "BLEU". Ce sont les deux lignes à ajouter. Vous déterminez où ils vont dans le programme :

PRINT "VOUS DEVEZ TAPER SOIT ROUGE OU BLEU" GOTO 50

Insérez les numéros de ligne.

ASTUCE : Les lignes doivent apparaître après que l'ordinateur a eu la possibilité de tester votre réponse pour "rouge" ou "bleu".

ASTUCE : les lignes doivent apparaître avant que l'ordinateur ne rende votre écran "rouge".

Réponse : Les lignes doivent venir après la ligne 50 et avant la ligne 100 :

80 PRINT "VOUS DEVEZ TAPER SOIT ROUGE OU BLEU" 90 GOTO 50

#### **PROGRAMME** Faites le vous même

Une fois que l'ordinateur a rendu l'écran rouge ou bleu, faites-le revenir en arrière et vous demander de taper à nouveau "rouge" ou "bleu".

ASTUCE : Vous devez modifier la ligne 110 et ajouter la ligne 210.

Voici la façon dont nous avons écrit ce programme.

10 SCREEN 9 20 COLOR 7,0 30 CLS 40 PRINT "VOULEZ-VOUS L'ECRAN ROUGE OU BLEU?" 50 INPUT C\$ 60 IF C\$ = "ROUGE" THEN 100 70 IF C\$ = "BLEU" THEN 200 80 PRINT "VOUS DEVEZ TAPER SOIT ROUGE OU BLEU" 90 GOTO 50 100 COLOR 7,4 110 GOTO 40 110 END 200 COLOR 7,1 210 GOTO 40

Tracez le chemin emprunté par l'ordinateur dans ce programme. Passez d'une ligne 10 à la suivante.

#### **RÈGLES SUR IF THEN ET GOTO**

If THEN est conditionnel. L'ordinateur "branche" uniquement si la condition est vraie.

GOTO est inconditionnel. L'ordinateur se branche toujours.

Même si cette section est courte. Vous avez appris un concept de programmation important. L'ordinateur prendra des décisions tout au long de cet apprentissage à Initiation au langage Basic.

#### Jeux de chances



Merci à un mot BASIC appelé RND, l'ordinateur peut jouer à presque n'importe quel jeu de hasard. Et même si vous ne voulez pas jouer à des jeux informatiques, vous voudrez apprendre deux mots que ce chapitre introduit : RND et LOCATE.

Vous trouverez également dans ce chapitre d'autres utilisations de IF THEN.

Pour que RND génère un nombre différent à chaque nouvelle exécution d'un programme insérez au débute du programme la ligne:

#### **5 RANDOMIZE TIMER**

Tapez ce programme :

#### 10 PRINT RND(1)

Note si vous utilisez PRINT RND(0), RND généra toujours le même nobre.

L'ordinateur vient de choisir un nombre aléatoire.

Exécutez-le plusieurs fois ... C'est comme si l'ordinateur tirait un nombre d'un chapeau. Le nombre qu'il choisit est imprévisible.

Pour choisir un nombre de 1 à N, on utilise

A=INT(N\*RND(1)+1)

où N est le nombre maximum à générer.

Exemple A=INT(10\*RND(1)+1) L'ordinateur généra un nombre aléatoire de 1 à 10.

Tapez et exécutez ce programme suivant. Appuyez sur BREAK (CTRL+C) lorsque vous vous assurez que les nombres sont aléatoires.

5 RANDOMIZE TIMER 10 PRINT RND(1) 20 GOTO 10

ou encore

5 RANDOMIZE TIMER 10 PRINT INT(49\*RND(1)+1) 20 GOTO 10

Comment pouvez-vous changer le programme pour obtenir des nombres aléatoires de 1 à 255 ?

10 PRINT A=INT(255\*RND(1)+1)

#### Un spectacle aléatoire

Juste pour le plaisir, demandez à l'ordinateur de composer une chanson composée de tonalités aléatoires.

5 RANDOMIZE TIMER 10 T= INT(255\*RND(1)+1) 20 IF T<37 THEN 10 30 SOUND T,5 40 GOTO 10

La ligne 20 vérifie si T est plus grand que 37. Car SOUND n'accepte pour son premier paramètre (freg) que les nombres de 37 à 32767 (T dans ce programme).

Pour le deuxième paramètre (durée) SOUND accepte les nombres de 0 à 65535. (5 T dans ce programme)

Ce programme compose une chanson composée de tonalités et de durée aléatoires:

5 RANDOMIZE TIMER 10 T= INT(255\*RND(1)+1) 20 IF T<37 THEN 10 25 D= INT(25\*RND(1)+1) 30 SOUND T,D 40 GOTO 10

# PROGRAMME À FAIRE VOUS-MÊME

Ajoutez quelques lignes pour que l'ordinateur affiche une couleur aléatoire  $(1 \cdot 16)$  juste avant qu'il n'émette chaque tonalité aléatoire.

5 RANDOMIZE TIMER 10 SCREEN 9 20 COLOR 7,0 30 CLS 40 T= INT(255\*RND(1)+1) 50 IF T<37 THEN 10 60 C = INT(16\*RND(1)+1) 70 COLOR 7,C 80 SOUND T,5 90 GOTO 40

Nous avons quelques jeux simples dans cette section. N'hésitez pas à utiliser votre imagination pour les intéresser ou inventez la vôtre.

#### **Roulette russe**

Dans ce jeu, un pistolet a 10 chambres. L'ordinateur choisit, au hasard, laquelle des 10 chambres contient la balle fatale.

Taper:

```
5 RANDOMIZE TIMER
10 PRINT "CHOISISSEZ VOTRE CHAMBRE (1-10)"
20 INPUT X
30 IF X = INT(10*RND(1)+1) THEN 100
40 SOUND 200,1
50 PRINT "--CLICK--"
60 GOTO 10
100 PRINT "--BANG--"
```

Tout d'abord, à la ligne 20, le joueur saisit X (un nombre de 1 à 10).

Ensuite, l'ordinateur compare X avec INT(10\*RND(1)+1) (un nombre aléatoire de 1 à 10).

Ensuite, il suit les "flèches":



• Si X est égal à RND(10), l'ordinateur passe à la ligne 100, la "routine morte".

• Si X n'est pas égal à INT(10\*RND(1)+1), l'ordinateur "clique" et retourne à la ligne 10, où vous obtenez une autre chance...

Améliorez la routine à la ligne 100.

Taper:

100 FOR T = 133 TO 38 STEP -5 110 PRINT" BANG!!!!!" 120 SOUND T, 1 130 NEXT T 140 CLS 150 LOCATE 5,20 155 PRINT "DESOLER VOUS AVEZ PERDU" 160 SOUND 38,50 170 LOCATE 8,15 180 PRINT "PROCHAINE VICTIME, S'IL VOUS PLAIT"

Exécutez le programme. Voici ce que fait la routine :

les lignes 100-130 donnent à l'ordinateur des tonalités descendantes et affiche BANG !!!!! maintes et maintes fois sur l'écran.

La ligne 140 efface l'écran.

Les lignes 150 et 170 utilisent un nouveau mot LOCATE pour positionner deux messages sur votre écran :

DESOLER VOUS AVEZ PERDU et PROCHAINE VICTIME, S'IL VOUS PLAÎT.

Les paramètres pour LOCATE LIN,COL

LIN: Le numéro de ligne soit entre 1 et 25

COL: Le numéro de la colonne soit entre 1 et 40 ou 1 et 80, en fonction de la largeur de l'écran.

Écrivez un programme qui affiche des lignes pour félicité le joueur s'il gagne.

Soit:

FELICITATION

VOUS AVEZ GAGNEZ

# Lancer les dés

Ce jeu demande à l'ordinateur de lancer deux dés. Pour ce faire, il doit trouver deux nombres aléatoires.

Taper:

```
5 RANDOMIZE TIMER
10 CLS
20 \text{ X} = \text{INT}(6*\text{RND}(1)+1)
30 \text{ Y} = \text{INT}(6*\text{RND}(1)+1)
40 R = X + Y
50 LOCATE 1,2:PRINT "DE 1"
60 LOCATE 1,10:PRINT "DE 2"
70 LOCATE 1,18:PRINT"VOUS AVEZ LANCER"
80 LOCATE 2,2:PRINT X
90 LOCATE 2,10:PRINT Y
100 LOCATE 2,25:PRINT R
110 PRINT
120 INPUT "VOULEZ-VOUS UN AUTRE LANCER OUI OU NON? ",A$
130 IF A$ = "OUI" THEN 10
140 IF A = "NON" THEN END
150 GOTO 120
```

La ligne 10 efface l'écran.

La ligne 20 choisit un nombre aléatoire de 1 à 6 pour le premier dé.

La ligne 30 choisit un nombre aléatoire pour l'autre dé.

La ligne 40 additionne les deux dés pour obtenir le résultat total.

Les lignes 50 à 100 impriment les résultats du lancer.

#### Un : permet d'exécuter plusieurs commandes sur la même ligne. Ici LOCATE ensuite PRINT.

La ligne 120 vous permet d'indiquer si vous voulez un autre lancer.

Si vous répondez "OUI", le programme passe à la ligne 10 et s'exécute à nouveau.

Si vous répondez "NON", le programme se termine.

Sinon, le programme passe à la ligne 120 pour avoir une réponce valide soit OUI ou NON.

#### Programme faites le vous même.

Puisque vous savez comment lancer les dés, il devrait être facile d'écrire un programme de "Craps".

Voici les règles du jeu (dans sa forme la plus simple:
1. Le joueur lance deux dés. Si le premier jet de dé est un 2 ("snake eyes"), un 3 {"cock-eyes"} ou un '2 ("boxcars"), le joueur perd et la partie est terminée.

2. Si le premier lancer est un 7 ou un 11 ("un naturel"), le joueur gagne et la partie est terminée.

3. Si le premier lancer est un autre nombre, il devient le "point" du joueur.

Le joueur doit continuer à rouler jusqu'à ce qu'il "fait le point" en obtenant à nouveau le même nombre pour gagner, ou en lançant un 7 et en perdant.

Vous en savez déjà plus qu'assez pour écrire ce programme. Fais le.

Faites en sorte que l'ordinateur l'affiche dans un format attrayant sur votre écran et tenez le joueur informé de ce qui se passe. Il peut vous prendre un certain temps, mais faites de votre mieux. Bonne chance!

Dans le programme remplacez L et C par vos coordonnées.

```
5 RANDOMIZE TIMER
10 CLS
20 \text{ A} = \text{INT}(6*\text{RND}(1)+1)
30 \text{ B} = \text{INT}(6*\text{RND}(1)+1)
40 R = A + B
50 LOCATE L,C:PRINT A
60 LOCATE L,C: PRINT B
70 LOCATE L,C: PRINT "VOUS AVEZ ROULE "; R
80 IF R = 2 THEN 600
90 IF R = 3 THEN 600
100 IF R = 12 THEN 600
110 IF R = 7 THEN 500
120 IF R = 11 THEN 500
130 FOR X = 1 TO 800
140 NEXT X
150 CLS
160 LOCATE L,C: PRINT "ROULER UN AUTRE" R "ET VOUS GAGNER"
170 LOCATE L,C: PRINT "ROULER UN 7 ET VOUS PERDER"
180 LOCATE L.C: PRINT "APPUYER SUR <ENTER> LORSOUE VOUS ETES PRET"
185 LOCATE L,C: PRINT "POUR VOTRE PROCHAIN LANCER"
190 INPUT A$
200 \text{ X} = \text{INT}(6*\text{RND}(1)+1)
210 \text{ Y} = \text{INT}(6*\text{RND}(1)+1)
220 Z = X + Y
225 CLS
230 LOCATE L,C: PRINT X
240 LOCATE L,C: PRINT Y
250 LOCATE L,C: PRINT "VOUS ROULER UN" Z
260 IF Z = R THEN 500
270 IF Z = 7 THEN 600
280 GOTO 180
500 FOR X = 1 TO 1000
510 NEXT X
515 CLS
```

520 LOCATE L,C: PRINT "VOUS ETES LE GAGNAT" 530 LOCATE L,C: PRINT "FILICITATION!!!" 540 GO TO 630 600 FOR X = 1 TO 1000 610 NEXT X 615 CLS 620 LOCATE L,C: PRINT "DÉSOLER VOUS PERDER" 630 LOCATE L,C: PRINT "GAME'S OVER"

#### Lecture



Votre ordinateur est un naturel à l'enseignement. Il est patient, infatigable et ne se trompe jamais.

Selon le programmeur (vous, bien sûr), cela peut aussi être imaginatif, réconfortant et enthousiaste.

En utilisant RND, apprenez les mathématiques.

Le programme ci-dessous vous entraîne sur les tables de multiplication, de 1 à 15, et vérifie vos réponses.

Taper:

```
5 RANDOMIZE TIMER

10 CLS

20 X=INT(15*RND(1)+1) 'Nombre entre 1 et 15

30 Y=INT(15*RND(1)+1) 'Nombre entre 1 et 15

40 PRINT "QUE DONNE " X "*" Y " ? "

45 INPUT A

50 IF A = X * Y THEN 90

60 PRINT "LA REPONSE EST" X*Y

70 PRINT "MEILLIEUR CHANCE LA PROCHAINE FOIS"

80 GOTO 100

90 PRINT "EXACTE !!!"

100 PRINT "APPUYER SUR <ENTER> QUAND VOUS ÊTES PRÊT POUR UN AUTRE"

105 INPUT A$

110 GOTO 10
```

#### Exercice

Ajoutez des lignes pour vérifier si A\$ est égale à "OUI" ou "O" pour entrer un autre nombre. Sinon le programme se termine.

\*\*\*\*\*

Pour sauvegarder un programme sur le disque utilisez SAVE:

SAVE "PROGRAM", A enregistre le programme actuel dans un fichier texte nommé PROGRAM.BAS .

Pour charger un programme utilisez LOAD:

LOAD "PROGRAM (Avec une seule ") charge le fichier programme nommé PROGRAM.BAS en mémoire

\*\*\*\*\*

Faites en sorte que le programme vous entraîne sur des problèmes d'addition de 1 à 100.

Voici les lignes que nous avons modifiées :

20 X=INT(**100**\*RND(1)+1) 30 Y=INT(**100**\*RND(1)+1) 40 PRINT "QUE DONNE " X "+" Y 45 INPUT A 50 IF A = X + Y THEN 90 60 PRINT "LA REPONSE EST " X + Y

Rendez le programme plus intéressant. Demandez-lui de garder un total cumulé de toutes les bonnes réponses.

Taper:

15 T=T+1 95 C=C+1 98 PRINT "IL Y A " C "REPONSES CORRECT SUR " T "REPONSES"

T est un "compteur". Il compte le nombre de questions qui vous sont posées. Lorsque vous démarrez le programme pour la première fois, T est égal à zéro.

Ensuite, chaque fois que l'ordinateur arrive à la ligne 15, il ajoute 1 à T.

C est aussi un compteur. Il compte vos bonnes réponses. Puisque C est à la ligne 95, l'ordinateur n'augmente pas C à moins que votre réponse ne soit correcte.

#### Programme faites-le vous même.

Rendez le programme plus amusant. Faites-lui effectuer une ou plusieurs des actions suivantes :

- 1. Appelez-vous par votre nom.
- 2. Récompensez votre bonne réponse avec un spectacle son et lumière.
- 3. Afficher le problème et les messages de manière attrayante sur votre écran. (Utilisez LOCATE pour cela.)
- 4. Gardez un total cumulé du pourcentage de bonnes réponses.
- 5. Terminez le programme si vous obtenez 10 bonnes réponses consécutives.

Utilisez votre imagination.

Note : Dans ce programme on utilise les : pour mettre plusieurs commandes sur la même ligne

Remplacez L et C de LOCATE par vos coordonnées.

```
5 SCREEN 9:COLOR 7,0:CLS:RANDOMIZE TIMER
6 LOCATE 1,5:PRINT "Quel est ton nom?";
8 INPUT N$
9 LOCATE L,C:PRINT "Ce petit test comprend 10 questions"
10 COLOR 7,0:CLS
15 T = T + 1
20 X=INT(100*RND(1)+1)
30 Y=INT(100*RND(1)+1)
40 LOCATE L,C:PRINT "Comment font " X "+" Y
45 INPUT A
50 IF A = X + Y THEN 82
60 LOCATE L,C:PRINT "La réponse est " X + Y
70 LOCATE L,C:PRINT "Meilleur chance la prochaine fois," N$
80 GOTO 100
82 COLOR 7,1:CLS
83 FOR M = 1 TO 4
84 SOUND 175,1
85 SOUND 200,1
86 NEXT M
87 CLS
90 LOCATE L,C:PRINT "Correcte, " N$ "! !!"
95 C = C + 1
97 LOCATE L,C:PRINT "C'EST"
98 LOCATE L,C:PRINT C " Réponses Correctes sur " T " Réponses "
99 LOCATE L,C:PRINT C/T*100 "% De bonnes réponses" :IF T=10 THEN END
100 LOCATE L,C:PRINT "Appuyez sur <ENTER> Quand vous êtes prêt"
102 LOCATE L,C:PRINT "POUR UN AUTRE"
105 INPUT A$
110 GOTO 10
```

Pour afficher des caractères on peut utiliser la commande CHR\$(X) où X est le code ascii du caractère.

PRINT CHR\$(144) affiche É majuscule.

Le programme ci-dessous affiche les caractères des codes ascii de 33 à 255 sur 4 colonnes. Il utilise le paramètre TAB de PRINT et une boucle FOR ... NEXT.

Ce programme utilise la variable P comme indicateur de position pour TAB, soit pour les positions 0, 15, 30 et 45. La variable P est augmentée de 15 après la ligne 30.

10 CLS:P=0 'Attribut la valeur 0 à P 20 FOR X= 33 TO 255 25 IF P=60 THEN P=0 'Vérifie si P=60 si oui alors P=0 30 PRINT TAB(P) X;"CHR= "CHR\$(X); 'Affiche X et le caractère pour la valeur de X 35 P=P+15 'Incrémente P de 15 pour le prochain affichage 40 NEXT X

Voir à la fin de cette section pour les codes ascii.

#### Tout d'abord, construisez le vocabulaire de votre ordinateur...

Pour construire le vocabulaire de votre ordinateur (afin qu'il puisse construire le vôtre !), tapez et exécutez ce programme :

10 DATA POMMES, ORANGES, POIRES 20 FOR X = 1 TO 3 30 READ F\$ 40 NEXT X

Que s'est-il passé... rien ?

Rien que vous puissiez voir, c'est-à-dire. Pour voir ce que fait l'ordinateur, ajoutez cette ligne et exécutez le programme :

35 PRINT "F\$ = : " F\$

La ligne 30 indique à l'ordinateur de :

- 1. Recherchez une ligne DATA.
- 2. LIRE le premier élément de la liste POMMES.
- 3. Donnez aux POMMES une étiquette F\$.
- 4. "Barrez" les POMMES.

La deuxième fois que l'ordinateur se gélifie à la ligne 30, on lui dit de faire la même chose :

- 1. recherchez une ligne DATA.
- 2. LISEZ le premier article cette fois, il s'agit d'ORANGES.
- 3. Donnez à ORANGES le label F\$.
- 4. "Barrez" les ORANGES.
- Et si vous voulez que l'ordinateur relise la même liste ?

Toutes les données sont déjà "barré" ...

Tapez:

60 GOTO 10

Exécutez le programme.

Vous obtenez une erreur : Out of DATA in 30

L'ordinateur a barré toutes les données.

Tapez cette ligne et lancez le programme :

**50 RESTORE** 

Maintenant, c'est comme si l'ordinateur n'avait jamais barré aucune donnée. Il lit la même liste encore et encore. Vous pouvez mettre des lignes DATA où vous voulez dans le programme. (Faites Break ou CTRL+C pour arrêter le programme)

Exécutez chacun de ces programmes. Ils fonctionnent tous de la même manière.

Note: N'oublier pas de taper NEW avant de taper un nouveau programme. NEW effaces toutes les donnés en mémoire, valeurs des variables et le programme.

Programme 1	Programme 2
10 DATA POMMES	10 DATA POMMES, ORANGES
20 DATA ORANGES	20 DATA POIRES
30 FOR X = 1 TO 3	30 FOR X = 1 TO 3
40 READ F\$	40 READ F\$
50 PRINT "F\$ = : " F\$	50 PRINT "F\$ = : " F\$
60 NEXT X	60 NEXT X
70 DATA PEARS	
Programme 3	Programme 4
Programme 3	Programme 4
Programme 3 30 FOR X = 1 TO 3	<b>Programme 4</b> 100 FOR X = 1 TO 3
<b>Programme 3</b> 30 FOR X = 1 TO 3 40 READ F\$	Programme 4 100 FOR X = 1 TO 3 110 READ F\$
Programme 3 30 FOR X = 1 TO 3 40 READ F\$ 50 PRINT "F\$ = : " F\$	Programme 4 100 FOR X = 1 TO 3 110 READ F\$ 120 PRINT "F\$ = :" F\$
Programme 3 30 FOR X = 1 TO 3 40 READ F\$ 50 PRINT "F\$ = : " F\$ 60 NEXT X	Programme 4 100 FOR X = 1 TO 3 110 READ F\$ 120 PRINT "F\$ = :" F\$ 130 NEXT X
Programme 3 30 FOR X = 1 TO 3 40 READ F\$ 50 PRINT "F\$ = : " F\$ 60 NEXT X 70 DATA POMMES	Programme 4 100 FOR X = 1 TO 3 110 READ F\$ 120 PRINT "F\$ = :" F\$ 130 NEXT X 140 DATA POMMES, ORANGES,
Programme 3           30 FOR X = 1 TO 3           40 READ F\$           50 PRINT "F\$ = : " F\$           60 NEXT X           70 DATA POMMES           80 DATA ORANGES	Programme 4 100 FOR X = 1 TO 3 110 READ F\$ 120 PRINT "F\$ = :" F\$ 130 NEXT X 140 DATA POMMES, ORANGES, POIRES

#### Maintenant, construisez votre vocabulaire

Voici quelques mots et définitions à apprendre :

10 DATA TACITE,HABITUELLEMENT NON PARLANT 20 DATA LOQUACE,TRES BAVARE 30 DATA BRUYANT, BRUIT FORT 40 DATA LACONIQUE,EXPRIME EN PEU DE MOTS 50 DATA EXPANSIF,QUI TEND A S'ETENDRE

Maintenant, demandez à l'ordinateur de sélectionner l'un de ces mots au hasard. Hmmm il y a dix articles. Peutêtre que cela fonctionne

1 RANDOMIZE TIMER 60 N= INT(10\*RND(1)+1) 70 FOR X = 1 TO N 80 READ A\$ 90 NEXT X 100 PRINT "LE MOT AU HASARD EST: " A\$

Exécutez le programme plusieurs fois. Cela ne fonctionne pas très bien. L'ordinateur est tout aussi susceptible de s'arrêter à une définition qu'à un mot.

Ce que l'ordinateur doit vraiment faire, c'est choisir un mot au hasard uniquement parmi les éléments 1, 3, S, 7 ou 9.

Heureusement, BASIC a un mot qui aide à cela.

Taper:

65 IF INT(N/2) = N/2 THEN N = N - 1

Maintenant, exécutez à nouveau le programme plusieurs fois. Cette fois, ça devrait marcher.

INT indique à l'ordinateur de ne regarder que la "partie entière" du nombre et d'ignorer la partie décimale. Par exemple, l'ordinateur voit INT(3.9) comme 3.

Supposons que N, le nombre aléatoire, est 10. La clause IF de la ligne 65 fait ceci :

INT(10/2) = 10/2INT(5) = 55 = 5

Ce qui précède est vrai : 5 est égal à S. Puisque c'est vrai, l'ordinateur complète la clause THEN. N est ajusté pour être égal à 9 (10 - 1).

Supposons maintenant que N, le nombre aléatoire, est 9. La clause IF de la ligne 65 fait ceci :

INT(9/2) = 9/2INT(4.5) = 4.54 = 4.5

Ce qui précède n'est pas vrai : 4 n'est pas égal à 4,5. Comme ce n'est pas vrai, l'ordinateur ne complète pas la clause THEN. N reste 9.

Outre la lecture d'un mot au hasard, l'ordinateur doit également lire la définition du mot. Ajoutez ces lignes à la fin du programme :

110 READ B\$ 120 PRINT "LA DEFINITION EST:" B\$

Exécutez maintenant le programme plusieurs fois.

Demandez à l'ordinateur d'imprimer un mot au hasard et sa définition après le suivant.

Ajoutez ceci au début du programme :

5 CLEAR 100

Cela réserve beaucoup "d'espace de caractères". Ajoutez ces lignes à la fin du programme :

130 RESTORE 140 GOTO 60

Cela permet à l'ordinateur de choisir un nouveau mot aléatoire et sa définition à partir d'un groupe "restauré" d'éléments de données.

Voici à quoi ressemble le programme :

```
1 RANDOMIZE TIMER
5 CLEAR 100
10 DATA TACITE, HABITUELLEMENT NON PARLANT
20 DATA LOQUACE, TRES BAVARE
30 DATA BRUYANT, BRUIT FORT
40 DATA LACONIQUE, EXPRIME EN PEU DE MOTS
50 DATA EXPANSIF, QUI TEND A S'ETENDRE
60 \text{ N} = \text{INT}(10 \text{*RND}(1) + 1)
65 IF INT(N/2) = N/2 THEN N = N - 1
70 \text{ FOR } X = 1 \text{ TO } N
80 READ A$
90 NEXT X
100 PRINT "LE MOT AU HASARD EST: " A$
110 READ B$
120 PRINT "LA DEFINITION EST:" B$
130 RESTORE
140 GOTO 60
```

Si vous voulez vous pouvez ajoutez quelques mots et définitions supplémentaires en ajoutant des lignes DATA.

Pour des variantes de ce programme, vous pouvez essayer des états et des capitales, des villes et des pays, des mots et des significations étrangers.

#### Programme faites-le vous même

Vous souhaitez compléter ce programme ?

Programmez-le pour que l'ordinateur :

- 1. Affiche uniquement la définition.
- 2. Vous demande le mot.
- 3. Compare le mot avec le mot aléatoire correct.
- 4. Vous indique si votre réponse est correcte. Si votre réponse est incorrecte, affiche le mot correct.

**1 RANDOMIZE TIMER** 5 CLEAR 100 10 DATA TACITE, HABITUELLEMENT NON PARLANT 20 DATA LOQUACE, TRES BAVARE 30 DATA BRUYANT, BRUIT FORT 40 DATA LACONIQUE, EXPRIME EN PEU DE MOTS 50 DATA EXPANSIF, QUI TEND A S'ETENDRE 60 N = INT(10 \*RND(1) + 1)65 IF INT(N/2) = N/2 THEN N = N - 1 70 FOR X = 1 TO N80 READ A\$ 90 NEXT X 110 READ B\$ 120 PRINT "QUE SIGNIFIE LE MOT: " B\$ **130 RESTORE** 140 INPUT R\$ 150 IF R\$ = A\$ THEN 180 160 PRINT "MAUVAIS" 170 PRINT "LE MOT JUSTE EST :" A\$ 180 GOTO 60 190 PRINT "CORRECT" 200 GOTO 60

	ASC cha	II control aracters		A	SCII p	orinta acters	ble				E	cha	led AS racters	CII		
00	NULL	(Null character)	32	space	64	@	96	(32)	128	ç	160	á	192	L	224	Ó
01	SOH	(Start of Header)	33	1	65	A	97	a	129	ü	161	i	193	1	225	ß
02	FTX	(Start of Text)	34	#	67	B	98	b	130	e	162	0 ú	194	τ	226	ò
04	EOT	(End of Trans.)	36	S	68	D	100	d	132	ä	164	ñ	196	-	228	õ
05	ENQ	(Enquiry)	37	%	69	E	101	е	133	à	165	Ñ	197	+	229	õ
06	ACK	(Acknowledgement)	38	8	70	F	102	f	134	å	166	a	198	ã	230	μ
07	BEL	(Bell)	39	,	/1	G	103	g	135	ç	167	•	199	A	231	p
09	HT	(Horizontal Tab)	41	1	73	ï	104	ï	137	ë	169	R	200	E	232	Ú
10	LF	(Line feed)	42	*	74	J	106	j	138	è	170	-	202	1	234	Û
11	VT	(Vertical Tab)	43	+	75	K	107	k	139	Ï	171	1/2	203	T	235	Ù
12	CR	(Form feed)	44	,	76	M	108	m	140	1	1/2	1/4	204	1	236	Ŷ
14	SO	(Shift Out)	46	÷.	78	N	110	n	142	Ä	174	w i	205	+	238	-
15	SI	(Shift In)	47	1	79	0	111	o	143	Å	175	39	207	n	239	*
16	DLE	(Data link escape)	48	0	80	P	112	р	144	É	176		208	ð	240	=
1/	DC1	(Device control 1)	49	1	81	Q	113	q	145	æ	1//		209	ÐÊ	241	±
19	DC3	(Device control 2)	51	3	83	S	115	S	147	ô	179	T	211	Ë	242	3/4
20	DC4	(Device control 4)	52	4	84	Т	116	t	148	ö	180	-	212	È	244	1
21	NAK	(Negative acknowl.)	53	5	85	U	117	u	149	ò	181	Á	213	1	245	§
22	ETR	(Synchronous Idle) (End of trans, block)	54	7	87	w	110	v	150	ů	182	À	214	i	246	÷
24	CAN	(Cancel)	56	8	88	x	120	x	152	ÿ	184	C	216	i	248	0
25	EM	(End of medium)	57	9	89	Y	121	у	153	Ö	185	4	217	٦	249	22
26	SUB	(Substitute)	58	:	90	Z	122	Z	154	U	186		218	Г	250	
27	ESC	(Escape)	59	:	91	1	123	{	155	ø	187	]	219		251	3
20	GS	(Group separator)	61	=	93	i	125	3	157	ø	189	¢	221		253	2
20		(Densel and the back	62	>	91	~	126	-	158	×	190	¥	222	i	254	-
30	RS	(Record separator)	02		34											
30 31 127 fre (sp	US DEL QUER	(Unit separator) (Unit separator) (Delete)	63 VO	? Wels (spani	95 acut	e ac	ccent		159 VOI	f wels	191 s with	1	223		255	nbsp
23 30 31 127 fre (sp Ñ Ŵ @ ¿? i ! !	RS US DEL	(Record separator) (Unit separator) (Delete) alt + 164 alt + 165 alt + 165 alt + 64 alt + 168 alt + 63 alt + 63 alt + 173 alt + 33 alt + 58 alt + 47 alt + 92	63 VOV	? Wels (spani á é í ó ú Á É Í Ó Ú	95 acut ish lar al al al al al al al al al al al al al	te ac nguag t + 16 t + 13 t + 16 t + 16 t + 16 t + 14 t + 14 t + 22 t + 23	ccent je) 50 51 52 53 51 52 53 51 52 53 51 52 53 51 52 53 51 52 53 51 52 53 51 52 53 51 52 53 53 51 53 54 54 55 55 56 56 56 56 56 56 56 56 56 56 56		IS9 VOV C ä ë ï ö ü Ä Ë ï Ö Ü	f liare a a a a a a a a a a a a a a a a a a	191 S With SSIS alt + 13 alt + 13 alt + 14 alt + 14 alt + 14 alt + 21 alt + 21 alt + 15 alt + 15	1 22 77 99 89 99 22 11 6 33 44	223	•	255	nbsp
230 30 31 127 fre (sp Ñ @ ;? i ! : / \	athe sym	(Record separator) (Unit separator) (Delete) (Delete) alt + 164 alt + 165 alt + 165 alt + 64 alt + 168 alt + 63 alt + 173 alt + 33 alt + 58 alt + 47 alt + 92 (Delete)	63 VOV	? Wels (spani á é í ó ú Á É Í Ó Ú	acut ish lar al al al al al al al al al children	te ac nguag t + 16 t + 13 t + 16 t + 16 t + 16 t + 16 t + 14 t + 22 t + 23 t + 23 t + 23 t + 23	ccent <sup>je)</sup> <sup>50</sup> <sup>51</sup> <sup>52</sup> <sup>53</sup> <sup>51</sup> <sup>54</sup> <sup>54</sup> <sup>53</sup> <sup>54</sup> <sup>54</sup> <sup>54</sup> <sup>53</sup> <sup>54</sup> <sup>54</sup> <sup>55</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>57</sup> <sup>56</sup> <sup>56</sup> <sup>57</sup> <sup>56</sup> <sup>56</sup> <sup>57</sup> <sup>56</sup> <sup>57</sup> <sup>56</sup> <sup>56</sup> <sup>57</sup> <sup>56</sup> <sup>56</sup> <sup>57</sup> <sup>56</sup> <sup>56</sup> <sup>57</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup> <sup>56</sup>		IS9 VOV d ä ë ï ö ü Ä Ë ï Ü U U Qu	f wels liare	191 S With SIS alt + 13 alt + 13 alt + 14 alt + 14 alt + 14 alt + 21 alt + 21 alt + 15 alt + 15 s ancis	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	223	•	255	nbsp
230 331 127 fre (sp Ñ @ ¿? i! : / \ m	athe sym	(Record separator) (Unit separator) (Delete) alt + 164 alt + 165 alt + 165 alt + 168 alt + 168 alt + 168 alt + 173 alt + 33 alt + 58 alt + 47 alt + 92 matical bols alt + 171	G2 G3 VON	? Wels (spani á é í ó ó ú Á É Í í ó ý ý	95 acut ish lar al al al al al al al al al al al al al	te ac nguag t + 16 t + 16 t + 16 t + 16 t + 16 t + 16 t + 14 t + 22 t + 23 l / tr ols lt + 30	ccent (e) 50 50 51 52 53 51 52 53 51 52 53 51 52 53 51 52 53 51 52 53 54 54 55 56 56 56 56 56 56 56 56 56		IS9 VOV C ä ë ï Ö Ü Ü u pa	f wels liare	191 S with esis alt + 13 alt + 13 alt + 14 alt + 12 alt + 14 alt + 21 alt + 21 alt + 15 alt + 15 s and thesis alt + 3	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	223	•	255	nbsp
ال 30 31 127 fre (sp ñ Ñ @ ¿? i ! / \ m ½	athe sym	(Record separator) (Unit separator) (Delete) alt + 164 alt + 165 alt + 165 alt + 168 alt + 168 alt + 168 alt + 173 alt + 33 alt + 58 alt + 47 alt + 92 matical bols alt + 171 alt + 172		? Wels (spani á é í ó ú Á É Í ó Ú Ú S S	95 acut ish lar al al al al al al al al crcia ymbo a	te ac nguag t + 16 t + 12 t + 21 t + 22 t + 23 l / tr ols lt + 36 t + 15	ccent (e) 50 50 51 52 53 51 52 53 51 52 53 51 52 53 51 52 53 51 52 53 53 54 55 56 56 56 56 56 56 56 56 56		IS9 VOV C ä ë ï Ö Ü Ü u pa	f liare	191 S with esis alt + 13 alt + 13 alt + 13 alt + 14 alt + 21 alt + 21 alt + 21 alt + 25 alt + 15 s and t + 15 s and t + 3 alt + 3 alt + 3 alt + 3	л 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	223	-	255	nbsp
230 330 31 127 fre (sp Ñ @ ¿? i! :/ \ %	athe sym	(Record separator) (Unit separator) (Delete) alt + 164 alt + 165 alt + 165 alt + 168 alt + 168 alt + 168 alt + 173 alt + 33 alt + 33 alt + 58 alt + 47 alt + 92 matical bols alt + 171 alt + 172 alt + 243	63 VO\ CO	? Wels (spani á ć í ó ú Á ć í ó ú ý ý s s f f	95 acut ish lar al al al al al al al al al al al al al		ccent (e) (6) (6) (1) (2) (3) (3) (4) (4) (4) (4) (4) (4) (4) (4		159 VOV C ä ë ï ö ü Ä Ë ï Ö Ü U u pa	f wels liare	191 S With SIS alt + 13 alt + 13 alt + 13 alt + 14 alt + 14 alt + 14 alt + 21 alt + 14 alt + 21 alt + 15 alt + 15 s and s alt + 3 alt + 3 alt + 4	1 22 77 29 29 20 20 20 20 20 20 20 20 20 20 20 20 20	223	-	255	nbsp
230 330 31 127 fre (sp Ñ @ ¿? i! :/ \ % % % %	athe sym	(Record separator) (Unit separator) (Delete) alt + 164 alt + 165 alt + 165 alt + 168 alt + 168 alt + 168 alt + 168 alt + 173 alt + 33 alt + 58 alt + 47 alt + 92 matical bols alt + 171 alt + 172 alt + 243 alt + 251		? Wels (spani á é í ó ú Á É Í Ó Ú Ó Ó Ú S S S S S	ercia ymbo al al al al al al al al al al al al al		ccent (e) 50 50 51 52 53 53 53 53 53 54 55 55 55 55 55 55 55 55 55		159 VOV d ä ë ï ö ü Ä Ë ï Ö Ü U u pa	f wels liare	191 S With SIS alt + 13 alt + 13 alt + 13 alt + 14 alt + 14 alt + 14 alt + 21 alt + 21 alt + 15 alt + 15 s and chesis alt + 3 alt + 4 alt + 4	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	223	-	255	nbsp
230 331 3127 fre (sp Ñ @ ¿? i!::/ \ % % %	athe sym	(Record separator) (Unit separator) (Delete) alt + 164 alt + 165 alt + 165 alt + 168 alt + 168 alt + 168 alt + 168 alt + 173 alt + 33 alt + 58 alt + 47 alt + 92 matical bols alt + 171 alt + 172 alt + 243 alt + 251 alt + 252		? Wels (spani á ć í ó ú Á É Í Ó Ú Ømme S) \$ £ ¥	95 acut ish lar al al al al al al al al al al al al al		ccent (e) 50 50 51 52 53 53 51 52 53 53 53 53 54 55 55 55 55 55 55 55 55 55		159 VOV d ä ë ï ö ü Ä Ë ï Ö Ü U u pa	f wels liare	191 S With S Sis alt + 13 alt + 13 alt + 14 alt + 12 alt + 14 alt + 21 alt + 14 alt + 21 alt + 15 s and s alt + 3 alt + 3 alt + 4 alt + 4 alt + 4 alt + 9	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	223	-	255	nbsp
230 331 127 fre (sp Ñ@ 2? i!:/ \ %4 %4 1 3 2	athe sym	(Record separator) (Unit separator) (Delete) alt + 164 alt + 165 alt + 165 alt + 168 alt + 168 alt + 168 alt + 173 alt + 33 alt + 58 alt + 47 alt + 92 matical bols alt + 171 alt + 172 alt + 243 alt + 251 alt + 252 alt + 253		? Wels (spani á ć í ó ú Á Á ć í ó Ú Ó Ú S S S S E E	95 acut ish lar al al al al al al al al al al al al al	te ac nguag t + 16 t + 16t + 16 t + 16 t + 16 t + 16 t + 16t + 16t + 16 t + 16t	ccent (e) 60 60 61 62 63 63 64 44 44 83 ade 6 6 6 6 7 9 9		IS9 VOV C a ë ï ö ü Ä Ë ï Ö Ü U Q U P a ' ' ( ) [ ]	f wels liare	191 S With S Sis alt + 13 alt + 13 alt + 14 alt + 12 alt + 14 alt + 21 alt + 21 alt + 21 alt + 21 alt + 15 S ano chesis alt + 3 alt + 4 alt + 4 alt + 4 alt + 9 alt + 9 alt + 9 alt + 9	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	223	-	255	nbsp
230 331 127 fre (sp Ñ@ 2? i!:/\ % % 1 3 2 f	athe sym	(Record separator) (Delete) (Unit separator) (Delete) alt + 164 alt + 165 alt + 164 alt + 168 alt + 168 alt + 173 alt + 33 alt + 58 alt + 47 alt + 58 alt + 47 alt + 92 matical bols alt + 171 alt + 172 alt + 243 alt + 251 alt + 253 alt + 159		? Wels (spani á é í ó ú Á É Í Ó Ú S S S S S S S S S S S S S S S S S S	ercia ymbo alt al al al al al al al al al alt alt	te ac nguag t + 16 t + 13 t + 16 t + 18 t + 180 t	ccent (e) 50 50 51 52 53 53 54 54 56 66 60 99 77 99 4		159 VOV d ä ë ï ö ü Ä Ë ï Ö Ü U u pa ' ( ) [ ] {	f wels liare	191 S With S Sis alt + 13 alt + 13 alt + 14 alt + 12 alt + 14 alt + 21 alt + 21 alt + 21 alt + 21 s and chesis alt + 3 alt + 4 alt + 4 alt + 4 alt + 9 alt + 9 alt + 12 chesis	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	223	-	255	nbsp
230 331 127 fre (sp Ñ @ ¿? i!: / ۱ ½ ¼ ¼ ¼ 1 3 2 f ±	athe sym	(Record separator) (Unit separator) (Delete) alt + 164 alt + 165 alt + 164 alt + 165 alt + 64 alt + 168 alt + 63 alt + 173 alt + 33 alt + 58 alt + 47 alt + 92 matical bols alt + 171 alt + 172 alt + 243 alt + 251 alt + 252 alt + 253 alt + 159 alt + 241	63 VO\ CO	? Wels (spani á ć í ó ú Á ć í ó ú ý S S S S S S S S S S S S S S S S S S	95 acut ish lar al al al al al al al al al al al al al	te ac nguag t + 16 t + 16t + 16 t + 16t + 16 t + 16 t + 16t + 16t + 16 t + 16t + 16t + 16 t	ccent ge) 60 60 60 61 62 63 63 64 64 65 66 09 97 99 44 66		159 VOV d ä ë ï ö ü Ä Ë ï ö Ü Ü u pa	f wels liare	191 S With S Sis alt + 13 alt + 13 alt + 14 alt + 12 alt + 14 alt + 21 alt + 21 alt + 21 alt + 21 alt + 15 s and t + 15 s alt + 3 alt + 3 alt + 4 alt + 4 alt + 4 alt + 9 alt + 12 alt + 15 s and chesis alt + 3 alt + 4 alt + 4 alt + 12 alt + 15 s and chesis alt + 3 alt + 4 alt + 12 alt + 15 s and chesis alt + 3 alt + 4 alt + 12 alt + 15 s and chesis alt + 3 alt + 4 alt + 21 alt + 15 s and chesis alt + 3 alt + 4 alt + 4 alt + 12 alt + 15 s and chesis alt + 3 alt + 4 alt + 4 alt + 12 alt + 15 s and chesis alt + 3 alt + 4 alt + 12 alt + 15 s and chesis alt + 12 alt + 12 al	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	223		255	nbsp
230 31 127 fresp i!:/\ % % 1 3 2 f ± x	athe sym	(Record separator) (Unit separator) (Delete) alt + 164 alt + 165 alt + 165 alt + 168 alt + 168 alt + 168 alt + 173 alt + 33 alt + 33 alt + 58 alt + 47 alt + 92 matical bols alt + 171 alt + 172 alt + 243 alt + 251 alt + 253 alt + 159 alt + 241 alt + 158		? Wels (spani á é í ó ú Á É Í Ó Ú Ó Í Ó Ú S S S S S S S S S S S S S S S S S S	ercia ymbo al al al al al al al al al al al al al	$\begin{array}{c} - \\ te a \\ nguag \\ t + 16 \\ t + 14 \\ t + 21 \\ t + 23 \\ t + 16 \\ t + 15 \\ t + 16 \\ t + $	ccent je) 50 50 51 52 53 53 54 54 55 55 55 55 55 55 55 55		159 VOV G ä ë ï ö ü Ä Ë ï Ö Ü U u pa " ' ( ) [ ] { } «	f wels liare	191 S With S Sis alt + 13 alt + 13 alt + 14 alt + 14 alt + 14 alt + 14 alt + 14 alt + 14 alt + 15 S ANC S AN	1 22 77 99 89 22 1 6 33 4 90 11 32 32 54	223		255	nbsp

