



QL

QL Abacus

CHAPITRE UN

Quelques mots sur QL Abacus

QL Abacus est un tableur. Parmi ses cas d'emploi, citons : la planification, les tableaux de statistiques, les bilans ou encore le stockage et l'analyse d'informations. Ces informations sont présentées sur une grille qui est en fait un immense tableau composé de 255 rangées et de 64 colonnes. Ce que vous voyez sur l'écran est une fenêtre affichant une partie de la grille. Les intersections des rangées et des colonnes forment plus de 16 000 cases ou cellules. Dans toute cellule ou bloc de cellules, vous pouvez introduire un texte ou stocker des chiffres ou des données.

Le grand avantage de QL Abacus relève de l'emploi de règles ou formules qui mettent en correspondance des blocs, des rangées ou des colonnes et même des cases données de la grille. Ainsi, les informations saisies en un endroit peuvent être immédiatement évaluées et représentées autre part.

Par exemple, vous choisirez douze colonnes pour représenter les mois de l'année avant d'inscrire les données des ventes sur une rangée intitulée « Ventes. Les deux rangées suivantes peuvent prévoir des formules permettant de calculer le coût des produits vendus (disons, un pourcentage des ventes augmenté d'un coût fixe) et les bénéfices. Les bénéfices mensuels sont alors calculés automatiquement chaque fois que vous tapez un chiffre de ventes. Les totaux annuels peuvent également être additionnés au moyen d'une autre formule; en cas de changement dans les ventes au mois de mars par exemple, vous obtenez alors sans délai un tableau totalement différent des résultats de l'année. QL Abacus procède automatiquement à tous les calculs.

Grâce aux commandes **EXP/IMP** de vos programmes QL, vos données sur QL Abacus peuvent être représentées sous forme de graphique ou de tableau.

QL Abacus est en fait pratiquement un langage de programmation visuel, mais très simple d'emploi. Il vous permet de manipuler du texte, des données ou des formules et d'utiliser des instructions d'entrée et de sortie ou des variables de texte.

QL Abacus est là pour vous aider. Donc, si vous êtes pris de doute appuyez sur la touche aide **F1**. Enfin, n'oubliez pas également que, si vous lancez une commande par erreur ou si vous changez d'avis en cours de route, vous avez toujours la possibilité de faire marche arrière en appuyant sur la touche **ESC**.

CHAPITRE 2

PREMIERS PAS

CHARGEMENT DE QL ABACUS

Pour le chargement de QL Abacus, conformez-vous aux consignes données dans la section intitulée « Présentation des logiciels du QL » Une fois QL ABACUS chargé, vous pouvez retirer la cartouche programme.

QL ABACUS ne fera d'accès sur la cartouche programme que dans 2 cas :

- Juste avant d'imprimer un document,
- Lorsque l'on fait appel au fichier d'aide.

Il faut donc dans ces 2 cas remettre la cartouche programme dans MDV1. Le reste du temps, les 2 Microdrives sont utilisables pour les documents.

Par défaut QL ABACUS considère que vous vous adressez à MDV2. Si vous désignez un document se trouvant sur MDV1, préfixez simplement son nom par « MDV1_ ».

Dès le programme chargé, vous verrez apparaître :

```
CHARGEMENT DE QL ABACUS
      Tableur
      version x.xx
      Copyright ©1984 PSION Ltd.
      Tous droits réservés
```

où x.xx est le numéro de version du logiciel, p.ex. 2.00.

Le message de copyright reste affiché pendant quelques secondes mais vous pouvez le faire disparaître instantanément en appuyant sur une touche quelconque.

Les informations obtenues en appuyant sur la touche **F1** ne sont pas chargées dans la mémoire de l'ordinateur avec le programme. Elles sont uniquement extraites de QL Abacus selon les besoins. Il est donc important de ne pas enlever la cartouche QL Abacus du Microdrive 1 si vous envisagez d'utiliser cette touche **AIDE (F1)**.

Une fois QL Abacus chargé, votre écran devrait se présenter comme indiqué sur la Figure 2.1. Nous appellerons cet affichage l'écran de saisie.

PRESENTATION SUR L'ECRAN

QL Abacus peut afficher 40, 64 ou 80 caractères par ligne. Si vous utilisez un téléviseur normal, la lisibilité du texte ne sera pas très bonne avec 80 caractères par ligne. Dans ce cas, descendez à 64 ou 40. Avec 64 ou 80 caractères par ligne, l'écran de saisie reste pratiquement le même. Avec 40 caractères par ligne, il prend l'aspect de la Figure 2.2.

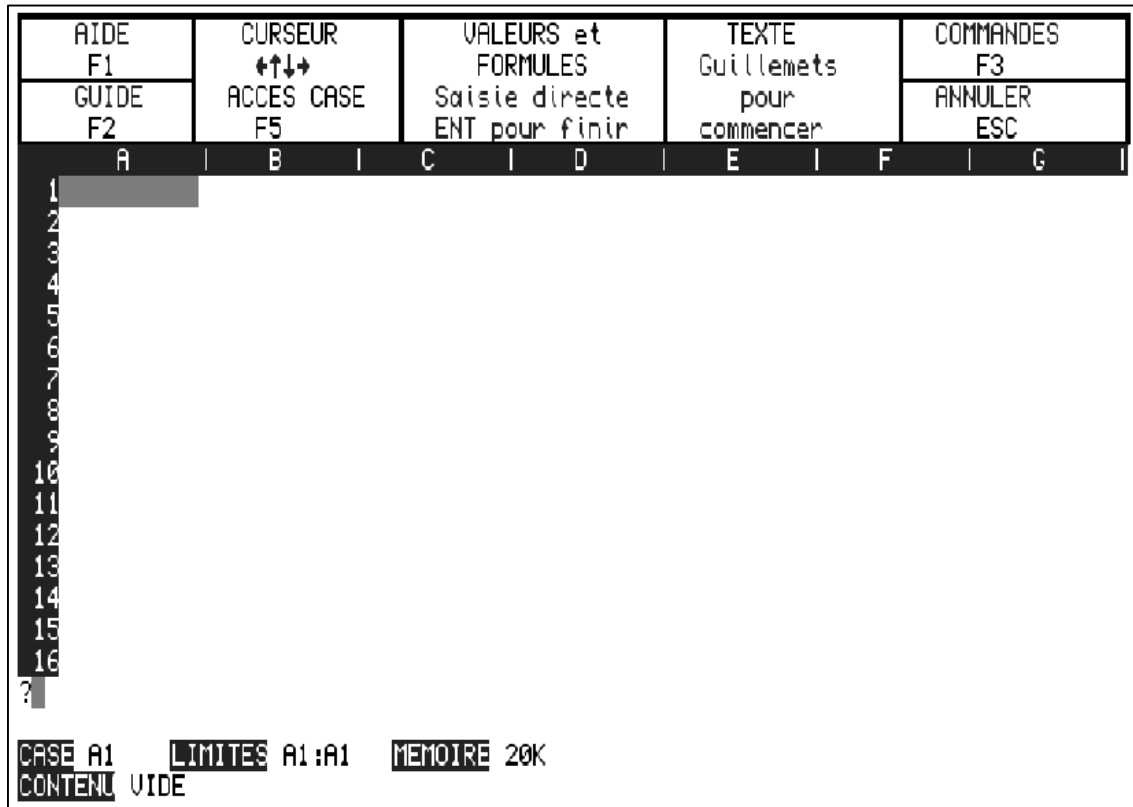


Figure 2.1 E écran de saisie sur un moniteur (80 caractères).