### L'écriture



Jusqu'à maintenant, vous avez probablement modifié les programmes de manière longue et ennuyeuse en les retapant. Si c'est le cas, vous serez content d'être arrivé à ce chapitre.

Vous apprendrez une nouvelle façon facile de changer les programmes en les « éditant ».

Ne jetez pas cette ligne, Modifier la !

#### La commande EDIT

Pour plus de détail voir la section PC-BASIC, Éditer des programmes

Imaginez que vous faites une erreur en tapant un programme. La ligne 50 se termine en quelque sorte :

#### 50 DABA EFFUSIF, TRAS DETREMPER

Vous pouvez modifier cette ligne à la dure, en la retapant, ou à la méthode simple, en l'éditant. Pour entrer dans le "mode d'édition" de la ligne 50, tapez :

EDIT 50

Vous voyez

# EDIT 50 50 DABA EFFUSIF, TRAS DETREMPER\_

Vous êtes maintenant en mode édition. Dans ce mode, vous pouvez utiliser n'importe laquelle des "touches d'édition" spéciales pour afficher ou modifier la ligne 50. Elles sont toutes répertoriées plus loin.

#### DÉPLACEZ-VOUS SUR LA LIGNE (Mouvement du curseur)

Lorsque vous éditez une ligne, vous pouvez utiliser les flèches, les touches DEL, INS et Backspace pour faire des changements dans cette ligne.

Pour effacer une ligne entrez le numéro de la ligne et appuyez sur ENTER

Ex. 10 + ENTER efface la ligne 10.

Appuyez plusieurs fois sur Backspace. Cette touche vous fait reculer et efface les caractères.

## 50 DABA EFFUSIF, TRAS DETREM

Faites Ctrl + C et EDIT 50 à nouveau

Déplacez le curseur au début de la ligne 50 avec la flèche ← et appuyez sur la BARRE D'ESPACEMENT, le 5 est effacé.

## **0 DABA EFFUSIF, TRAS DETREMPER**

Faites Ctrl + C et EDIT 50 à nouveau

Avec la flèche ← déplacez le curseur sous le A et tapez E. Le A est remplcé pour un E.

## 50 DABA EFFUSIF, TRES DETREMPER

Appuyez sur ENTER, la ligne est modifiée enregistrée.

Faites EDIT 50 à nouveau, vous voyez le changement.

## 50 DABA EFFUSIF, TRES DETREMPER

Changer le B de DABA pour un D.

Pour insérer un caractère appuyer sur INS à l'endroit désiré.

Pour changer EFFUSIF en EFFUSIVE, déplacer d'abord le curseur sous le F et TAPER V, ensuite appuyez sur INS et taper E et ENTER.

Faites EDIT 50 à nouveau, vous voyez le changement.

## 50 DADA EFFUSIVE, TRES DETREMPER

Pour ajouter des caractères à la ligne faites EDIT 50 à nouveau et tapez les caractères à ajoutés.

Ajouter ,MUSQUELER, FORT et appuyer sur ENTER.

Faites EDIT 50 à nouveau, vous voyez le changement.

## 50 DADA EFFUSIVE, TRES DETREMPER ,MUSQUELER, FORT

Pour effacer une ligne taper le numéro de la ligne + ENTER.

Pour effacer la ligne 50, taper 50 + ENTER.

Faites EDIT 50 à nouveau, vous voyer le message d'erreur Undefined line number.

Ce qui signifie que la ligne 50 n'existe PLUS. Faites attention à ne pas effacer une ligne par erreur !!!

## EDIT 50 Undefined line number Ok

Pour effacer une ou plusieurs lignes qui se suivent faites DELETE numéro de ligne - numéro de ligne2

DELETE 20-60 efface les lignes de 20 à 60

DELETE 120- efface les lignes de 120 jusqu'à la fin.

## Faites attention à ne pas effacer des lignes par erreur !!!

### Clés spéciales

Les touches suivantes ont un effet spécial dans l'environnement de travail :

$\uparrow$ or Ctrl + 6	Déplace le curseur vers le haut, sauf sur la ligne du haut.			
↓ or Ctrl + -	Déplace le curseur vers le bas, sauf à la ligne 24.			
← or Ctrl + ]	Déplace le curseur vers la gauche. Le bord gauche de l'écran s'enroule, sauf au rang supérieur.			
$\rightarrow$ or Ctrl + /	Déplace le curseur vers la droite. Le bord droit de l'écran s'enroule, sauf au rang 24.			
$Ctrl + \leftarrow ou Ctrl + B$	Passe à la première lettre du mot précédent. Les mots sont composés de lettres A—Z et chiffres 0—9.			
$Ctrl + \rightarrow ou Ctrl + F$	Passe à la première lettre du mot suivant.			
Tab or Ctrl + I	Déplace le curseur sur le taquet de tabulation suivant. Les taquets de tabulation ont une largeur de 8 colonnes.			
Backspace ou Ctrl + H	Supprime le caractère à gauche du curseur, déplacez tous les autres caractères sur la ligne logique une position vers la gauche et modifier les attributs de ces caractères à l'attribut courant. Au bord gauche de l'écran, cela fait la même chose que Del.			
Del ou Ctrl + Backspace	Supprime le caractère au niveau du curseur et décaler tous les autres caractères d'une position vers la gauche, changeant les attributs en courant.			
Esc or Ctrl + [	Supprimer la ligne logique courante.			
Ctrl + End ou Ctrl + E	Supprime tous les caractères depuis le curseur jusqu'à la fin de la ligne logique.			
Ctrl + Break ou Ctrl + C ou Ctrl + Scroll Lock	Passe à la première colonne de la ligne suivante, sans exécuter ni stocker la ligne sous le curseur.			

Enter ou Ctrl + M	Exécute ou stocke la ligne logique courante. La ligne complète sur l'écran est considéré comme faisant partie de la commande, même si vous n'avez pas tapé ce. Une ligne commençant par un chiffre est stockée en tant que ligne de programme.		
End ou Ctrl + N	Déplace le curseur à la première position après la fin de la ligne logique.		
Home ou Ctrl + K	Déplace le curseur en haut à gauche de l'écran.		
Ctrl + Home ou Ctrl + L	Efface l'écran et déplacez le curseur en haut à gauche de l'écran.		
Ctrl + Enter ou Ctrl + J	Passe à la première colonne de la ligne suivante, reliant les deux lignes en une seule ligne logique.		
Ctrl + G	Bip le haut-parleur.		
Pause ou Ctrl + Num Lock	Pause. Appuyez sur une autre touche pour reprendre. La dernière pression de touche ne sera pas détectée par BASIC.		
Ctrl + Prt Sc	Bascule la sortie d'écran en écho vers l'imprimante (ou un autre périphérique attaché à LPT1 :).		
Shift + Prt Sc	Imprime l'écran.		
Ins or Ctrl + R	Bascule le mode d'insertion. En mode insertion, les caractères sont insérés plutôt qu'écrasé à l'emplacement actuel. Si l'insertion fait que la ligne étendre la largeur physique de l'écran, la ligne logique s'étend sur ligne suivante. Les touches fléchées quittent le mode d'insertion.		

## **Clés alternatives**

Dans PC-BASIC, la touche F12 peut être utilisée pour saisir des touches spéciales qui ne sont pas présentes sur certains claviers.



Vous pouvez écrire votre programme avec un éditeur de texte comme Bloc-Notes de Windows et ensuite utiliser la commande LOAD" de PC-BASIC. Exemple LOAD"Blackjac pour charger le programme BLACKJAC.BAS.

Pour sauvegarder un programme sur le disque utilisez SAVE:

SAVE "PROGRAM", A enregistre le programme actuel dans un fichier texte nommé PROGRAM.BAS .

Pour charger un programme utilisez LOAD:

LOAD "PROGRAM (Avec une seule ") charge le fichier programme nommé PROGRAM.BAS en mémoire.

Pour avoir la liste des fichiers faites FILES.

Dans l'explorateur de fichier de Windows vous pouvez sélectionner et copier le nom du fichier à charger dans PC-BASIC.

Par la suite dans PC-BASIC vous appuyer sur F3 LOAD"; ensuite F11 + V Coller le nom du fichier depuis le presse-papiers; et ensuite faites F2 (RUN) si vous voulez exécuter immédiatement le programme.

#### Votre numéro est en ligne !

#### La commande RENUM

Vous pouvez désormais tout modifier sur une ligne de programme, même le numéro de ligne.

Mais pour renuméroter plusieurs lignes, vous pouvez même le faire avec RENUM.

Pour voir comment fonctionne RENUM, tapez ce petit programme :

10 PRINT "C'EST LA PREMIERE LIGNE" 20 PRINT "C'EST LA DEUXIEME LIGNE" 30 PRINT "VOICI UNE AUTRE LIGNE" 40 GOTO 10

Maintenant, renumérotez-le. Taper:

RENUM 100

Lister le programme et vous voyez les nouveaux numéros de ligne commençant par 100.

La ligne 100 est ce que nous appelons la nouvelle ligne :

100 PRINT "C'EST LA PREMIERE LIGNE" 110 PRINT "C'EST LA DEUXIEME LIGNE" 120 PRINT "VOICI UNE AUTRE LIGNE" 130 GOTO 100

Notez que même la référence du numéro de ligne GOTO est renumérotée.

Renumérotez à nouveau le programme avec une nouvelle ligne de 200.

Tapez :

RENUM 200,120

Ici, la nouvelle ligne est 200, mais la renumérotation commence à la ligne 120. La ligne 120 est ce que nous appelons la ligne de départ ;

100 PRINT "C'EST LA PREMIERE LIGNE" 110 PRINT "C'EST LA DEUXIEME LIGNE" 200 PRINT "VOICI UNE AUTRE LIGNE" 210 GOTO 100

Renuméroter le programme une fois de plus en lui donnant un incrément de 50 entre chaque ligne :

RENUM 300,,50

Ici, la nouvelle ligne est 300, puisque vous avez omis la ligne de départ, BASIC renumérote l'ensemble du programme. L'incrément entre les lignes est de 50 :

300 PRINT "C'EST LA PREMIERE LIGNE" 350 PRINT "C'EST LA DEUXIEME LIGNE" 400 PRINT "VOICI UNE AUTRE LIGNE" 450 GOTO 300

Voici la "syntaxe" de la commande RENUM :

RENUM [nouveau numéro],[ancien numéro][,incrément]]

Commentaires:

nouveau numéro est le premier numéro de ligne à utiliser dans la nouvelle séquence. La valeur par défaut est 10.

ancien numéro est la ligne du programme en cours où la renumérotation doit commencer. La valeur par défaut est la première ligne du programme.

incrément est l'incrément à utiliser dans la nouvelle séquence. La valeur par défaut est 10.

RENUM modifie également toutes les références de numéro de ligne suivant les instructions ELSE, GOTO, GOSUB, THEN, ON...GOTO, ON...GOSUB, RESTORE et RESUME pour refléter les nouveaux numéros de ligne.

Si un numéro de ligne inexistant apparaît après l'une de ces instructions, le message d'erreur "Undefined line x in y" s'affiche.

La référence de numéro de ligne incorrecte x n'est pas modifiée par RENUM, mais le numéro de ligne y peut être modifié.

RENUM ne peut pas être utilisé pour modifier l'ordre des lignes de programme (par exemple, RENUM 15,30 lorsque le programme comporte trois lignes numérotées 10, 20 et 30) ou pour créer des numéros de ligne supérieurs à 65529. Une erreur "Illegal function call " se produit.

Essayez d'autres variantes de cette commande.

Taper:

RENUM ,,20

Cela renumérote tout votre programme. La nouvelle ligne est 10 et l'incrément est 20 :

10 PRINT "C'EST LA PREMIERE LIGNE" 30 PRINT "C'EST LA DEUXIEME LIGNE" 50 PRINT "VOICI UNE AUTRE LIGNE" 70 GOTO 10

Tapez RENUM 40,30

Ici, la nouvelle ligne est 40 ; la ligne de départ est 30 ; et l'incrément est de 10 :

10 PRINT "C'EST LA PREMIERE LIGNE" 40 PRINT "C'EST LA DEUXIEME LIGNE" 50 PRINT "VOICI UNE AUTRE LIGNE" 60 GOTO 10

Tapez RENUM 5,40 et vous obtenez une erreur Illegal function call.

En effet, le résultat déplacerait la ligne 40 avant la ligne 10.

#### Arithmétique



Résoudre rapidement et avec précision de longs problèmes mathématiques est une tâche que votre ordinateur accomplit facilement. Avant de taper des formules longues et difficiles, cependant, il y a quelques raccourcis que vous voudrez utiliser.

Un moyen simple de gérer des formules mathématiques compliquées consiste à utiliser des "sous-routines",

Tapez et exécutez ce programme :

10 PRINT "EXECUTION DU PROGRAMME PRINCIPAL" 20 GOSUB 500 30 PRINT "MAINTENANT DE RETOUR DANS LE PROGRAMME PRINCIPAL" 40 END 500 PRINT "EXECUTION DU SOUS-PROGRAMME" 510 RETURN

AX(BY + C) - D + E(G/W) - F

GOSUB 500 indique à l'ordinateur d'aller au sous-programme qui commence à la ligne 500.

RETURN indique à l'ordinateur de renvoyer au mot BASIC qui suit immédiatement GOSUB.

Supprimez la ligne 40 et voyez ce qui se passe lorsque vous exécutez le programme.

Si vous l'avez fait, votre écran affiche :

EXECUTION DU PROGRAMME PRINCIPAL EXECUTION DU SOUS-PROGRAMME MAINTENANT DE RETOUR DANS LE PROGRAMME PRINCIPAL EXECUTION DU SOUS-PROGRAMME RETURN without GOSUB in 510

Voyez-vous pourquoi la suppression de END à la ligne 40 provoque cette erreur ?

Au début, le programme fonctionne comme avant. Il passe au sous-programme de la ligne 500 puis revient à la ligne PRINT qui suit immédiatement GOSUB.

Ensuite, puisque vous avez supprimé FIN, il passe à la ligne suivante - le sous-programme de la ligne 500. Cette fois, cependant, il ne sait pas où retourner. C'est parce qu'il ne fait que "passer" dans le sous-programme ; il n'est pas envoyé au sous-programme par une ligne GOSUB.

Ce sous-programme élève un nombre à n'importe quelle puissance :

5 CLS 10 INPUT "TAPEZ UN NOMBRE "; N 20 INPUT "TAPEZ LA PUISSANCE A LAQUELLE VOUS VOULEZ L'AUGMENTER "; P 30 GOSUB 2000 40 PRINT: PRINT N "A LA PUISSANCE" P "EST" E 50 GOTO 10 2000 REM FORMULE POUR ELEVER UN NOMBRE A LA PUISSANCE A 2010 E = 1 2020 FOR X = 1 TO P 2030 E = E \* N 2040 NEXT X 2050 IF P = 0 THEN E = 1 2060 RETURN

Avec GW-Basic vous pouvez utiliser des minuscules qui prend en charge les accents en minuscules.

5 CLS 10 INPUT "Tapez un nombre "; N 20 INPUT "Tapez la puissance à laquelle vous voulez l'augmenter "; P 30 GOSUB 2000 40 PRINT: PRINT N "à la puissance " P "est" E 50 GOTO 10 2000 REM Formule pour élever un nombre à la puissance A 2010 E = 1 2020 FOR X = 1 TO P 2030 E = E \* N 2040 NEXT X 2050 IF P = 0 THEN E = 1 2060 RETURN

Sont également introduits dans ce programme :

Les deux-points (:), à la ligne 40.

Vous pouvez combiner des lignes de programme en utilisant les deux-points pour les séparer. La ligne 40 contient les deux lignes :

PRINT et PRINT N "à la puissance " P "est" E

REM, dans Line 2000. REM ne signifie rien pour l'ordinateur. Placez les lignes REM où vous voulez dans votre programme pour vous aider à vous souvenir de ce que fait le programme ; ils ne font aucune différence dans le fonctionnement du programme.

Pour voir par vous-même, ajoutez ces lignes et exécutez le programme :

5 REM CECI EST UN PROGRAMME PARTICULIER,17 REM CETTE LIGNE CHANGERA-T-ELLE LE PROGRAMME?45 REM La ligne suivante garde le sous-programme séparé

Vous pouvez aussi utiliser ' à la place de REM pour insérer des commentaires.

Essayez ceci:

50 GOTO 10 REM Retour à la ligne 10

50 GOTO 10 ' Retour à la ligne 10

#### Programme faites-le vous même

Modifiez le programme ci-dessus pour que l'ordinateur imprime un tableau de carrés (un nombre à la puissance 2) pour les nombres, disons, de 2 à 10.

5 CLS 7 PRINT "Table des carrés " 8 PRINT 10 P = 220 FOR N = 2 TO 10 25 GOSUB 2000 30 PRINT N "\*" N "=" E, 40 NEXT N **50 END** 2000 REM Formule pour élever un nombre à la puissance A 2010 E=1 2020 FOR X = 1 TO P 2030 E = E \* N2040 NEXT X 2050 IF P = 0 THEN E = 1 **2060 RETURN** 

### Donnez un peu d'aide à l'ordinateur

À mesure que les formules mathématiques deviennent plus complexes, votre ordinateur a besoin d'aide pour les comprendre.

Par exemple, que se passe-t-il si vous voulez que l'ordinateur résolve ce problème :

Diviser la somme de 13 + 3 par 8

Vous voudrez peut-être que l'ordinateur arrive à la réponse de cette façon :

13+3 / 8=16/8=2

Mais, à la place, l'ordinateur arrive à une autre réponse.

Tapez cette ligne de commande et voyez :

PRINT 13 + 3 / 8

Ce qui donne 13.375

L'ordinateur résout les problèmes logiquement, en utilisant ses propres règles :

## **RÈGLES SUR L'ARITHMÉTIQUE**

L'ordinateur résout les problèmes arithmétiques dans cet ordre :

1. Premièrement. il résout toutes les opérations de multiplication et de division.

2. En dernier. il résout les opérations d'addition et de soustraction.

3. S'il y a égalité (c'est-à-dire plus d'une opération de multiplication/division ou d'addition/soustraction), il résout les opérations de gauche à droite.

Dans le problème ci-dessus, l'ordinateur suit ses règles :

D'abord, il fait la division (3/8 = .375)

Ensuite, il fait l'addition (13 + .375 = 13.375)

Pour que l'ordinateur résolve le problème différemment, vous devez utiliser des parenthèses.

Tapez cette ligne :

PRINT (13 + 3) / 8

Ce qui donne 2

Chaque fois que l'ordinateur voit une opération entre parenthèses, il résout cette opération avant de résoudre les autres.

## EXERCICE DE MATHÉMATIQUES POUR L'ORDINATEUR

Que pensez-vous que l'ordinateur affichera comme réponses à chacun de ces problèmes ?

PRINT 10 - (5 - 1) /2 PRINT 10 - 5 - 1 / 2 PRINT (10 - 5 - 1) /2 PRINT (10 - 5) - 1 /2 PRINT 10 - (5 - 1 /2)

Achevé? Tapez chacune des lignes de commande pour vérifier vos réponses.

Et si vous voulez que l'ordinateur résolve ce problème ?

Divisez 10 moins la différence de 5 moins 1 par 2

Vous demandez en fait à l'ordinateur de faire ceci :

(10-(5-1)) / 2

Lorsque l'ordinateur détecte un problème avec plusieurs jeux de parenthèses, il résout les parenthèses intérieures puis passe aux parenthèses extérieures. En d'autres termes, il fait ceci :



### **RÈGLES SUR LES PARENTHÈSES**

1. L'ordinateur résout d'abord les opérations entre parenthèses, avant d'en résoudre d'autres.

2. L'ordinateur résout d'abord les parenthèses intérieures. Ensuite, il en sort.

#### EXERCICE DE MATHÉMATIQUES INFORMATIQUE

Insérez des parenthèses dans le problème ci-dessous afin que l'ordinateur affiche 28 comme réponse :

PRINT 30 -9- 8 -7- 6

Réponse

PRINT 30 - (8 - (8 - (7 - 6)))

Le programme ci-dessous utilise deux sous-programmes. C'est pour ceux d'entre vous qui économisent en mettant la même somme d'argent à la banque chaque mois :

5 CLS 10 INPUT "Votre acompte mensuel "; M 20 INPUT "Taux d'intérêt annuel de la banque "; I 30 I = I/12 \* 0.01 40 INPUT "Nombre de dépôts " ; NP 50 GOSUB 1000 60 PRINT "Tu vas avoir \$" FV " dans" NP "Mois" **70 END** 1000 REM Formule d'intérêt mensuel composé 1010 N = 1 + I1020 GOSUB 2000 1030 FV = M \* E1040 RETURN 2000 REM Formule pour élever un nombre à une puissance 2010 E = 12020 FOR X = 1 TO NP 2030 E = E \* N2040 NEXT X 2050 IF NP = 0 THEN E = 12060 RETURN

Notez qu'un sous-programme "appelle" un autre. Cela convient à l'ordinateur tant que :

Il y a un GOSUB pour envoyer l'ordinateur à chaque sous-programme, et il y a un RETURN à la fin de chaque sous-programme.

Ci-dessous, vous trouverez des sous-routines mathématiques utiles que vous pouvez ajouter à vos programmes.

Après l'exécution du programme ci-haut, l'ordinateur affiche Tu vas avoir \$ 6.740905 dans 36 Mois.

Avec la fonction PRINT USING, on l'ordinateur affiche la valeur FV arrondi à la deuxième décimal.

60 PRINT "Tu vas avoir \$";USING "##.##"; FV;: PRINT " dans" NP "Mois"

### On peut utiliser l'opérateur ^ pour élever un nombre à une puissance.

Ce programme donne le même résultat:

5 CLS 10 INPUT "Votre acompte mensuel " ; M 20 INPUT "Taux d'intérêt annuel de la banque " ; I 30 I = I/12 \* 0.01 40 INPUT "Nombre de dépôts " ; NP 50 FV=M\*(1+I)^NP 60 PRINT "Tu vas avoir \$" FV " dans" NP "Mois"

#### **PRINT USING**

Pour Afficher des chaînes ou des nombres en utilisant un format spécifié.

Syntaxe:

PRINT USING expressions de chaîne; liste d'expressions[;]

Commentaires:

expressions de chaîne est une chaîne littérale ou une variable composée de caractères de formatage spéciaux. Les caractères de formatage déterminent le champ et le format des chaînes ou des nombres imprimés.

liste d'expressions se compose de la chaîne ou des expressions numériques séparées par des points-virgules.

Champs de chaîne

Les trois caractères suivants peuvent être utilisés pour formater le champ de chaîne :

!

Spécifie que seul le premier caractère de la chaîne doit être imprimé.

\n espaces\

Spécifie que 2+n caractères de la chaîne doivent être imprimés.

Si les barres obliques inverses sont saisies sans espaces, deux caractères sont imprimés ; si les barres obliques inverses sont tapées avec un espace, trois caractères sont imprimés, et ainsi de suite.

Si la chaîne est plus longue que le champ, les caractères supplémentaires sont ignorés.

Si le champ est plus long que la chaîne, la chaîne est justifiée à gauche dans le champ et complétée par des espaces à droite. Par exemple:

```
10 A$="LOOK": B$="OUT"
30 PRINT USING "!"; A$; B$
40 PRINT USING"\ \"; A$; B$
50 PRINT USING"\ \"; A$; B$;"!!"
RUN
LO
LOOKOUT
LOOK OUT!!
```

## &

Spécifie un champ de chaîne de longueur variable. Lorsque le champ est spécifié avec &, la chaîne est sortie exactement comme l'entrée.

Par exemple:

10 A\$="LOOK": B\$="OUT" 20 PRINT USING "!"; A\$ 30 PRINT USING "&"; B\$ RUN LOUT

#### **Champs numériques**

Les caractères spéciaux suivants peuvent être utilisés pour formater le champ numérique :

#

Un signe dièse est utilisé pour représenter chaque position de chiffre. Les postes de chiffres sont toujours pourvus. Si le nombre à imprimer comporte moins de chiffres que les positions spécifiées, le nombre est justifié à droite (précédé d'espaces) dans le champ.

Un point décimal peut être inséré à n'importe quelle position dans le champ. Si la chaîne de format spécifie qu'un chiffre doit précéder la virgule décimale, le chiffre est toujours imprimé (comme 0 si nécessaire). Les nombres sont arrondis si nécessaire.

Par exemple:

PRINT USING "##.##";.78 0.78 PRINT USING "###.##";987.654 987.65 PRINT USING "##.##" ;10.2,5.3,66.789,.234 10.20 5.30 66.79 0.23 Dans le dernier exemple, trois espaces ont été insérés à la fin de la chaîne de format pour séparer les valeurs imprimées sur la ligne.

+ Un signe plus au début ou à la fin de la chaîne de format entraîne l'impression du signe du nombre (plus ou moins) avant ou après le nombre.

- Un signe moins à la fin du champ de format entraîne l'impression des nombres négatifs suivis d'un signe moins.

Par exemple:

PRINT USING"+##.##";-68.95,2.4,55.6,-9 -68.95 +2.40 +55.60 -0.90 PRINT USING"##.##-";-68.95,22.449,-7.01 68.95 22.45 7.01-

\*\*

Un double astérisque au début de la chaîne de format entraîne le remplissage des espaces de début du champ numérique par des astérisques. Le \*\* spécifie également deux positions de chiffres supplémentaires.

Par exemple:

```
PRINT USING "**#.#";12.39,-0.9,765.1
*12.4* -09765.1
```

## \$\$

Un signe de dollar double au début de la chaîne de format entraîne l'impression d'un signe dollar immédiatement à gauche du nombre formaté. Le \$\$ spécifie deux positions de chiffres supplémentaires, dont l'une est le signe dollar.

Le format exponentiel ne peut pas être utilisé avec \$\$.

Les nombres négatifs ne peuvent pas être utilisés à moins que le signe moins ne soit à la droite.

Par exemple:

PRINT USING "\$\$###.##";456.78 \$456.78

\*\*\$

Le \*\*\$ au début d'une chaîne de format combine les effets des deux symboles ci-dessus. Les espaces de début sont remplis d'astérisques et un signe dollar est imprimé avant le nombre. \*\*\$ spécifie trois positions de chiffres supplémentaires, dont l'une est le signe dollar.

Par exemple:

#### PRINT USING "\*\*\$##.##";2.34

\*\*\*\$2.34

Une virgule à gauche du point décimal dans la chaîne de format entraîne l'impression d'une virgule à gauche de chaque troisième chiffre à gauche du point décimal. Une virgule à la fin de la chaîne de format est imprimée dans le cadre de la chaîne.

PRINT USING "####,.##";1234.5

1,234.50

^ ^ ^ ^

,

Quatre carets peuvent être placés après les caractères de position des chiffres pour spécifier le format exponentiel.

Les quatre carets permettent d'imprimer E+xx. Toute position de la virgule décimale peut être spécifiée. Les chiffres significatifs sont justifiés à gauche et l'exposant est ajusté. À moins qu'un signe + ou un signe + ou - à la fin ne soit spécifié, une position de chiffre est utilisée à gauche de la virgule décimale pour imprimer un espace ou un signe moins.

Par exemple:

```
PRINT USING "##.##^^^^";234.56
2.35E+02
PRINT USING ".####^^^^-";888888
OK
PRINT USING "+.##^^^^";123
+.12E+03
```

\_

Un trait de soulignement dans la chaîne de format entraîne la sortie du caractère suivant sous forme de caractère littéral.

Par exemple:

PRINT USING "\_!##.##\_!";12.34 !12.34!

Le caractère littéral lui-même peut être un trait de soulignement en plaçant "\_" dans la chaîne de format.

%

Un signe de pourcentage est imprimé devant le nombre si le nombre à imprimer est supérieur au champ numérique spécifié.

Si l'arrondi fait que le nombre dépasse le champ, un signe de pourcentage est imprimé devant le nombre arrondi.

Par exemple:

PRINT USING "##.##";111.22 %111.22 PRINT USING ".##"';.999 %1.00

Si le nombre de chiffres spécifié dépasse 24, une erreur "Illegal function call" se produit.