Pour commencer, QL Abacus choisit automatiquement 64 ou 80 caractères, suivant que vous avez appuyé sur F1 ou F2 à la mise sous tension de votre QL.

QL Abacus fonctionne exactement de la même façon dans les trois modes d'affichage. Par souci de simplicité, la plupart des schémas de ce Manuel représentent un écran de 80 caractères.



Figure 2.2 L'écran de saisie - 40 caractères.

La fenêtre.

La zone centrale de l'écran, *la fenêtre*, présente une section de la grille.

Au sommet de la fenêtre se situe une ligne comportant un certain nombre de lettres. Ces lettres identifient les colonnes verticales de la grille, soit, dans notre cas, les colonnes A, B, C, etc. La fenêtre est également bordée de chiffres de 1 à 15. Ces chiffres identifient les rangées de cases.

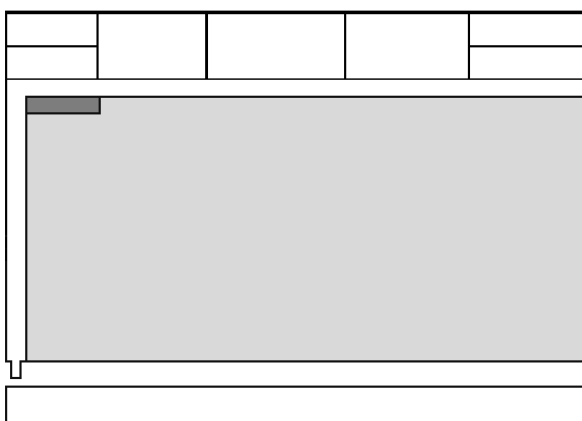


Figure 2.3 La fenêtre.

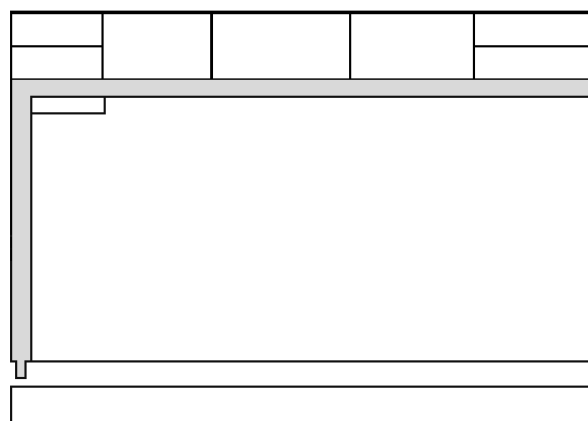


Figure 2.4 Les références de la grille.

Une case est donc identifiée par une lettre et un chiffre ; c'est ce que nous appellerons une *référence de case*. Par exemple, A1 se rattache à la cellule située à l'intersection de la colonne A et de la rangée 1 (en haut à gauche de la fenêtre).

Comme vous l'aurez remarqué, cette case diffère des autres en ce qu'elle renferme un rectangle rouge. Ce rectangle, intitulé curseur, repère la *cellule en cours*, c'est à dire celle qui reçoit les données que vous tapez.

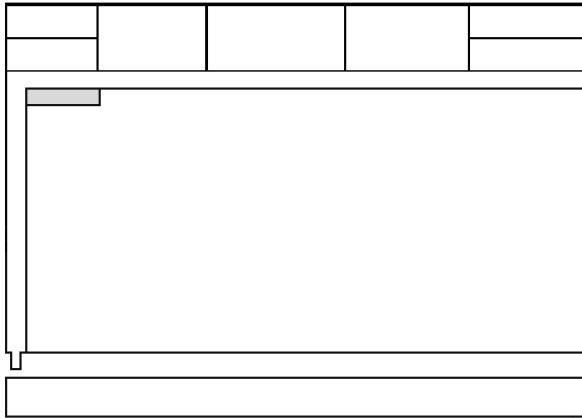


Figure 2.5 Le curseur.

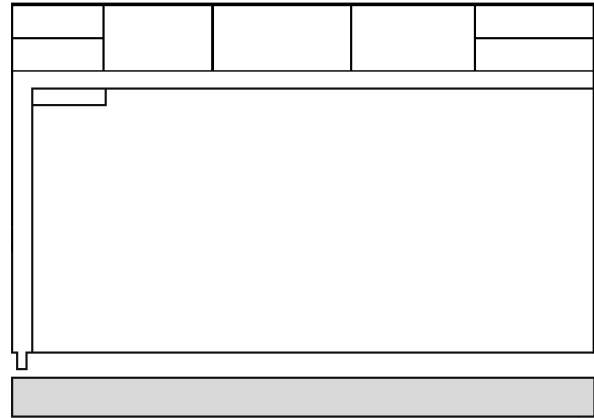


Figure 2.6 La zone d'état.

La zone d'état

La *zone d'état*, qui occupe la partie inférieure de l'écran, vous renseigne sur l'état de la grille en cours.

Elle indique la référence de la case en cours et son contenu. Cette case est vide quand vous venez de charger le programme. La zone d'état vous signale en outre la surface de grille que vous avez utilisée (en vous donnant la référence de la case située à l'angle inférieur droit de cette section) ainsi que la capacité de mémoire qu'il vous reste.

La zone de contrôle

La zone de contrôle vous rappelle la fonction des touches communes aux quatre logiciels QL : F1 pour demander de l'aide, F2 pour faire disparaître/réapparaître le guide, F3 pour sélectionner une commande et ESC pour abandonner une opération en cours. Elle explique de plus trois fonctions propres à QL Abacus, à savoir :

- comment déplacer le curseur;
- comment entrer des données ou une formule;
- comment taper un texte.

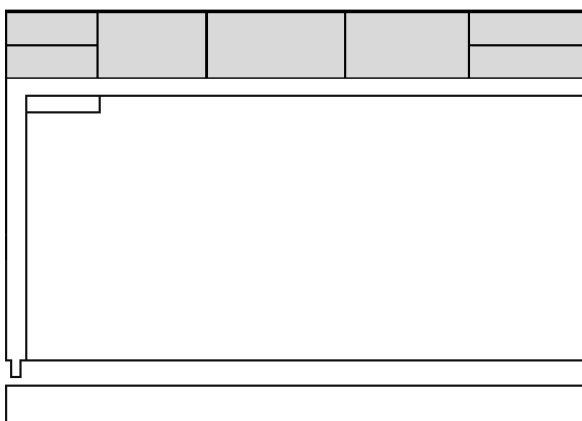


Figure 2.7 La zone de contrôle.