## **Opérateurs et fonctions Mathématiques**

Code	Opération	Résultat
x ^ y	Exponentiation	x élevé à la puissance y
x * y	Multiplication	Produit de x et y
x / y	Division	Quotient de x et y
$\mathbf{x} \setminus \mathbf{y}$	Division tronquée à l'entier	Quotient entier de x et y
x MOD y	Modulo	Reste entier de x par y (avec le signe de x)
x + y	Addition	Somme de x et y
x - y	Soustraction	Différence de x et y
+ y	Unaire Plus	Valeur de y
- y	Négation	Valeur négative de y

Fonction	
ABS	Valeur absolue
ATN	Arc tangente
COS	Cosinus
EXP	Exponentiel
FIX	Troncature
INT	Sol
LOG	Un algorithme naturel
SIN	Sinus
SGN	Signe

SQR	Racine carrée
TAN	Tangente

Pour plus de détails voir <u>Guide de langage</u>

## Pour convertir les degrés en radians

radians = degrés  $\times$  PI / 180°

## Tableau de conversion des degrés en radians

Dégrées (°)	Radians (rad)	Radians (rad)
0°	0 rad	0 rad
30°	$\pi/6$ rad	0.5235987756 rad
45°	$\pi/4$ rad	0.7853981634 rad
60°	$\pi/3$ rad	1.0471975512 rad
90°	$\pi/2$ rad	1.5707963268 rad
120°	2π/3 rad	2.0943951024 rad
135°	$3\pi/4$ rad	2.3561944902 rad
150°	$5\pi/6$ rad	2.6179938780 rad
180°	$\pi$ rad	3.1415926536 rad
270°	$3\pi/2$ rad	4.7123889804 rad
360°	$2\pi$ rad	6.2831853072 rad

Des Mots, Des Mots, Des Mots...



Une grande compétence de l'ordinateur est son don avec les mots, il peut inlassablement tordre et combiner les mots comme vous le souhaitez. Avec ce cadeau, vous pouvez l'amener à lire, écrire et même parler.

Combiner des mots

Tapez et exécutez ce programme :

5 CLS 10 PRINT "TAPEZ UNE PHRASE" 20 INPUT S\$ 30 PRINT "VOTRE PHRASE A" LEN(S\$) " CHARACTERES" 40 INPUT "ENTREZ OUI POUR UNE AUTRE PHRASE" ; A\$ 50 IF A\$="OUI" THEN 10

Impressionné ? LEN(S\$) calcule la longueur de la chaîne S\$, votre phrase. L'ordinateur compte chaque caractère de la phrase, y compris les espaces et les signes de ponctuation.

Effacez le programme et exécutez ceci, qui compose un poème (en quelque sorte):

5 CLS 10 A\$ = "Une Rose" 20 B\$ = " " 30 C\$ = "C'est une Rose" 40 D\$ = B\$ + C\$ 50 E\$ = "Et ainsi de suite" 60 F\$ = A\$ + D\$ + D\$ + B\$ + E\$ 70 PRINT F\$

Ici, le signe plus (+) combine des chaînes. Par exemple, D ("EST UNE ROSE") est une combinaison de B + C.

Vous pouvez rencontrer deux problèmes lors de la combinaison de chaînes. Ajoutez la ligne suivante et exécutez le programme. Cela montre les deux problèmes:

80 G = F\$ + F\$ + F\$ + F\$ + F\$ + F\$

Lorsque l'ordinateur arrive à la ligne 80, il affiche le premier problème avec cette ligne :

String too long in 80

Au démarrage, l'ordinateur ne réserve que 200 caractères d'espace pour travailler avec des chaînes. La ligne 80 lui demande de travailler avec 343 caractères. Pour réserver de la place pour autant de caractères et plus (jusqu'à 500), ajoutez cette ligne au début du programme et exécutez :

#### 1 CLEAR 500

Maintenant, lorsque l'ordinateur arrive à Line 80, il dispose de suffisamment d'espace de chaîne, mais affiche le deuxième problème avec cette ligne : String too long in 80.

Une chaîne ne peut pas contenir plus de 255 caractères. Lorsque vous stockez plus de 255 caractères, vous devez placer ces caractères dans plusieurs chaînes.

#### **Mots tordus**

Maintenant que vous pouvez combiner des chaînes, essayez de séparer une chaîne.

Tapez et exécutez ce programme :

1 CLS 10 INPUT "Tapez un mot" ; W\$ 20 PRINT "La première lettre est : "LEFT\$ (W\$,1) 30 PRINT "Les 2 dernières lettres sont : " RIGHT\$ (W\$,2) 40 GOTO 10

Voici comment fonctionne le programme :

À la ligne 10 vous entrez un mot. On assume que le mot est MACHINE:

Aux lignes 20 et 30, l'ordinateur calcule la première lettre de gauche et les deux dernières lettres de droite de la chaîne :

#### MACHINE

LEFT\$ (W\$,1) RIGHT\$ (W\$,2)

Exécutez le programme plusieurs fois pour voir comment cela fonctionne.

Ajoutez maintenant cette ligne au programme :

#### 5 CLEAR 500

Afin que votre ordinateur réserve suffisamment d'espace pour travailler avec des chaînes. Exécutez à nouveau le programme. Cette fois, saisissez une phrase plutôt qu'un mot.

#### EXERCICE DE PROGRAMMATION

Comment changeriez-vous les lignes 20 et 30 pour que l'ordinateur vous donne les 5 premières lettres et les 6 dernières lettres de votre chaîne ?

20 \_\_\_\_\_

30 \_\_\_\_\_

Réponses:

20 PRINT "Les cinq premières lettres sont :" LEFT\$ (W\$,5)

30 PRINT "Les six dernières lettres sont :" RIGHT\$ (W\$,6)

Effacez votre programme et tapez celui-ci : (Rappelez-vous comment effacer un programme ? Taper: NEW)

5 CLS 10 CLEAR 500 20 INPUT "TAPEZ UNE PHRASE" ; S\$ 30 PRINT "TAPEZ UN NUMERO DE 1 A " LEN(S\$) 40 INPUT X 50 PRINT "LA CHAINE COMMENCE AVEC LE CARACTERE" X 60 PRINT "LA CHAINE COMMENCE AVEC LE CARACTERE" X 60 PRINT "TAPEZ UN NUMERO DE 1 A " LEN(S\$) - X + 1 70 INPUT Y 80 PRINT "LA CHAINE SERA" Y "CARACTERES DE LONG" 80 PRINT "LA CHAINE EST:" MID\$(S\$,X,Y) 100 GOTO 20

Exécutez ce programme plusieurs fois pour voir si vous pouvez en déduire le fonctionnement de MID\$.

Voici comment fonctionne le programme :

À la ligne 20, supposons que vous saisissiez ICI UNE CHAINE :

À la ligne 30, l'ordinateur calcule d'abord la longueur de S\$, qui est de 14 caractères. Il vous demande ensuite de choisir un nombre de 1 à 14. Supposons que vous choisissiez 5.

A la ligne 60, l'ordinateur vous demande de choisir un autre nombre de 1 à 10(14-5+1).

Supposons que vous choisissiez 4.

À la ligne 90, l'ordinateur vous donne une "chaîne médiane" de S\$ qui commence au 6e caractère et comporte quatre caractères :

1 2 3 4 5 6 7 8 9 10 11 12 13 14ICI UNE CHAINE  $\leftarrow 4 \rightarrow$ 

MID\$(S\$,6,4)

Pour un autre exemple de MID, effacez le programme et exécutez ceci :

5 CLS 10 INPUT "TAPEZ UNE PHRASE " ; S\$ 20 INPUT "TAPEZ UN MOT DANS LA PHRASE " ; W\$ 30 L = LEN (W\$) 40 FOR X = 1 TO LEN(S\$) 50 IF MID\$(S\$,X ,L) = W\$ THEN 90 60 NEXT X 70 PRINT "VOTRE MOT N'EST PAS DANS LA PHRASE" 80 END 90 PRINT W\$ " --COMMENCE AU CARACTERE NO. " X

Vous pouvez utiliser ce type de programme pour trier les informations.

Par exemple, en séparant les chaînes, vous pouvez parcourir une liste de diffusion pour les adresses.

Voici comment fonctionne le programme :

À la ligne 20, supposons que vous saisissiez le mot **UNE** pour W\$. A la ligne 30, l'ordinateur compte la longueur de W\$ : 3 caractères.

Aux lignes 40 à 90 (la boucle FOR/NEXT), l'ordinateur compte chaque caractère de S\$, en commençant par le caractère 1 et en terminant par le caractère LEN(S\$), qui est 14.

Chaque fois que l'ordinateur compte un nouveau caractère, il regarde un nouveau le milieu de chaîne. Chaque chaîne médiane commence au caractère X et a une longueur de L(3) caractères.

Par exemple, lorsque X est égal à 1, l'ordinateur regarde cette chaîne médiane :

1234567891011121314ICI UNE CHAINE  $\leftarrow 2$ 

MID\$(S\$,1,2)

La quatrième fois dans la boucle, lorsque X est égal à 4, l'ordinateur regarde cette chaîne médiane :

 $\begin{array}{c} 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \\ I \ C \ I \ U \ N \ E \ C \ H \ A \ I \ N \ E \\ \leftarrow 4 \end{array}$ 

MID\$(S\$,4,2)

Lorsque X est égal à 5, l'ordinateur trouve finalement UNE, la chaîne intermédiaire qu'il recherche.

#### Programme faites-le vous même

Commencez le programme avec:

### 10 A\$ = "CHANGER UNE PHRASE."

Ajoutez une ligne qui insère ceci au début de A\$ :

C'EST FACILE DE

Ajoutez une autre ligne qui affiche la nouvelle phrase :

### C'EST FACILE DE CHANGER UNE PHRASE

Voici notre programme :

5 CLS 10 A\$ = "MODIFIER UNE PHRASE." 20 B\$ = "C'EST FACILE DE" 30 C\$ = B\$ + " " + A\$ 40 PRINT C\$

### Programme faites-le vous même

Ajoutez au programme ci-dessus pour faire:

Trouvez le début de:

UNE PHRASE

Supprimez la sous-chaîne ci-dessus pour former cette nouvelle chaîne :

C'EST FACILE DE CHANGER

Ajoutez ces mots à la fin:

#### TOUT CE QUE TU VEUX

Affichez la chaîne nouvellement formée :

#### IL EST FACILE DE CHANGER TOUT CE QUE VOUS VOULEZ

ASTUCE : Pour former la chaîne IL EST FACILE DE CHANGER, vous devez obtenir la partie gauche de la chaîne IL EST FACILE DE CHANGER UNE PHRASE.

Ce programme est la base d'un programme de "traitement de texte", un programme populaire qui réduit les dépenses de dactylographie.

Réponse:

5 CLS 10 A\$ = "MODIFIER UNE PHRASE." 20 B\$ = "C'EST FACILE DE" 30 C\$ = B\$ + " " + A\$ 40 PRINT C\$ 50 Y = LEN ("A SENTENCE") 60 FOR X =1 TO LEN(C\$) 70 IF MID\$(C\$,X,Y) = "UNE PHRASE" THEN 90 80 NEXT X 85 END 90 D\$ = LEFT\$ (C\$,X - 1) 100 E\$ = D\$ + "TOUT CE QUE VOUS VOULEZ" 110 PRINT E\$

#### Programme faites-le vous même

Écrivez un programme qui :

Vous demande de saisir une phrase.

Vous demande de saisir

(1) une expression dans la phrase à supprimer et

(2) une expression pour la remplacer.

Affiche la phrase modifiée.

Cela peut prendre un certain temps, mais vous avez tout ce dont vous avez besoin pour l'écrire.

```
5 CLS
10 PRINT "TAPEZ UNE PHRASE :"
15 INPUT S$
20 PRINT "TAPEZ UNE PHRASE A SUPPRIMER"
23 INPUT D$
25 L = LEN(D\$)
30 PRINT "TAPEZ UNE PHRASE DE REMPLACEMENT"
35 INPUT R$
40 FOR X = 1 TO LEN(S$)
50 IF MID$(S$,X,L) = D$ THEN 100
60 NEXT X
70 PRINT D$ " N'EST PAS DANS VOTRE PHRASE"
80 GOTO 20
100 \text{ E} = \text{X} - 1 + \text{LEN} (\text{D}\$)
110 \text{ NF} = LEFT$(S$,X-1) + R$ + RIGHT$(S$,LEN(S$) - E)
120 PRINT "LA NOUVELLE PHRASE EST :"
130 PRINT NF$
```



En utilisant un mot nommé **INKEY\$**, vous pouvez faire en sorte que l'ordinateur « surveille », « chronomètre » ou « teste » constamment ce que vous tapez.

Tapez et exécutez ce programme :

5 CLS 10 A\$ = INKEY\$ 20 IF A\$<>"" GOTO 50 'Si différant de vide 30 PRINT "VOUS N'APPUYEZ SUR RIEN" 40 GOTO 10 50 PRINT "LA TOUCHE QUE VOUS AVEZ APPUYÉE EST---- " A\$

INKEY\$ vérifie si vous appuyez sur une touche. Il le fait en une fraction de seconde.

Au moins les 20 premières fois qu'il vérifie, vous n'avez appuyé sur rien (""), la ligne 10 étiquette la touche sur laquelle vous appuyez comme A\$.

Ensuite, l'ordinateur prend une décision :

Si A\$ est égal à rien (""), il affiche VOUS N'APPUYEZ SUR RIEN et revient à la ligne 10 pour vérifier à nouveau le clavier.

Si A\$ est égal à quelque chose (tout sauf ""), l'ordinateur va à la ligne 50 et affiche la touche.

Ajoutez cette ligne et exécutez le programme :

60 GOTO 10

Peu importe à quelle vitesse vous êtes, l'ordinateur est plus rapide !

Effacez la ligne 30 pour voir sur quelles touches vous appuyez.
## **Battre l'ordinateur**

Tapez ce programme :

```
5 RANDOMIZE TIMER

10 X = INT(4*RND(1)+1)

20 Y = INT(4*RND(1)+1)

30 PRINT "QU'EST-CE QUE FONT" X " + " Y

40 T = 0

50 A$ = INKEY$

60 T = T + 1

70 SOUND 128,1

80 IF T = 15 THEN 200

90 IF A$ = "" THEN 50

100 GOTO 10

200 CLS

210 SOUND 180,30

220 PRINT "TROP TARD"
```

Voici comment fonctionne le programme :

Aux lignes 10, 20 et 30, l'ordinateur affiche deux nombres aléatoires et vous demande leur somme.

La ligne 40 définit T sur O. T est une minuterie.

La ligne 50 vous donne votre première chance de répondre à la question en une fraction de seconde.

La ligne 60 ajoute 1 à T, la minuterie. T est maintenant égal à 1. La prochaine fois que l'ordinateur arrive à la ligne 60, il ajoute à nouveau 1 à la minuterie pour que T soit égal à 2. Chaque fois que l'ordinateur exécute la ligne 60, il ajoute 1 à T.

La ligne 70 est là juste pour te rendre nerveux.

La ligne 80 indique à l'ordinateur que vous avez 15 chances de répondre. Une fois que T est égal à 15, le temps est écoulé. L'ordinateur vous insulte avec les lignes 200, 210 et 220.

La ligne 90 indique que si vous n'avez pas encore répondu, l'ordinateur doit revenir en arrière et vous donner une autre chance.

L'ordinateur n'atteint la ligne 100 que si vous répondez. La ligne 100 le renvoie pour un autre problème.

Comment pouvez-vous faire en sorte que l'ordinateur vous donne trois fois plus de temps pour répondre à chaque question ?

Réponse:

En changeant cette ligne :

80 IF T = 45 THEN 200

Vérification de vos réponses

Comment pouvez-vous demander à l'ordinateur de vérifier si votre réponse est correcte ?

Cela fonctionnerait-il?

100 IF A\$ = X + Y THEN 130 110 PRINT "INCORRECTE " X "+" Y "=" X + Y 120 GOTO 10 130 PRINT "CORRECT" 140 GOTO 10

Si vous exécutez ce programme (et répondez à temps), vous obtiendrez ce message d'erreur :

Type mismatch in 100

C'est parce que vous ne pouvez pas rendre une chaîne (A) égale à un nombre (X + Y). Vous devez en quelque sorte changer A en un nombre.

Changez la ligne 100 en tapant :

100 IF VAL (A\$) = X + Y THEN 130

VAL(A\$) convertit A\$ en sa valeur numérique.

Si A\$ est égal à la chaîne "5 :' par exemple, VAL(A\$) est égal au nombre 5. Si VAL(A\$) est égal à la chaîne "C :' VAL(A\$) est égal au nombre 0. ("C" n'a pas de valeur numérique.)

Pour rendre le programme plus difficile, modifiez ces lignes :

10 X = INT(49\*RND(1)+1) 20 Y = INT(49\*RND(1)+1) 45 B\$ = "" 90 B\$ = B\$ + A\$ 95 IF LEN(B\$) <> 2 THEN 50 100 IF VAL(B\$) = X + Y THEN 130

## Un test de dactylographie informatique

Voici un programme qui chronomètre la vitesse à laquelle vous tapez :

```
10 CLS
20 INPUT "Appuyez <ENTER> Lorsque vous êtes prêt à saisir cette phrase "; E$
30 PRINT "C'est maintenant le temps pour tous les hommes bons"
40 T = 1
50 A$ = INKEY$
60 IF A$ = "" THEN 100
70 PRINT A$;
80 B$ = B$ + A$
90 IF LEN(B$) = 51 THEN 120
100 T = T + 1
110 GOTO 50
```

120 S = T/74 130 M = S/60 140 R=9/M 150 PRINT 160 PRINT "Vous avez type à -- "R" --Mots/Min"

la ligne 40 met T, le temporisateur, à 1.

la ligne 50 vous donne votre première chance de taper une touche (A\$). Si vous n'êtes pas assez rapide, la ligne 60 envoie le programme à la ligne 100 et ajoute 1 à la minuterie.

la ligne 70 imprime la clé que vous avez tapée.

la ligne 80 forme une chaîne nommée B\$. Chaque fois que vous tapez une touche (A\$), le programme l'ajoute à B\$. Par exemple, si la première clé que vous saisissez est "N", alors :

A\$ = "C" et B\$= B\$+A\$ B\$= ""+"C" B\$ = "C"

Si la clé suivante que vous saisissez est " ' ", alors :

A\$ = " ' " et B\$ = B\$ + A\$ B\$="C"+" ' " B\$ = "C' "

Si la troisième clé que vous saisissez est "E", alors :

A\$ = "e" et B\$ "" "C"" + "e" B\$ = "C'e"

Lorsque la longueur de B\$ est de 51 (la longueur de (**C'est maintenant le temps pour tous les hommes bons**), le programme suppose que vous avez fini de taper la phrase et passe à la ligne 120 pour calculer vos mots par minute.

les lignes 120, 130 et 140 calculent votre vitesse de frappe. Ils divisent T par 74 (pour obtenir les secondes), S par 60 (pour obtenir les minutes). Ils divisent ensuite les neuf mots par M pour obtenir les mots par minute.

#### Plus de base



Avant d'en finir avec les "bases", vous devez connaître quelques mots supplémentaires.

Le premier est STOP.

Tapez et amusez-vous ce programme:

10 A = 1 20 A = A + 1 30 STOP 40 A = A \* 2 50 STOP 60 GOTO 20

L'ordinateur démarre l'exécution du programme. Lorsqu'il arrive à la ligne 30, il affiche :

Break in 30

Vous pouvez maintenant taper une ligne de commande pour voir ce qui se passe.

Par exemple, tapez :

PRINT A

L'ordinateur affiche 2, la valeur de A lorsque le programme est à la ligne 30.

Tapez maintenant :

CONT

L'ordinateur poursuit le programme. Lorsqu'il obtient 10 Ligne 50, il affiche :

Break in 50

Tapez :

PRINT A

Cette fois, l'ordinateur affiche 4, la valeur de A à la ligne 50.

Tapez à nouveau CONT et l'ordinateur s'arrête à nouveau à la ligne 30. Si vous le faites afficher à nouveau A, il affiche la valeur 5 la valeur de A à la ligne 30 la deuxième fois dans le programme.

L'insertion de lignes STOP dans votre programme vous aide à comprendre pourquoi cela ne fonctionne pas comme prévu. Lorsque vous corrigez le programme, supprimez les lignes STOP.

## Pour les longs programmes • • •

Effacez la mémoire et tapez :

PRINT FRE(0)

L'ordinateur affiche l'espace de stockage restant dans la mémoire de l'ordinateur.

Lorsque vous tapez un long programme, vous souhaiterez que l'ordinateur affiche de temps en temps l'espace de stockage restant pour vous assurer que vous ne manquez pas de mémoire.

# Aide à la saisie

Tapez ce programme :

10 INPUT "Tapez 1, 2, ou 3"; N 20 ON N GOSUB 100, 200, 300 30 GOTO 10 100 PRINT "Vous avez tapez 1" 110 RETURN 200 PRINT "Vous avez tapez 2" 210 RETURN 300 PRINT "Vous avez tapez 3" 310 RETURN

Exécuter.

ON ... GOSUB à la ligne 20 fonctionne comme trois lignes :

18 IF N = 1 THEN GOSUB 100 20 IF N = 2 THEN GOSUB 200 22 IF N = 3 THEN GOSUB 300

ON ... GOSUB regarde le numéro de ligne suivant ON - dans ce cas N.

Si N vaut 1, l'ordinateur passe au sous-programme en commençant par le premier numéro de ligne suivant GOSUB.

Si N vaut 2, l'ordinateur passe au sous-programme en commençant par le deuxième numéro de ligne.

Si N vaut 3, l'ordinateur passe au sous-programme commençant au troisième numéro de ligne.

Et si N vaut 47 Comme il n'y a pas de numéro de quatrième ligne, l'ordinateur passe simplement à la ligne suivante du programme.

Voici un programme qui utilise ON ... GOSUB:

**1 RANDOMIZE TIMER** 5 FOR P = 1 TO 600: NEXT P 10 CLS: X = INT(100\*RND(1)+1): Y = INT(100\*RND(1)+1)20 PRINT"(1) ADDITION" 30 PRINT "(2) SOUSTRACTION" 40 PRINT "(3) MULTIPLICATION" 50 PRINT "(4) DIVISION" 60 INPUT "QUEL EXERCICE (1-4)"; R **70 CLS** 80 ON R GOSUB 1000, 2000, 3000, 4000 90 GOTO 5 1000 PRINT "COMMENT FONT " X "+" Y **1010 INPUT A** 1020 IF A =X + Y THEN PRINT "CORRECT" ELSE 1030 PRINT "ERREUR" **1040 RETURN** 2000 PRINT "COMMENT FONT " X "-" Y **2010 INPUT A** 2020 IF A = X-Y THEN PRINT "CORRECT" ELSE PRINT "ERREUR" **2030 RETURN** 3000 PRINT "COMMENT FONT " X "\*" Y **3010 INPUT A** 3020 IF A = X\*Y THEN PRINT "CORRECT" ELSE PRINT "ERREUR" 3030 RETURN 4000 PRINT "COMMENT FONT " X "/" Y "AVEC 7 DECIMALES" **4010 INPUT A** 4020 IF A = X/Y THEN PRINT "CORRECT" ELSE PRINT "ERREUR" **4030 RETURN** 

Remarquez le mot ELSE aux lignes 1020, 2020, 3020 et 4020. Vous pouvez utiliser ELSE si vous voulez que l'ordinateur fasse quelque chose de spécial lorsque la condition n'est pas vraie. À la ligne 1020, si votre réponse A est égale à X + Y, alors l'ordinateur imprime CORRECT ou bien il imprime ERREUR.

Vous pouvez utiliser ON ... GOTO de la même manière que ON ... GOSUB. La seule différence est que ON GOTO envoie l'ordinateur à un autre numéro de ligne plutôt qu'à un sous-programme.

Voici une partie d'un programme utilisant ON ... GOTO

10 CLS 20 PRINT "(1) CRAZY EIGHTS" 30 PRINT "(2) 500" 40 PRINT "(3) HEARTS" 50 PRINT "QUE VOULEZ-VOUS JOUER ? " 60 INPUT A 65 CLS 70 ON A GOTO 1000, 2000, 3000 1000 PRINT "JEU CRAZY EIGHTS" 1010 END 2000 PRINT "JEU 500" 2010 END 3000 PRINT "JEU HEARTS" 3010 END

Le travail indique-t-il "AND" ou "OR" ?

Quiconque parle anglais connaît la différence entre « AND (et)» et « OR (ou) », même votre ordinateur.

Par exemple, supposons qu'il y ait une offre d'emploi en programmation.

Le travail nécessite :

Un diplôme en programmation AND (et) Expérience en programmation

Effacez la mémoire et tapez :

10 PRINT "AS-TU --" 20 INPUT "UN DIPLOME EN PROGRAMMATION"; D\$ 30 INPUT "EXPERIENCE EN PROGRAMMATION"; E\$ 40 IF D\$ = "OUI" AND E\$ = "OUI" THEN PRINT "VOUS AVEZ LE TRAVAIL" ELSE PRINT "DESOLÉ, NOUS NE POUVONS PAS VOUS ENGAGER" 50 GOTO 10

Exécutez le programme. Vous pouvez répondre aux questions de cette façon :

RUN AS-TU --UN DIPLOME EN PROGRAMMATION? NON EXPERIENCE EN PROGRAMMATION? OUI DESOLÉ, NOUS NE POUVONS PAS VOUS ENGAGER

Supposons maintenant que les exigences changent de sorte que "AND" devienne "OR".

Le travail nécessite désormais :

Un diplôme en programmation OR (ou) Expérience en programmation

Pour effectuer cette modification dans le programme, tapez :

40 IF D\$ = "OUI" OR E\$ = "OUI" THEN PRINT "VOUS AVEZ LE TRAVAIL" ELSE PRINT "DESOLÉ, NOUS NE POUVONS PAS VOUS ENGAGER"

Exécutez le programme et voyez quelle différence AND et OR font :

RUN AS-TU --UN DIPLOME EN PROGRAMMATION? NON EXPERIENCE EN PROGRAMMATION? OUI VOUS AVEZ LE TRAVAIL

# Plus d'arithmétique

Ces mots peuvent sauver de nombreuses lignes de programme :

## SNG

SGN vous indique si un nombre est positif. négatif, ou zéro :

10 INPUT "Tapez un nombre"; X 20 IF SGN(X) = 1 THEN PRINT "POSITIF" 30 IF SGN(X) = 0 THEN PRINT "ZERO" 40 IF SGN(X) = -1 THEN PRINT "NEGATIF" 50 GOTO 10

# ABS

ABS vous indique la valeur absolue d'un nombre (la magnitude du nombre sans égard à son signe).

Taper:

```
10 INPUT "Tapez un nombre"; X
20 PRINT "La valeur absolue de X est " ABS(X)
30 GOTO 10
```

Exécutez le programme en saisissant les mêmes nombres que ceux ci-dessus.

## STR\$

STR\$ convertit un nombre en une chaîne. Exemple:

10 INPUT "Tapez un nombre"; N 20 A\$ = STR\$(N) 30 PRINT A\$ + ".. Est maintenant une chaîne"

#### **Exposants**

Tapez et exécutez ce programme pour voir comment l'ordinateur traite les très grands nombres :

10 X =1 20 PRINT X; 30 X = X \* 10 40 GOTO 20 Remarquez l'erreur Overflow à la fin. L'ordinateur ne peut pas gérer les nombres supérieurs à IE +38 ou inférieurs à -1E+38.

L'ordinateur imprime des nombres très grands ou très petits en "notation exponentielle". "Un milliard" (1 000 000 000), par exemple, devient 1E+09, ce qui signifie "le chiffre 1 suivi de neuf zéros".

Si une réponse sort "5E-6", vous devez décaler la virgule décimale. qui vient après le 5, six places vers la gauche, en insérant des zéros si nécessaire.

Techniquement, cela signifie 5.10-6, ou 5 millionièmes (.000005).

La notation exponentielle est simple une fois que vous vous y êtes habitué. Vous y trouverez un moyen facile de suivre des nombres très grands ou très petits sans perdre la virgule décimale.

## Félicitations, Programmeur !

Vous avez maintenant appris les "bases" et pouvez sans aucun doute écrire des programmes décents. La section suivante vous aidera à ajouter du piquant à vos programmes avec des graphismes et de la musique.

**Allons au Point** 

L'une des fonctionnalités les plus intéressantes de GW-BASIC est sa capacité à afficher des graphiques précis, variés et faciles à utiliser appelés "graphiques haute résolution".

À quel point ces graphiques sont-ils faciles à utiliser ?

Eh bien, commençons par l'élément graphique le plus basique (jeu de mots) - un point et construisons à partir de là.

GW-BASIC simplifie l'affichage d'un point sur l'écran.

Tapez le programme suivant et voyez :

10 SCREEN 9 20 CLS 30 PSET (10,20),7 40 GOTO 40

Exécutez maintenant le programme. L'écran devrait être chamois, et si vous regardez attentivement, vous pouvez voir un petit point blanc dans le coin supérieur gauche. Ce point a été mis là par le PSET (sélection d'un point dans la ligne 30).

PSET affiche un point à un endroit spécifié sur l'écran lors de l'utilisation du mode graphique.

Syntaxe:

PSET(x,y)[,couleur]

Commentaires:

(x,y) représente les coordonnées du point.

couleur est la couleur du point.

Les coordonnées peuvent être données sous forme absolue ou relative.

La forme absolue (x absolu, y absolu) est plus courante et se réfère directement à un point sans tenir compte du dernier point référencé.

Par exemple:

(10,10)

Forme relative

STEP (x décalage, y décalage) est un point relatif au point le plus récent référencé.

Par exemple:

STEP(10,10)

Les valeurs de coordonnées peuvent être au-delà du bord de l'écran. Cependant, les valeurs en dehors de la plage d'entiers (-32768 à 32767) provoquent une erreur "Overflow".

(0,0) est toujours le coin supérieur gauche et (639,349) est le coin inférieur droit pour SCREEN 9.

Voir les instructions <u>SCREEN, COLOR, PALETTE</u> pour plus d'informations.

Même si vous ne pouvez pas le voir, l'ordinateur a divisé votre écran en une grille de près de 223 011 points - 639 en largeur et 349 en bas - afin que vous puissiez placer un point précisément là où vous le souhaitez.

Regardez à nouveau la ligne 30 et voyez comment PSET spécifie la position du point (10 à gauche et 20 en bas):

# 30 PSET (10,20),7

Voici la déclaration que vous utiliseriez pour définir un point rouge au centre de l'écran :

PSET (319,174),4

Ajoutez maintenant une ligne de programme qui définit un point rouge dans le coin inférieur droit (639 audessus et 349 en bas). Est-ce la ligne que vous avez utilisée ?

PSET (639,349),4

Si oui, félicitations !

Vous avez fait valoir votre point de vue. Exécutez votre programme et vous verrez.

Ce qui suit trace une ligne diagonale de (0,0) à (100,100) dans SCREEN 1  $(320 \times 200 \text{ pixels})$ .

10 CLS 20 SCREEN 1 30 FOR I=0 TO 100 40 PSET (I,I) 50 NEXT 60 LOCATE 14,1

Le programme ci-dessous trace une ligne diagonale de (0,0) à (100,100) dans SCREEN 1,2,7,8,9

10 SCREEN 1 20 LOCATE 1,5:PRINT "SCREEN 1, 320 × 200 pixels, 40 caractères" 30 GOSUB 1000 40 SCREEN 2 50 LOCATE 1,5:PRINT "SCREEN 2, 640 × 200 pixels 80 caractères" 60 GOSUB 1000 70 SCREEN 7 80 LOCATE 1,5:PRINT "SCREEN 7, 320 × 200 pixels, 40 caractères" 90 GOSUB 1000 **100 SCREEN 8** 110 LOCATE 1,5:PRINT "SCREEN 8, 640 × 200 pixels, 80 caractères" 120 GOSUB 1000 **130 SCREEN 9** 140 LOCATE 1,5:PRINT "SCREEN 9, 640 × 350 pixels, 80 caractères" 150 GOSUB 1000 160 LOCATE 14,1 170 SCREEN 0 180 END 1000 FOR I=0 TO 100 1010 PSET (I,I) 1020 NEXT 1030 LOCATE 14,1:PRINT"Appuyez sur une touche pour continuer" 1040 A\$=INKEY\$:IF A\$="" THEN 1040 **1050 RETURN** 

Mais qu'en est-il de la couleur ?

À présent, vous avez probablement compris que vous pouvez changer les couleurs en remplaçant c par un nombre différent dans la plage de 0 à 15.

Voici la liste des 16 principales couleurs

0 Noir 1 Bleu 2 Vert 3 Cyan 4 Rouge 5 Magenta 6 Marron 7 Blanc 8 Gris 9 Bleu clair 10 Vert clair 11 Cyan clair 12 Rouge clair 13 Magenta clair 14 Jaune 15 Blanc haute intensité

Pour SCREEN 9, il y a 64 couleurs ....

## Maintenant, vous le voyez ... maintenant, vous ne le voyez pas

Des suppositions comment désactiver un point?

Voici un indice : c'est facile et c'est lié à la couleur.

Vous voyez, vous n'éteignez pas vraiment le point, vous changez simplement sa couleur pour qu'il se fonde dans l'arrière-plan.

Vous faites cela avec une nouvelle déclaration :

PRESET (réinitialisation du point). PRESET "sait" que vous voulez utiliser la couleur d'arrière-plan, vous n'avez donc pas besoin de donner la couleur.

PRESET (h,v) réinitialise un point sur l'écran graphique actuel

#### Programme faites-le vous même

Apprenez à connaître les positions des points sur votre écran de télévision en utilisant une feuille de travail de l'écran graphique (Papier millimétrique).

Sélectionnez plusieurs points sur la feuille de travail, identifiez-les en fonction de leurs coordonnées (X,Y) et affichez-les à l'écran à l'aide du programme que nous avons utilisé pour vous aider à démarrer.

Ne modifiez aucune ligne de programme sauf celles qui contiennent P5ET(h,v),c.

10 SCREEN 9 20 CLS 30 PSET (10,20),7 40 GOTO 40

## Programme faites-le vous même

Rappelez-vous la fonction RND (aléatoire) ?

Sinon, révisez-la, voir <u>Jeux de chances</u>; puis écrivez un court programme qui remplit l'écran avec des points et des couleurs aléatoires.

5 SCREEN 1 '320 × 200 pixels 10 RANDOMIZE TIMER 20 CLS 30 X = INT(320\*RND(1)+1) 40 Y = INT(200\*RND(1)+1) 50 C = INT(15\*RND(1)+1) 60 PSET(X,Y),C 70 GOTO 30

## Le dernier Point

Avant de terminer cette section, nous voulons faire une dernière remarque. Vous pouvez utiliser POINT pour savoir de quelle couleur est n'importe quel point à l'écran.

POINT(x,y)

5 SCREEN 7 '320 × 200 pixels 10 RANDOMIZE TIMER 20 CLS 30 X = INT(320\*RND(1)+1) 40 Y = INT(200\*RND(1)+1) 50 C = INT(15\*RND(1)+1) 60 PSET(X,Y),C 70 IF POINT (X,Y)=12 THEN GOTO 90 80 GOTO 30 90 PRINT "La couleur du point "X Y" est Rouge clair"

## Tenez cette ligne !



Ainsi, vous pouvez mettre <u>un point sur l'écran</u> --- même plusieurs points. Mais quel genre de point de départ estce, vous vous demandez peut-être, lorsque vous êtes désireux de créer de "vrais" graphiques.

Pour répondre à cette question, pensez à certains de vos tout premiers "dessins" sur papier. Peut-être s'agissait-il d'images détaillées de clowns, de phoques dressés et d'autres choses merveilleuses.

Comment avez-vous dessiné de telles merveilles ? Probablement en connectant un tas de points.

Et c'est exactement comme ça que votre ordinateur "dessine". Vous lui dites quels points se connectent, et il trace une ligne.

## C'est une ligne que vous avez

Une façon de dire à l'ordinateur de tracer une ligne entre les points est d'utiliser l'instruction LINE de GW-BASIC. Pour voir LINE au travail, modifiez le programme qui définit les points. (Pour plus de commodité, appelez le programme "lignes".)

10 SCREEN 9 '640 × 350 pixels 64 Couleurs 20 CLS 30 LINE (0,0)-(639,349),12 40 GOTO 40

Exécutez maintenant le programme. L'écran doit afficher une ligne rouge clair allant du coin supérieur gauche au coin inférieur droit sur un fond noir.

Que diriez-vous de changer la direction de la ligne pour qu'elle aille du coin inférieur gauche au coin supérieur droit ?

Vous avez probablement déjà compris celui-ci, mais au cas où, voici la nouvelle ligne 30 :

30 LINE (0,349)-(639,0),12

## X marque l'endroit

Qu'en est-il des lignes qui se croisent ?

Réinsérez la ligne 30 d'origine qui a tracé la première ligne. (Tout d'abord, renumérotez-le en tant que ligne 25.)

Ensuite, exécutez le programme. Votre écran affiche-t-il 2 lignes orange qui se croisent au centre ?

En fait, vous pouvez mettre autant de lignes à l'écran que vous le souhaitez, une fois que vous avez appris le format.

C'est ici:

LINE [(x1,y1)]-(x2,y2) [,[attribut][,B[F]]

x1,y1 et x2,y2 spécifient les extrémités d'une ligne.

Le mode de résolution est déterminé par l'instruction SCREEN.

L'attribut spécifie la couleur ou l'intensité du pixel affiché (voir les instructions SCREEN, COLOR, PALETTE).

B (boîte) dessine une boîte avec les points (x1,y1) et (x2,y2) aux coins opposés.

BF (boîte remplie) dessine une boîte (comme ,B) et remplit l'intérieur avec des points.

Noter

Si l'attribut n'est pas spécifié, deux virgules doivent être utilisées avant B ou BF.

Maintenant faites dessiner un carré en ajoutant B à la fin de la ligne 30 et ensuite un carré rempli en ajoutant BF

30 LINE (0,349)-(639,0),12,B et exécuter le programe

ensuite

30 LINE (0,349)-(639,0),12,BF

## Programme faites-le vous même

Écrivez un programme qui crée une boîte avec une paire de lignes se croisant au centre.

Tout comme à l'époque du point à point, vous souhaiterez souvent tracer une ligne qui commence au point de fin de la dernière ligne. Chaque fois que c'est le cas, vous pouvez omettre le point de départ. L'ordinateur démarre automatiquement au point final défini par la dernière instruction LINE ou, si vous n'avez pas encore utilisé LINE dans le programme, à au milieu de l'écran soit pour SCREEN 9 (319,174).

Voici un exemple:

30 LINE (0,0)-(639,349),12 35 LINE -(639,0),10

La ligne 30 trace une ligne de (0,0) à (639,349).

La ligne 35 trace alors une autre ligne, celle-ci de (639,349) au point (639,0).

Que vous incluiez ou non le point de départ, vous devez faire précéder le point de fin d'un trait d'union (-)

Que diriez-vous de laisser tomber une ligne ?

Nous avons discuté des points de départ et d'arrivée de la ligne.

Pour effacer une ligne utilisez la couleur de fond

10 SCREEN 9:CLS 20 LINE (0,0)-(639,349),12 30 A\$=INKEY\$:IF A\$="" THEN 30 40 LINE (0,0)-(639,349),0 50 GOTO 50

Pour un exemple de programme QUEBEC.BAS

## C'est la couleur avec un C majuscule, O majuscule, majuscule

À la setion <u>Allons au Point</u>, nous avons expliqué comment utiliser le paramètre c de la commande PSET pour changer la couleur d'un point. Mais nous parlons aussi depuis un certain temps des couleurs de premier plan (foreground) et d'arrière-plan (background). Il est maintenant temps de les expliquer davantage.

Naturellement, si vous utilisez beaucoup une couleur, vous ne voulez pas avoir à la spécifier chaque fois que vous mettez quelque chose à l'écran. Avec la fonction COLOR, vous n'avez pas à le faire.

Dans certaines limites, la fonction graphique COLOR permet de définir les couleurs de premier plan/arrièreplan.

Voici son format :

## COLOR [foreground][,[background]

Couleur 0 Noir 1 Bleu 2 Vert 3 Cyan 4 Rouge 5 Magenta 6 Marron 7 Blanc 8 Gris 9 Bleu clair 10 Vert clair 11 Cyan clair 12 Rouge clair 13 Magenta clair 14 Jaune

15 Blanc haute intensité

Pour LINE si l'attribut de la couleur est omit, la couleur des points de la lignes sera celle de la couleur du fond du premier plan (foreground).

Pour voir COLOR en action, tapez ce programme:

5 SCREEN 9:CLS 10 COLOR 2:Print "Vert" 20 LINE (0,0)-(639,349) 30 A\$=INKEY\$:IF A\$="" THEN 30 40 COLOR 3:Print "Cyan" 50 LINE (0,0)-(639,349) 60 GOTO 50

## Programme faites-le vous même

Prêt pour votre propre programmeur "lignes"

Pouvez-vous construire une maison?

Assurez-vous d'ajouter :

Une porte, bien sûr.

Au moins une fenêtre. (N'oubliez pas d'allumer ou d'éteindre les lumières.)

Une cheminée. (Vous n'aurez pas besoin d'un ramoneur, pas encore en tout cas !)

La conception globale vous appartient (Cape Cod, Ranch ou autre), mais nous avons inclus un exemple de programme de maison (bonne vue, pas d'animaux). Ne vous inquiétez pas des poignées de porte ; nous les ajouterons plus tard.

Assurez-vous de sauvegarder ce programme, car vous en aurez besoin plus tard. (Vous trouverez qu'il est beaucoup plus facile de dessiner la maison si vous tracez ses points sur une feuille de travail d'écran graphique.)

10 SCREEN 7

20 COLOR 10:CLS 'Permet de définir la couleur de l'avant plan (des points) 30 LINE (72,168)-(200,72),,B 'FRAME 40 LINE (72,72)-(136,36) 'ROOF 45 LINE (200,72)-(136,36) 'ROOF 50 LINE (120,168)-(152,100),,B 'DOOR 55 LINE (152,60)-(168,36),,BF 'CHIMNEY 60 LINE (165,128)-(191,100),,B 'WINDOW 65 LINE (165,128)-(191,100),,B 'WINDOW 65 LINE (178,128)-(178,100) 'WINDOW PART 70 LINE (165,114)-(191,114) 'WINDOW PART 75 LINE (85,128)-(111,100),,B 'WINDOW 80 LINE (85,114)-(111,114) 'WINDOW PART 85 LINE (98,100)-(98,128) 'WINDOW PART 90 GOTO 90 Le même programme sans la commande COLOR

10 SCREEN 7 20 CLS 30 LINE (72,168)-(200,72),10,B 'FRAME 40 LINE (72,72)-(136,36),10 'ROOF 45 LINE (200,72)-(136,36),10 'ROOF 50 LINE (120,168)-(152,100),10,B 'DOOR 55 LINE (152,60)-(168,36),10,BF 'CHIMNEY 60 LINE (165,128)-(191,100),10,B 'WINDOW 65 LINE (178,128)-(178,100),10 'WINDOW PART 70 LINE (165,114)-(191,114),10 'WINDOW PART 75 LINE (85,128)-(111,100),10,B 'WINDOW 80 LINE (85,114)-(111,114),10 'WINDOW PART 85 LINE (98,100)-(98,128),10 'WINDOW PART 90 GOTO 90

Vous pouvez changer l'attribut couleur de ligne pour avoir une maison plus colorée ....

# Programme faites-le vous même

Cela devrait être un véritable défi pour vous.

Comme vous le savez, une ligne droite est la distance la plus courte entre deux points. Eh bien, mettez quelques milles de plus entre nos deux points. Utilisez LIGNE pour tracer une ligne en zigzague.

10 SCREEN 7 20 CLS 30 Y=0 40 FOR X = 0 TO 200 STEP 10 50 OY = Y 60 Y = 30-OY 70 LINE (X,100-Y)-(X+10,100-OY),12 80 NEXT 90 GOTO 90

# L'écran d'argent



Êtes-vous prêt à découvrir une autre déclaration ?

Si c'est le cas, éteignez les lumières et beurrez le pop-corn, car nous sommes sur le point de lever le rideau sur le grand écran.

## Un mot sur la mémoire vidéo

Chaque fois que vous souhaitez afficher une image sur votre moniteur, l'ordinateur stocke l'image de l'écran dans la "mémoire vidéo". Le circuit du moniteur de l'ordinateur "lit" alors l'image de l'écran et l'affiche sur votre moniteur.

La mémoire vidéo "normale" est suffisamment grande pour le texte (lettres et chiffres) mais pas pour les graphiques (cercles, lignes, cases, etc.).

Par conséquent, l'ordinateur dispose de deux mémoires vidéo : une pour le texte et une pour les graphiques.

## Éclairage de l'écran d'argent

Jetez un coup d'œil au programme ci-dessous pendant une seconde. Concentrez-vous sur l'instruction SCREEN à la ligne 10 :

10 SCREEN 9 20 CLS 30 LINE (0,0)-(639,349),12 40 GOTO 40

SCREEN indique à l'ordinateur d'afficher une image d'écran sur votre téléviseur. Le type d'écran qu'il affiche dépend des instructions que vous lui donnez :

Tout d'abord, vous indiquez à l'ordinateur s'il doit utiliser l'écran du moniteur pour du texte (comme des lettres ou des chiffres) ou des graphiques (comme des lignes et des cercles).

Deuxièmement, vous indiquez à l'ordinateur quel "ensemble de couleurs" utiliser.

SCREEN [mode]

Mode = 1,2,3, 7,8 ou 9

Dans le programme changez la ligne 10 pour SCREEN 0

## 10 SCREEN 0

Vous obtenez l'erreur Illegal function call in 30, car SCREEN 0 est un mode texte uniquement.

Maintenant exécutez le programme ci-dessous:

10 SCREEN 1 20 LOCATE 1,5:PRINT "SCREEN 1, 320 × 200 pixels, 40 caractères" 30 GOSUB 1000 40 SCREEN 2 50 LOCATE 1,5:PRINT "SCREEN 2, 640 × 200 pixels 80 caractères" 60 GOSUB 1000 70 SCREEN 7 80 LOCATE 1,5:PRINT "SCREEN 7, 320 × 200 pixels, 40 caractères" 90 GOSUB 1000 **100 SCREEN 8** 110 LOCATE 1,5:PRINT "SCREEN 8, 640 × 200 pixels, 80 caractères" 120 GOSUB 1000 **130 SCREEN 9** 140 LOCATE 1,5:PRINT "SCREEN 9, 640 × 350 pixels, 80 caractères" 150 GOSUB 1000 160 LOCATE 14.1 170 SCREEN 0 180 END 1000 FOR I=0 TO 100 1010 PSET (I,I) 1020 NEXT 1030 LOCATE 14,1:PRINT"Appuyez sur une touche pour continuer" 1040 A\$=INKEY\$:IF A\$="" THEN 1040 **1050 RETURN** 

Exécutez le programme ci-dessous:

5 SCREEN 9:CLS 10 COLOR 2:Print "Vert" 20 LINE (0,0)-(639,349) 30 A\$=INKEY\$:IF A\$="" THEN 30 40 COLOR 3:Print "Cyan" 50 LINE (0,0)-(639,349) 60 GOTO 50 Revoyons les caractéristiques des différents modes de SCREEN:

# SCREEN 0

Mode texte uniquement Format texte  $40 \times 25$  ou  $80 \times 25$  avec une taille de boîte de caractères de  $8 \times 8$  ( $8 \times 14$  avec EGA) Affectation de 16 couleurs à l'un des 2 attributs Affectation de 16 couleurs à l'un des 16 attributs (avec EGA)

# SCREEN 1

Graphiques à résolution moyenne de  $320 \times 200$  pixels Format texte  $40 \times 25$  avec une taille de boîte de caractères de  $8 \times 8$ Affectation de 16 couleurs à l'un des 4 attributs Prend en charge à la fois EGA et CGA 2 bits par pixel

# SCREEN 2

Graphiques haute résolution  $640 \times 200$  pixels Format texte  $80 \times 25$  avec une taille de boîte de caractères de  $8 \times 8$ Affectation de 16 couleurs à l'un des 2 attributs Prend en charge à la fois EGA et CGA 1 bit par pixel

# SCREEN 7

Graphiques à résolution moyenne de 320 × 200 pixels Format texte 40 × 25 avec une taille de boîte de caractères de 8 × 8 2, 4 ou 8 pages de mémoire avec respectivement 64K, 128K ou 256K de mémoire installées sur l'EGA Affectation de l'une des 16 couleurs à 16 attributs EGA requis 4 bits par pixel

# SCREEN 8

Graphiques haute résolution 640 × 200 pixels Format texte 80 × 25 avec une taille de boîte de caractères de 8 × 8 1, 2 ou 4 pages de mémoire avec respectivement 64K, 128K ou 256K de mémoire installées sur l'EGA Affectation de l'une des 16 couleurs à 16 attributs EGA requis 4 bits par pixel

## SCREEN 9

Graphiques à résolution améliorée de 640 × 350 pixels Format texte 80 × 25 avec une taille de boîte de caractères de 8 × 14 Affectation soit de 64 couleurs à 16 attributs (plus de 64K de mémoire EGA), soit de 16 couleurs à 4 attributs (64K de mémoire EGA) Deux pages d'affichage si 256 Ko de mémoire EGA sont installés EGA requis 2 bits par pixel (mémoire EGA 64K) 4 bits par pixel (mémoire EGA supérieure à 64K)

Mode	Résolution	Plage d'attributs	Gamme de couleurs	Mémoire EGA	Pages	Taille de la page
0	40-colonne texte	NA	0-15 <sup>a</sup>	NA	1	2К
	80-colonne texte	NA	0-15 <sup>a</sup>	NA	1	4K
1	320×200	0-3 <sup>b</sup>	0-3	NA	1	16K
2	640×200	0-1 <sup>b</sup>	0-1	NA	1	16K
				64K	2	
7	320×200	0-15	0-15	128K	4	32K
				256K	8	
				64K	1	
8	640×200	0-15	0-15	128K	2	64K
				256K	4	
		0-3	0-15	64K	1	64K
9	640×350	0-15	0-63	128K	1	128K
		0-15	0-63	256K	2	
10	640×350	0-3	0-8	128K	1	128K
				256K	2	

a Les nombres compris entre 16 et 31 sont des versions clignotantes des couleurs 0-15. b Attributs applicables uniquement avec EGA.

# La commande COLOR

COLOR [foreground][,[background]

foreground = premier plan background = arrière-plan

## **Attributs et couleurs**

Pour divers modes d'écran et configurations matérielles d'affichage, différents paramètres d'attribut et de couleur existent. (Voir l'instruction PALETTE pour une discussion sur l'attribut et le numéro de couleur.)

La majorité de ces configurations d'attributs et de couleurs sont résumées dans le tableau suivant :

Attributs et couleurs par défaut pour la plupart des modes d'écran

#### Tableau 3

Attributs pour le Mode		Affichage couleur		Affichage monochrome		
1,9	2	0,7,8,9 <sup>b</sup>	Numéro <sup>c</sup>	Couleur	Numéro <sup>c</sup>	Couleur
0	0	0	0	Noir	0	Off
		1	1	Bleu		(Souligné) <sup>a</sup>
		2	2	Vert	1	On <sup>a</sup>
		3	3	Cyan	1	On <sup>a</sup>
		4	4	Rouge	1	On <sup>a</sup>
		5	5	Magenta	1	On <sup>a</sup>
		6	6	Marron	1	On <sup>a</sup>
		7	7	Blanc	1	On <sup>a</sup>
		8	8	Gris	0	Off
		9	9	Bleu clair		Haute intensité(Souligné)
		10	10	Vert clair	2	Haute intensité
1		11	11	Cyan clair	2	Haute intensité
		12	12	Rouge clair	2	Haute intensité
2		13	13	Magenta clair	2	Haute intensité
		14	14	Jaune	2	Haute intensité
3	1	15	15	Blanc haute intensité	0	Off

a Désactivé lorsqu'il est utilisé pour l'arrière-plan.

b Avec mémoire EGA > 64K.

c Uniquement pour le mode 0 monochrome.

Les couleurs de premier plan par défaut pour les différents modes sont données dans le tableau suivant :

Tableau 4

Couleurs de premier plan par défaut

# Attribut de premier plan par défaut Couleur de premier plan par défaut

Screen mode	Affichage couleur/étenduª	Affichage monochrome	Affichage couleur/étenduª	Affichage monochrome
0	7	7	7	1
1	3	NA	15	NA
2	1	NA	15	NA
7	15	NA	15	NA
8	15	NA	15	NA
9	3 <sup>b</sup>	NA	63	NA
10	NA	3	NA	8

a écran couleur amélioré IBM b 15 si supérieur à 64K de mémoire EGA NA=Non Applicable



Voir SCREEN, COLOR, PALETTE pour plus de détails.

## Programme faites-le vous même

Comprenez-vous SCREEN ?

Si c'est le cas, écrivez un programme qui passe de l'écran texte à l'écran graphique. Vous voudrez peut-être mettre une boucle dans le programme afin qu'il change le jeu de couleurs après avoir parcouru le programme en boucle. De cette façon, vous pouvez voir toutes les fonctionnalités de SCREEN au travail.

10 SCREEN 0 20 PRINT "Écran texte" 30 A\$=INKEY\$:IF A\$="" THEN 30 40 FOR Y = 1 TO 2 **50 SCREEN Y** 60 LOCATE 19,1:PRINT "Écran" Y 70 LOCATE 20,1:PRINT "Voici un carré ?" 80 LINE (50,50)-(120,100), Y,B 90 A\$=INKEY\$:IF A\$="" THEN 90 100 NEXT Y 110 FOR Y = 7 TO 9120 SCREEN Y:COLOR Y+1 130 LOCATE 19,1:PRINT "Écran" Y 140 LOCATE 20,1:PRINT "Voici un carré ?" 150 LINE (50,50)-(120,100), Y,B 160 A\$=INKEY\$:IF A\$="" THEN 160 170 NEXT Y

**Trouver les bonnes pages** 



En écrivant <u>Initiation au langage Basic</u> nous avons « stocké » des sections sur des pages. Certaines section nécessitent plus de pages ; certaines moins.

Dans le même sens, GW-BASIC stocke les écrans graphiques sur des blocs de mémoire graphique appelés "pages". Certains écrans nécessitent plus de pages ; certains moins.

Le tableau ci-dessous indique le nombre de pages pour chaque mode.

Mode	Résolution	Plage d'attributs	Gamme de couleurs	Mémoire EGA	Pages	Taille de la page
0	40-colonne texte	NA	0-15 <sup>a</sup>	NA	1	2K
0	80-colonne texte	NA	0-15ª	NA	1	4K
1	320×200	0-3 <sup>b</sup>	0-3	NA	1	16K
2	640×200	0-1 <sup>b</sup>	0-1	NA	1	16K
				64K	2	
7	320×200	0-15	0-15	128K	4	32K
				256K	8	
				64K	1	
8	640×200	0-15	0-15	128K	2	64K
				256K	4	
		0-3	0-15	64K	1	64K
9	640×350	0-15	0-63	128K	1	1291/
		0-15	0-63	256K	2	
10	C10, 270	0-3	0-8	128K	1	128K
	040×330			256K	2	

## Nouveaux Paramètres de SCREEN

SCREEN [mode] [,[colorswitch]][,[apage]][,[vpage]]

## colorswitch

Pour les moniteurs composites et les téléviseurs, le commutateur de couleurs est une expression numérique qui est vraie (non nulle) ou fausse (zéro). Une valeur de zéro désactive la couleur et permet l'affichage d'images en noir et blanc uniquement. Une valeur différente de zéro autorise la couleur. La signification de l'argument **colorswitch** est inversée en mode SCREEN 0.

Pour les configurations matérielles qui incluent un EGA et suffisamment de mémoire pour prendre en charge plusieurs pages d'écran, deux arguments sont disponibles. Ces arguments **apage** et **vpage** déterminent les pages mémoire "actives" et "visuelles". La page active est la zone en mémoire où les instructions graphiques sont écrites ; la page visuelle est la zone de mémoire qui s'affiche à l'écran.

L'animation peut être réalisée en alternant l'affichage des pages graphiques. L'objectif est d'afficher la page visuelle avec une sortie graphique terminée, tout en exécutant des instructions graphiques dans une ou plusieurs pages actives. Une page s'affiche uniquement lorsque la sortie graphique de cette page est terminée.

10 CLS:SCREEN 7,, 1, 2 'travailler à la page 1, afficher la page 2 20 CLS:LOCATE 19,1:PRINT "Page 2" 30 LOCATE 20,1:PRINT "Voici un carré ?" 40 LINE (50,50)-(120,100),2,B 50 A\$=INKEY\$:IF A\$="" THEN 50 60 SCREEN 7,, 2, 1 'travailler à la page 2, afficher la page 1 70 CLS:LOCATE 19,1:PRINT "Page 1" 80 LOCATE 20,1:PRINT "Voici un carré ?" 90 LINE (50,50)-(130,100),14,B 100 A\$=INKEY\$:IF A\$="" THEN 100 110 10 SCREEN 7,, 2

## Haut et bas, haut et bas

Vous pensez probablement que votre ordinateur est un peu fou, mais maintenant nous allons prouver que c'est un vrai yo-yo.

En fait, vous pouvez appeler ce programme "Yo-Yo". Entrez et exécutez-le.

10 SCREEN 9,, 2, 1 'travailler à la page 2, afficher la page 1 20 COLOR 10:CIRCLE (128,15),20 30 SCREEN 9,, 1,2 'travailler à la page 1, afficher la page 2 40 LINE (128,0)-(128,100),12 50 COLOR 10:CIRCLE (128,115),20 60 FOR X = 1 TO 100:NEXT X 70 GOTO 10

# PCOPY

Pour copier une page d'écran sur une autre dans tous les modes d'écran.

PCOPY sourcepage, destinationpage

Exemples:

Cela copie le contenu de la page 1 vers la page 2 :

**PCOPY 1,2** 

Aller en cercles



Est-ce que toutes ces discussions sur SCREEN et COLOR, vous font tourner en rond ?

Si oui, vous n'avez encore rien vu !

Par exemple, vous pouvez créer un cercle ou une ellipse complet(e), ou un cercle ou une ellipse partiel(le) à l'aide d'une seule instruction, CIRCLE.

Voici la syntaxe de CERCLE :

CIRCLE(xcenter, ycenter), radius[,[color][,[start],[end][,aspect]]]

xcenter et ycenter sont les coordonnées x et y du centre de l'ellipse, et radius est le rayon (mesuré le long de l'axe principal) de l'ellipse. Les quantités xcenter et ycenter peuvent être des expressions. Les attributs de centre peuvent utiliser des coordonnées absolues ou relatives.

color spécifie la couleur de l'ellipse. Sa valeur dépend du mode d'écran en cours. Voir les instructions <u>SCREEN</u>, <u>COLOR, PALETTE</u> pour plus d'informations sur l'utilisation des couleurs dans différents modes d'écran.

Les paramètres d'angle de début et de fin sont des arguments en radian entre  $-2\pi$  et  $2\pi$  qui spécifient où le dessin de l'ellipse doit commencer et se terminer. Si début ou fin est négatif, l'ellipse est reliée au point central par une ligne et les angles sont traités comme s'ils étaient positifs (notez que cela est différent de l'ajout de  $2\pi$ ).

aspect décrit le rapport du rayon x au rayon y (x:y). Le rapport d'aspect par défaut dépend du mode d'écran, mais donne un cercle visuel dans l'un ou l'autre des modes graphiques, en supposant un rapport d'aspect d'écran standard de 4:3. Si le rapport d'aspect est inférieur à 1, alors le rayon est donné en x-pixels.

S'il est supérieur à 1, le rayon est donné en pixels y. Dans de nombreux cas, un rapport d'aspect de 1 donne de meilleures ellipses en mode de résolution moyenne. Cela permet également de dessiner l'ellipse plus rapidement. L'angle de départ peut être inférieur à l'angle de fin.

Voir CIRCLE, LINE, WINDOW pour plus de détails.

Pour dessiner un cercle, vous n'avez besoin que du centre (x,y) et du rayon (radius), qui est la distance du centre en points.

Tout d'abord, comptez sur l'axe x, puis vers le bas sur l'axe y pour localiser le centre souhaité. Alors. une fois que vous avez spécifié ce point, indiquez le rayon du cercle. Le plus grand rayon qui tient sur l'écran dépend de la résolution, si le rayon est supérieur à l'écran, le cercle "natte" contre les bords de l'écran.

Tapez et exécutez ce programme:

5 CLS 10 SCREEN 1:PRINT "SCREEN 1" 20 GOSUB 1000 30 SCREEN 2:PRINT "SCREEN 2" 40 GOSUB 1000 50 SCREEN 7:PRINT "SCREEN 7" 60 GOSUB 1000 70 SCREEN 8:PRINT "SCREEN 8" 80 GOSUB 1000 90 SCREEN 9:PRINT "SCREEN 9" 100 GOSUB 1000 110 END 1000 CIRCLE(100,100), 50 1010 A\$=INKEY\$:IF A\$ = "" THEN 1010 1020 RETURN

## Programme faites-le vous même

Faites un programme pour générez une cible. Vous pouvez faire celui-ci de deux manières:

Ajoutez une ligne de programme distincte pour chaque cercle concentrique, mais utilisez un centre commun (coordonnées x,y).

Utilisez une boucle FOR ... NEXT avec un STEP 10 pour que l'ordinateur fasse le travail pour vous.

10 SCREEN 1 20 CLS 30 FOR RAYON = 1 TO 100 STEP 10 40 CIRCLE (128, 86),RAYON 50 NEXT RAYON 60 A\$=INKEY\$:IF A\$=""THEN 60

## Programme faites-le vous même

Avez-vous encore le programme de la maison que vous avez construite ?

Comment comptez-vous entrer dans la maison sans poignée de porte ? Utilisez CERCLE pour mettre une poignée de porte sur la porte d'entrée. Votre feuille de travail sur l'écran graphique est utile pour localiser le point exact dont vous avez besoin.

Remarque : Si vous utilisez une résolution moyenne ou basse, un cercle suffisamment petit pour servir de poignée de porte n'a pas beaucoup de détails. Exécutez le programme en SCREEN 9 pour plus de détails.

## **Colorier le cercle**

Après avoir décidé du rayon du cercle, choisissez sa couleur.

10 SCREEN 8 20 CLS 30 CIRCLE (128,86) ,50

Maintenant, pour un peu de variété, changez la couleur en cyan (couleur 3) :

30 CIRCLE (128,86) ,50,3

C'est aussi simple que ça! Vous pouvez changer la couleur du cercle en n'importe laquelle des couleurs disponibles.

## Mettre la pression

Avez-vous déjà pris un Hula-Hoop, un pneu de vélo ou une roue de buggy et l'avez-vous pressé à deux mains pour former une ellipse ?

De même, vous pouvez changer le cercle sur votre écran en une ellipse en utilisant l'option rapport hauteur/largeur (aspect).



La largeur de l'ellipse est égale au rayon. La hauteur est déterminée par aspect. Si aspect vaut 1, l'ordinateur dessine un cercle. Si aspect est supérieur à 1, il dessine une ellipse plus haute que large. Si aspect est inférieur à 1, il dessine une ellipse plus large que haute.

Par exemple, ce programme dessine un cercle :

10 SCREEN 8 20 CLS 30 CIRCLE (128,86) ,50 Si toutefois, vous modifiez aspect comme indiqué ici, le programme dessine une ellipse verticale :

30 CIRCLE (128,86),50,,,,8

Si vous modifiez aspect montré ici, il dessine une ellipse horizontale :

30 CIRCLE (128,86),50,,,,**.1** 30 CIRCLE (128,86),50,,,,1/8

Maintenant dessinons saturne:

10 SCREEN 8 20 CLS 30 CIRCLE (128,86) ,50 40 CIRCLE (128,86),50,,,,8

## Du début à la fin

Supposons que vous vouliez dessiner seulement une partie d'une ellipse (un arc). Pour ce faire, vous devez lister le centre de l'ellipse (x,y), son rayon (r), et son rapport hauteur/largeur (aspect).

Si vous le souhaitez, vous pouvez faire précéder aspect de la couleur (color).

Remarque : Pour tracer un arc , vous devez spécifier aspect. Pour un arc normal, utilisez aspect 1.

À partir des informations ci-dessus, l'ordinateur connaît l'emplacement, la largeur et la hauteur de l'ellipse. Maintenant, vous pouvez lui dire quelle portion de l'ellipse dessiner.

Pour ce faire, spécifiez le début de l'arc (start, 0 à 1) et la fin (end, 0 à 1) de l'arc, en suivant le tableau cidessous. Gardez à l'esprit que l'ordinateur dessine toujours dans le sens des aiguilles d'une montre.

CIRCLE(xcenter, ycenter), radius[,[color][,[start],[end][,aspect]]]

Les paramètres d'angle de début et de fin sont des arguments en radian entre  $-2\pi$  et  $2\pi$  qui spécifient où le dessin de l'ellipse doit commencer et se terminer. Si début ou fin est négatif, l'ellipse est reliée au point central par une ligne et les angles sont traités comme s'ils étaient positifs (notez que cela est différent de l'ajout de  $2\pi$ ).

Voici un cercle avec les angles en degrés et en radians:



0 degré = 0 radians et 90 degré =  $\pi/2$  ( $\pi/2$  = 1,57

pour convertir les degrés en radians

 $rad = deg * \pi/180$  soit pour 90°  $rad = 90 * \pi/180 = 1.57$  rad

30 CIRCLE (128,86),30,1,0,1.57,1



## Programme faites-le vous même

La nuit est-elle tombée sur la maison que vous avez bâtie ? Si c'est le cas, vous voudrez peut-être faire la lumière sur le sujet en mettant un croissant de lune dans le coin.

Cela nécessite deux arcs qui se croisent et quelques essais et erreurs de votre part.

10 CIRCLE (200,40),30,15,1.57,4.71,2 20 CIRCLE (200,40),30,15,1.57,4.71,1

Modifiez les deux lignes ci-dessous et insérez-les dans le programme maison

## Programme maison

10 SCREEN 7 20 CLS 30 LINE (72,168)-(200,72),10,B 'FRAME 40 LINE (72,72)-(136,36),10 'ROOF 45 LINE (200,72)-(136,36),10 'ROOF 50 LINE (120,168)-(152,100),10,B 'DOOR 55 LINE (152,60)-(168,36),10,BF 'CHIMNEY 60 LINE (165,128)-(191,100),10,B 'WINDOW 65 LINE (178,128)-(178,100),10 'WINDOW PART 70 LINE (165,114)-(191,114),10 'WINDOW PART 75 LINE (85,128)-(111,100),10,B 'WINDOW 80 LINE (85,114)-(111,114),10 'WINDOW PART 85 LINE (98,100)-(98,128),10 'WINDOW PART 90 GOTO 90

## Programme faites-le vous même

Peut-être qu'il fait froid, ainsi que sombre, autour de votre maison. Si c'est le cas, allumez un feu dans la cheminée et montrez la fumée sortant de la cheminée. (Utilisez CIRCLE pour générer une spirale qui simule la fumée.)

## Autre

Exemple 1:

```
1 ' Cela dessinera 17 ellipses
10 CLS
20 SCREEN 1
30 FOR R=160 TO 0 STEP-10
40 CIRCLE (160,100),R,,,,5/18
50 NEXT
```

Exemple 2:

```
10 'Cela dessinera 5 sphères

20 GOTO 160

50 IF VERT GOTO 100

60 CIRCLE (X,Y),R,C,,,.07

70 FOR I = 1 TO 5

80 CIRCLE (X,Y),R,C,,,I*.2:NEXT I

90 IF VERT THEN RETURN

100 CIRCLE (X,Y),R,C,,,1.3

110 CIRCLE (X,Y),R,C,,,1.9

120 CIRCLE (X,Y),R,C,,,3.6

130 CIRCLE (X,Y),R,C,,,9.8

140 IF VERT GOTO 60

150 RETURN

160 CLS: SCREEN 1: COLOR 0,1: KEY OFF: VERT=0

170 X=160: Y=100: C=1: R=50: GOSUB 50
```

```
180 X=30: Y=30: C=2: R=30: GOSUB 50
190 X=30: Y=169: GOSUB 50
200 X=289: Y=30: GOSUB 50
210 X=289: Y=169: GOSUB 50
220 LINE (30,30)-(289,169),1
230 LINE (30,169)-(289,30),1
240 LINE (30,169)-(289,30),1,B
250 Z$=INKEY$: IF Z$="" THEN 250
```

Le gros pinceau



Vous pensez peut-être que nous avons oublié qu'il s'agit d'un ordinateur couleur. Jusqu'à présent, ça a été un peu ici et une tache ou deux là. Vous ne créerez jamais un chef-d'œuvre de cette façon ! Eh bien, il est temps de se détendre un peu et de peindre la ville, en rouge, puis au moins un orange vif.

La fonction graphique PAINT vous permet de "peindre" n'importe quelle forme avec n'importe quelle couleur disponible.

Voici la syntaxe de PAINT :

PAINT (x,y)[,couleur[,couleur bordure][,couleur arrière plan]]

(x,y) est le point de départ

Si l'ordinateur atteint une bordure autre que celle de la couleur spécifiée, il peint par-dessus cette bordure.

5 SCREEN 7 10 CLS 20 REM Exemple de PAINT 30 LINE (0,0)-(255,181),12 40 LINE (0,181)-(255,0),10 50 CIRCLE (100,86),80,12 60 PAINT (50,50),5,12 70 GOTO 70

Avant d'exécuter le programme, pouvez-vous prédire les résultats ?

Les lignes 30 et 40 forment les lignes d'intersection. La ligne 50 génère un cercle dont le centre est au point d'intersection des deux lignes. Cette partie devrait être facile, mais qu'en est-il de PAINT dans Line 60 ?

Si vous avez deviné que l'ordinateur passe en position d'écran (50,50) et peint en Magenta jusqu'à ce que la peinture atteigne une bordure Rouge clair, vous avez raison !

Supprimez la ligne 30, puis exécutez le programme. Maintenant que vous redéfinissez les bordures, l'ordinateur peint le cercle en entier.

## Programme faites-le vous même

Avez-vous toujours votre maison ? Elle a probablement l'air un peu simple, peut-être même minable. Pourquoi ne pas l'embellir avec de la peinture ?

Voir programme maison Aller en cercles

Ajoutez un garage à votre maison, puis utilisez PAINT pour monter et descendre la porte du garage. Puisque l'action de peinture monte toujours en premier. cela prend un raffinement de votre part. Ajoutez un délai avant et après l'ouverture.

(Avec CIRCLE, ajoutez le soleil.)

# Tracer la ligne quelque part



Vous savez déjà comment créer des lignes, des ellipses et des boîtes. Maintenant, aimeriez-vous apprendre un raccourci pour faire certaines de ces choses ?

Le raccourci est DRAW, qui vous permet de tracer une ligne (ou une série de lignes) en spécifiant la direction, l'angle et la couleur, le tout dans la même ligne de programme !

Voici la syntaxe de DRAW :

Commandes de mouvement :

Chacune des commandes de mouvement suivantes commence le mouvement à partir de la position graphique actuelle. Il s'agit généralement de la coordonnée du dernier point graphique tracé avec une autre commande GML, LINE ou PSET.

La position actuelle est par défaut au centre de l'écran (160,100) en résolution moyenne SCREEN 7 ; (320,175) en haute résolution SCREEN 9 lorsqu'un programme est exécuté.

Les commandes de mouvement se déplacent sur une distance de facteur d'échelle \*n, où la valeur par défaut pour n est 1 ; ainsi, ils se déplacent d'un point si n est omis et que le facteur d'échelle par défaut est utilisé.

## Mouvements de commande

Un haut Dn vers le bas Ln gauche R à droite En diagonale vers le haut et vers la droite Fn en diagonale vers le bas et vers la droite Gn en diagonale vers le bas et vers la gauche Hn en diagonale vers le haut et vers la gauche

Cette commande se déplace comme spécifié par l'argument suivant :

Mx,y Déplacement absolu ou relatif. Si x est précédé d'un + ou d'un -, x et y sont ajoutés à la position graphique courante et reliés à la position courante par une ligne. Sinon, une ligne est tracée au point x, y à partir de la position actuelle.