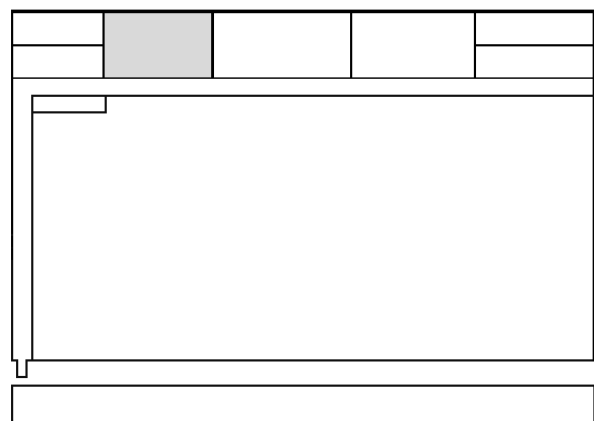


Figure 2.8 Déplacement du curseur.

LE CURSEUR

Quatre touches fléchées assurent le déplacement du curseur sur la grille. Appuyez sur la flèche de droite ; le curseur se déplace d'une colonne à droite et B1 est maintenant la case en cours. Appuyez ensuite sur la flèche gauche ; le curseur revient dans la case A1. Si vous appuyez de nouveau sur la flèche gauche, il ne se passera rien puisque vous êtes à l'extrême gauche de la grille.

Essayez maintenant de déplacer le curseur à l'extrême droite de la grille. Vous remarquerez que, quand vous appuyez sur la flèche droite, ce n'est pas le curseur qui se déplace, mais les lettres inscrites au sommet de la fenêtre qui défilent. Donc, quand le curseur doit sortir de la zone affichée à l'écran, la fenêtre se déplace le long de la grille afin de maintenir le curseur en vue.

Vous n'utiliserez ces touches fléchées que pour déplacer le curseur d'une ou deux cases. Mais elles sont très lentes quand il s'agit de balayer la grille entière. Dans un tel cas, il est plus pratique d'aller directement à la case recherchée. Pour ce faire, il suffit d'appuyer sur la touche **F5** pour obtenir l'option accès à et alors taper la référence de la case avant d'appuyer sur **ENTREE**.

Exemple : demandez à QL Abacus de déplacer le curseur à la case D11. Appuyez tout d'abord sur **F5**. Juste sous la fenêtre, vous voyez alors apparaître : **accès à> A1**. QL Abacus vous propose ainsi de faire passer le curseur à l'angle gauche supérieur de la grille. Si vous acceptez sa suggestion (il vous suffit d'appuyer sur **ENTREE**), le curseur s'y rendra automatiquement. Mais, pour déplacer le curseur à une autre case, tapez sa référence, soit, dans notre cas :

d11

et appuyez sur **ENTREE**. Peu importe que vous tapiez « d » en minuscules ou en majuscules; QL Abacus accepte les deux. La référence de case que vous venez d'introduire remplace celle proposée par QL Abacus et le curseur s'y rend directement.

Au moyen de cette même option, ramenez maintenant le curseur à l'angle supérieur gauche. Vous pouvez cette fois accepter sa suggestion (1l) et il vous suffit de taper :

F5 **ENTREE**

Vous revenez ainsi à l'affichage précédent et le curseur est de nouveau dans la case A1 en haut à gauche de la fenêtre.

Déplacez maintenant le curseur à la case Y1. Vous tapez :

F5 y1 **ENTREE**

Observez les lettres identifiant les colonnes au sommet de la fenêtre. Vous remarquerez que la première colonne à droite de la colonne Z est repérée par AA, la suivante par AB, etc. Ainsi, vous pouvez consulter plus de 26 colonnes.

La grille compte au total 64 colonnes. La colonne AZ est suivie des colonnes BA, BB, etc., la dernière étant BL.

Vous pouvez également faire descendre le curseur pour trouver la dernière rangée. Mais le chemin est long : il y a en effet 255 rangées !

SAISIE DE CHIFFRES

Ramenez le curseur à la case A1 puis tapez :

100

mais n'appuyez pas encore sur **ENTREE**. Vous remarquerez que, dans la zone de contrôle, l'option « VALEUR et FORMULE » est en surbrillance ; QL Abacus vous demande ainsi de confirmer votre action. Le guide **valeur** apparaît également juste sous la fenêtre.

Tout ce que vous tapez, ainsi que le texte que QL Abacus affiche quand vous utilisez une

commande, apparaît sur cette ligne. Nous l'appellerons la **ligne de saisie**.

La ligne de saisie présente un petit rectangle. Ce rectangle indique l'endroit où apparaîtra le prochain caractère que vous taperez; c'est le curseur de saisie, à ne pas confondre avec le curseur principal de la fenêtre. Si vous faites une erreur de frappe sur cette ligne, vous la rectifierez au moyen du système de correction expliqué dans la section « Présentation des logiciels du QL ».

Quand vous appuyez sur la touche **ENTREE**, la valeur 100 est transférée à la case en cours ; la ligne de saisie est effacée pour recevoir d'autres données. Vous remarquerez que la valeur 100 apparaît alors dans la zone d'état en bas de l'écran.

SAISIE DE TEXTE

Le mode de saisie d'un texte dans une case est identique à celui adopté pour les chiffres, à cette différence près qu'un texte doit être précédé de guillemets. Dès que vous tapez des guillemets, QL Abacus répond en mettant en surbrillance l'option TEXTE dans la zone de contrôle et en indiquant **texte** sur la ligne de saisie. Vous tapez alors ce que vous voulez faire apparaître dans la case et vous appuyez ensuite sur **ENTREE**. Ne fermez pas les guillemets. Essayez maintenant de saisir du texte dans quelques cases. Remarquez notamment ce qui se passe quand vous tapez par exemple

un chiffre

et

texte

Un chiffre s'affiche à la droite de la case tandis qu'un texte apparaît à gauche. De plus, la zone d'état précise le type d'information tapée, numérique ou autre, dans la case en cours.

LES COMMANDES

Pour sélectionner une commande, appuyez tout d'abord sur **F3**.

La section centrale de la zone de commande affiche alors une liste, dite menu, de toutes les commandes. C'est ce que nous appellerons le *menu des commandes* qui est représenté à la Figure 2.13.

Nous traiterons plus en détail de ces commandes par la suite mais voyons déjà brièvement deux d'entre elles : **Abandon** pour effacer le tableau en cours et **Quitte** pour abandonner QL Abacus et revenir sous SuperBASIC.

AIDE F1	MENU Abandon	Exéc. Format	Imprime Justif.	Ouvre Quitte	Tri Union	COMMANDES F3
GUIDE F2	Copie Docu.	Grille Hublot	Modifie Notation	Repro. Sauve	Vide	ANNULER ESC
A	B	C	D	E	F	G
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

Commande>

CASE A1 LIMITE A1:A1 MEMOIRE 20K
CONTENU VIDE

Figure 2.9 Le menu des commandes

Entraînez-vous tout d'abord avec la commande **Abandon**. Appuyez sur **F3** et repérez **Abandon** sur le menu affiché. Quand vous appuyez sur la touche **A**, le mot « abandonner » apparaît sur la ligne de saisie. Inutile de taper le nom de la commande en entier la première lettre suffit. Autre changement : le menu de la commande **Abandon** est alors affiché dans la zone de contrôle. Pour annuler cette commande, appuyez sur **ESC**.