Les commandes de préfixe suivantes peuvent précéder n'importe laquelle des commandes de mouvement cidessus :

B Déplace, mais ne trace aucun point.

N Déplace, mais revient à la position d'origine lorsque vous avez terminé.

Plus tôt, vous avez appris à créer une boîte à l'aide de LINE. Pour ce faire, vous avez peut-être dû effectuer des calculs difficiles avec la feuille de travail de l'écran graphique pour localiser les points de départ et d'arrivée nécessaires.

Avec DRAW, vous devez localiser uniquement le point de départ, puis indiquer à l'ordinateur dans quelle direction dessiner et jusqu'où le faire.

Si vous omettez le point de départ, l'ordinateur démarre à la dernière position DRAW ou, si vous n'avez pas encore utilisé DRAW, au centre de l'écran.

DRAW "BM128,96;U25;R25;D25;L25"

Ce programme a probablement remplacé votre chien comme votre meilleur ami.

5 SCREEN 7 10 CLS 25 DRAW "BM128,96;U25;R25;D25;L25" 30 GOTO 30

Pour rendre le programme plus facile à lire, on a séparé les instructions de mouvement avec un point-virgule (;). Vous n'avez pas besoin de le faire. Vous devez cependant, séparez toujours les coordonnées (x,y) par une virgule.

Presto! Pouvez-vous deviner pourquoi le coin inférieur gauche du carré est à (128,96) ?

Regardez les deux premiers chiffres à l'intérieur des guillemets.

La commande de mouvement, M, indique à l'ordinateur quel point commencer à dessiner pour plus tard.
M x,y indique à l'ordinateur à quel point commencer à dessiner: x est la position horizontale (0 à 255). y est la position verticale (0 à 191).

Remarque : Toujours préfacer M par la lettre B ; si vous ne le faites pas, des lignes indésirables apparaissent.

B Déplace, mais ne trace aucun point.

Le programme ci-dessus indique à l'ordinateur de commencer à dessiner à (128,96), de dessiner (U) 25 points vers le haut, d(R) 25 points de plus vers la droite, de descendre (D) 25 points de plus et enfin (L) 25 points vers la gauche.

Remarque : Si vous omettez la longueur de la ligne, l'ordinateur utilise 1 comme longueur.

# Placer le carré sur le bord

(Lignes diagonales)

Au lieu de tracer des lignes horizontales et verticales, placez le carré sur l'un de ses coins. Pour ce faire, remplacez E, F, C et H par U, R, l et D à la ligne 25 :

25 DRAW "BM128,96;E25;F25;G25;H25"

Ce TIRAGE commence également à (128,96). Au lieu de monter, cependant, la première ligne s'incline à 45 degrés; l'ordinateur dessine les 3 lignes suivantes selon leurs angles désignés.

### Programme faites-le vous même

Vous savez déjà que votre ordinateur est la vedette du spectacle, mais pouvez-vous le prouver en dessinant une étoile ? Utilisez les commandes de mouvement DRAW pour les lignes perpendiculaires et diagonales.

```
5 SCREEN 7
10 CLS
30 DRAW "BM68,116;E20;BE20;E20;F20;BF20;F20;L40;BL40;L40;BU40;R40;BR40;
R40;G20;BG20;G20;H20;BH20;H20;BM128,96;NU40;ND40;NE20;NF20;NG20;NH20;N140;R40"
40 GOTO 40
```

On peut écrire la ligne 30 sur deux lignes:

5 SCREEN 7 10 CLS 20 DRAW "BM68,116;E20;BE20;E20;F20;BF20;F20;L40;BL40;L40;BU40;R40;BR40;R40;G20;BG20;G20;H20;BH20; H20" 30 DRAW "BM128,96;NU40;ND40;NE20;NF20;NG20;NH20;NI40;R40" 40 GOTO 40

# Absolu M y Relatif M

Supposons que vous dessiniez un carré et que vous vouliez en dessiner un autre à proximité.

Vous savez exactement à quelle distance vous voulez placer le deuxième carré, mais vous ne voulez pas avoir à localiser les coordonnées (x,y).

Une autre forme de la commande M consiste à spécifier un mouvement "relatif" au lieu d'un mouvement "absolu". Jusqu'à présent, vous avez utilisé le mouvement absolu ; vous avez spécifié des points en fonction de leurs coordonnées (x,y).

En utilisant le mouvement relatif, vous pouvez spécifier des points par rapport au point courant le dernier point dessiné!.

Voici la syntaxe pour mouvement relatif :

M sign x-offset, y-offset vous permet de spécifier des points relatifs à l'actuel point. point h-offset est la distance pour se déplacer horizontalement à partir de la position actuelle.

Si vous le faites précéder d'un signe plus (+), la position h est incrémentée de la valeur spécifiée.

Si vous la faites précéder d'un signe moins (-), la position h est décrémentée.

y-offset est la distance à parcourir verticalement à partir du courant. Si vous faites précéder y-offset d'un signe plus (+) ou si vous omettez le signe, la position s'incrémente de la quantité spécifiée. Si vous la précédez avec un signe moins (-), la position y décrémente.

Par exemple. si vous souhaitez créer la deuxième case à une position relative à celle ou à la première case dans le programme "Carré" (redéfini), vous pouvez ajouter cette ligne :

30 DRAW "BM+15,15;U25;R25;D25;L25"

5 SCREEN 7 10 CLS 25 DRAW "BM128,96;U25;R25;D25;L25" 30 DRAW "BM+15,15;U25;R25;D25;L25" 40 GOTO 40

Lorsque l'ordinateur exécute la ligne 30, la position de tirage actuelle est (128,96), qui est la dernière position de DRAW de la ligne 25. Ainsi, le coin inférieur gauche de la nouvelle case est à (128+15,96+15) ou (143,111).

Modifiez la ligne 30 comme suit :

30 DRAW "BM+15,-15;U25;R25;D25;L25"

Exécutez le programme. Le point de départ du nouveau carré est (128 + 15,96-15) ou (143,81).

### Programme faites-le vous même

Après toute cette activité intense, vous êtes probablement prêt à vous rafraîchir. alors pourquoi ne pas utiliser DRAW pour créer un glaçon (un cube de glace) ?

Vous pouvez généraliser le cube entier en utilisant DRAW. Ou vous pouvez incorporer quelques commandes LINE dans le programme. Essayez d'utiliser à la fois le mouvement absolu et relatif.

10 SCREEN 7 20 CLS 25 DRAW "BM40,80;U40R40D40L40" 30 DRAW "BM+20,20;U40;R40;D40;L40" 40 LINE (60,100)-(40,80),15 50 LINE (60,60)-(40,40),15 60 LINE (100,60)-(80,40),15 70 LINE (100,100)-(80,80),15 80 GOTO 80

# Faire pencher la balance

Que se passe-t-il si les figures que vous dessinez sont trop grandes ou trop petites ?

La solution est simple. Votre ordinateur possède une fonction intégrée qui vous permet de « redimensionner » (vers le haut ou vers le bas) tout affichage généré par DRAW. Tout ce que vous avez à faire est d'utiliser la commande Sx dans la chaîne.

Lorsque vous utilisez l'option de réduction, l'ordinateur arrondit la longueur de ligne résultante au nombre entier le plus proche. si ce n'est pas déjà un nombre entier.

Sx définit le facteur d'échelle. x peut être compris entre 1 et 255. x est divisé par 4 pour obtenir le facteur d'échelle. Le facteur d'échelle est multiplié par les distances données avec les commandes U, D, L, R, E, F, G, H ou M relatives pour obtenir la distance réelle parcourue. La valeur par défaut pour S est 4.

1 = échelle 1/4 2 = échelle 2/4 3 = échelle 3/4 4 = 4/4 pleine échelle 5 = 5/4 (échelle 125 %) 8 = échelle 8/4 (double) 12 = 12/4 (triple) échelleelc.

Si vous omettez x, l'ordinateur utilise 4 (4/4 = I).

Après avoir entré une commande Sx, l'ordinateur met à l'échelle toutes les commandes de mouvement absolu et relatif en conséquence jusqu'à ce que vous en saisissiez une autre.

Faites en sorte que vos "Lignes" raffinées dessinent à nouveau un seul carré. Pour ce faire, supprimez la ligne 30 et modifiez la ligne 25 comme suit :

5 SCREEN 7 10 CLS 25 DRAW "S2;BM128,96;U25;R25;D25;L25" 40 GOTO 40

Exécutez le programme.

Le carré dans le coin inférieur gauche doit être la moitié de la taille que vous avez spécifiée.

Pour voir à quel point un carré peut être petit ou grand, exécutez le programme suivant :

5 SCREEN 9 10 CLS 20 FOR **SCALE** = 1 TO 62 30 S\$ = "S" + STR\$(**SCALE**) + ";" 35 DRAW S\$ + "BM10,340U20R20D20L20" 40 NEXT SCALE 50 GOTO 50

Ne commettez pas l'erreur de penser que le plus petit carré est celui spécifié à la ligne 35. Celui que nous avons spécifié est le quatrième à partir du bord Pour SCALE=4.

# Colorie-moi

L'option C de DRAW vous permet de spécifier la couleur d'une ligne particulière.

Cn Définir la couleur n. Voir les instructions <u>SCREEN, COLOR, PALETTE</u> pour des discussions sur les couleurs, les nombres et les attributs valides.

Tout d'abord, tapez le programme "Lignes":

5 SCREEN 7 10 CLS 30 DRAW "BM128,96;U25;R25;D25;L25" 40 GOTO 40

Revenez à la pleine échelle soit en remplaçant 52 par 54, soit en supprimant 52. Ensuite, juste à l'intérieur du premier jeu de guillemets de la ligne 30, insérez :

C6

30 DRAW "C6;BM128,96;U25;R25;D25;L25"

Exécutez le programme. Affiche-t-il un carré Marron sur un fond noir ?

Remplacez le C6 (dans la ligne 30 du programme) par C9 et exécutez le programme. Le carré est-il devenu Bleu clair?

C doit prendre la forme suivante :

Cx permet de spécifier la couleur d'une ligne x est le code couleur (0 à 15). Si vous ometez x, l'ordinateur utilise la couleur de premier plan.

Vous pouvez insérer Cx n'importe où dans l'instruction DRAW. Toutes les actions qui suivent sont de la couleur que vous spécifiez. Par exemple, modifiez la ligne 30:

30 DRAW "C6;BM128,96;U25;C9;R25;C4;D25;L25"

Exécutez le programme. Le programme affiche un carré de 3 couleurs.

Si vous souhaitez effacer une ligne, tracez une autre ligne par-dessus en utilisant la couleur d'arrière-plan.

# Quel est votre angle ?

Une autre option disponible avec DRAW est A. Cette option vous permet de spécifier l'angle auquel une ligne doit être tracée. Après avoir inclus A dans la commande DRAW, l'ordinateur dessine toutes les lignes suivantes avec le déplacement angulaire spécifié par Ax jusqu'à ce que vous spécifiez le contraire.

Votre programme devrait être :

5 SCREEN 7 10 CLS 30 DRAW "C6;BM128,96;U25;C9;R25;C4;D25;L25" 40 GOTO 40

Voici la syntaxe de la commande A :

Ax vous permet de spécifier l'angle d'une ligne

x est le code d'angle (0 à 3). Tous les angles sont mesurés dans le sens des aiguilles d'une montre.

0 = 0 degrés 1 = 90 degrés 2 = 180 degrés 3 = 270 degrés

Si vous omettez Ax, l'ordinateur utilise A0.

Pour illustrer cela, modifiez la ligne 30 du programme:

30 DRAW "A0;C6;BM128,96;U25"

Exécutez le programme. Votre écran affiche une ligne verticale de 25 points de long.

Changez maintenant la ligne 30 :

30 DRAW "A1;C6;BM128,96;U25"

Exécutez le programme. La ligne est maintenant horizontale.

Si vous voulez que la prochaine ligne que vous dessinez soit une ligne "blanche" ou invisible, incluez l'option B.

Par exemple, disons que vous dessinez des lettres de l'alphabet et que vous êtes prêt pour la lettre C qui n'est rien d'autre qu'un carré avec le côté droit vide.

Modifiez la ligne 30 comme suit pour que le programme génère un tel chiffre :

30 DRAW "A1;C6;BM128,96;U25; B; C9;R25;C4;D25;L25"

Exécutez le programme. N'oubliez pas que seule la ligne suivant immédiatement le B est vide.

Exécutez le programme. N'oubliez pas que seule la ligne suivant immédiatement le B est vide.

# Programme faites-le vous même

Imprimez votre nom sur l'écran en utilisant DRAW. Cela signifie que vous devrez rester dans l'écran graphique. Bien sûr, il serait plus facile d'écrire votre nom sur l'écran de texte, mais vous ne pouvez pas avoir du "vrai" texte et des graphiques en même temps.

10 SCREEN 7 20 CLS 25 DRAW "BM50,50;L30D30R30D30L30" 'S 30 DRAW "BM90,50D60R30U60" 'U 40 DRAW "BM160,50D60R30BU60L30D30R30" 'E 50 GOTO 50

# **Quoi!** Plus d'options?

Une autre des nombreuses fonctionnalités de DRAW est N, l'option "pas de mise à jour". N indique à l'ordinateur de revenir à sa position d'origine (actuelle) après avoir tracé la ligne suivante. Pour voir cela, modifiez la ligne 30:

30 DRAW "C6;BM128,96;N;U25;N;C9;N;R25;C4;N;D25;N;L25"

Exécutez le programme. L'ordinateur trace une ligne de 25 points vers le haut à partir de (128,96}. Il revient ensuite à (128,96), trace la ligne suivante, revient, trace la suivante, et ainsi de suite.

En conséquence, quatre lignes partent du centre de l'écran, chacune dans une direction différente (haut, droite,bas et gauche).

### Programme faites-le vous même

En utilisant l'option N de DRAW (et CIRCLE), demandez au compilateur de dessiner un gâteau de 8 pièces. Une fois que vous avez fait cela, découpez la tranche de tarte et mettez-la de côté.

10 SCREEN 7 20 CLS 30 DRAW "BM98,96;NU80;NE56;NR80;NF56;ND80;NG56;NL80;NH56" 40 CIRCLE (98,96),80,1,,,1 50 CIRCLE (135,110),80,1,5.45,0 'Les angles sont en radian 60 LINE (135,110)-(190,160),15 70 LINE (135,110)-(213,110),15 80 GOTO 80

# Constantes de chaîne vs Variables de chaîne

Comme indiqué précédemment, la chaîne qui suit DRAW peut être soit une constante, soit dans les exemples précédents, soit une variable.

Pour utiliser une variable de chaîne, faites précéder l'instruction DRAW d'une ligne de programme qui identifie la variable en tant que chaîne ; puis remplacez la chaîne dans DRAW.

Par exemple, ajoutez la ligne 25 et modifiez la ligne 30 comme suit :

5 SCREEN 7 10 CLS 25 A\$ = "C6;BM128,96;U25;C9;R25;C4;D25;L25" 30 DRAW A\$ 40 GOTO 40

Exécutez le programme. L'ordinateur affiche une case de (25 x 25) dont le coin inférieur gauche est au centre de l'écran.

GW-BASIC offre une variante à cela, appelée l'action "exécuter" (X).

Pendant que vous exécutez une routine DRAW, l'action d'exécution vous permet d'exécuter une autre chaîne DRAW, puis de revenir et de terminer la première opération. Pour ce faire, laissez la ligne 25 telle quelle afin qu'elle définisse A\$ ; puis changez la ligne 30. Les deux lignes sont:

5 SCREEN 7 10 CLS 25 A\$ = "C6;BM128,96;U25;C9;R25;C4;D25;L25" 30 DRAW "BM95,50;U25;R25;XA\$;D25;L25" 40 GOTO 40

Exécutez le programme. L'ordinateur commence à tracer à (95,50) une ligne qui s'étend vers le haut (U25) puis vers la droite (R25). Il exécute ensuite A\$ de manière à dessiner un carré de 25 x 25, en commençant par al (128,96). Après avoir exécuté A\$, il revient à la chaîne d'origine (actuelle) et termine son exécution (D25,125).

### Programme faites-le vous même

Faites-le vous-même ci-haut montre que vous pouvez simuler du texte (lettres) sur l'écran graphique en dessinant les lettres. Utilisez DRAW pour créer les 26 lettres de l'alphabet. Stockez les commandes DRAW dans des chaînes. Utilisez ensuite l'action "exécuter" (X) pour organiser les lettres en mots.

Voici un exemple <u>ALPHA1.BAS</u>

Avez-vous toujours votre maison (Voir <u>Le gros pinceau</u>) ? Si c'est le cas, chargez à nouveau le programme et utilisez DRAW pour ouvrir et fermer la porte avant.

Voici un exemple MAISON3.BAS

Voir aussi <u>RID.BAS</u>

# DRTRS-80

Get et Put : l'affichage est allé dans ce tableau

Dans les sections précédentes, vous avez appris quelques façons de déplacer des personnages d'un écran à un autre, mais aucune n'est très efficace. N'ayez pas peur; il y a un meilleur tableau. Cela a à voir avec GET et PUT.

En utilisant ces instructions, vous pouvez "obtenir" une zone rectangulaire de l'écran, stocker son contenu dans un "tableau" (une zone de mémoire), puis le "remettre" où vous voulez sur l'écran. C'est la meilleure méthode pour simuler le mouvement.

Un qui ? Un quoi? Les tableaux sont couverts dans la section Passer aux affaires, plus loin.

Nous utilisons le terme "rectangle" pour désigner la zone qui contient l'affichage graphique\_ ou le parcours, vous ne pouvez pas réellement voir le rectangle. Il va falloir le visualiser.

Les formats pour GET et PUT sont :

GET (x1,y1)-(x2,y2),tableau

x1,y1 est le coin supérieur gauche du rectangle.

x2,y2 est le coin inférieur droit du rectangle.

tableau est une zone en mémoire qui stocke le rectangle.

# PUT(x,y),tableau,[,action]

x,y est le coin supérieur gauche du rectangle.

# action peut être PSET, PRESET, AND, OR, XOR.

Le verbe d'action est utilisé pour interagir avec l'image transférée avec l'image déjà à l'écran. PSET transfère les données sur l'écran textuellement.

PRESET est identique à PSET sauf qu'une image inversée (noir sur blanc) est produite.

AND transfère l'image uniquement si une image existe déjà sous l'image transférée.

OR superpose que l'image existe.

XOR est un mode spécial souvent utilisé pour l'animation. XOR provoque l'inversion des points sur l'écran là où un point existe dans l'image du tableau. Ce comportement est exactement comme le curseur à l'écran. XOR est particulièrement utile pour l'animation. Lorsqu'une image est placée deux fois sur un arrière-plan complexe, l'arrière-plan est restauré tel quel. Un objet peut être déplacé sur l'écran sans effacer l'arrière-plan. Le mode d'action par défaut est XOR.

Pour plus d'informations sur les effets dans les différents modes, consultez les instructions <u>SCREEN, COLOR,</u> <u>PALETTE</u>.

L'animation d'un objet est généralement effectuée comme suit :

Placez le ou les objets sur l'écran.

Recalculez la nouvelle position du ou des objets.

Placez le(s) objet(s) sur l'écran une deuxième fois à l'ancien(s) emplacement(s) pour supprimer l'ancienne image(s).

Retournez à l'étape 1, cette fois en plaçant le ou les objets au nouvel emplacement.

Le mouvement effectué de cette manière laisse l'arrière-plan inchangé. Le scintillement peut être réduit en minimisant le temps entre les étapes 4 et 1, et en s'assurant qu'il y a suffisamment de temps entre les étapes 1 et 3. Si plus d'un objet est animé, traitez chaque objet à la fois, une étape à la fois.

S'il n'est pas important de préserver l'arrière-plan, l'animation peut être réalisée à l'aide du verbe d'action PSET.

Laissez une bordure autour de l'image (lorsqu'elle est obtenue pour la première fois) aussi grande ou plus grande que la distance maximale sur laquelle l'objet se déplacera.

Ainsi, lorsqu'un objet est déplacé, cette bordure efface effectivement tous les points. Cette méthode peut être un peu plus rapide que la méthode utilisant XOR décrite ci-dessus puisqu'un seul PUT est nécessaire pour déplacer un objet. Cependant, l'image à PUT doit être plus grande que l'image existante.

Tapez et exécutez ce programme pour voir comment fonctionnent GET et PUT :

10 SCREEN 7 15 CLS 25 DIM V(20,20) 30 CLS:CIRCLE (20,20),10 32 FOR DLAY = 1 TO 600: NEXT DLAY 35 GET (10,10)-(30,30),V 40 CLS 42 FOR DLAY = 1 TO 600: NEXT DLAY 45 PUT (110,110),V 50 FOR DLAY = 1 TO 600: NEXT DLAY 60 GOTO 30

Le programme dessine un cercle sur une partie de l'écran puis le déplace vers une autre. Pour ce faire, l'ordinateur :

1. Crée un tableau nommé V en mémoire (Ligne 25). Le tableau V est assez grand pour stocker un rectangle de 20 X 20.

- 2. Dessine un cercle sur l'écran (ligne 30).
- 3. Obtient un rectangle 20 X 20 contenant le cercle et le stocke dans le tableau V (ligne 35).
- 4. Efface l'écran (Ligne 40),
- 5. Affiche le rectangle 20 X 20 (stocké dans le tableau V) à l'écran.

# Stockage du rectangle

Comme vous pouvez le voir dans le programme ci-dessus, GET et PUT utilisent un tableau pour stocker le rectangle. Donc, avant d'utiliser GET ou PUT, vous devez créer ce tableau.

L'instruction DIM vous permet de faire.

Pour spécifier les valeurs maximales des indices de variable de tableau et allouer le stockage en conséquence.

Syntaxe:

DIM variable(subscripts)[,variable(subscripts)]...

EX. DIM A(12) une dimension, B(5,6) deux diamantions, C(3,8,5) trois démentions ... etc.

Commentaires:

Si un nom de variable de tableau est utilisé sans instruction DIM, la valeur maximale de son ou ses indices est supposée être 10. Si un indice supérieur au maximum spécifié est utilisé, une erreur "Subscript out of range " se produit.

Le nombre maximal de dimensions pour un tableau est de 255.

La valeur minimale d'un indice est toujours 0, sauf indication contraire avec l'instruction OPTION BASE.

Un tableau, une fois dimensionné, ne peut pas être redimensionné dans le programme sans d'abord exécuter une instruction CLEAR ou ERASE.

L'instruction DIM définit tous les éléments des tableaux spécifiés sur une valeur initiale de zéro et doit être mis avant l'utilisation du tableau.

# **OPTION BASE**

Pour déclarer la valeur minimale des indices de tableau.

Syntaxe:

OPTION BASE n

Commentaires:

n est 1 ou 0. La base par défaut est 0.

Si l'instruction OPTION BASE 1 est exécutée, la valeur la plus basse qu'un indice de tableau peut avoir est 1.

Un indice de tableau ne peut jamais avoir une valeur négative.

OPTION BASE génère une erreur uniquement si vous modifiez la valeur de base. Cela permet aux programmes chainés d'avoir des instructions OPTION BASE tant que la valeur n'est pas modifiée par rapport au paramètre initial.

# Note

Vous devez coder l'instruction OPTION BASE avant de pouvoir définir ou utiliser des tableaux. Si une tentative est faite pour modifier la valeur de base de l'option après l'utilisation d'un tableau, une erreur se produit.

Les erreurs

n n'est pas un chiffre 0 ou 1 : Syntax error.

L'OPTION BASE 1 est appelée mais un tableau a déjà été alloué auparavant : Duplicate definition.

OPTION BASE est appelée plus d'une fois avec un index de départ différent : Duplicate definition.

Quelle doit être la taille du tableau ?

Cela dépend de la taille d'un rectangle que vous voulez "obtenir" ou "mettre":

Largeur = y1 - x1longueur = y2 - x2

Par exemple, l'instruction GET du programme ci-dessus utilise (10,10) et (30,30) pour spécifier un rectangle. Ainsi, le rectangle est de 20 X 20 : Il a une largeur et une longueur de 20. L'instruction PUT utilise le même rectangle de taille : 20 X 20.

La taille d'un rectangle que vous pouvez stocker dans un tableau dépend de la quantité de mémoire dont vous disposez.

Chaque point, lorsqu'il est stocké dans un tableau, consomme 5 octets de mémoire. Dans un système de 16K RAM, vous ne pouvez pas stocker plus de 1400 points dans un tableau. Si votre programme est long, vous devrez peut-être le réduire.

### Ne mettez pas ce que vous voyez

Vous avez maintenant mis un rectangle sur l'écran dans un sens avec l'action PSET. (Lorsque vous ne spécifiez pas d'autre action, l'ordinateur utilise PSET).

Il y a plus d'une façon, cependant, de mettre des rectangles sur l'écran.

Pour voir comment fonctionnent les autres actions. Commencez par exécuter ce programme. Il met 15 rectangles sur l'écran avec l'action P5ET.

10 DIM V (30,30) 20 SCREEN 7 25 CLS 30 CIRCLE (128,86),30,4 35 PAINT (128,85),2,4 40 PAINT (128,97),3,4 45 GET(98,81)-(128,100),V 50 CLS 55 FOR I = 150 TO 1 STEP -10 60 PUT (I,81-I/5),V,PSET 65 NEXT I 70 GOTO 70

PSET définit et réinitialise chaque point tel qu'il se trouve dans le rectangle du tableau. Chaque rectangle qu'il place à l'écran est le même que celui stocké dans le tableau.

Modifiez maintenant la ligne 60 de différentes manières pour essayer d'autres actions. Essayez d'abord PRESET.

# 60 PUT (I,81-I/5),V,PRESET

PRESET définit et réinitialise l'inverse de chaque point dans le rectangle du tableau. Chaque rectangle qu'il place à l'écran est l'inverse de celui stocké dans le tableau.

Essayez l'action OR :

60 PUT (I,81-I/5),V,OR

OR définit chaque point qui est soit

(1) défini dans le rectangle arrière ou

(2) déjà défini dans la position où il tire le rectangle d'écran.

Chaque rectangle qu'il place sur l'écran a tous les points définis qui sont stockés dans le tableau plus ce qui est actuellement à l'écran.

Pour un effet étrange, essayez l'action XOR.

60 PUT (I,81-I/5),V,XOR

XOR est un mode spécial souvent utilisé pour l'animation. XOR provoque l'inversion des points sur l'écran là où un point existe dans l'image du tableau. Ce comportement est exactement comme le curseur à l'écran. XOR est particulièrement utile pour l'animation. Lorsqu'une image est placée deux fois sur un arrière-plan complexe, l'arrière-plan est restauré tel quel. Un objet peut être déplacé sur l'écran sans effacer l'arrière-plan. Le mode d'action par défaut est XOR.

Essayez l'option AND avec ce programme, et vous ne verrez rien :

60 PUT (I,81-I/5),V,AND

AND définit chaque point qui

(1) est défini dans le tableau et

(2) est déjà défini sur l'écran à la position où il place le rectangle.

Tous les points qui ne remplissent pas ces deux conditions sont réinitialisés. Dans ce cas, chaque rectangle AND mis à l'écran a tous les points réinitialisés - vous ne voyez rien.

### Programme faites-le vous même

Utilisez GET et PUT pour envoyer un vaisseau spatial sur l'écran et à travers ses "limites extérieures". Vous voudrez peut-être ajouter un quelques astéroïdes et extraterrestres pour rendre le voyage plus excitant !

```
15 SCREEN 7
20 CLS
25 DIM V(35,35)
30 X=10: Y=10
35 DRAW
"BM10,10;S2;H10;R15;F10;R20;F10;G10;L20;G10;L15;E10;U20;D4:NL8;D4;NL12;D4NL16;D4;NL12D4;N
L8:"
40 GET (X-X,Y-Y)-(X*3.5,Y*3.5),V
45 A$=INKEY$: IF A$="" THEN 45 'APPUYEZ SUR N'IMPORTE QUELLE TOUCHE POUR
COMMENCER
50 CLS
55 FOR A = 10 TO 200 STEP 5
60 PUT (X+A,Y),V,PSET
65 NEXT A
70 CLS
75 GOTO 55
```

### Jouez-le encore



Donc vous pensez que votre ordinateur est un bon artiste, hein ?

Eh bien, vous n'avez encore rien entendu !

Attendez de découvrir ses talents musicaux ! Prêt?

Alors allons-y pour travailler et JOUER.

La fonction PLAY de votre ordinateur vous permet non seulement d'écouter de la musique, mais aussi de la composer.

Écoutez attentivement

Voici la syntaxe pour PLAY :

PLAY music lit la valeur de la musique, " expression de chaîne comprenant les éléments suivants :

A-G [#,+,-] A-G sont des notes. # ou + après une note produit un dièse ; - produit un plat.

Toute note suivie de #, + ou - doit faire référence à une touche noire sur un piano.

### L(n)

Définit la longueur de chaque note. L4 est une noire, L1 est une ronde, et ainsi de suite. n peut être compris entre 1 et 64.

La longueur peut également suivre la note pour modifier la longueur de cette note uniquement. A16 est équivalent à L16A.

**MF** Musique de premier plan. Les instructions PLAY et SOUND (Voir <u>Les premiers pas</u>) doivent être exécutées au premier plan. C'est-à-dire que chaque note ou son suivant n'est pas démarré tant que la note ou le son précédent n'est pas terminé. Il s'agit de la valeur par défaut initiale.

**MB** Fond de musique. Les instructions PLAY et SOUND doivent s'exécuter en arrière-plan. C'est-à-dire que chaque note ou son est placé dans un tampon permettant au programme BASIC de continuer son exécution pendant que la musique joue en arrière-plan. Jusqu'à 32 notes (ou silences) peuvent être jouées en arrière-plan en même temps.

MN Musique normale. Chaque note joue sept huitièmes du temps déterminé par L (longueur).

ML Music legato. Chaque note joue la période complète définie par L.

MS Musique saccadée. Chaque note joue les trois quarts du temps déterminé par L.

**N(n)** Joue la note n. n peut varier de 0 à 84. Dans les 7 octaves possibles, il y a 84 notes. n mis à 0 indique un silence.

**O**(**n**) Octave 0 définit l'octave actuelle. Il y a 7 octaves (0 à 6). La valeur par défaut est 4. Le do médian est au début de l'octave 3.

**P**(**n**) Pause. P peut varier de 1 à 64.

T(n) Tempo. T définit le nombre de L4 en une minute. n peut varier de 32 à 255. La valeur par défaut est 120.

# . (point)

Un point après une note augmente le temps de jeu de la note de 3/2 fois la période déterminée par L (longueur de note) fois T (tempo). Plusieurs points peuvent apparaître après une note, et le temps de jeu est mis à l'échelle en conséquence.

Par exemple, A. fera jouer la note A une fois et demie le temps de jeu déterminé par L (longueur de la note) fois T (le tempo); deux points placés après A (A..) feront jouer la note à 9/4 fois sa valeur attribuée ; un A à trois points (A...) à 27/8, etc.

Des points peuvent également apparaître après un P (pause) et augmenter la durée de la pause comme décrit cidessus.

# Xstring ;

Exécute une sous-chaîne, où chaîne est une variable affectée à une chaîne de commandes PLAY.

En raison de la lenteur du taux d'interruption de l'horloge, certaines notes ne jouent pas à des tempos plus élevés ; par exemple, 1,64 à T255. Ces combinaisons note/tempo doivent être déterminées par l'expérimentation.

>n Un symbole supérieur à précédant la note n joue la note dans l'octave immédiatement supérieure. <n Un symbole inférieur à précédant la note n joue la note dans l'octave inférieure suivante.

Note

Les arguments numériques suivent la même syntaxe décrite sous l'instruction DRAW.

n en tant qu'argument peut être une constante, ou il peut s'agir d'une variable avec = devant (= variable).

Un point-virgule est requis après la variable et également après la variable dans Xstring.

#### **Comparons les notes** (NOTE)

De toute évidence, vous pouvez avoir de la musique sans notes. PLAY offre deux façons de spécifier la note précise dont vous avez besoin.

Le premier moyen, et probablement le plus simple, de jouer la note souhaitée est d'entrer l'une des notes de musique standard - A, B, C, D, E, For G. Pour indiquer une note aiguë, suivez la note avec un signe plus (+) ou avec le signe dièse (#).

Pour indiquer un bémol, faites-le suivre d'un signe moins (-).

Par exemple, A représente A naturel ; A # est A dièse; et A - est A bémol.

Tapez ce qui suit pour voir (entendre ?) ce que nous voulons dire :

PLAY "A"

Pour entendre le changement que peuvent faire un dièse et un bémol, entrez ces lignes :

PLAY "A;Aa" PLAY "A-;A;A#;A;A-"

Vous pouvez faire la même chose avec les sept notes (A-G) sur l'échelle, sauf C et B. Puisque B# = C, vous devez utiliser C. De même, puisque C - = B, vous devez utiliser B.

### Notes entières, demi temps, quart de temps. (LONGUEUR DE LA NOTE)

Pour choisir la longueur de note, utilisez L suivi d'un nombre de 1 à 64. Le nombre 1, par exemple, désigne une ronde, 2 une blanche, 4 une noire, 8 une croche, 16 une double croche.

En fait, vous pouvez utiliser n'importe quel nombre de 1 à 64. (Qui a déjà entendu parler d'une 1/1 5ème note ?)

Variez les longueurs de notes pour produire un roulement de tambour. Taper:

PLAY "L2;A;L4;A;A;L2;A;A"

L1 Ronde 0

L2 demi temps

L3 1/4 note pointée

L4 1/4 note

L8 1/8 note

L16 1/16 note

L32 1/32 note

•••

L64 1/64 note

L2 indique une demi temps ; L4 un quart de temps.

# PLAY "L1;A;A#;A-"

Notez que vous n'avez pas besoin de répéter l'option L pour chaque note. PLAY utilise la valeur de note actuelle jusqu'à ce que vous saisissiez une autre commande pour lui dire le contraire.

En fait, la plupart des options PLAY discutées dans le reste de cette section utilisent une valeur "actuelle" jusqu'à ce que vous les changiez.

Juste pour le plaisir, essayez de jouer trois notes 1/64 sur A :

PLAY "L64;A;A#;A-"

Maintenant c'est staccato.

# J'adore cette note pointée

Si vous lisez la musique, vous connaissez déjà les "notes pointées". Le point vous indique d'augmenter la longueur de la note de la moitié de sa valeur normale.

Par exemple, une note 1/4 pointée est égale à une note "3/8".

Vous pouvez jouer une telle note en ajoutant une point (.) ou une série de périodes (...J au nombre L. Chaque point augmente la longueur de note de la moitié de sa valeur normale.

Par exemple :

1/2 de 1/4 = 1/8

1/4. = 1/4 + 1/8 = une note 3/8

Essaye ça:

### PLAY "L4;A.;L8;C;L4;E.;L8;C;E;C;E;C;L4;A"

### Jouez par numéro

N(x) Joue la note x. x peut varier de 0 à 84. Dans les 7 octaves possibles, il y a 84 notes. x mis à 0 indique un silence.

10 FOR X = 0 TO 84 20 PRINT "NOTE#"; X 30 PLAY "N" + STR\$(X) 40 NEXT X

# **Montons (ou descendons) d'une octave ou deux** (OCTAVE)

Notre simple octave (octave 2) sonne bien, mais, la variété étant le piment de la musique, il devient un peu ennuyeux lorsqu'il est joué encore et encore (comme un piano avec seulement 12 touches).

Pour changer d'octave, utilisez la lettre O suivie d'un nombre compris entre 1 et 5. (Tout nombre en dehors de cette plage entraîne une erreur d'appel de fonction illégale.)

Si vous ne spécifiez pas l'octave, l'ordinateur utilise automatiquement l'octave 2, qui inclut le do médian. essayons de jouer une gamme de C simple :

# PLAY "CDEFGABAGFEDCBA"

Qu'est-il arrivé? G est la note la plus haute de l'octave 2, donc quand l'ordinateur a atteint A, il est passé au début de l'octave. Pour entrer dans l'Octave 3, essayez ceci :

# PLAY "CDEF;O3;GABA;O2;GFEDCBA"

# Une minute de silence, s'il vous plait (PAUSE)

Peut-être que ce dernier petit programme serait plus facile à écouter si toutes les notes n'étaient pas jouées ensemble. Utilisez la fonction P (pause) pour quelques instants de silence entre les notes et voyez si elles sonnent mieux.