Revenez maintenant au menu de commandes en appuyant sur **F3** et tapez ensuite **A** pour rappeler la commande **Abandon**; cette fois, par contre, appuyez alors sur **ENTREE** pour effacer la grille. La grille est vide et le curseur dans la case A1; vous pouvez reprendre à zéro.

Quand vous voulez abandonner QL Abacus et repasser sous SuperBASIC, vous devez utiliser la commande **Quitte**. Procédez comme avec la commande **Abandon** : appuyez sur **F3** puis tapez la première lettre **Q**. Comme, dans ce cas, vous perdrez toutes les données introduites dans la grille, QL Abacus vous donne toujours l'option de revenir sur votre décision grâce à la touche **ESC**.

CHAPITRE 3

CASES, RANGEES, COLONNES ET BLOCS

L'un des grands intérêts de QL Abacus est qu'il vous permet de manipuler, en une seule opération, des rangées et des colonnes entières ou même des blocs de cases. Pour ce faire, il vous suffit d'employer de simples expressions pour, par exemple, remplir toute une rangée de cases. Les valeurs inscrites dans les cases peuvent être identiques ou varier de manière régulière.

Certaines de ces possibilités et leur mode d'emploi sont précisés dans ce chapitre.

CASES

La case est l'unité de base dans laquelle QL Abacus enregistre des informations. Chaque case peut contenir une rubrique d'information, soit du texte, un nombre ou une formule.

Pour chaque case contenant des informations, QL Abacus note également comment elles doivent être affichées. Vous pouvez par exemple afficher des chiffres ou du texte à gauche, au centre ou à droite de la case et visualiser les chiffres selon différents formats.

La justification

La commande **Justification** sert à modifier la position du contenu d'une case, c'est à dire à choisir la position de chiffres ou d'un texte.

Introduisez la valeur 100 dans la case A1. Mettez ensuite en œuvre la commande **Justif** en appuyant sur **F3** puis sur la touche **J**. QL Abacus vous demande alors de choisir entre l'option cases pleines ou l'option cases vides. Choisissez l'option cases pleines en appuyant sur **ENTREE**. Vous avez ensuite à choisir entre texte ou nombres. Vous choisissez nombres et appuyez donc sur **N**. Du fait que QL Abacus vous propose un alignement à gauche, acceptez en appuyant sur **ENTREE**.

En dernier lieu, QL Abacus vous demande d'indiquer le bloc de cases concerné. Appuyez simplement sur **ENTREE**. La valeur 100 déjà dans la case A1 s'inscrit alors à gauche.

Nota : vous pouvez modifier le format numérique ou la justification numérique d'une case contenant déjà du texte. Dans l'immédiat, vous ne remarquerez aucune différence. Cependant, si par la suite vous remplacez le texte par des chiffres, ces chiffres seront affichés et justifiés selon le format voulu. Il en est de même dans le cas de la justification du texte dans une case contenant alors des données numériques.

Cases vides

Pour QL Abacus, une case vide n'existe pas; elle n'est pas enregistrée en mémoire et n'a donc aucune propriété. Rien ne se passera si vous sélectionnez l'option cases pleines après avoir passé la commande **Justif** ou **Notation** en référence à une case vide. Si des chiffres y sont par la suite introduits, ils sont affichés selon le format par défaut, le format général.

Pour changer ces dispositions implicites, vous devez utiliser l'option cases vides de la commande Justif ou Notation (ou des deux). Par exemple, pour sélectionner un format par défaut de pourcentage avec une décimale, employez l'option cases vides de la commande **Notation** : vous tapez donc **F3**, **N** et **V**. Les choix qui vous sont proposés sont identiques à ceux de l'option cases pleines mais il ne vous est pas demandé de préciser un bloc de cases.

L'option cases vides de la commande **Justif** fonctionne de même. Ici encore, vous n'avez pas à préciser un bloc de cases du fait que QL Abacus emploie la nouvelle affectation par défaut chaque fois que vous introduisez des données dans une case jusqu' alors vide.

Les nouvelles affectations par défaut restent en vigueur jusqu'à ce que vous les changiez ou jusqu'à ce que vous repassiez sous SuperBASIC après avoir terminé votre travail sur QL Abacus.

Pour retrouver les affectations par défaut de base (chiffres justifiés à droite, texte à gauche et chiffres affichés sous format général), procédez comme suit :

F3 J V N D [chiffres justifiés à droite]

F3 J V **ENTREE** **ENTREE** [texte justifié à gauche]

F3 N V G [chiffres affichés selon le format général]

RANGÉES

Il vous faudra fréquemment inscrire une valeur donnée dans plusieurs cases d'une même rangée, ou du moins des valeurs variant régulièrement. Vous pouvez dans ce cas donner à ces cases une référence que nous appellerons **identificateur**. On compte deux tels identificateurs : **rang** et **col** qui correspondent aux cases de la rangée ou de la colonne en cours, c'est à dire la rangée ou la colonne où se trouve le curseur.

Exemple : inscrivons la valeur 100 dans la première rangée de la colonne B à la colonne D. Dans ce cas, l'identificateur est **rang**. Amenez le curseur dans la case A1 puis tapez :

rang = 100 **ENTREE**

Dès que vous appuyez sur **ENTREE**, un guide, « DE », apparaît sur la ligne de saisie; il vous propose de remplir la rangée à partir de la colonne A (où se trouve le curseur). Le programme suggère toujours une solution raisonnable pour le point de départ, solution que vous pouvez accepter en appuyant simplement sur **ENTREE**. Cependant, dans notre exemple, nous voulons commencer à partir de la colonne B. Vous devez donc taper :

B **ENTREE**

La ligne de saisie change alors pour vous signaler que la ligne sera remplie à partir de la colonne B et un autre guide, « JUSQU'A », vous propose de terminer à la colonne BL (la dernière colonne de la grille), BL ne nous convient pas puisque nous voulons terminer à la colonne D. Vous tapez donc :

D **ENTREE**

Ces instructions suffisent et elles sont exécutées : la valeur 100 apparaît dans chaque case de B1 à D1 incluse et la ligne de saisie est dégagée pour que vous y inscriviez d'autres données.

COLONNES

Pour remplir une colonne, vous suivez une méthode très semblable. La seule différence est ici que vous identifiez une colonne par une ou deux lettres et non plus par un chiffre. Supposons que nous voulions inscrire « bonjour » dans chaque case de la colonne D, de la rangée 5 à la rangée 11. Nous employons dans ce cas le second identificateur, **col**. Amenez le curseur à la case D5 et tapez :

col = "bonjour" **ENTREE**

Cette fois, QL Abacus vous suggère le bon point de départ puisque c'est là que se trouve le curseur. Vous acceptez en appuyant sur **ENTREE**. Comme point d'arrêt, il vous propose la rangée 255 ; vous demandez la rangée 11 en tapant :

11 **ENTREE**

« bonjour » apparaît dans les cases 5 à 11 et la ligne de saisie est dégagée dans l'attente de vos nouvelles données.

Chaque fois que vous utilisez la référence **col**, le système vous demande de préciser la première et la dernière rangées concernées. Vous les spécifiez ou, comme toujours, vous pouvez accepter les suggestions de QL Abacus.

Vous pouvez également utiliser les identificateurs **rang** et **col** pour spécifier un bloc de cases concerné par une fonction quelconque.

Par exemple, inscrivez des chiffres dans toutes les cases du rectangle dont l'angle supérieur gauche est la case A1 et dont **ENTREE** l'angle inférieur droit est la case C3 (soit neuf chiffres au total), amenez le curseur à la case D1 puis tapez :

col = somme(rang)

Dans chaque case de la colonne D s'inscrira le total des valeurs introduites dans toutes les cases de la rangée correspondante. Mais avant, QL Abacus doit connaître les blocs de **rang** et **col** ; il vous pose deux questions. Il vous demande d'abord d'indiquer le bloc de colonnes pour **rang**, vous suggérant ici D, ce qui ne vous convient pas. Vous tapez **A** puis appuyez sur **ENTREE**. QL Abacus vous propose de terminer ensuite à D, ce que vous changez en tapant C puis **ENTREE**. Il vous demande enfin de délimiter **col**, vous proposant tout d'abord 1, ce qui est correct, puis 255 (ou 11 si vous tapez cet exemple sans avoir « abandonné » le précédent). Vous tapez alors 3 puis appuyez, comme toujours, sur. QL Abacus calcule alors le total de chacune de ces trois rangées et affiche les résultats dans les cases de la colonne D.

ETIQUETTES

Dans nos exemples jusqu'ici, les rangées et les colonnes étaient uniquement identifiées par leurs chiffres et leurs lettres de référence. Il existe une autre méthode d'identification très utile. Il s'agit d'étiquettes, c'est à dire des mots de votre choix. Ces étiquettes vous permettent de faire référence à des rangées, des colonnes ou des cases données.

Un texte introduit dans une case peut faire office d'étiquette. Au lieu d'utiliser un identificateur dans une commande ou une formule, vous tapez le nom d'étiquettes. L'avantage évident est qu'il est plus facile de garder des noms en mémoire que des chiffres.

Cette méthode simplifie considérablement la mise au point et l'utilisation d'une grille. L'emploi de ces étiquettes vous est expliqué dans les deux sections suivantes.

Etiquettes de rangées et de colonnes

Selon le contenu des autres cases d'une grille, une étiquette peut identifier une rangée et de colonnes ou une colonne. Quand vous donnez une étiquette à une rangée ou à une colonne, QL Abacus explore la grille en-dessous et à droite de la case contenant l'étiquette. La case la plus proche renfermant un chiffre, en-dessous ou à droite de l'étiquette, indique s'il s'agit d'une rangée ou d'une colonne. Ce système est plus simple à comprendre sur les Figures 3.1 et 3.2. Sur la Figure 3.1, l'étiquette identifie une rangée, tandis que sur la Figure 3.2, elle correspond à une colonne.

Dans des situations plus complexes, s'il existe par exemple des chiffres à la droite et en-dessous de l'étiquette, le chiffre le plus proche (d'après le nombre de case séparant ce chiffre de l'étiquette) indique s'il s'agit d'une rangée ou d'une colonne. Si les deux chiffres sont équidistants de l'étiquette, QL Abacus affiche le message suivant :

Aucune rangée ou colonne de ce nom

et attend que vous appuyiez sur la barre d'espace. QL Abacus remplace alors le texte de votre formule sur la ligne de saisie pour que vous la corrigiez.

Remplacez alors la référence insuffisante par **rang** ou **col** puis appuyez de nouveau sur **ENTREE**.

Un conseil : changez vos étiquettes ambiguës pour que QL Abacus n'ait pas à régler de nouveau ce problème.

AIDE F1	CURSEUR ↑↓←→	VALEURS et FORMULES	TEXTE Guillemets pour commencer	COMMANDES F3		
GUIDE F2	ACCES CASE F5	Saisie directe ENT pour finir		ANNULER ESC		
A	B	C	D	E	F	G
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
?						
CASE A1 LIMITES A1:C4 MEMOIRE 20K						
CONTENU VIDE						

Figure 3.1 Etiquetage d'une rangée.

AIDE F1	CURSEUR ↑↓←→	VALEURS et FORMULES	TEXTE Guillemets pour commencer	COMMANDES F3		
GUIDE F2	ACCES CASE F5	Saisie directe ENT pour finir		ANNULER ESC		
A	B	C	D	E	F	G
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
?						
CASE A1 LIMITES A1:C4 MEMOIRE 20K						
CONTENU VIDE						

Figure 3.2 Etiquetage d'une colonne.

AIDE F1	CURSEUR ↑↓←→	VALEURS et FORMULES	TEXTE Guillemets pour commencer	COMMANDES F3		
GUIDE F2	ACCES CASE F5	Saisie directe ENT pour finir		ANNULER ESC		
A	B	C	D	E	F	G
1		MARS				
2						
3						
4	COUTS	1000.00				
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
?						

CASE A1 LIMITES A1:C4 MEMOIRE 20K
CONTENU VIDE

Figure 3.3 Affectation d'une étiquette à une case

Affectation d'étiquettes à des cases

Des étiquettes vous servent également à identifier des cases données ; dans ce cas cependant, deux étiquettes sont nécessaires. Dans l'exemple suivant, la case C4 peut être identifiée par « Mars » et « coûts » !

La référence se compose des noms des deux étiquettes séparés par un point, (par exemple : Mars.Coûts). Inutile ici de taper les noms en entier ni de faire la distinction entre majuscules et minuscules. Il suffit en outre de prévoir assez de lettres pour éviter toute confusion avec une autre référence. Pour notre exemple, l'identification « mar.coû » conviendrait parfaitement. Peu importe également l'ordre des étiquettes, la même case pourrait très bien être identifiée par « coû.mar ».

BLOCS

De plus, vous pouvez composer une instruction pour identifier un rectangle entier, ce que nous appelons ici un **bloc** de cases.

Une référence de bloc compte deux parties. La première est la référence de la colonne et de la rangée de la case supérieure gauche du bloc. La seconde partie, séparée de la première par deux points, est la référence de la colonne et de la rangée de la case inférieure droite du bloc. Exemple :

A2:D27

Vous utiliserez par exemple cette référence de bloc dans la commande **Copie** pour copier un bloc ailleurs sur le tableau.

De nombreuses commandes vous demandent de spécifier la référence d'un bloc afin d'identifier les cases concernées. Les possibilités étant beaucoup plus vastes que dans le cas d'une référence de colonne ou de rangée, QL Abacus ne propose aucune suggestion. Vous devez donc taper toute la référence de bloc.