Pour mettre une pause entre les notes, utilisez P suivi d'un nombre de 1 à 64.

Les longueurs de pause correspondent aux longueurs de note avec une différence importante. Vous ne pouvez pas utiliser de points (points) avec P. Pour compenser, tapez simplement une série de pauses. Par exemple, pour obtenir une pause de 3/8, tapez P4P8.

# 10 PLAY "A;P2;A;P2;A;P2;A;P2;A;P2;A;P2"

En fait, une pause de demi temps (P2) entre tous ces A ne les fait pas sonner beaucoup mieux, mais vous devriez avoir une idée de la façon dont P fonctionne.

# Il est temps de prendre le rythme

(TEMPO)

Pour l'instant, le programme de test ressemble à ceci :

# 5 CLS

10 PLAY "A;P2;A;P2;A;P2;A;P2;A;P2;A;P2"

C'est passable, sinon agréable, mais le tempo (vitesse) est un peu lent. Vous pouvez augmenter ou diminuer le tempo avec T et un nombre de 1 à 255.

Si vous ne spécifiez pas de tempo, votre ordinateur utilise automatiquement T2.

T Tempo définit le nombre de L4 en une minute. n peut varier de 32 à 255. La valeur par défaut est 120.

Commencez par ralentir le tempo du programme :

# 5 CLS 10 PLAY "T32;A;P2;A;P2;A;P2;A;P2;A;P2;A;P2"

Accélérez-le en changeant T32 en T45. Maintenant, c'est plus comme ça.

Que diriez-vous de l'accélérer au maximum, 255, et d'exécuter le programme.

Cela n'a pas pris longtemps, n'est-ce pas ?

# Exécution de la sous-chaîne (X)

Vous souvenez-vous de l'option d'exécution (X) de DRAW ? PLAY a une fonctionnalité similaire qui vous permet d'exécuter la sous-chaîne, puis de revenir à la chaîne d'origine et de la compléter.

La fonction execute prend la forme suivante :

XA \$;

A\$ contient une séquence de commandes et de fonctions de lecture normales. X dit à l'ordinateur de JOUER A\$.

Réorganisez le programme de démonstration afin qu'il exécute une sous-chaîne :

```
5 CLS
10 A$ = "A;A#;A-"
20 B$ = "O5;XA$;"
30 C$ = "O1;XA$;XB$;"
40 PLAY C$
```

Exécutez le programme et suivez son exécution.

Remarque : Chaque fois que vous utilisez la fonction d'exécution, un point-virgule(;) doit suivre le signe dollar (\$). Dans cet exemple, vous pouvez supprimer tous les points-virgules à l'exception de ceux qui suivent le signe dollar.

# Une note supplémentaire

+,-,<>

Non, nous n'allons pas lancer une nouvelle note, comme H ou L sur vous. Nous avons juste une dernière façon d'utiliser certaines des options de PLAY. Avec O (octave), T (tempo) et L (longueur de note), vous pouvez utiliser l'un des suffixes suivants au lieu d'ajouter un chiffre :

+ Ajoute 1 à la valeur actuelle.

- Soustrait 1 de la valeur actuelle.

>n Un symbole supérieur à précédant la note n joue la note dans l'octave immédiatement supérieure. <n Un symbole inférieur à précédant la note n joue la note dans l'octave inférieure suivante.

Utilisez l'exemple de programme pour en savoir plus sur ces fonctionnalités.

5 CLS 10 PLAY "A;A#;A-"

Remarquez que la ligne 10 définit le tempo. Exécutez le programme une fois juste pour avoir une oreille attentive. Rien n'a changé; c'est comme toujours, maintenant A+ à la ligne 10 et ajouter la ligne 20.

10 PLAY "A;A#;A-;A+" 20 GOTO 10

Maintenant changez la ligne 10:

10 PLAY "A>;A#;A-;A+"

Lancez le programme. Le signe > augmente automatiquement la valeur de A à chaque fois que la ligne 10 est exécutée.

# **Rouler**, Beethoven

Après tout le travail acharné que vous avez fait ces derniers temps, vous méritez une sérénade.

Connaissez-vous toutes les fonctions de PLAY ? Si c'est le cas, regardez-les travailler dans le programme suivant et voyez si vous pouvez nommer cette chanson !

5 CLS 100 A\$ ="C;E;F;L1;G;P4;L4;C;E;F;L1;G" 105 B\$= "P4;L4;C;E;F;L2;G;E;C;E;L1;D" 110 C\$ ="P8;L4;E;E;D;L2;C;L4;C;L2;E" 115 D\$ ="L4;G;G;G;L1;F;L4;E;F" 120 E\$ ="L2;G;E;L4;C;L8;D;D+;D;E;G;A;L1;O3;C" 125 X\$ = "XA\$;XB\$;XC\$;XD\$;XE\$;" 130 PLAY X\$

Reconnaissez-vous cette chanson ? Habillez-le un peu en ajoutant ces lignes :

10 PRINT STRING\$ (32,"\*") 20 PRINT "WHEN THE SAINTS" 30 PRINT "GO MARCHING IN" 35 PRINT STRING\$ (32,"\*") 40 FOR X = 1 TO 500: NEXT X 45 CLS 50 PRINT "OH WHEN THE SAINTS" 55 PRINT "OH WHEN THE SAINTS" 60 PRINT "OH WHEN THE SAINTS, GO MARCHIN IN" 65 PRINT "YES I WANT TO BE IN THAT NUMBER" 70 PRINT "WHEN THE SAINTS GO MARCHIN IN"

Exécutez le programme maintenant et chantez avec votre ordi.

# Programme faites-le vous même

Notre interprétation de "Saints" sonne bien, mais ce n'est pas le vrai style de la Nouvelle-Orléans. Jazzez-le selon vos propres goûts musicaux. Essayez de changer d'octave ou d'ajouter quelques dièses ou bémols.

Essayez vos propres arrangements musicaux. Nous en avons inclus plusieurs dans les exemples de programmes.

```
100 A$ = "C;E;F;L1;G;P4;L4;C;E;F;L1;G"

105 B$ = "P4;L4;C;E;F;L2;G;E;C;E;L1;D"

110 C$ = "P4;L4;D+;L8;E;G;E;P8;L4;C;L8;D;D+"

115 D$ = "L4;E;C;L2;O3;C;L8;O3;D;L8;O2;B-"

120 E$ = "G;E;L4;G;L1;F;P4;L8;G;F;E;F"

125 F$ = "L2;G;E;L4;C;L8;D;D+;E;G;L4;A;L1;O3;C"

130 X$ = "XA$;XB$;XC$;XD$;XE$;XF$;"

140 PLAY X$
```



# Accéder à un fichier séquentiel

Votre première et principale tâche est de stocker vos informations en permanence sur le disque.

Prêt à s'organiser? Nous continuerons avec votre collection de livres. Voici une petite liste de livres :

TRAVAIL
 LE BERCEAU DU CHAT
 PETIT C'EST BEAU
 INTRODUCTION A GW-BASIC

Si vous avez lu votre introduction, vous savez comment sauvegarder des programmes de base sur le disque. Pour enregistrer des informations, vous avez besoin d'un programme qui suit ces étapes :

# ÉTAPES POUR ENREGISTRER DES INFORMATIONS SUR LE DISQUE

- 1. Ouvrez la communication sur le disque afin que vous puissiez envoyer l'information vers un fichier
- 2. Enregistrer toutes les informations dans le fichier.
- 3. Fermez la communication avec le fichier sur le disque.

Lancez le programme avec cette ligne :

10 OPEN "O",1,"Livres"

Cela ouvre le fichier 1 "Livres" pour que vous puissiez "sortir" ("O") les informations en mode de sortie séquentiel.

Quelles que soient les informations que vous produisez, l'ordinateur les stocke sur le disque dans un "fichier" nommé LIVRES. Sortez maintenant les informations.

Taper:

15 CLS: PRINT "ENTREZ VOS LIVRES - TAPEZ <XX> UNE FOIS TERMINÉ" 10 OPEN "O",1,"Livres.txt" 20 INPUT "TITRE"; T\$ 30 PRINT #1, T\$ 40 GOTO 15

La ligne 30 "enregistre" (sort) vos titres de livres --- pas sur l'écran, mais sur le disque.

Fermez ensuite les communications. Taper:

25 IF T\$ = "XX" OR T\$ = "xx" THEN 50 50 CLOSE #1

L'ordinateur ferme alors la communication avec le fichier.

Ajoutez trois lignes supplémentaires au programme :

Exécutez ensuite le programme. L'ordinateur ouvre un "fichier" sur le disque et le nomme LIVRES. Le programme demande alors des titres. Taper:

TITRE? TRAVAIL

TITRE? LE BERCEAU DU CHAT

TITRE? PETIT C'EST BEAU

# TITRE? INTRODUCTION A GW-BASIC

Chaque fois que vous saisissez un titre, l'ordinateur l'imprime dans un emplacement spécial de la mémoire réservé au disque. Lorsque vous avez terminé, l'ordinateur imprime tous les titres dans le fichiers (Ligne 30) puis ferme la communication avec le fichier (Ligne 50).

Vos titres de livres sont maintenant tous enregistrés sur disque dans un fichier nommé Livres.txt. Pour les relire en mémoire, suivez à peu près les mêmes étapes.

# ÉTAPES POUR LA SAISIE D'INFORMATIONS À PARTIR DU FICHIER

1. Ouvrez un fichier afin de pouvoir saisir des informations à partir d'un fichier.

- 2. Vérifiez si vous êtes à la fin du fichier.
- 3. Entrez les informations du fichier du disque.
- 4. Répétez les étapes 2 et 3 jusqu'à ce que vous atteigniez la fin du fichier.

Tapez:

80 OPEN "I", #1, "Livres.txt" 90 INPUT #1,B\$ 100 PRINT B\$

La ligne 90 entre le premier titre de livre (B\$ du fichier Livres stocké sur bande. (Le nom de la variable que vous choisissez ne fait aucune différence.) La ligne 100 affiche ce titre sur votre écran.

Pour vérifier la fin du fichier et fermer le fichier, ajoutez ces lignes :

85 IF EOF(1) THEN 120 110 GOTO 85 120 CLOSE #1

La ligne 85 indique si vous êtes à la fin de ce fichier (dans ce CAS, le fichier Livres). aller à 120 et fermer le fichier.

Notez que EOF(1) vient avant la ligne INPUT #1. Si c'est après INPUT #1, vous obtiendrez une erreur IE error après la fin du fichier.

Listez cette dernière partie du programme en tapant list. Ça devrait ressembler à ça:

80 OPEN "I", #1, "Livres.txt" 85 IF EOF(1) THEN 120 90 INPUT #1,B\$ 100 PRINT B\$ 110 GOTO 85 120 CLOSE #1

Exécutez maintenant cette partie du programme.

L'ordinateur entre les éléments du disque. Une fois terminé, il affiche les quatre éléments sur votre écran.

# Un catalogue électronique de cartes

Supposons que vous deviez modifier le programme afin qu'il puisse également stocker les auteurs et les sujets des livres :

Titre	Auteur	Sujet
TRAVAIL	Studs Terkel	Sociologie
LE BERCEAU DU CHAT	Kurt Vonnegut	Fiction
PETIT C'EST BEAU	E. F. Shumacher	Économie
INTRODUCTION A GW-	Michel Gaudet	Informatique
BASIC		

Commencez par changer la partie "sortie" du programme (la première moitié). Tapez ces lignes :

15 CLS: PRINT "ENTREZ VOS LIVRES - TAPEZ <XX> UNE FOIS TERMINÉ" 10 OPEN "O",1,"Livres2.txt" 20 INPUT "TITRE"; T\$ 25 IF T\$ = "XX" OR T\$ = "xx" THEN 50 26 INPUT "AUTEUR"; A\$ 27 INPUT "SUJET"; S\$ 30 PRINT #1, T\$; "," A\$ "," ; S\$ 'Les variables dans le fichier doivent être séparées par des virgules 40 GOTO 15 50 CLOSE #1

Modifiez ensuite la partie "INPUT" du programme. Tapez ces lignes :

80 OPEN "I", #1, "Livres2.txt" 85 IF EOF(1) THEN 120 90 INPUT #1,T\$,A\$,S\$ 100 PRINT T\$,A\$,S\$ 110 GOTO 85 120 CLOSE #1

Profitez maintenant de cette organisation.

Par exemple, demandez au programme d'imprimer une liste de livres sur un sujet donné. Ajoutez ces lignes :

130 CLS 140 INPUT "QUEL SUJET"; C\$ 150 CLS: PRINT C\$, "Livres" : PRINT 160 OPEN "I", #1, "Livres2.txt" 170 IF EOF(1) THEN 210 180 INPUT #1,T\$,A\$,S\$ 190 IF S\$ = C\$ THEN PRINT "Titre",T\$,"Auteur", A\$ 200 GOTO 170 210 CLOSE #1

### Programme faites-le vous même

No.	Date	Payable à	Compte	Montant \$
101	5/13	Safeway	Nourriture	52.60
102	5/13	Amoco	Voiture	32.70
103	5/14	Joe's Cafe	Nourriture	10.32
104	5/17	American Airline	Vacances	97.50
105	5/19	Holiday Inn	Vacances	72.30

Supposons que vous ayez ces informations:

Écrivez un programme qui envoie tous les chèques sur le disque. Ensuite, faites-lui entrer le fichier à partir du disque afin que vous puissiez taper un compte, comme la nourriture, et l'ordinateur vous indiquera le montant total que vous avez dépensé en nourriture.

#### 5 CLS

10 OPEN "O" ,#1, "Cheques.txt" 15 PRINT "ENTREZ VOS CHEQUES - TAPEZ <XX> UNE FOIS TERMINÉ" 20 INPUT "NUMERO:"; N\$ 25 IF N\$ = "XX" OR N\$ = "xx" THEN 90 30 INPUT "DATE:"; D\$ 40 INPUT "PAYABLE A :"; P\$ 50 INPUT "COMPTE:"; S\$ 60 INPUT "MONTANT: \$"; A 70 PRINT#1,N\$;","D\$;","P\$",";S\$",";A 80 GOTO 15 90 CLOSE #1 92 CLS: T = 095 INPUT "QUEL COMPTE"; B\$ 120 OPEN "I", #1, "Cheques.txt" 130 IF EOF (1) THEN 170 140 INPUT #1, N\$, D\$,P\$,S\$,A 150 IF B = S\$ THEN T = T + A 160 GOTO 130 170 CLOSE #1 180 PRINT "TOTAL POUR "; B\$; "EST EN \$" ;T

# Fichiers à accès aléatoire

# Accéder à un fichier à accès aléatoire

Les étapes de programme suivantes sont nécessaires pour accéder à un fichier aléatoire :

1. Ouvrez le fichier en mode R :

OPEN "R", #1, "filename", 32

2. Utilisez l'instruction FIELD pour allouer de l'espace dans le tampon aléatoire aux variables qui seront lues à partir du fichier :

FIELD, #1, 20 AS N\$, 4 AS A\$, 8 AS P\$

Dans cet exemple, les 20 premières positions (octets) dans le tampon de fichier aléatoire sont allouées à la variable de chaîne N\$. Les 4 positions suivantes sont attribuées à A\$ ; les 8 suivants à P\$.

Noter

Dans un programme qui exécute à la fois INPUT et OUTPUT sur le même fichier aléatoire, vous pouvez souvent utiliser une seule instruction OPEN et une seule instruction FIELD.

3. Utilisez l'instruction GET pour déplacer l'enregistrement souhaité dans la mémoire tampon aléatoire.

GET #1, CODE%

Les données dans la mémoire tampon sont maintenant accessibles par le programme.

4. Reconvertissez les valeurs numériques en nombres à l'aide des fonctions de conversion : CVI pour les nombres entiers, CVS pour les valeurs à simple précision et CVD pour les valeurs à double précision.

PRINT N\$ PRINT CVS(A\$)

•••

Le programme de l'exemple 2 accède au fichier aléatoire, INFOFILE, qui a été créé dans l'exemple 1. En entrant le code à 3 chiffres, les informations associées à ce code sont lues à partir du fichier et affichées.

# Exemple 2

10 OPEN "R",#1,"INFOFILE",32 20 FIELD #1, 20 AS N\$, 4 AS A\$, 8 AS P\$ 30 INPUT "2-DIGIT CODE";CODE% 40 GET #1, CODE% 50 PRINT N\$ 60 PRINT USING "\$\$###.##";CVS(A\$) 70 PRINT P\$:PRINT 80 GOTO 30 Avec des fichiers aléatoires, la fonction LOC renvoie le numéro d'enregistrement actuel. Le numéro d'enregistrement en cours est le dernier numéro d'enregistrement utilisé dans une instruction GET ou PUT.

Par exemple, la ligne suivante termine l'exécution du programme si le numéro d'enregistrement courant dans le fichier n°1 est supérieur à 99 :

# IF LOC(1)>99 THEN END

L'exemple 3 est un programme d'inventaire qui illustre l'accès aléatoire aux fichiers. Dans ce programme, le numéro d'enregistrement est utilisé comme numéro de pièce et il est supposé que l'inventaire ne contiendra pas plus de 100 numéros de pièce différents.

Les lignes 900-960 initialisent le fichier de données en écrivant CHR\$(255) comme premier caractère de chaque enregistrement. Ceci est utilisé plus tard (ligne 270 et ligne 500) pour déterminer si une entrée existe déjà pour ce numéro de pièce.

Les lignes 130 à 220 affichent les différentes fonctions d'inventaire exécutées par le programme. Lorsque vous tapez le numéro de fonction souhaité, la ligne 230 se branche sur le sous-programme approprié.

# Exemple 3

120 OPEN"R",#1,"INVEN.DAT",39 125 FIELD#1,1 AS F\$,30 AS D\$, 2 AS Q\$,2 AS R\$,4 AS P\$ 130 PRINT: PRINT "FUNCTIONS:": PRINT 135 PRINT 1,"INITIALIZE FILE" 140 PRINT 2,"CREATE A NEW ENTRY" 150 PRINT 3."DISPLAY INVENTORY FOR ONE PART" 160 PRINT 4,"ADD TO STOCK" 170 PRINT 5, "SUBTRACT FROM STOCK" 180 PRINT 6,"DISPLAY ALL ITEMS BELOW REORDER LEVEL" 220 PRINT: PRINT: INPUT" FUNCTION"; FUNCTION 225 IF (FUNCTION<1)OR(FUNCTION>6) THEN PRINT "BAD FUNCTION NUMBER":GOTO 130 230 ON FUNCTION GOSUB 900,250,390,480,560,680 240 GOTO 220 250 REM BUILD NEW ENTRY 260 GOSUB 840 270 IF ASC(F\$) <> 255 THEN INPUT"OVERWRITE"; A\$: IF A\$ <> "Y" THEN RETURN 280 LSET F\$=CHR\$(0) 290 INPUT "DESCRIPTION"; DESC\$ 300 LSET D\$=DESC\$ 310 INPUT "QUANTITY IN STOCK";Q% 320 LSET Q\$=MKI\$(Q%) 330 INPUT "REORDER LEVEL":R% 340 LSET R\$=MKI\$(R%) 350 INPUT "UNIT PRICE";P 360 LSET P\$=MKS\$(P) 370 PUT#1,PART% 380 RETURN 390 REM DISPLAY ENTRY 400 GOSUB 840 410 IF ASC(F\$)=255 THEN PRINT "NULL ENTRY":RETURN

420 PRINT USING "PART NUMBER ###";PART% 430 PRINT D\$ 440 PRINT USING "QUANTITY ON HAND #####";CVI(Q\$) 450 PRINT USING "REORDER LEVEL #####";CVI(R\$) 460 PRINT USING "UNIT PRICE \$\$##.##":CVS(P\$) 470 RETURN 480 REM ADD TO STOCK 490 GOSUB 840 500 IF ASC(F\$)=255 THEN PRINT "NULL ENTRY":RETURN 510 PRINT D\$: INPUT "QUANTITY TO ADD"; A% 520 O%=CVI(O\$)+A% 530 LSET Q\$=MKI\$(Q%) 540 PUT#1.PART% 550 RETURN 560 REM REMOVE FROM STOCK 570 GOSUB 840 580 IF ASC(F\$)=255 THEN PRINT "NULL ENTRY":RETURN **590 PRINT D\$** 600 INPUT "OUANTITY TO SUBTRACT":S% 610 Q%=CVI(Q\$) 620 IF (0%-S%)<0 THEN PRINT "ONLY";0%;"IN STOCK" :GOTO 600 630 Q%=Q%-S% 640 IF Q% = < CVI(R\$) THEN PRINT "QUANTITY NOW";Q%;"REORDER LEVEL";CVI(R\$) 650 LSET Q\$=MKI\$(Q%) 660 PUT#1,PART% 670 RETURN 680 REM DISPLAY ITEMS BELOW REORDER LEVEL4 690 FOR I=1 TO 100 710 GET#1.I 720 IF CVI(Q\$)<CVI(R\$) THEN PRINT D\$;"QUANTITY";CVI(Q\$) TAB(50) "REORDER LEVEL";CVI(R\$) 730 NEXT I 740 RETURN 840 INPUT "PART NUMBER"; PART% 850 IF(PART% < 1)OR(PART% > 100) THEN PRINT "BAD PART NUMBER":GOTO 840 ELSE GET#1,PART%:RETURN 890 END 900 REM INITIALIZE FILE 910 INPUT "ARE YOU SURE"; B\$: IF B\$ <> "Y" THEN RETURN 920 LSET F\$=CHR\$(255) 930 FOR I=1 TO 100 940 PUT#1,I 950 NEXT I 960 RETURN

### Gérer les nombres



Avez-vous essayé d'écrire des programmes pour gérer beaucoup d'informations ?

Si c'est le cas, vous serez heureux d'apprendre que GW-BASIC dispose d'un outil facile à gérer pour suivre les informations.

Supposons, par exemple. vous voulez écrire un programme qui vous permette de gérer ces informations :

District	Votes pour le candidat A
1	143
2	215
3	125
4	331
5	442
6	324
7	213
8	115
9	318
10	314
11	223
12	152
13	314
14	92

Jusqu'à maintenant, vous avez utilisé des variables pour stocker des informations en mémoire. Par exemple, pour comptabiliser les votes des trois premiers districts, tapez :

$$A = 143$$
  $B = 215$   $C = 125$ 

Mais voici un meilleur type de variable que vous pouvez utiliser. Tapez:

$$A(1) = 143$$
  $A(2) = 215$   $A(3) = 125$ 

Chacune des variables ci-dessus a un "indice" (1), (2) et (3). Mis à part la façon dont ils utilisent l'indice, ces variables fonctionnent de la même manière que toutes les autres variables. A voir par vous-même. tapez ces deux lignes :

PRINT A; B; C PRINT A(1); A(2) ; A(3)

Maintenant, jetez un coup d'œil et comparez les deux programmes ci-dessous. Les deux fonctionnent de la même manière : le programme 1 utilise des "variables simples". Le programme 2 utilise des "variables indicées".

#### PROGRAM 1

5 CLS

10 DATA 14,215,125,331,442 20 DATA 324,213,115,318,314 30 DATA 223,152,314,92 40 READ A ,B,C,D,E 50 READ F,G,H,I,J 60 READ K,L,M,N 70 INPUT "DISTRICT NO. (1-14)"; Z 75 IF Z>14 THEN 70 80 IF Z=1 THEN PRINT A "VOTES" 90 IF Z=2 THEN PRINT B "VOTES" 100 IF Z=3 THEN PRINT C "VOTES" 110 IF Z=4 THEN PRINT D "VOTES" 120 IF Z=5 THEN PRINT E "VOTES" 130 IF Z=6 THEN PRINT F "VOTES" 140 IF Z=7 THEN PRINT G "VOTES" 150 IF Z=8 THEN PRINT H "VOTES" 160 IF Z=9 THEN PRINT I "VOTES" 170 IF Z=10 THEN PRINT J "VOTES" 180 IF Z= 11 THEN PRINT K "VOTES" 190 IF Z= 12 THEN PRINT L "VOTES" 200 IF Z= 13 THEN PRINT M "VOTES" 210 IF Z= 14 THEN PRINT N "VOTES" 220 GOTO 70

PROGRAM 2

5 CLS 10 DATA 14,215,125,331,442 20 DATA 324,213,115,318,314 30 DATA 223,152,314,92 40 DIM A(14) 50 FOR X = 1 TO 14 60 READ A (X) 70 NEXT X 80 INPUT "DISTRICT NO. (1-14)"; Z 85 IF Z<0 OR Z > 14 THEN 80 90 PRINT A(Z) "VOTES" 100 GOTO 80

Le programme 1 est lourd à écrire. Le programme 2 est court et simple à écrire.

Entrez et exécutez le programme 2. Voici comment cela fonctionne :

La ligne 40 réserve de l'espace pour une liste d'informations appelée "tableau (Array)" nommé A-avec 14 éléments indicés.

Les lignes 50 et 70 configurent une boucle pour compter de 1 à 14. la ligne 60 lit les 14 votes dans le tableau A :



La ligne 80 vous demande d'entrer un indice et la ligne 90 affiche l'élément que vous avez demandé.

Maintenant que vous avez stocké des informations dans un tableau, il est facile de les gérer. Par exemple, vous pouvez ajouter ces lignes, qui vous permettent de modifier les informations :

92 INPUT "VOULEZ-VOUS DES VOTES AJOUTER A CECI OUI ou NON "; R\$
94 IF R\$ = "NON" OR R\$ = "non" THEN 80
96 INPUT "COMBIEN DE VOTES DE PLUS"; X
97 A(Z) = A(Z) + X
98 PRINT "TOTAL DES VOTES POUR LE DISTRICT" Z "EST MAINTENANT" A(Z)

Ou vous pouvez ajouter ces lignes pour afficher les informations :

72 INPUT "VOUS VOULEZ VOIR TOUS LES TOTAUX ";S74 IF S = "OUI" OR S = "oui" THEN GOSUB 110 100 GOTO 72 110 PRINT "DISTRICT", "VOTE5" 120 FOR X = 1 TO 14 130 PRINT X, A(X) 140 NEXT X 150 RETURN

### Un deuxième tableau

Supposons que vous souhaitiez également suivre les votes d'un deuxième candidat ---- Candidat B :

Résultats des élections

District	Votes pour le candidat A	Votes pour le candidat B			
1	143	678			
2	215	514			
3	125	430			
4	331	475			
5	442	302			
6	324	520			
7	213	613			
8	115	694			
9	318	420			
10	314	518			
11	223	370			
12	152	412			
13	314	460			
14	92	502			

Pour ce faire, ajoutez un autre tableau au programme. Appelez-le tableau (Array) B. Le programme suivant enregistre les votes pour le candidat A (Array A) et le candidat B (Array B) :

5 CLS

```
10 DATA 14,215,125,331,442
20 DATA 324,213,115,318,314
30 DATA 223,152,314,92
40 DATA 678,514,430,475,302
50 DATA 520,613,694,420,518
60 DATA 370,412,460,502
70 DIM A(14), B(14)
80 FOR X = 1 TO 14
90 READ A (X)
100 NEXT X
110 FOR X = 1 TO 14
120 READ B (X)
130 NEXT X
140 INPUT "DISTRICT NO. (1-14)"; Z
145 IF Z<0 OR Z > 14 THEN 140
150 INPUT "CANDIDAT A OU B "; R$
160 IF R = "A" OR R = "a" THEN PRINT A(Z)
170 IF R = "B" OR R = "b" THEN PRINT B(Z)
180 GOTO 140
```

### Programme faites-le vous même

Écrivez un programme d'inventaire qui garde la trace de 12 articles numérotés de 1 à 12 et la quantité que vous avez de chaque article.

5 CLS 10 DATA 33,12,42,13,15,23 20 DATA 25,30,33,27,14,8 30 DIM I(12) 40 FOR X = 1 TO 12 50 READ I (X) 60 NEXT X 70 INPUT" ARTICLE NO. "; N 75 IF N<0 OR N > 12 THEN 70 80 PRINT "INVENTAIRE POUR L'ARTICLE " N "EST" I(N) 90 GOTO 70

### **Distribuez les cartes**

Pour garder une trace de 52 "cartes", vous devez utiliser un tableau. Effacez votre programme et tapez et exécutez celui-ci :

40 FOR X = 1 TO 52 50 C = INT(52\*RND(1)+1) 90 PRINT C; 100 NEXT X

L'ordinateur distribue 52 "cartes" aléatoires, mais si vous regardez attentivement, vous voyez que certaines cartes sont identiques.

Pour vous assurer que l'ordinateur ne distribue chaque qu'une seule fois, vous pouvez créer un autre tableau -Array T - qui garde une trace de chaque carte distribuée. Ajoutez ces lignes :

5 DIM T(52) 10 FOR X = 1 TO 52 20 T (X) = X 30 NEXT X

Les lignes ci-dessus construisent le tableau T et y placent toutes les 52 cartes : T(1) = 1, T(2) = 2, T(3) = 3 ... T(52) = 52.

Ajoutez ensuite quelques lignes qui "effacent" chaque carte du tableau T après qu'elle a été distribuée. Taper:

60 D(X) = C80 T(C) = 0

Maintenant, l'ordinateur ne peut pas distribuer deux fois la même carte au hasard. Par exemple, supposons que l'ordinateur distribue d'abord un deux. La ligne 80 change la valeur de T(2)s de 2 à 0.

Supposons ensuite que l'ordinateur en distribue deux autres. Puisque T(2) est maintenant égal à 0. La ligne 60 retourne à la ligne 50 pour distribuer une autre carte.

Exécutez le programme. Notez comment l'ordinateur ralentit à la fin du pont. Il doit essayer plusieurs cartes différentes avant d'en trouver une qu'il n'a pas encore distribuée.

Pour jouer à un jeu de cartes, vous devez garder une trace des cartes qui ont été distribuées. Vous pouvez le faire en construisant un autre tableau-Array D.

Ajoutez ces lignes, qui stockent toutes les cartes, dans l'ordre où elles sont distribuées, dans Array D :

7 DIM D(52) 60 D(X) = C 90 PRINT D(X);

### Programme faites-le vous même

Ajoutez des lignes au programme pour qu'il n'affiche que votre "main" - les 5 premières cartes distribuées.

**1 RANDOMIZE TIMER** 5 DIM T(52) 7 DIM D(52) 10 FOR X = 1 TO 52 20 T (X) = X30 NEXT X 34 CLS 36 PRINT " DISTRIBUTION DES CARTES" 40 FOR X = 1 TO 52 50 C = INT(52\*RND(1)+1)55 IF T (C) = 0 THEN 50 60 D(X) = C70 SOUND 128,1 80 T(C) = 090 PRINT D(X); 100 NEXT X 110 CLS 120 PRINT "Votre Main" 130 FOR X = 1 TO 5 140 PRINT D(X) " "; 150 NEXT X

### Tableau à deux dimensions

Le tableau des résultats des élections pour les deux candidats peut être représenté dans un seul tableau à deux dimensions.

		Colonne 1	Colonne 2
No Ligne	District	Votes pour le candidat A	Votes pour le candidat B
1	1	143	678
2	2	215	514
3	3	125	430
4	4	331	475
5	5	442	302
6	6	324	520
7	7	213	613
8	8	115	694
9	9	318	420
10	10	314	518
11	11	223	370
12	12	152	412
13	13	314	460
14	14	92	502

La variable tableau A a deux dimensions, 14 lignes et 2 collonnes

70 DIM A(14,2)

Le programme ci-dessous attribut les valeurs pour la première colonne et ensuite pour la deuxième

5 CLS 10 DATA 14,215,125,331,442 20 DATA 324,213,115,318,314 30 DATA 223,152,314,92 40 DATA 678,514,430,475,302 50 DATA 520,613,694,420,518 60 DATA 370,412,460,502 70 DIM A(14,2) 80 FOR X = 1 TO 14 90 READ A(X,1) 100 NEXT X 110 FOR X = 1 TO 14 120 READ A(X,2) 130 NEXT X 140 PRINT "Résultats des élections" 150 PRINT "Candidat A", "Candidat B" 160 FOR X = 1 TO 14 170 PRINT TAB(4);A(X,1);TAB(18);A(X,2) 180 NEXT X

# Créer un fichier de résultats

10 CLS 20 OPEN "O",1,"Result.txt" 30 PRINT #1,"District";",Candidat A,";"Candidat B" 40 FOR X = 1 TO 14 50 PRINT "ENTREZ LES VOTES - DISTRICT ";X 60 INPUT "Votes pour le Candidat A ";A 70 INPUT "Votes pour le Candidat B ";B 80 PRINT #1,X; ","A;","B 90 NEXT X 100 CLOSE #1 110 OPEN "I", #1, "Result.txt" 120 INPUT #1,D\$,A\$,S\$ 130 PRINT D\$,A\$,S\$ 140 FOR X = 1 TO 14 150 INPUT #1,D,A,B 160 PRINT D,A,B 170 NEXT X 180 CLOSE #1

Maintenant faisons un programme créer un fichier de données brute pour le tableau ci-dessous:

District	Population	Α	B	%A	%B	%Population	%A Pop	%B Pop
	Рор					%Рор		
A(X)	A(X,1)	A(X,2)	A(X,3)	A(X,4)	A(X,5)	A(X,6)	A(X,7)	A(X,8)
	Р	Α	В	PA	PB	PP	PPA	PPB
1	100	51	39					
2	125	92	22					
3	98	8	81					
4	130	51	69					
5	209	102	93					
6	55	1	50					
7	145	112	21					
8	126	82	39					
9	156	51	99					
10	127	120	2					
11	99	52	33					
12	108	89	8					
13	123	23	87					
14	100	49	48					

(Votes Total) VT = A + B, PA = A/VT, PB = B/VT, PP = VT / Pop, PPA= A/Pop, PPB = B/Pop
5 REM Création du fichier de données brutes 10 CLS 20 OPEN "O",1,"StatB.txt" 30 FOR X = 1 TO 14 40 PRINT "ENTREZ LES VOTES - DISTRICT ":X 50 INPUT "Nombre de la population ";P 60 INPUT "Votes pour le Candidat A ";A 70 INPUT "Votes pour le Candidat B ";B 80 VT = A+B90 PA=INT(A/VT\*1000)/10 100 PB=INT(B/VT\*1000)/10 110 PP = INT(VT/P\*1000)/10120 PPA = INT(A/P\*1000)/10130 PPB = INT(B/P\*1000)/10190 PRINT #1,P; ","A;","B;","PA;","PB;","PP;","PPA;","PPB 200 NEXT X 210 CLOSE #1 220 CLS 230 OPEN "I",1,"StatB.txt" 240 FOR X = 1 TO 14 250 INPUT #1,P,A,B,PA,PB,PP,PPA,PPB 260 PRINT P; ", "A;", "B;", "PA;", "PB;", "PP;", "PPA;", "PPB 270 NEXT X 280 CLOSE #1

Maintenant faisons un programme avec un menu qui utilise un tableau pour:

1. Afficher le nombre de votes pour un district en particulier.

2. Afficher le nombre de votes pour tous les districts.

3. Quitter le programme

5 CLS

10 A\$ = "District": B\$="Pop": C\$="%A" : D\$="%B": E\$= "%Pop": F\$="%APop": G\$="%BPop" 20 DIM A(14,8) 30 OPEN "I",1,"StatB.txt" 50 FOR X = 1 TO 14 60 INPUT #1,A(X,1),A(X,2),A(X,3),A(X,4),A(X,5),A(X,6),A(X,7),A(X,8) 70 NEXT X 80 CLOSE #1 90 CLS:PRINT "Faites votre choix 100 PRINT "1. Afficher le nombre de votes pour un district en particulier." 110 PRINT "2. Afficher le nombre de votes pour tous les districts." 120 PRINT "3. Quitter le programe." 140 INPUT "Quel numéro ";N 150 IF N = 1 THEN GOSUB 1000 160 IF N = 2 THEN GOSUB 2000 170 IF N = 3 THEN END 190 GOTO 90 1000 CLS: INPUT "Pour quel district ";X

1010 IF X<1 OR X>14 THEN 1000 1020 PRINT A\$ TAB(12) B\$ TAB(19) "A" TAB(24) "B" TAB(29) C\$ TAB(35) D\$ TAB(41) E\$ TAB(49) F\$ TAB(58) G\$ 1030 PRINT X TAB(11) A(X,1) TAB(18) A(X,2) TAB(23) A(X,3) TAB(27) A(X,4) TAB(33) A(X,5) TAB(40) A(X,6) TAB(48) A(X,7) TAB(58) A(X,8) 1040 PRINT:PRINT "Appuyez sur une touche pour continuer" 1050 K\$=INKEY\$:IF K\$="" THEN 1050 1060 RETURN 2000 CLS 2010 PRINT A\$ TAB(12) B\$ TAB(19) "A" TAB(24) "B" TAB(29) C\$ TAB(35) D\$ TAB(41) E\$ TAB(49) F\$ TAB(58) G\$ 2020 FOR X = 1 TO 14 2030 PRINT X TAB(11) A(X,1) TAB(18) A(X,2) TAB(23) A(X,3) TAB(27) A(X,4) TAB(33) A(X,5) TAB(40) A(X,6) TAB(50) A(X,7) TAB(60) A(X,8) 2040 NEXT X 2050 PRINT:PRINT "Appuyez sur une touche pour continuer" 2060 K\$=INKEY\$:IF K\$="" THEN 1050 2070 RETURN

Maintenant voici un programme pour créer un fichier avec un entête.