Vous disposez pour ce faire de quatre méthodes, à savoir :

- 1) Avec les chiffres et les lettres des rangées et des colonnes, p.ex.: A1:C7
- 2) Avec des étiquettes, p.ex.: janvier.ventes:mars.coûts
- 3) Avec une combinaison de ces deux méthodes, p.ex. : A1:mars.coûts
- 4) Avec un identificateur de bloc, p.ex.: rang (ou col).

Cet identificateur se rattache à la rangée (ou colonne) contenant le curseur. Dans ce cas, QL Abacus peut proposer un point de départ et d'arrêt.

AIDE F1		COPIE le contenu d'un groupe de cases ailleurs sur la grille. Délimiter ainsi le bloc à copier: (case sup. gauche):(case inf. droite)					COMMANDES F3	
GUIDE F2							ANNULER ESC	
	A	B	C	D	E	F	G	I
1								
2								
3								
4		1		2		3		
5		4		5		6		
6		7		8		9		
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								

Commande>copier le bloc b3:d5

CASE A1 LIMITE A1:D5 MEMOIRE 20K
CONTENU VIDE

Figure 3.4 Une référence de bloc

DES PRECISIONS SUR LES CHIFFRES ET LE TEXTE

Maintenant que vous savez comment spécifier à QL Abacus une case ou un bloc de cases, voyons comment vous pouvez modifier la présentation du contenu et, en premier lieu, comment sont enregistrés les chiffres. Amenez le curseur à la case A1 et tapez le nombre 123.456.

QL Abacus enregistre tous les nombres avec une précision de 16 chiffres significatifs. Un maximum de 14 peut être affiché; les deux derniers permettent simplement de garantir l'exactitude de l'affichage du calcul. Bien que QL Abacus calcule et enregistre tous les nombres avec cette exactitude, vous n'êtes bien sûr pas obligé d'avoir tous les chiffres significatifs à l'écran.

Sélectionnez la commande **NOTATION** (en appuyant sur **F3** puis sur la touche **N**). Vous disposez de deux options : cases pleines ou cases vides. Tapez alors **ENTREE** pour choisir l'option cases pleines.

QL Abacus vous propose alors différentes possibilités d'affichage.

Appuyez sur la touche **M** pour obtenir le format Monétaire. QL Abacus vous demande alors comment vous désirez afficher des valeurs négatives, vous suggérant de les faire précéder d'un signe moins. Vous pouvez accepter cette proposition en appuyant sur **ENTREE**. Mais, si vous

préférez les afficher entre parenthèses, il vous faut taper la lettre **R** Dans cet exemple, peu importe notre choix; choisissons néanmoins l'option du signe moins.

QL Abacus vous demande ensuite quel est le bloc de cases concerné. Vous répondez en précisant une référence de bloc (p. ex. A1:B3) ou simplement d'une seule case. QL Abacus essaie toujours de deviner à l'avance quel bloc vous intéresse mais ne peut pas toujours le faire et suggère alors simplement le bloc A1:B1. Ce bloc correspond en fait à la case A1. Vous accepterez cette proposition en appuyant sur **ENTREE** ou vous indiquerez votre propre référence avant de taper sur **ENTREE**.

Supposons que QL Abacus vous propose par défaut le bloc A1:A1. Les opérations se déroulent comme suit :

F3 N **ENTREE** M **ENTREE** **ENTREE**

Juste avant que vous tapiez **ENTREE** pour la troisième fois, la ligne de saisie doit afficher :

```
Commande>notation,pleines,monétaire,signe moins,blac A1:A1
```

Quand vous appuyez sur **ENTREE**, l'affichage dans la case A1 devient \$123.46 bien que la valeur réelle (123.456) soit toujours indiquée dans la zone d'état. QL Abacus vous ramène automatiquement à l'affichage principal. Le format monétaire arrondit toujours le nombre à deux décimales près et le précède du symbole de la devise, Vous pouvez toujours transformer \$ en F ou autre au moyen des options de la commande **Hublot** (\$ étant la devise affectée par défaut).

Donnons maintenant à l'affichage dans la case A1 le format Entier. Faites appel ici encore à l'option cases pleines de la commande **Notation** mais appuyez sur E. Vous pouvez toujours choisir d'indiquer des nombres négatifs précédés d'un signe moins ou entre parenthèses ; choisissons les parenthèses en tapant P puis **ENTREE** (ce qui n'affecte toujours que la case A1).

L'opération se déroule comme suit :

F3 N **ENTREE** E P **ENTREE** **ENTREE**

et la ligne de saisie affiche :

```
Commande>notation,pleines,entière,parenthèses,bloc A1:A1
```

123 apparaît maintenant dans la case en format entier; la virgule décimale et les chiffres suivants sont éliminés.

Passons maintenant au format Décimal. Avec ce format, comme avec ceux que nous étudierons ensuite, vous ne pouvez pas présenter des valeurs négatives entre parenthèses. Mais vous devez indiquer au système (sauf dans le cas du format Général) le nombre de chiffres que vous désirez afficher après la virgule. Disons ici cinq chiffres après la virgule. Choisissez l'option cases pleines de la commande **Notation**. Décimal est le format par défaut. Il vous suffit donc d'appuyer sur **ENTREE** puis de préciser cinq chiffres après la virgule. Enfin, en réponse au guide « bloc », appuyez sur **ENTREE** de nouveau pour accepter la proposition par défaut. L'opération doit se dérouler comme indiqué ci-après et donner les affichages correspondants sur la ligne de saisie :

F3 N **ENTREE** **ENTREE** 5 **ENTREE** **ENTREE**

```
Commande>notation,pleines,décimale,décimales 5,bLoc A1:A1
```

Comme spécifié, 123.45600 apparaît maintenant dans la case A1.

Faites de nouveau appel à cette commande mais en tapant cette fois **P** pour demander le format Pourcentage. Spécifiez une décimale et sélectionnez la case A1. L'opération se déroule comme suit :

F3 N **ENTREE** P 1 **ENTREE** **ENTREE**

L'écran affiche maintenant 12345.6%. L'option Pourcentage produit un nombre multiplié par 100 et suivi du signe %. A noter que la valeur mémorisée, et d'ailleurs indiquée dans la zone d'état, est toujours 123.456 quel que soit l'affichage dans la case.

Essayons maintenant le format Scientifique, ou exponentiel, avec trois chiffres après la virgule. Tapez :

F3 N **ENTREE** S 3 **ENTREE** **ENTREE**

Avant que vous tapiez votre troisième **ENTREE**, la ligne de saisie affiche :

Commande>notation,pleines,scientifique,décimales3,bloc A1:A1

après avoir tapé le troisième **ENTREE**, 1.235E+02 apparaît dans la case.

Le format scientifique vous permet d'afficher des nombres trop grands ou trop petits pour le format décimal. Le nombre est traité comme une puissance de 10 multipliée par une valeur comprise entre 1 et 10.

Par exemple, 2 300 000 000 peut être considéré comme étant 2.3 multiplié par 1 000 000 000, 1 000 000 000 étant ensuite élevé à la puissance 9 (dix multiplié neuf fois). Ainsi, en format scientifique, 2 300 000 000 peut être inscrit sous la forme : 2.3 E+09. Il en est de même avec des chiffres très petits qui sont dans ce cas traités avec une puissance négative. Par exemple, 0.000123, soit 1.23 divisé par 10 000 (dix puissance 4), devient 1.23 E-04 en format scientifique.