5 CLS 10 A\$ = "District": B\$="Pop": C\$="%A" : D\$="%B": E\$= "%Pop": F\$="%APop": G\$="%BPop" 20 DIM A(14,8) 30 OPEN "I",1,"StatB.txt" 50 FOR X = 1 TO 14 60 INPUT #1,A(X,1),A(X,2),A(X,3),A(X,4),A(X,5),A(X,6),A(X,7),A(X,8) 70 NEXT X 80 CLOSE #1 90 OPEN "O",1,"StatF.txt" 100 PRINT #1,A\$; TAB(12); B\$; TAB(20); "A"; TAB(29); "B" ;TAB(37); C\$; TAB(43); D\$; TAB(50); E\$ ;TAB(56); F\$ ;TAB(64); G\$ 110 FOR X = 1 TO 14 120 PRINT #1, X; TAB(14) USING"\*\*#";A(X,1); 130 PRINT #1, TAB(21) USING"\*\*#";A(X,2); 140 PRINT #1, TAB(28) USING"\*\*#";A(X,3); 150 PRINT #1, TAB(35) USING"\*\*.#";A(X,4); 160 PRINT #1, TAB(43) USING"\*\*.#";A(X,5); 170 PRINT #1, TAB(51) USING"\*\*.#";A(X,6); 180 PRINT #1, TAB(60) USING"\*\*.#";A(X,7); 190 PRINT #1, TAB(74) USING"\*\*.#";A(X,8) 200 NEXT X 210 PRINT:PRINT "Terminer" 220 CLOSE #1

### Gestion des mots



Dans la dernière section, vous avez utilisé des tableaux pour gérer les nombres. Ici, vous utiliserez des tableaux pour gérer les mots en modifiant, mettant à jour et imprimant un essai entier.

Commencez par une simple liste de mots : une liste de courses :

1. OEUFS	7. TOMATES
2. BACON	8. PAIN
3. POMMES DE TERRE	9. LAIT
4. SEL	10. FROMAGE
5. SUCRE	11. POISSON
6. LAITUE	12 JUS

Attribuez chaque mot à une variable indicée - cette fois utilisez une variable de chaîne indicée. Par exemple, pour les trois premiers éléments, saisissez :

S(1) = OEUFS S(2) = BACON S(3) = POMMES DE TERRE

Le signe de dollar est la seule différence entre ces variables indicées et celles de la dernière section.

Pour voir comment les éléments sont stockés, tapez :

PRINT S\$(1),S\$(2), S\$(3)

Construisez maintenant un programme qui lit ces mots dans un tableau nommé S\$ puis les affiche :

5 DIM S(12)10 DATA OEUFS, BACON, POMMES DE TERRE, SEL 20 DATA SUCRE, LAITUE, TOMATES, PAIN 30 DATA LAIT, FROMAGE, POISSON, JUS 40 FOR X = 1 TO 12 50 READ S(X)60 NEXT X 70 CLS:PRINT "Liste d'épicerie:" 80 FOR X = 1 TO 12 90 PRINT X; S(X)100 NEXT X

#### Programme faites-le vous même

Ajoutez des lignes au programme ci-dessus afin de pouvoir modifier n'importe quel élément de cette liste.

105 INPUT "QUEL NUMÉRO D'ARTICLE. VOULEZ-VOUS CHANGER, 0 POUR AUCUN"; N 110 IF N=0 THEN END 115 IF N<1 OR N > 12 THEN 110 120 INPUT "QU'EST-CE QUE L'ARTICLE DE REMPLACEMENT"; S\$(N) 130 GOTO 70

Voici un programme qui utilise un tableau pour écrire les paroles des chansons.

5 DIM A\$(4) 10 CLS:PRINT "TAPEZ 4 LINGES" 20 FOR X = 1 TO 4 30 INPUT A\$(X) 40 NEXT X 50 CLS 60 PRINT "C'EST VOTRE CHANSON:" 70 PRINT 80 FOR X = 1 TO 4 90 PRINT X;" "; A\$(X) 100 NEXT X

Ajoutez quelques lignes pour pouvoir réviser n'importe quelle ligne.

110 PRINT 120 INPUT "QUELLE LIGNE VOULEZ-VOUS RÉVISER, 0 POUR AUCUNE";L 125 IF L=0 THEN END 130 IF L<1 OR L > 4 THEN 120 135 PRINT "TAPEZ LA LIGNE DE REMPLACEMENT" 140 INPUT A\$(L) 150 GOTO 50

## Rédaction d'un essai (un roman, dissertation)

Maintenant que vous avez appris à utiliser les tableaux de chaînes, il sera facile d'écrire un programme qui stocke et édite ce que vous tapez. Tapez ce programme :

```
1 CLEAR 1000

5 DIM A$(50)

10 CLS:PRINT "TAPEZ UN PARAGRAPHE"

20 PRINT "APPUYEZ </> UNE FOIS TERMINÉ"

30 X = 1

40 A$ = INKEY$

50 IF A$ = "I" THEN 40

60 PRINT A$;

70 IF A$ = "/" THEN 110

80 A$(x) = A$(x) + A$

90 IF A$ = "." THEN X = X + 1

100 GOTO 40
```

110 CLS 120 PRINT "Votre paragraphe:" 130 PRINT 140 FOR Y = 1 TO X 150 PRINT A\$(Y); 160 NEXT Y

Exécutez le programme. Pour voir comment chaque phrase est stockée, tapez ces lignes :

PRINT A\$(1) PRINT A\$(2) PRINT A\$(3)

Voici comment fonctionne le programme :

La ligne 1 dégage beaucoup d'espace caractère.

La ligne 5 attribut de l'espace pour la variable tableau jusqu'à 50 phrases.

La ligne 30 met X = 1. X est utilisé pour étiqueter toutes les phrases.

La ligne 40 vérifie si vous avez appuyer sur une touche. Si A = "" la ligne 50 revois le programme à la ligne 40.

La ligne 60 affiche la touche que vous avez pressé.

La ligne 70 renvois le programme à la ligne qui affiche votre paragraphe lorsque vous appuyez sur /

La ligne 80 construit une chaîne et l'étiquette avec X. X = 1 jusqu'à ce que vous pressiez sur un point (.). À ce moment le programme met X = X + 1.

Par exemple:

si la première touche que vous pressez est L.

#### A\$(1) = "L"

la seconde touche que vous pressez est A.

A\$(1) = "L" + "A" = "LA"

Supposons que lorsque A\$(1) est égal à LES ROSES SONT ROUGES, vous appuyez sur un point. A\$(1) alors égale à la phrase entière; LES ROSES SONT ROUGES. Ensuite, la prochaine lettre est dans A\$(2).

La ligne 140 - 160 affiche votre paragraphe.

#### Programme faites-le vous même

Voici une touche (mais cela peut être fait !) Pour ceux qui sont intrigués par le traitement de texte, modifiez le programme ci-dessus pour que vous puissiez ;

1 Imprimez n'importe quelle phrase, 2. Révisez n'importe quelle phrase

```
1 CLEAR 1000
5 DIM A$(50)
7 CLS
10 PRINT "TAPEZ UN PARAGRAPHE"
20 PRINT "APPUYEZ </>
UNE FOIS TERMINÉ"
30 X = 1
40 \text{ A} = INKEY$
50 IF A$ = "" THEN 40
60 PRINT A$;
70 IF A$ = "/" THEN 105
80 A(X) = A(X) + A
90 IF A$ = "." OR A$ = "?" OR A$ = "!" THEN X= X + 1
100 GOTO 40
105 PRINT: PRINT
110 INPUT "(1) PRINT OU (2) REVISER";R
120 CLS
130 ON R GOSUB 1000, 2000
140 GOTO 105
1000 REM PRINT PARAGRAPHE
1010 \text{ FOR } Y = 1 \text{ TO } X
1020 PRINT A$(Y);
1030 NEXT Y
1035 A$=INKEY$:IF A$ = "" THEN 1035
1040 RETURN
2000 REM REVISSÉ PARAGRAPHE
2010 \text{ FOR } Y = 1 \text{ TO } X
2020 PRINT Y "--" A$(Y)
2030 NEXT Y
2040 INPUT "NUMÉRO DE PHARSE À RÉVISER";S
2045 IF S> X OR S < 1 THEN 2040
2050 PRINT A$(S)
2060 PRINT "SAISIR LA PHRASE À SUPPRIMER"
2070 INPUT D$
2080 L=LEN (D$)
2090 PRINT "TAPEZ UNE PHRASE DE REMPLACEMENT"
2100 INPUT R$
2110 FOR Z =1 TO LEN(A$(S))
2120 IF MID$(A$(S),Z,L) = D$ THEN 2160
2130 NEXT Z
2140 PRINT D$ "--- N'EST PAS DANS VOTRE PHRASE"
2150 GOTO 2060
2160 \text{ E} = \text{Z} - 1 + \text{LEN}(\text{D}\$)
2170 \text{ A}(S) = \text{LEFT}(A(S), Z-1) + R + \text{RIGHT}(A(S), LEN(A(S))-E)
2180 RETURN
```



Tout commis au dossier sait qu'il est plus facile de trouver des informations triées par ordre alphabétique. Tapez ce programme et exécutez-le. jusqu'à ce que vous soyez convaincu que l'ordinateur peut classer par ordre alphabétique :

5 CLS

10 INPUT "TAPEZ LE PREMIER MOTS"; A\$ 15 INPUT "TAPEZ LE DEUXIÈME MOTS"; B\$ 20 IF A\$ < B\$ THEN PRINT A\$ " VIENT AVANT " B\$ 30 IF A\$ > B\$ THEN PRINT A\$ " VIENT APRÈS " B\$ 40 IF A\$ = B\$ THEN PRINT "LES DEUX MOTS SONT LES MÊMES" 50 INPUT "UN AUTRE TEST ? O/N "; R\$ 60 IF R\$ = "o" OR R\$="O" THEN 10

Avec les chaînes, les signes supérieur à (>), inférieur à (<) et égal (=) ont une nouvelle signification. Ils indiquent laquelle des deux chaînes vient avant l'autre dans l'ordre alphabétique :

< précède alphabétiquement < = précède ou est le même alphabétiquement > suit par ordre alphabétique > = suit ou est le même par ordre alphabétique = est le même

Puisque l'ordinateur peut classer par ordre alphabétique, il est facile d'écrire un programme de tri.

Trier un tableau avec la méthode Bubble - Sort

Ce programme utilise WHILE ... WEND, FOR ... NEXT et IF ... THEN

SWAP permet de changer les valeurs entre deux variables

Tapez et lancez ce programme, qui contient 5 mots :

5 CLS 10 DIM A\$(5) 20 FOR I = 1 TO 5 30 INPUT "TAPEZ UN MOT "; A\$(I) 40 NEXT I 45 FLIPS=1 50 **WHILE** FLIPS 60 FLIPS=0 70 FOR N=1 TO 4 80 IF A\$(N)>A\$(N+1) THEN SWAP A\$(N), A\$(N+1): FLIPS=1 90 NEXT N 100 WEND 110 FOR N=1 TO 5 120 PRINT A\$(N), 'La virgule permet d'afficher en colonne 130 NEXT N

Cette section montre une manière simple de trier. Si vous devez trier de nombreux éléments, vous pouvez rechercher des méthodes de tri plus rapides.



Tableau 3 dimensions

Vous pouvez continuer avec autant de dimensions que vous le souhaitez dans une variable tableau. Vous n'êtes limité que par la quantité d'informations que vous pouvez stocker dans la mémoire de l'ordinateur.

Ajoutez une troisième dimension au tableau: les groupes d'intérêt. Voir Enregistrement

Voici les informations :

VOTES DU GROUPE D'INTÉRÊT 1

	Candidat 1	Candidat 2
District 1	143	678
District 2	215	514
District 3	125	430

VOTES DU GROUPE D'INTÉRÊT 2

	Candidat 1	Candidat 2
District 1	525	54
District 2	318	157
District 3	254	200

## VOTES DU GROUPE D'INTÉRÊT 3

	Candidat 1	Candidat 2
District 1	400	119
District 2	124	300
District 3	75	419

Pour mettre tout cela dans la mémoire de votre ordinateur, tapez:

5 DIM V(3,3,2) 10 DATA 143, 678, 215, 514, 125, 430 20 DATA 525, 54, 318, 157, 254, 200 30 DATA 400,118,124,300,75,418 40 FOR G = 1 TO 3 50 FOR D = 1 TO 3 60 FOR C = 1 TO 2 70 READ V(G,D,C) 80 NEXT C 90 NEXT D 100 NEXT G 105 CLS 110 INPUT "Groupe d'intérêt No. (1-3)"; G 120 IF G < 1 OR G > 3 THEN 110 130 INPUT "DISTRICT NO. (1-3)"; D 140 IF D < 1 OR D > 3 THEN 130 150 INPUT "CANDIDAT NO. (1-2)"; C 160 IF C < 1 OR C > 2 THEN 150 170 PRINT V(G,D,C) 180 GOTO 110

Exécutez le programme et testez les indices. Les lignes 40 à 100 lisent tous les votes dans le tableau V, en leur donnant chacun trois indices :

Le premier indice est le groupe d'intérêt (Groupes d'intérêt 1-3), Le deuxième indice est le district (Districts 1-3). Le troisième indice est le candidat (candidats 1-2).

Voici la mémoire de l'ordinateur:

$\forall (1,1,1) \rightarrow 143$	V(1,1,2)-+678
V(1,2,1)->215	V(1,2,2)->514
V(1,3,1)→125	V(1,3,2)-+430
V(2,1,1)->525	V(2,1,2)-+54
V(2,2,1)→318	V(2,2,2)-+157
V(2,3,1)-+254	V(2,3,2)-200
V(3,1,1)-+400	V(3,1,2)-+119
V(3,2,1)→124	V(3,2,2)->300
V(3,3,1)→75	V(3,3,2)-+419

Par exemple, 678 est maintenant étiqueté V(1, 1, 2). Cela signifie que 678 est du groupe d'intérêt 1, du district 1 et du candidat 2.

Pour profiter des trois dimensions, supprimez les lignes 110-180 et tapez :

110 R=0 120 IF R = 0 THEN 140 130 A\$=INKEY\$:IF A\$="" THEN 130 140 CLS:PRINT: PRINT "TAPEZ <1> POUR GROUPE" 150 PRINT "<2> POUR DISTRICT OU <3> POUR CANDIDAT" 160 P = 0 : INPUT R170 ON R GOSUB 1000,2000,3000 180 GOTO 120 1000 INPUT "GROUPE (1-3)"; G 1010 IF G<1 OR G>3 THEN 1000 1020 CLS 1030 PRINT "VOTES POUR LE GROUPE" G 1040 PRINT "CAND. 1 " 1050 FOR D = 1 TO 3 1060 PRINT TAB(P) "DIST. " D; 1065 P = P + 151070 NEXT D 1080 P=0:C=1 1090 FOR D = 1 TO 3 1100 PRINT TAB(P) V(G,D,C); 1105 P = P + 151110 NEXT D 1120 PRINT:PRINT:PRINT "CAND. 2 " 1125 P=0 1130 FOR D = 1 TO 3 1140 PRINT TAB(P) "DIST. " D; 1150 P = P + 151160 NEXT D 1170 P=0:C=1 1180 FOR D = 1 TO 3 1190 PRINT TAB(P) V(G,D,C); 1200 P = P + 151210 NEXT D 1220 RETURN 2000 INPUT · "DISTRICT(1-3)"; D 2010 IF D<1 OR D>3 THEN 2000 2020 CLS 2030 PRINT "VOTES POUR DIST. " D 2040 PRINT "CAND. 1 " 2050 PRINT "CAND. 2" 2060 FOR G = 1 TO 3 2070 PRINT "GROUP" G 2080 FOR C = 1 TO 2 2100 PRINT V(G,D,C) 2110 NEXT C 2120 NEXT G

2130 RETURN 3000 INPUT "CANDIDAT (1-2)"; C 3010 IF C<1 OR C>2 THEN 3000 3020 CLS 3030 PRINT "VOTES POUR CAND. " C 3040 PRINT "DIST. 1 "; 3050 PRINT "DIST. 2 "; 3060 PRINT "DIST. 3" 3070 FOR G = 1 TO 3 3080 PRINT "GROUPE" G; 3090 FOR D = 1 TO 3 3100 PRINT V(G,D,C); 3110 NEXT D 3120 NEXT G 3130 RETURN

#### Programme faites-le vous même

Écrivez un programme pour distribuer les cartes en utilisant un tableau à deux dimensions.

Faites de la première dimension la couleur de la carte {1-4) et de la deuxième dimension la valeur de la carte (1-13).

```
5 RANDOMIZE TIMER
10 DIM S$(4),N$(13),T(4,13)
20 DATA TREFLE, COEUR, CARREAU, PIQUE
30 \text{ FOR } X = 1 \text{ TO } 4
40 READ S$(X)
50 NEXT X
60 DATA AS,2,3,4,5,6,7,8,9,10, VALET, DANE, ROI
70 FOR X = 1 TO 13
80 READ N$(X)
90 NEXT X
100 \text{ FOR } S = 1 \text{ TO } 4
110 FOR N= 1 TO 13
120 T(S,N) = (S-1) * 13 + N
130 NEXT N ,S
140 FOR X = 1 TO 52
150 \text{ S} = \text{INT}(4*\text{RND}(1)+1): \text{N} = \text{INT}(13*\text{RND}(1)+1)
160 \text{ IF T}(S,N) = 0 \text{ THEN } 150
170 T(S,N) = 0
180 PRINT N$(N) "-" S$(S),
190 NEXT X
```

#### Le jeu des chiffres



Votre ordinateur comprend plusieurs fonctions mathématiques avancées. Cette section donne un aperçu des fonctions et montre comment les utiliser.

Avant de continuer, cependant, vous devez connaître quelques fonctions et définitions décrites ci-dessous.

## Exponentiation

Λ

Rapide! Qu'est-ce que 1,5 au carré ? Que diriez-vous de 77 au cube ? Si vous ne savez pas. demander à l'ordinateur. Chaque fois que vous souhaitez élever un nombre à la puissance n, suivez ce format :

nombre (^) puissance

nombre est la base (le nombre que vous souhaitez élever à la puissance n, il peut s'agir de n'importe quelle expression numérique.

La puissance est l'exposant auquel la base est élevée. Il peut s'agir de n'importe quelle expression numérique.

Remarque : L'exponentiation a priorité sur les autres opérateurs. Par exemple, si l'ordinateur calcule - 2^2, le résultat est un nombre négatif. Pour élever - 2 à la puissance 2d "correctement" (ce qui donne un nombre positif), mettez - 2 entre parenthèses.

Commencez par 77 au cube (77^3). Après avoir regardé le bloc de syntaxe, pouvez-vous donner la commande ? Votre réponse devrait être 456533.

Si votre écran ressemble à ceci, vous êtes sur la bonne voie :

PRINT 77^3 456533 Ok Essayez d'élever 10 à la puissance 10. L'écran affiche :

1E + 10

Puisque 10 000 000 a plus de 9 chiffres significatifs, l'ordinateur est passé à la notation E.

Que diriez-vous de 100 à la puissance 100?

L'écran affiche-t-il un Overflow (débordement) ?

Overflow 1.701412E+38 Ok

Cela signifie que la réponse est trop grande pour que l'ordinateur puisse la gérer. Tout ce qui se trouve en dehors de la plage provoque une erreur de dépassement, soit

entier 2 bytes -32768-32767

virgule flottante simple précision 4 bytes  $\pm 2.938726 \cdot 10^{-39} \pm 1.701412 \cdot 10^{38}$ 

virgule flottante double précision 8 bytes  $\pm 2.938735877055719 \cdot 10^{-39} - \pm 1.701411834604692 \cdot 10^{38}$ 

## Programme faites-le vous même

Écrivez un programme court qui affiche le carré de chaque nombre entier de 1 à 10.

5 CLS 10 FOR NUMBER = 1 TO 10 20 PRINT NUMBER^2 30 NEXT NUMBER

## SQR

SQR vous permet de trouver la racine carrée d'un nombre. Voici sa syntaxe :

SQR (nombre)

nombre est zéro ou n'importe quel nombre positif.

Par exemple, si vous voulez la racine carrée de 100, tapez :

## PRINT SQR(100)

et vous découvrirez (si vous ne le saviez pas déjà) que la réponse est 10.

Écrivez un autre programme court pour afficher la racine carrée de chaque dixième nombre de 100 à 0.

5 CLS 10 FOR NUMBER = 100 TO 1 STEP -10 20 PRINT SQR(NUMBER) 30 NEXT NUMBER

### Fonctions trigonométriques

Regardez ce triangle. Vous l'utiliserez tout au long de la discussion sur les fonctions trigonométriques.



La trigonométrie a de nombreuses applications pratiques. Par exemple, imaginez que votre triangle est en fait le toit d'une maison que vous construisez. Les fonctions trigonométriques peuvent vous aider à déterminer soit la longueur des chevrons, soit la pente du toit (la « pente »). Donc, si les mathématiques vous rebutent mais que construire des choses vous excite, cette section pourrait être exactement ce que vous cherchez.

Notez que nous avons étiqueté les angles avec le préfixe A et les côtés avec le préfixe S. L'angle A, par exemple, est AA ; le côté opposé est SA.

En utilisant le triangle, nous pouvons définir les fonctions trigonométriques communes de la manière suivante :

Sinus de AA = SIN (AA) = SA/SCCosinus de AA = COS (AA) = SB/SCTangente de AA = TAN (AA) = SA/SB

#### **Degrés vs Radians**

Pour définir un angle, vous pouvez utiliser l'une des deux unités de mesure. L'unité la plus courante est le degré; l'unité "plus technique" est le radian.

Votre ordinateur suppose que tous les angles sont mesurés en radians. Étant donné que les radians peuvent vous être quelque peu étrangers, vous pouvez les convertir en degrés (et vice versa) de cette façon :

Voici un cercle avec les angles en degrés et en radians:



Degrés en Radians : Degrés / 57,29577951 (Deg/(180/ $\pi$ )) = deg \*  $\pi$ /180 Radians en Degrés : Radians \* 57.29577951 (Rad\*(180/ $\pi$ )) = rad /  $\pi$ /180

0 degré = 0 radians et 90 degré =  $\pi/2$  ( $\pi/2$  = 1,57)

Les exemples de programmes de ce chapitre incluent un "convertisseur" qui prend les degrés que vous avez saisis et les convertit automatiquement en radians (et vice versa dans certains cas).

## SIN

Cela se prononce sinus.

Sa syntaxe est :

SIN (angle)

angle est la taille de l'angle en radians.

Étant donné la longueur d'un côté et les tailles de deux angles, vous pouvez utiliser SIN pour déterminer les longueurs des autres côtés.

Entrez et exécutez le programme suivant, en saisissant toutes les valeurs.

## 5 CLS

10 INPUT "QUEL EST L'ANGLE A (AA)"; AA: IF AA<=0 OR AA>=180 THEN 100 20 INPUT "QUEL EST L'ANGLE (AB)"; AB: IF AA <=0 OR AB >=180 THEN 100 30 INPUT "QUEL EST LE COTÉ (SC)"; SC: IF SC <=0 THEN 100 40 AC = 180-(AA+AB) 'VALEUR DE L'ANGLE AC 50 IF (AA+AB+AC) <> 180 THEN 100 'TRIANGLE=180 DÉGRÉES 60 AA=AA/57.29577951: AB=AB/57.29577951:AC=AC/57.29577951 'CONVERTIR DES DEGRÉS EN RADIANS 70 SA=((SIN(AA))/(SIN(AC))) \* SC: IF SA<0 THEN 100 80 SB=((SIN(AB))/(SIN(AC))) \* SC: IF SB<0 THEN 100 90 PRINT "COTÉ A (SA) EST " SA "DE LONG": PRINT "COTÉ (SB) EST " SB "DE LONG": GOTO 10 100 PRINT "DÉSOLÉ. CE N'EST PAS UN TRIANGLE. RÉESSAYER": GOTO 10

Lorsque l'ordinateur vous demande AB et AC. entrer les degrés des angles. Si vous entrez un nombre négatif ou un nombre supérieur ou égal à 180, l'ordinateur passe à la ligne 100. Il imprime ensuite le message et demande à nouveau les tailles.

Si vous entrez un nombre négatif pour Sc, il fait LA même chose.

Puisque vous ne connaissez pas la taille de AC, l'ordinateur la calcule automatiquement à la ligne 40. Si la somme des trois angles n'est pas égale à 180 degrés, l'ordinateur prend les mesures appropriées dans la ligne 50. La ligne 60 convertit les degrés en radians pour que l'ordinateur puisse effectuer les calculs de sinus.

## **Ondes sinusoïdales**

Vous avez peut-être déjà vu des ondes sinusoïdales. Ils sont utilisés pour indiquer le courant alternatif et d'autres conditions électriques. Exécutez le programme suivant pour voir une onde sinusoïdale "à défilement horizontal"

1 '\*\*\* Onde sinusoïdale \*\*\* 10 CLS 15 SCREEN 7 20 LINE (0,86)-(255,86) 25 PI=3.141592 30 A1 =-4\*PI 35 A2=4\*PI 40 N=180 45 R=50 50 X = (A2-A1) / N55 F=255/(A2-A1) 60 FOR I = A1 TO A2 STEP X 65 X=I\*F 70 Y=R\*SIN(I) 75 PSET ((X+140),(80+Y)) 80 NEXT I 90 GOTO 90

# COS

La fonction cosinus est liée à la fonction sinus et a la syntaxe est la suivante :

Cos (angle)

angle EST la taille de l'angle en radians.

Étant donné les longueurs de deux côtés sans la taille d'un angle, vous pouvez utiliser le cosinus pour déterminer la longueur du troisième côté d'un triangle, comme illustré ici :

## 5 CLS 10 INPUT "QUEL EST L'ANGLE C (AC)"; AC:IF AC<=0 OR AC>=180 THEN 100 20 AC=AC / 57.29577951'CONVERTIR DES DEGRÉS EN RADIANS 30 INPUT "QUEL EST LE COTÉ A (SA)"; SA:IF SA<=0 THEN 100 40 INPUT "QUEL EST LE COTÉ B (SB)"; SB:IF SB=<0 THEN 100 50 SC = ((SA^2)+(SB^2))-(2\*(SA\*SB\*COS(AC))): IF SC<0 THEN 100 60 PRINT "COTÉ C (SC} EST " SQR(SC) "DE LONG":GOTO 10 100 PRINT "DÉSOLÉ, CE N'EST PAS UN TRIANGLE. RÉESSAYER": GOTO 10

5 CLS

```
10 FOR A = -180 TO 179 STEP 10

15 RD=A/57.29577951

30 CP=COS(RD)*14+16.5 'COS POSITION

40 SP=SIN(RD)*14+16.5 'SIN POSITION

50 IF SP<=CP THEN 70

60 PRINT TAB(CP); "C";TAB(SP);"S": GOTO 80

70 PRINT TAB(SP) ;"S" ;TAB(CP) ;"C"

80 NEXT A

90 GOTO 10
```

## Programme faites-le vous même

Le cosinus peut faire des courbes par lui-même. Réécrivez le programme "Onde sinusoïdale" afin qu'il trace COS au lieu de SIN. Quelle est la différence entre les deux ?

## TAN

La troisième fonction trigonométrique, TAN, permet de calculer la tangente d'un angle. Voici sa syntaxe :

TAN (angle)

angle est la taille de l'angle en radians.

Vous pouvez utiliser TAN pour déterminer, entre autres, un côté d'un triangle, étant donné un autre côté et un angle.

Entrez et exécutez ce programme :

5 CLS

10 INPUT "QUEL EST LE COTÉ B (SB)"; SB:IF SB<=0 THEN 100 20 INPUT "QUEL EST L'ANGLE A (AA)"; AA:IF AA<=0 OR AA>=180 THEN 100 30 AA=AA/57.28577851 'CONVERTIR DES DEGRÉS EN RADIANS 40 SA=SB\*(TAN(AA)): IF SA<=0 THEN 100 50 PRINT "COTÉ A (SA) EST " SA "DE LONG":GOTO 10 100 PRINT "DÉSOLÉ, CE N'EST PAS UN TRIANGLE. RÉESSAYER": GOTO 10

La clé de ce programme, bien sûr, est la ligne 40, où la tangente de AA est multipliée par la longueur de SB pour déterminer la longueur de SA.

## ATN

ATN (arc tangente) est l'inverse de TAN et a la syntaxe suivante :

ATN (angle) angle est la taille de l'angle en radians.

Le programme suivant utilise ATN et TAN pour calculer deux angles inconnus d'un triangle lorsque deux côtés et un angle sont connus.

```
10 CLS
```

```
20 INPUT "QUEL EST LE COTÉ A (SA)"; SA:IF SA<=0 THEN 150
30 INPUT "QUEL EST LE COTÉ C (SC)"; SC:IF SC<=0 THEN 150
40 INPUT "QUEL EST L'ANGLE B (AB)"; AB:IF AB<=0 OR AB>=180 THEN 150
50 X=(180-AB) 'AA+AC=180-AB
60 X=X/57.28577851 'CONVERTIR DES DEGRÉS EN RADIANS
70 Y=((SA-SC)/(SA+SC))*TAN(X/2)
80 Z=ATN(Y)
90 AA=(X/2)+(Z)
100 AC=(X/2)-(Z)
110 AA=AA*57.29577951 'CONVERTIR DES DEGRÉS EN RADIANS
120 AC=AC*57.29577951 'CONVERTIR DES DEGRÉS EN RADIANS
130 PRINT "ANGLE A (AA) EST " AA "DÉGRÉES"
140 PRINT "ANGLE C (AC) EST " AC "DÉGRÉES":GOTO 20
150 PRINT "DÉSOLÉ, CE N'EST PAS UN TRIANGLE. RÉESSAYER": GOTO 20
```

TAN ((AA-AC)/2) est égal à ((SA-SC)/(SA + SO) \* TAN ((AA + ACl/2). Notez également qu'il a fallu convertir les radians "de l'ordinateur" en "vos" degrés (Lignes 110 et 120).

## LOG

LOG renvoie le logarithme naturel d'un nombre. C'est l'inverse de EXP, donc X = LOC(EXP(X)). Voici la syntaxe de LaC :

LOG (nombre) nombre est supérieur à zéro. Le logarithme d'un nombre est la puissance à laquelle une "base" donnée doit être élevée pour obtenir le résultat du nombre.

Les "LOG" sont utiles dans les problèmes scientifiques et mathématiques. Dans la fonction LOG, si vous omettez la base, l'ordinateur suppose que vous spécifiez Base e (2.718281828).

Pour trouver le logarithme d'un nombre 10 d'une autre base, B. utilisez cette formule :

LOG base B (x) = LOG e (x) / LOG e (B)

Par exemple. LOG (32768)/LOG(2) renvoie le logarithme 10 Base 2 de 32768. (Il renvoie la puissance à laquelle 2 est élevé pour obtenir 32768.)

Essayez ceux-ci :

PRINT LOG(1) PRINT LOG (100) PRINT LOG (2.718281828)

Calculez le LOG de chacun des nombres suivants : a) 1003 b) 74,9865 c) 3,354285

a.) ?LOG (1003) 6.910751

b.) ?LOG(74.8865) 4.315974

c,) ? LOG(3.354285) 1.210239

Calculez le log à Base de chacun des nombres suivants :

a) 1 b)10 c) 100 d) 500 e) 0.1 f) 1001 log e x Hint: log 10 x = log e 10

5 CLS 10 INPUT "QUEL NOMBRE "; NUMBER 15 X=LOG(NUMBER)/LOG(10) 20 PRINT "LE LOG BASE 10 DE" NUMBER "EST" X 25 GOTO 10

a.) 0

Remarque : Le log de 1 dans n'importe quelle base est 0.

b.) 1
c.) 2
d.) 2.69897
e.) - 1
f.) 3.000435

## EXP

La fonction EXP renvoie l'exponentielle naturelle d'un nombre (enombre).

EXP est l'inverse de LOG ; donc X = EXP(LOG(X)). Voici la syntaxe d'EXP :

EXP (nombre) nombre est inférieur à 87,3365.

Exécutez ce programme pour voir EXP au travail.

10 CLS 20 INPUT "ENTREZ X"; X 30 PRINT "EXP(X)=" ; EXP(X) 40 GOTO 20

## FIX

C'est impressionnant quand votre ordinateur porte un nombre jusqu'à 9 chiffres significatifs, surtout quand 8 de ces nombres sont à droite de la virgule décimale.

Cependant, parfois, vous ne voudrez peut-être pas tous ces chiffres ; vous ne voudrez peut-être que la partie entière (le nombre à gauche de la virgule décimale).

FIX vous permet de geler ce nombre entier en coupant simplement tous les chiffres à droite de la virgule décimale. Voici la syntaxe de FIX :

FIX (nombre)

Par exemple, tapez :

PRINT FIX (2.26413851)

Le résultat est 2 Ok Voici un programme qui décompose un nombre en ses parties entières et fractionnaires.

```
10 CLS
20 INPUT"UN NOMBRE COMME X.YZ" ; X
30 W=FIX(X)
40 F=ABS (X) -ABS (W)
50 PRINT"LA PARTIE ENTIERE EST ";W
60 PRINT"LA PARTIE FRACTIONNELLE EST "; F
70 GOTO 20
```

# INT

Pour tronquer une expression à un nombre entier.

Syntaxe: INT(x)

Les nombres négatifs renvoient le nombre le plus bas suivant.

Les fonctions FIX et CINT renvoient également des valeurs entières.

Exemples:

PRINT INT(98. 89) 98

PRINT INT(-12. 11) -13

## SGN

Pour retourner le signe de x.

Syntaxe:

SGN(x)

x est une expression numérique quelconque.

Si x est positif, SGN(x) renvoie 1.

Si x vaut 0, SGN(x) renvoie 0.

Si x est négatif, SGN(x) renvoie -1.

Cette instruction est similaire, mais pas la même que SIN(x), qui renvoie une fonction trigonométrique en radians, plutôt qu'en uns et zéros.

Exemples:

10 INPUT "Entrez une valeur", x 20 ON SGN(X)+2 GOTO 100, 200, 300

GW-BASIC passe à 100 si X est négatif, 200 si X est 0 et 300 si X est positif.

## **DEF FN**

GW-BASIC a une fonction numérique, DEF FN, qui ne ressemble à aucune autre dont nous avons parlé jusqu'à présent. DEF FN vous permet de créer votre propre fonction mathématique.

Vous pouvez utiliser votre nouvelle fonction de la même manière que n'importe quelle fonction disponible (SIN, COS, TAN, etc.). Une fois que vous avez utilisé DEF FN pour définir une fonction, vous pouvez la mettre en œuvre dans votre programme en attachant le préfixe FN au nom que vous attribuez à la nouvelle fonction.