La dernière option est le format Général qui apparaîtra dans la case A1 quand vous aurez tapé :

F3 N **ENTREE** G **ENTREE**

La ligne de saisie indique alors :

Commande>notation,pleines,générale,bloc A1:A1

En format Général, vous n'avez pas non plus à préciser le nombre de décimales. QL Abacus choisit un format d'affichage raisonnable pour chaque nombre et essaie de les présenter aussi précisément que possible dans l'espace disponible.

Avant d'en terminer avec la commande **Notation**, essayons d'afficher le nombre présent dans la case A1 avec neuf décimales. Vous tapez :

F3 N **ENTREE** **ENTREE** 9 **ENTREE** **ENTREE**

apparaît alors dans la case A1 vous indiquant qu'il n'y a pas assez d'espace pour tout afficher. Dans ce cas, vous pouvez changer le format d'affichage ou augmenter la largeur de la colonne.

Effacez maintenant le tableau avec la commande **Abandon**. Le curseur étant dans la case A1, tapez :

"Ce texte est très long

Bien que le texte soit trop long pour être contenu dans une seule case, il apparaît en entier en chevauchant d'autres cases. Entrez maintenant le chiffre 1 dans la case B1. Le texte est coupé à la fin de la case A1 pour permettre l'enregistrement de votre chiffre. Ramenez le curseur à la case A1 et vous remarquerez que votre texte est toujours mémorisé dans la zone d'état.

Ramenez le curseur à la case B1 et effacez-la avec la commande **Vide**. Vous devez ici préciser le bloc de cases dont le contenu est à effacer. Vous ne voulez effacer que le contenu de la case B1 et tapez (en acceptant la suggestion de bloc de QL Abacus) :

F3 V **ENTREE**

Maintenant que la case B1 est vide, le texte inscrit dans la case A1 réapparaît en entier.

CHAPITRE 4

FONCTIONS ET FORMULES FONCTIONS

QL Abacus vous propose plusieurs *fonctions* prédéfinies vous permettant de procéder à certains calculs sur le contenu d'une ou de plusieurs cases. Une fonction implique des valeurs d'entrée, dites *arguments*, d'après lesquelles un résultat donné est calculé. Le résultat est considéré être la valeur que la fonction *renvoie*.

Avec QL Abacus, vous devez présenter vos arguments entre parenthèses après avoir défini la fonction; si votre fonction exige plusieurs arguments, vous les séparerez par une virgule. La plupart des fonctions du système vous renvoient une valeur numérique, comme par exemple la fonction **somme()**. A titre d'argument, cette fonction prend une référence de bloc et renvoie ensuite une valeur numérique égale à la somme des valeurs numériques inscrites dans toutes les cases du bloc.

D'autres fonctions, **mois()** par exemple, renvoient un texte. Ainsi, **mois(1)** devient « Janvier ». Certaines fonctions n'exigent aucun argument; vous devez néanmoins toujours taper les parenthèses. Par exemple, la fonction **pi()** vous fournit la valeur numérique de cette constante mathématique (3.14 approximativement).

Deux fonctions sont particulièrement utiles : **col()** et **rang()**. Elles renvoient le numéro de la colonne (ou de la rangée) traversant la case qui renferme cette fonction. Nous les utiliserons fréquemment dans des exemples au chapitre suivant.

Ainsi, de la colonne A, **col()** renvoie la valeur 1, de la colonne B, la valeur 2, etc. La fonction **rang()** indique simplement le numéro de la rangée.

Prenons un exemple et utilisons les deux fonctions **mois()** et **col()** pour libeller des colonnes de la grille. Notre but est ici d'inscrire les titres janvier, février, etc. en haut des colonnes B à M. La fonction **col()** nous fournira l'argument requis par la fonction **mois()** de sorte que la valeur diffère à chaque colonne. Vous tapez donc (après avoir effacé le tableau) :

```
rang = mois(col())
```

et vous appuyez sur **ENTREE**. Quand QL Abacus vous demande où commence et se termine le bloc, indiquez B et M. Vous remarquerez que nous n'obtenons pas exactement le résultat recherché en ce que, si les étiquettes s'affichent bien à partir de la colonne B, elles commencent avec le mois de février, et non pas janvier. Ceci relève du fait que, dans la colonne B, **col()** renvoie la valeur 2 et que **mois(2)** correspond à février il nous suffit alors de modifier notre instruction de sorte que 1 soit soustrait de la valeur renvoyée par **col()** avant de calculer le mois. Vous tapez alors :

```
rang = mois(col()-1)
```

sans oublier de toujours appuyer sur **ENTREE** et de préciser de B à M.

FORMULES

Une formule vous sert généralement à rattacher le contenu d'une case à celui d'une ou plusieurs autres cases de la grille. Cette caractéristique de QL Abacus est particulièrement utile puisqu'elle vous permet de décrire très simplement les calculs les plus complexes.

La saisie d'une formule dans une case est identique à celle de nombres : vous amenez le curseur dans la case en question, vous tapez la formule et vous appuyez sur **ENTREE**. QL Abacus suppose que tout ce qui n'est pas un nombre (qui ne commence pas par un chiffre) ou un texte (qui ne commence pas par des guillemets) est une formule.

Amenez le curseur à la case B3 et introduisez le nombre 100; amenez ensuite le curseur à la

case C3 et introduisez le nombre 200. Amenez-le ensuite à la case D3 et tapez la formule suivante :

B3 + C3

Deux changements ont lieu quand vous appuyez sur **ENTREE**. Tout d'abord, la valeur 300 s'inscrit dans la case D3, représentant la somme du contenu dans B3 et dans C3. Vous remarquerez ensuite que la zone d'état en bas de l'écran rappelle votre formule de calcul. Une case contenant une formule indique toujours le résultat du calcul. Pour retrouver la formule, il vous suffit de placer le curseur dans cette case et de lire ce qu'indique alors la zone d'état.

Les autres exemples d'emploi de formules font appel au système d'étiquettes et aux identificateurs de bloc rang et col. Ces méthodes de saisie d'informations sont considérablement plus efficaces que celle demandant les chiffres et lettres de référence de cases.

Attention : une formule numérique sans référence de case n'est pas mémorisée en tant que formule. Dans un tel cas, QL Abacus calcule sa valeur et l'enregistre en tant que simple chiffre. Par exemple, 37 + 100/20 est enregistré en tant que 42 et non pas en tant que formule.