Voici la syntaxe pour DEF FN :

DEF FN nom (liste de variables) - le nom de la formule est le nom que vous attribuez à la fonction que vous créez.

La liste des variables contient une "variable factice" pour chaque variable à utiliser par la fonction.

La formule définit l'opération sur les termes des variables données dans la liste des variables.

Remarque : Les noms de variable qui apparaissent dans la formule servent uniquement à définir la formule ; elles n'affectent pas les variables de programme portant le même nom. Vous ne pouvez avoir qu'un seul argument dans un appel de formule, par conséquent, DEF FN ne doit contenir qu'une seule variable.

Vous ne pouvez utiliser DEF FN que dans un programme, pas dans le mode l'immédiat.

Par exemple, une opération mathématique que vous avez dû utiliser plusieurs fois dans cette ection est la conversion degré-radian. Ne serait-il pas agréable que l'ordinateur le fasse pour vous ?

Si vous modifiez l'exemple de programme que nous avons utilisé pour ATAN, vous verrez comment créer un DEF FN qui convertit les degrés en radians.

7 DEF FNR(X)=X/57.29577951 60 X=FNR(X): 110 AA=FNR(AA): AC=FNR(AC)

Effacez la ligne 120.

7 DEF FNR(X)=X/57.29577951 10 CLS 20 INPUT "QUEL EST LE COTÉ A (SA)"; SA:IF SA<=0 THEN 150 30 INPUT "QUEL EST LE COTÉ C (SC)"; SC:IF SC<=0 THEN 150 40 INPUT "QUEL EST L'ANGLE B (AB)"; AB:IF AB<=0 OR AB>=180 THEN 150 50 X=(180-AB) 'AA+AC=180-AB 60 X=FNR(X): 70 Y=((SA-SC)/(SA+SC))\*TAN(X/2) 80 Z=ATN(Y) 90 AA=(X/2)+(Z) 100 AC=(X/2)-(Z) 110 AA=FNR(AA): AC=FNR(AC) 130 PRINT "ANGLE A (AA) EST " AA "DÉGRÉES" 140 PRINT "ANGLE C (AC) EST " AC "DÉGRÉES":GOTO 20 150 PRINT "DÉSOLÉ, CE N'EST PAS UN TRIANGLE. RÉESSAYER": GOTO 20

Vous pouvez voir tout de suite combien cela économise en tapant, puisque vous n'avez dû entrer 57.29577951 qu'une seule fois. Chaque fois que FNR est appelé en cours d'utilisation, l'ordinateur insère automatiquement les valeurs que vous avez utilisées et effectue l'opération prescrite.

#### Programme faites-le vous même

Utilisez DEF FN pour :

- 1. Convertissez les radians en degrés.
- 2. Créez une fonction mathématique qui décompose les nombres.

1.) DEFFNR (X) = X\*57. 28577851

2.)

```
5 CLS
10 DEF FNC(X) = X^3
20 INPUT "QUEL NOMBRE VOULEZ-VOUS METTRE AU CUBE ";X
30 X=FNC(X)
40 PRINT X
50 FOR A = 1 TO 75
55 NEXT A
610 GOTO 20
```

## Voici un résumé des fonctions trigonométriques

Veuillez noter que les informations ci-dessus peuvent ne pas être exhaustives.



Quantity	Standard Formulas	BASIC Statement
Total Degrees of a Triangle	$180^\circ = A + B + C$	TTL = AA + AB + AC
Solve for Area Given Side a, Angles B and C	$A = 180 - (B + C)$ $a^{2} \sin B \cdot \sin C$	AA = 180 - (AE + AC) [then convert AA, AB and AC to radians]
and C	Area =2 sin A	$AREA = SA \diamond 2 \cdot SIN(AB) \cdot SIN(AC)/(2 \cdot SIN(AA))$
Given Sides a, b and c	$s = \frac{1}{2}(a+b+c)$	S = (SA + SB + SC)/2
	Area = $\sqrt{s(s-a)(s-b)(s-c)}$	AREA = SQR(S*(S-SA)*(S-SB)*(S-SC))
Law of Sines	$\frac{a}{b} = \frac{\sin A}{\sin B} \text{ or } a = \frac{\sin A}{\sin B} \cdot b$	$SA = (SIN(AA)/SIN(AB)) \cdot SB$
Law of Cosines	$a^{2} = b^{2} + c^{2} - 2bc \cdot \cos A \text{ or}$ $a = \sqrt{b^{2} + c^{2} - 2bc \cdot \cos A}$	$SA = SQR(SB \leftrightarrow 2 - SC \leftrightarrow 2 - 2 \cdot SB \cdot SC \cdot COS(AA))$
Law of Tangents	a-c tan 1/2(A-C)	REM Y = TAN((AA - AC)/2)
	$a+c = \frac{1}{\tan \frac{1}{2}(A+C)}$	$Y = (SA - SC)/(SA + SC) \cdot TAN((AA + AC)/2)$
9	$\tan \frac{1}{2}(A-C) = \frac{a-c}{a+c} \cdot \tan \frac{1}{2}(A+C)$	
Given Three Sides,	$s = \frac{1}{2}(a+b+c)$	S = (SA + SB + SC)/2
Solve for an Angle	$r = \sqrt{\frac{(s-a)(s-b)(s-c)}{s}}$	$R = SQR((S-SA) \cdot (S-SB) \cdot (S-SC)/S)$
	$A = 2 \arctan\left(\frac{r}{s-a}\right)$	$AA = 2 \cdot ATN(R/(S - SA))$
Quadratic	$ax^2 + bx + c = 0$	$REM A \cdot X \diamond 2 + B \cdot X + C = 0$
Equalions	$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$	$Z = B \Delta Z - 4 * A * C$ X1 = (-B + SOR(Z))/(2 * A)  'IF Z > = 0 X2 = (-B - SQR(Z))/(2 * A)  'IF Z > = 0
Algebraic	$(a^{x})^{y} = a^{xy}$	$Z = (A \ominus X) \ominus Y$ or $Z = A \ominus (X \cdot Y)$
Equations	$a^{x} = \frac{1}{a^{x}}$	$Z = A \Leftrightarrow (-X)$ or $Z = 1/(A \Leftrightarrow X)$
	$\log x^{y} = y \cdot \log x$	$Z = LOG(X \leftrightarrow Y)$ or $Z = Y \cdot LOG(X)$
	$\log xy = \log x + \log y$	$Z = LOG(X \cdot Y)$ or $Z = LOG(X) + LOG(Y)$
	$\log \frac{x}{y} = \log x - \log y$	Z = LOG(X Y) or $Z = LOG(X) - LOG(Y)$

Function	Function Expressed in Terms of Extended Color BASIC Functions. x is in radians.
SECANT	SEC(X) = 1/COS(X)
COSECANT	CSC(X) = 1/SIN(X)
COTANGENT	COT(X) = 1/TAN(X)
INVERSE SINE	$ARCSIN(X) = ATN(X/SQR(-X^*X + 1))$
INVERSE COSINE	$ARCCOS(X) = -ATN(X/SQR(-X^*X+1)) + 1.5708$
INVERSE SECANT	ARCSEC(X) = ATN(SQR(X*X - 1)) + (SGN(X) - 1)*1.5708
INVERSE COSECANT	$ARCCSC(X) = ATN(1/SQR(X^*X - 1)) + (SGN(X) - 1)^*1.5708$
INVERSE COTANGENT	ARCCOT(X) = -ATN(X) + 1.5708
HYPERBOLIC SINE	SINH(x) = (EXP(X) - EXP(-X))/2
HYPOBOLIC COSINE	COSH(X) = (EXP(X) + EXP(-X))/2
HYPERBOLIC TANGENT	$TANH(X) = -EXP(-X)/(EXP(X) + EXP(-X))^{*}2 + 1$
HYPERBOLIC SECANT	SECH(X) = 2/(EXP(X) + EXP(-X))
HYPERBOLIC COSECANT	CSCH(X) = 2/(EXP(X) - EXP(-X))
HYPERBOLIC COTANGENT	$COTH(X) = EXP(-X)/(EXP(X) - EXP(-X))^{*}2 + 1$
INVERSE HYPERBOLIC SINE	$ARGSINH(X) = LOG(X + SQR(X^*X + 1))$
INVERSE HYPERBOLIC COSINE	$ARGCOSH(X) = LOG(X + SQR(X^*X - 1))$
INVERSE HYPERBOLIC TANGENT	ARGTANH(X) = LOG((1 + X)/(1 - X))/2
INVERSE HYPERBOLIC SECANT	$ARGSECH(X) = LOG((SQR(-X^*X + 1) + 1)/X)$
INVERSE HYPERBOLIC COSECANT	$ARGCSCH(X) = LOG((SGN(X)^{*}SQR(X^{*}X + 1) + 1)/X)$
INVERSE HYPERBOLIC COTANGENT	ARGCOTH(X) = LOG((X + 1)/(X - 1))/2

Inverse Sine	-1 <x<1< th=""></x<1<>
Inverse Cosine	-1 <x<1< td=""></x<1<>
Inverse Secant	X < -or X > 1
Inverse Cosecant	X<-1 or X>1
Inverse Hyper. Cosine	X>1
Inverse Hyper. Tangent	X*X<1
Inverse Hyper. Secant	0 <x<1< td=""></x<1<>
Inverse Hyper, Cosecant	X<>0
Inverse Hyper. Cotangent	X*X>1

Certaines valeurs spéciales sont mathématiquement indéfinies, mais nos fonctions peuvent fournir des valeurs non valides :

TAN et SEC de 90 et 270 degrés COT et SCS de 0 et 180 degrés Les autres valeurs qui ne sont pas disponibles de ces fonctions sont :

ARCSIN(-1) = -PI/2ARCSIN(1) = PI/2ARCCOS(-1) = PIARCCOS(1) = 0ARCSEC(-1) = -PIARCSEC(1) = 0ARCCSC(-1) = -PI/2ARCCSC(1) = PI/2





Plus tôt, nous avons longuement discuté de la chaîne. C'est maintenant le temps pour plus d'informations sur les fonctions de chaîne de caractères de GW-BASIC.

## STRING\$

Zing devient STRING\$ ... et lorsque vous l'utilisez pour créer une chaîne de caractères, vous pouvez produire des graphiques, des tableaux et tout autre affichage textuel.

La syntaxe de STRING\$ est la suivante :

STRING\$ (longueur,caractère)

longueur est un nombre compris entre 0 et 255.

caractère est soit une expression de chaîne pour un caractère, soit une expression numérique pour un code ASCII. Si vous utilisez une constante de chaîne, placez-la entre guillemets.

Le nombre de caractères affichés dépend du nombre que vous spécifiez en longueur. Les caractères utilisés dépendent du caractère ou du code ASCII que vous spécifiez.

Par exemple:

5 CLS 10 X\$ = STRING\$(13,"\*") 20 PRINT X\$; "LIGNES"; X\$ 30 FOR X = 1 TO 1000: NEXT X 40 SCREEN 7 50 LINE (0,0)-(255,191) 60 LINE (0,191)-(255,0) 70 GOTO 70

La ligne 10 attribue à X\$ la valeur STRINGS (13,"\*") - une chaîne de 13 astérisques.

La ligne 20 indique à l'ordinateur d'afficher X\$, puis le mot LIGNES, suivi à nouveau de X\$. Puisque X\$ est égal à 13 astérisques ("\*"). ces caractères sont imprimés avant et après LIGNES.

Quoi? Vous voulez embellir encore plus le titre! Très bien, ajoutez cette ligne:

```
15 Y$ = STRING$(31,42): PRINT Y$
```

Cette fois, vous dites à l'ordinateur d'afficher le caractère représenté par le code ASCII 42. Et, comme vous l'avez probablement deviné, le code ASCII 42 représente un astérisque.

## Programme faites-le vous même

Avez-vous déjà rédigé des listes pour cocher les tâches que vous ou d'autres personnes devez effectuer ? En utilisant STRINGS, écrivez un programme qui crée une liste de vérification.

5 CLS 10 X\$ = STRING\$(30,"-") 20 PRINT TAB(7) "MATH" TAB(14) "ORAL" TAB(22) "LECTURE" 30 PRINT X\$ 40 PRINT "BILL" TAB(8) "X" 50 PRINT X\$ 60 PRINT "SUE" TAB(16) "X" 70 PRINT X\$ 80 PRINT "JON" TAB(16) "X" 90 PRINT X\$ 100 PRINT X\$ 100 PRINT X\$ 100 PRINT X\$ 120 GOTO 120

#### Je pense que je vois quelques cordes devant ! (INSTR)

Si vous souhaitez rechercher une deuxième chaîne dans une chaîne, utilisez INSTR.

La syntaxe d'INSTR est :

INSTR (Position, recherche-chaîne, cible)

position spécifie la position dans la chaîne de recherche à laquelle la recherche doit commencer (0 à 255). Si vous omettez la position, l'ordinateur commence automatiquement au premier caractère.

recherche-chaîne est la chaîne à rechercher.

cible est la chaîne à rechercher.

INSTR renvoie un 0 si l'une des conditions suivantes est vraie :

La position est supérieure au nombre de caractères dans la recherche-géniteur. recherche-chaîne est nulle. Il ne peut pas trouver la cible.

Regardez comment STR fonctionne dans le programme suivant :

5 CLEAR 500 10 CLS 15 INPUT "TEXTE A RECHERCHER";S\$ 20 INPUT "TESTE CIBLE";T\$ 25 C=0: P=1 'P = POSITION 30 F = INSTR(P,S\$,T\$) 35 IF F=0 THEN 60 40 C=C+1 45 PRINT LEFT\$ (S\$ ,F-1 )+STRING\$(LEN(T\$), CHR\$(42)) + RIGHT\$(S\$,LEN(S\$) -F- LEN(T\$)+1 ) 50 P=F+LEN (T\$) 55 IF P<=LEN(S\$)-LEN(T\$)+1 THEN 30 60 PRINT "TROUVER"; C;"OCCURRENCES"

Ce qui suit est un exemple d'exécution. Cependant, vous pouvez saisir le texte dont vous avez besoin.

TEXTE A RECHERCHER? ALLO LA MONDE TESTE CIBLE? LA ALLO \*\* MONDE TROUVER 1 OCCURRENCES Ok

Écrivez un programme qui renvoie la première et la seconde occurrences du B dans ABCDEB.

5 CLS 10 X\$ = "ABCDEB" 20 Y\$="B" 30 PRINT INSTR(X\$,Y\$) ; INSTR(4,X\$,Y\$)

Le programme de stockage de données suivant contient une liste de diffusion de noms et d'adresses. C'est un moyen simple de stocker des informations. Notez que nous avons économisé de l'espace de stockage en ne mettant pas d'espaces entre les mots.

Cela rend la lecture difficile pour vous, mais pas pour l'ordinateur.

Notez également que nous attribuons un astérisque ("") aux codes postaux afin que l'ordinateur ne les confond pas avec les numéros de rue.

Dans ce cas, nous recherchons les noms et adresses de toutes les personnes qui vivent dans la zone spécifiée par le code postal 650. Par conséquent, \*650 est la cible (A\$).

10 A\$ = "\*650" 15 X = 1 20 X\$ = "JAMES SMITH,6550HARISON,DALLASTX\*75002:SUE SIM,RT3.GRAVIOSMO\*65084; LYDIA LONG,3445SMITHST,ASBURYNJ\*32004:BOB STRONG,BOX 60,EDMONTONALBERTACA:TIMMY DUNTON, PIERMONTMO\*67078" 50 P = INSTR(X\$,A\$): PRINT P

Exécutez le programme. Votre écran affiche : 61 OK

Cela vous indique que la chaîne contient un nom et une adresse dont vous avez besoin.

Ne changez jamais de cheval dans la corde médiane (MID\$)

L'instruction MID\$ vous offre une puissante capacité d'édition de chaînes en vous permettant de remplacer une partie d'une chaîne par une autre. La syntaxe de MID\$ est la suivante :

MID\$ (ancienne chaîne, position, longueur) = nouvelle chaîne

ancienne chaîne est le nom de variable de la chaîne à remplacer.

position est le numéro de la position du premier caractère à être changé.

longueur est un nombre de caractères à remplacer.

Si vous omettez longueur, l'ordinateur remplace tout ancienne chaîne, nouvelle chaîne est la chaîne qui remplace la partie spécifiée de ancienne chaîne.

Remarque : Si nouvelle chaîne contient moins de caractères que la longueur spécifiée, l'ordinateur remplace la totalité de nouvelle chaîne. nouvelle chaîne a toujours la même longueur que ancienne chaîne.

Pour voir ce que nous voulons dire, exécutez ce programme :

5 CLS 10 A\$ = "KANSAS CITY. MO" 20 MID\$(A\$,14)="KS" 30 PRINT A\$

La ligne 10 attribue à A\$ la valeur KANSAS CITY, MO. Ensuite, la ligne 20 indique à l'ordinateur d'utiliser MID\$ pour remplacer une partie de l'ancienne chaîne (A\$) par KS, à partir de la position 14.

Changez la position 14 en 8 et exécutez le programme. Le résultat est:

KANSAS CITY. MO

Ajoutez maintenant l'option de longueur à la ligne 20 :

20 MID\$(A\$,14,2)="KS"

Notez que cela n'affecte pas le résultat puisque nouvelle chaîne et ancienne chaîne contiennent tous les deux deux caractères.

Changez la longueur en 1:

20 MID\$(A\$,14,1)="KS"

L'ordinateur remplace un seul caractère dans ancienne chaîne, en utilisant le premier caractère dans KS.

Vous constaterez que MID\$ est doublement efficace lorsqu'il est utilisé avec INSTR. En utilisant les deux, vous pouvez "rechercher et détruire" du texte. recherches INSTR ; MID\$ modifie ou "détruit".

Le programme suivant illustre cela :

5 CLS 10 INPUT "ENTRER UN MOIS ET UN JOUR (MM/DD) ";X\$ 20 P = INSTR(X\$,"/") 30 IF P = 0 THEN 10 40 MID\$(X\$,P,1)= "-" 50 PRINT X\$ " EST PLUS FACILE À LIRE. N'EST-CE PAS? "

Dans ce programme, IN5TR recherche une barre oblique (/). Lorsqu'il en trouve une, MID\$ la remplace par un trait d'union (-).

Imaginez que vous travailliez dans une compagnie de téléphone à l'époque où les centraux téléphoniques passaient de caractères alpha à 10 caractères numériques.

Écrivez un programme qui utilise MID\$ pour remplacer tous les échanges alpha par des nombres. Assurez-vous de libérer suffisamment d'espace pour les chaînes ou vous obtiendrez une ERREUR.

5 CLS 10 DIM TBL\$(26) 20 FOR I=0 TO 25 30 READ TBL\$ (I) : NEXT I 40 PRINT "ENTRER LE NUMÉRO DE TÉLÉPHONE À L'ANCIENNE" 50 INPUT N\$ 60 IF N\$=" " THEN 40 70 FOR I=1 TO LEN(N\$) 80 C = MID (N\$,I,1) 90 IF C\$< "A" OR C\$ >"Z" THEN 120 100 C\$=TBL\$(ASC(C\$)-65) 110 MID\$(N\$,I)=-C\$ 120 NEXT I 130 PRINT "LE NOUVEAU STYLE = ";N\$ 140 REM A B C D E F 150 DATA "2", "2", "2", "3", "3", "3" 160 REM G H I J K L

170 DATA "4", "4", "4", "5", "5", "5" 180 REM M N 0 P Q R 190 DATA "6", "6", "6", "7", "Q", "7" 200 REM S T U V W X 210 DATA "7", "8", "8", "8", "9", "9" 220 REM Y Z 230 DATA "9", "Z"

## Entre dans une porte et sort par l'autre



Les instructions d'entrée/sortie vous permettent d'envoyer des données (du clavier à l'ordinateur, de l'ordinateur à l'écran et de l'ordinateur à l'imprimante. Ces fonctions sont principalement utilisées à l'intérieur des programmes pour saisir des données et afficher des résultats et des messages.

## Un lecteur de ligne (LINE INPUT)

La première instruction d'entrée/sortie est LINE INPUT, pour saisir une ligne entière (jusqu'à 255 caractères) à partir du clavier dans une variable de chaîne, en ignorant les délimiteurs.

Sa syntaxe est la suivante :

Pour saisir une ligne entière (jusqu'à 255 caractères) à partir du clavier dans une variable de chaîne, en ignorant les délimiteurs.

LINE INPUT [;][chaîne d'invite;]variable de chaîne

la chaîne d'invite est un littéral de chaîne, affiché à l'écran, qui permet à l'utilisateur de saisir pendant l'exécution du programme.

Un point d'interrogation n'est imprimé que s'il fait partie d'une chaîne d'invite.

La variable de chaîne accepte toutes les entrées depuis la fin de l'invite jusqu'au retour de charriot. Les blancs de fin sont ignorés.

LINE INPUT est presque identique à l'instruction INPUT, sauf qu'elle accepte des caractères spéciaux (tels que des virgules) dans l'entrée de l'opérateur pendant l'exécution du programme.

Si une séquence de saut de ligne/retour de charriot (cette commande uniquement) est rencontrée, les deux caractères sont saisis et renvoyés en écho. La saisie des données se poursuit.

Si LINE INPUT est immédiatement suivi d'un point-virgule, appuyer sur la touche RETURN ne déplacera pas le curseur à la ligne suivante.

Une entrée de ligne peut être échappée en tapant CTRL-C. GW-BASIC revient au niveau de commande et affiche OK.

Taper CONT reprend l'exécution à la ligne LINE INPUT.

Avec LINE INPUT, vous pouvez saisir des données de chaîne sans vous soucier d'inclure accidentellement des délimiteurs tels que des virgules, des guillemets et des deux-points.

L'ordinateur accepte tout. En effet, certaines situations nécessitent que vous saisissiez des virgules, des guillemets et des blancs de début dans le cadre des données.

Exemples:

LINE INPUT X\$

vous permet d'entrer X\$ sans afficher d'invite.

LINE INPUT "LAST NAME, FIRST NAME? ";N\$

affiche l'invite « NOM, PRENOM ? » et saisit les données.

Les virgules ne terminent pas la chaîne d'entrée. Notez que l'invite inclut le point d'interrogation et l'espace suivant.

Pour mieux comprendre LINE INPUT, entrez et exécutez le programme suivant :

10 CLEAR 300: CLS 20 PRINT TAB(8); "ÉNONCÉ D'ENTRÉE DE LIGNE":PRINT 30 PRINT: PRINT "\*\*\* Entrez du texte \*\*\*" 40 ' \*\*\* Obtenez la chaîne, puis imprimez-la \*\*\* 50 A\$ = "" 'FIXE A\$ À CHAÎNE NULLE 60 LINE INPUT "==> "; A\$ 70 IF A\$ = "" THEN END' SI TOUJOURS NUL, STOP! 80 PRINT A\$ 90 GOTO 50

## Impression personnalisée (PRINT USING)

Pour imprimer des chaînes ou des nombres en utilisant un format spécifié.

Syntaxe:

PRINT USING string expressions; list of expressions[;]

Commentaires:

string expressions est une chaîne littérale ou une variable composée de caractères de formatage spéciaux. Les caractères de formatage déterminent le champ et le format des chaînes ou des nombres imprimés.

list of expressions se compose de la chaîne ou des expressions numériques séparées par des points-virgules.

## Champs de chaîne

Les trois caractères suivants peuvent être utilisés pour formater le champ de chaîne :

! Spécifie que seul le premier caractère de la chaîne doit être imprimé.

## \n espaces\

Spécifie que 2+n caractères de la chaîne doivent être imprimés.

Si les barres obliques inverses sont saisies sans espaces, deux caractères sont imprimés ; si les barres obliques inverses sont tapées avec un espace, trois caractères sont imprimés, et ainsi de suite.

Si la chaîne est plus longue que le champ, les caractères supplémentaires sont ignorés.

Si le champ est plus long que la chaîne, la chaîne est justifiée à gauche dans le champ et complétée par des espaces à droite.

Par exemple:

10 A\$="LOOK": B\$="OUT"

30 PRINT USING "!"; A\$; B\$

40 PRINT USING"\\"; A; B

50 PRINT USING"\\"; A\$; B\$;"!!"

RUN LO LOOKOUT LOOK OUT!! &

Spécifie un champ de chaîne de longueur variable. Lorsque le champ est spécifié avec &, la chaîne est sortie exactement comme l'entrée.

Par exemple:

10 A\$="LOOK": B\$="OUT"

20 PRINT USING "!"; A\$

30 PRINT USING "&"; B\$

RUN LOUT

## **Champs numériques**

Les caractères spéciaux suivants peuvent être utilisés pour formater le champ numérique :

#

Un signe dièse est utilisé pour représenter chaque position de chiffre. Les postes de chiffres sont toujours pourvus.

Si le nombre à imprimer comporte moins de chiffres que les positions spécifiées, le nombre est justifié à droite (précédé d'espaces) dans le champ.

Un point décimal peut être inséré à n'importe quelle position dans le champ.

Si la chaîne de format spécifie qu'un chiffre doit précéder la virgule décimale, le chiffre est toujours imprimé (comme 0 si nécessaire). Les nombres sont arrondis si nécessaire.

Par exemple:

PRINT USING "##.##";.78 0.78 PRINT USING "###.##";987.654 987.65 PRINT USING "##.##" ;10.2, 5.3, 66.789, .234 10.20 5.30 66.79 0.23

Dans le dernier exemple, trois espaces ont été insérés à la fin de la chaîne de format pour séparer les valeurs imprimées sur la ligne.

+ Un signe plus au début ou à la fin de la chaîne de format entraîne l'impression du signe du nombre (plus ou moins) avant ou après le nombre.

Un signe moins à la fin du champ de format entraîne l'impression des nombres négatifs suivis d'un signe moins.

Par exemple:

PRINT USING"+##.##";-68.95, 2.4, 55.6, -9 -68.95 +2.40 +55.60 -0.90 PRINT USING"##.##-";-68.95, 22.449, -7.01 68.95 22.45 7.01-

\*\*

Un double astérisque au début de la chaîne de format entraîne le remplissage des espaces de début du champ numérique par des astérisques. Le \*\* spécifie également deux positions de chiffres supplémentaires.

Par exemple:

PRINT USING "\*\*#.#";12.39, -0.9, 765.1 \*12.4\* -09 765.1

## \$\$

Un signe de dollar double au début de la chaîne de format entraîne l'impression d'un signe dollar immédiatement à gauche du nombre formaté. Le \$\$ spécifie deux positions de chiffres supplémentaires, dont l'une est le signe de dollar. Le format exponentiel ne peut pas être utilisé avec \$\$. Les nombres négatifs ne peuvent pas être utilisés à moins que le signe moins soit vers la droite.

Par exemple:

PRINT USING "\$\$###.##";456.78 \$456.78

\*\*\$

Le \*\*\$ au début d'une chaîne de format combine les effets des deux symboles ci-dessus. Les espaces de début sont remplis d'astérisques et un signe dollar est imprimé avant le nombre. \*\*\$ spécifie trois positions de chiffres supplémentaires, dont l'une est le signe dollar.

Par exemple:

PRINT USING "\*\*\$##.##" ;2.34 \*\*\*\$2.34

,

Une virgule à gauche du point décimal dans la chaîne de format entraîne l'impression d'une virgule à gauche de chaque troisième chiffre à gauche du point décimal. Une virgule à la fin de la chaîne de format est imprimée dans le cadre de la chaîne.

PRINT USING "####.##";1234.5 1234.50

#### ^ ^ ^ ^

Quatre carets peuvent être placés après les caractères de position des chiffres pour spécifier le format exponentiel. Les quatre carets permettent d'imprimer  $\mathbf{E}+\mathbf{x}\mathbf{x}$ .

Toute position de la virgule décimale peut être spécifiée. Les chiffres significatifs sont justifiés à gauche et l'exposant est ajusté. À moins qu'un signe + ou un signe + ou - à la fin ne soit spécifié, une position de chiffre est utilisée à gauche de la virgule pour imprimer un espace ou un signe moins.

Par exemple:

```
PRINT USING "##.##^^^^";234.56
2.35E+02
PRINT USING ".####^^^^-";888888
OK
PRINT USING "+.##^^^^";123
+.12E+03
```

Notez que dans les exemples ci-dessus, la virgule n'est pas utilisée comme délimiteur avec le format exponentiel.

\_

Un trait de soulignement dans la chaîne de format entraîne la sortie du caractère suivant sous forme de caractère littéral.

Par exemple:

PRINT USING "\_!##.##\_!";12.34 !12.34!

Le caractère littéral lui-même peut être un trait de soulignement en plaçant "\_" dans la chaîne de format.

%

Un signe de pourcentage est imprimé devant le nombre si le nombre à imprimer est supérieur au champ numérique spécifié.

Si l'arrondi fait que le nombre dépasse le champ, un signe de pourcentage est imprimé devant le nombre arrondi.

Par exemple:

```
PRINT USING "##.##";111.22
%111.22
PRINT USING ".##"';.999
%1.00
```

Si le nombre de chiffres spécifié dépasse 24, une erreur "Illegal function call" se produit.
Exemple de programme

5 CLS 10 INPUT "REVENU ";I 15 INPUT "DÉPENSES ";E 20 N = I -E 'GAIN OU PERTE NET 25 A\$ = "\$\$####.##" 30 B\$ = "\$\$####.##" 35 C\$ = "+\$\$####.##" 40 CLS: PRINT "RAPPORT MENSUEL SUR LA SITUATION ÉCONOMIQUE" 45 PRINT STRING\$(40,"-") 50 PRINT " REVENU", 55 PRINT "DÉPENSES", 60 PRINT "TOTAL (+) OU (-)" 65 PRINT STRING\$(40,"-") 70 PRINT USING A\$; I; 75 PRINT TAB(13) USING B\$; E; 80 PRINT TAB(30) USING C\$; N 90 PRINT:PRINT

## POS

POS est une fonction d'entrée/sortie qui renvoie la position actuelle du curseur sur l'écran.

Voici sa syntaxe :

POS(c)

c est un argument factice.

PRINT POS(X)

Une façon d'utiliser POS consiste à désactiver la fonction "wrap-around" sur l'écran ou l'imprimante. Faire cela empêche les mots d'être cassés au milieu. D'autre part, il raccourcit nécessairement la longueur de la ligne.

Exécutez le programme suivant pour voir POS au travail :

10 CLS 20 WIDTH 80 30 A\$=INKEY\$:IF A\$=""THEN GOTO 30 ELSE PRINT A\$; 40 IF POS(X)>10 THEN PRINT CHR\$(13); 50 GOTO 30

Provoque un retour de charriot après l'impression du 10e caractère sur chaque ligne de l'écran.

## Périphérique

Avez-vous déjà pensé à votre écran vidéo comme un périphérique "de sortie" et à votre clavier comme un périphérique "d'entrée" ?

#### Temps après la minuterie (TIMER)

Pour renvoyer des nombres à virgule flottante simple précision représentant le nombre de secondes écoulées depuis minuit ou la réinitialisation du système.

Votre ordinateur dispose également d'un "minuteur" intégré qui mesure le temps en soixantièmes de seconde (environ). Au moment où vous allumez l'ordinateur, la minuterie commence à compter à zéro.

A tout instant, vous pouvez voir le décompte de la minuterie en utilisant la fonction TIMER

Taper:

## PRINT TIMER

Les fractions de secondes sont calculées au degré le plus proche possible. La minuterie est en lecture seule.

Une horloge est intégrée à chaque ordinateur, fonctionnant à une fréquence définie. Cette horloge est, bien, cadencée à des cycles par seconde, avec des unités en hertz, et est une mesure de la vitesse du processeur de l'ordinateur.

Mais l'horloge d'un ordinateur peut aussi, avec un peu de programmation astucieuse, donner l'heure. GW-BASIC rend cela facile : une variable de chaîne prédéfinie, appelée TIME\$, est instantanément déclarée et initialisée chaque fois que l'interpréteur est chargé. TIME\$, lorsqu'il est imprimé à l'écran, affiche l'heure au format suivant :

## HEURE : MINUTE : SECONDE

où les heures sont en temps militaire (c'est-à-dire l'horloge de 24 heures). Par exemple, 14h53 et 22 secondes apparaîtraient comme suit :

#### 16:53:22

Ce serait bien si nous avions quelques options, cependant.

Par exemple : et si nous voulions voir uniquement les minutes ou les secondes ? Et si nous voulions convertir l'heure militaire au format 12 heures ?

L'utilisation de la fonction MID\$ sur la variable TIME\$ nous permet d'extraire les heures, les minutes et/ou les secondes de l'horloge interne.

Pour convertir l'horloge de 24 heures en une horloge de 12 heures, l'heure doit d'abord être transformée en une variable numérique - la fonction VAL de la line 70 accomplit cette tâche. Ensuite, une série d'instructions conditionnelles (lignes 75 à 100) dispense de l'heure milliaire.

Le programme s'exécute indéfiniment, à moins que vous n'appuyiez sur la touche Échap pour le terminer. La variable INKEY\$ sollicite l'entrée de l'utilisateur sur la line 110 ; si autre chose que CHR\$(27) (une fonction extraite de la table ASCII, dans laquelle 27 = Échap) est pressé sur le clavier, le programme revient à la line 30.

10 KEY OFF 20 CLS 30 LOCATE 1,1:PRINT "VARIOUS FORMS OF THE TIME:" 40 LOCATE 2,1:PRINT TIME\$ 50 LOCATE 3,1:PRINT "SECONDS: ";MID\$(TIME\$,7,2) 60 LOCATE 4,1:PRINT "MINUTES: ";MID\$(TIME\$,4,2) 70 HOUR=VAL(MID\$(TIME\$,1,2)) 75 IF HOUR=0 THEN HOUR=12 80 IF HOUR>12 THEN HOUR=HOUR-12:LOCATE 5,1:PRINT "HOUR: ";HOUR; "PM" 90 IF HOUR>12 THEN LOCATE 5,1:PRINT "HOUR: ";HOUR ;"PM" 100 IF HOUR<12 THEN LOCATE 5,1:PRINT "HOUR: ";HOUR ; "AM" 110 I\$=INKEY\$ 120 IF I\$<>CHR\$(27) THEN 30

Maintenant que nous pouvons manipuler la variable de chaîne TIME\$, la prochaine étape logique serait de créer une horloge analogique à l'écran. En effet, plus tard, nous le ferons.

#### Constantes hexadécimales et octales

#### HEX\$

Pour renvoyer une chaîne qui représente la valeur hexadécimale de l'argument numérique.

v=HEX(x)

HEX\$ convertit les valeurs décimales comprises entre -32768 et +65535 en une expression de chaîne hexadécimale comprise entre 0 et FFFF.

Les nombres hexadécimaux sont des nombres en base 16, plutôt qu'en base 10 (nombres décimaux). L'annexe C et l'annexe G du guide de l'utilisateur GW-BASIC(<u>PC-BASIC</u>) contiennent plus d'informations sur les hexadécimaux et leurs équivalents.

x est arrondi à un entier avant l'évaluation de HEX\$(x). Voir la fonction OCT\$ pour les conversions octales.

Si x est négatif, la forme de complément à 2 (binaire) est utilisée.

10 CLS: INPUT "TAPEZ UN NOMBRE DÉCIMAL";X 20 A\$=HEX\$(X) 30 PRINT X "DÉCIMAL EST "A\$" HEXADÉCIMAL" RUN TAPEZ UN NOMBRE DÉCIMAL? 32 32 DÉCIMAL IS 20 HEXADÉCIMAL 10 CLS:P=0 20 FOR X= 32 TO 255 25 IF P=60 THEN P=0 30 PRINT TAB(P)X;"HEX= "HEX\$(X); 35 P=P+15 40 NEXT X

# OCT\$

Pour convertir une valeur décimale en une valeur octale.

OCT\$(x)

x est arrondi à un nombre entier avant l'évaluation de OCT\$(x).

Cette instruction convertit une valeur décimale comprise entre -32768 et +65535 en une expression de chaîne octale.

Les nombres octaux sont des nombres en base 8 plutôt qu'en base 10 (nombres décimaux).

10 PRINT OCT\$(18)

RUN 22

Le décimal 18 est égal à l'octal 22.