SIMPLE EXEMPLE DE CALCUL DE CASH-FLOW

AIDE F1	CURSEUR ↑↓←→	VALEURS et FORMULES	TEXTE Guillemets pour commencer	COMMANDES F3			
GUIDE F2	ACCES CASE F5	Saisie directe ENT pour finir		ANNULER ESC			
A	B	C	D	E	F	G	
1	Janvier	Février	Mars	Avril	Mal	Juin	
2	Ventes	F10000.00	F10500.00	F11025.00	F11576.25	F12155.06	F12762.82
3	Coûts	F7220.00	F7495.00	F7783.75	F8086.94	F8405.28	F8739.55
4	Bénéfices	F2780.00	F3005.00	F3241.25	F3489.31	F3749.78	F4023.27

Figure 4.1 : Simple analyse de trésorerie

Effacez le tableau avec la commande **Abandon** puis réinscrivez les mois de B1 à M1 comme ci-avant.

Amenez ensuite le curseur dans la case A2, introduisez le texte « Ventes » et la valeur 10000 dans B2. Amenez alors le curseur dans C2 et tapez la formule suivante :

rang=ventes.janvier*1.05

QL Abacus vous propose le bloc de colonnes C à M; acceptez en appuyant à deux reprises sur **ENTREE**. Vous remarquerez que QL Abacus sait que la rangée se termine à la colonne M puisque c'est là que se termine la rangée des mois. Quand vous aurez appuyé deux fois sur **ENTREE**, différentes valeurs apparaîtront dans la rangée 2 à partir de la colonne C et la zone d'état en bas de l'écran indiquera la formule B2 *1,05.

En déplaçant le curseur le long de la rangée, vous observerez que la formule donnée pour chaque case est légèrement différente. Dans chacun des cas, la formule prend le contenu de la case juste à gauche et le multiplie par 1.05; le résultat de cette multiplication est alors inscrit dans la case en cours. Autrement dit, la formule dans la case E2 se rattache à la case D2, celle dans H2 à la case G2, etc.

Toutes les formules fonctionnent de même, sauf indication contraire de votre part. Chaque formule garde en mémoire les positions relatives de toutes les cases auxquelles elle se rattache. Quand une même formule est appliquée à plusieurs cases, les références sont adaptées pour donner une *référence de case relative*.

Pourquoi était-il nécessaire d'inscrire la première valeur de 1000 dans B2 ? Pour deux raisons : pour que l'étiquette « Ventes » soit identifiée comme référence de rangée et pour que la formule sache quelle première valeur utiliser.

Amenez maintenant le curseur dans A3 et introduisez le texte « Coûts » : Tapez ensuite la formule :

```
coûts = ventes * 0.55 + 1720
```

Cette formule calcule les coûts d'après deux éléments : les frais de production (55% des ventes) et les coûts fixes d'un total de 1720.00.

Acceptez les points de départ et d'arrêt proposés (colonnes B à M) et appuyez sur **ENTREE**. Du fait que le contenu de la rangée est défini d'après sa référence « Ventes » l'étiquette « Coûts » est également prise comme référence de rangée et porte sur le même bloc que « Ventes ».

Faites de nouveau avancer le curseur le long de la rangée et observez les différentes formules qui apparaissent en bas de l'écran afin de comprendre comment sont obtenus les résultats.

Enfin, inscrivez le texte « Bénéfices » dans la case A4 et tapez la formule suivante :

```
bénéfices = ventes-coûts
```

et acceptez ici encore le bloc proposé (colonnes B à M). QL Abacus se charge du reste et vous présente, sous un format simple, des résultats complets. Si vous désirez donner à cet affichage le format monétaire, vous faites appel à la commande Notation, comme suit :

```
F3 notation,pleines,monétaire,signe moins,bloc B2:M4
```

et l'affichage des premières colonnes doit être identique à celui présenté sur la Figure 4,1.

CALCUL AUTOMATIQUE

Maintenant que vous avez procédé à cette simple analyse de trésorerie, changez le nombre inscrit dans la case B2 (Ventes.janvier).

Amenez le curseur dans B2; le plus simple est d'appuyer sur **F5** puis de taper la référence de la case (soit B2, soit ven.jan), avant d'appuyer sur **ENTREE**. Tapez alors n'importe quel nombre. Appuyez sur **ENTREE** ; tous les chiffres du tableau sont modifiés !

Toutes les formules des cases sont recalculées automatiquement chaque fois que vous introduisez de nouvelles données. Du fait que, dans notre exemple, toutes les formules se rattachent directement ou indirectement à la valeur contenue dans B2, toutes les valeurs sont modifiées quand vous changez le contenu de cette case. (N'oubliez pas que nous avons supposé que les ventes augmenteraient de 5% par mois.)

Vous pouvez annuler ce système d'auto-calcul avec la commande **Format**, ce qui est particulièrement utile quand, par exemple, votre grille compte de nombreuses formules complexes et que vous ne voulez pas patienter pendant les calculs chaque fois que vous changez une valeur.

Sélectionnez la commande **Format** en appuyant sur **F3** puis sur **F**. L'écran vous propose plusieurs options qui sont rapportées sur la Figure 4.2. Pour choisir une option, il suffit de taper la première lettre de son nom. Pour l'auto-calcul, vous tapez donc **A** pour l'annuler ou le rétablir selon les indications de l'écran. Pour revenir au tableau, appuyez sur **ENTREE**.

AIDE F1	FORMAT permet de personnaliser ABACUS Taper la 1ère lettre de l'option voulue	COMMANDES F3
GUIDE F2	Faire ENTREE pour finir	ANNULER ESC
AUTO-CALCUL après chaque entrée.....		oui
VIDE si nul.....		non
ORDRE de calcul (rang. ou col.).....		RANG
ECRAN 80,64,40 colonnes (8,6,4).....		80
NOUVELLE page pour chaque tableau.....		oui
INTERLIGNE d'impression.....		0
LIGNES par page.....		64
MONNAIE (p.ex F,\$).....		F
CARACTERES par ligne.....		80

Figure 4.2 La commande Format

L'effet de l'auto-calcul étant maintenant annulé, vous remarquerez que, si vous changez la valeur dans B2, le contenu des autres cases ne variera pas.

Vous pouvez en outre obliger le système à recalculer toutes les formules de la grille avec la commande **Exec**. Essayons de l'utiliser alors que l'auto-calcul est toujours annulé. Appuyez donc sur **F3** puis sur **E** et toutes les valeurs des cases sont recalculées.

Avant de poursuivre, rappelons le système d'auto-calcul avec, ici encore, la commande **Format**. Tapez **A** pour rétablir l'auto-calcul et appuyez sur **ENTREE** pour revenir au tableau.

CHAPITRE 5

EXEMPLES

Les possibilités d'emploi de QL Abacus vous seront maintenant décrites au moyen d'exemples. Ces exemples ont été choisis pour démontrer, d' une part le fonctionnement du programme, et d'autre part, l'ample éventail de ses cas d'emploi. Le meilleur moyen de faire connaissance de QL Abacus est de le mettre à l'épreuve. C'est dans cet esprit que les exemples suivants ont été mis au point.

Nous vous conseillons d'effectuer vous-même chacun de ces exemples. Vous apprendrez ainsi quelque chose de nouveau à chaque fois et ces exercices vous aideront à garder en mémoire les informations fournies. Ce faisant, il est fort possible que vous songiez à des modifications ou des perfectionnements qui seront utiles à vos propres cas d'emploi.

Dans tous ces exemples, le texte, les nombres et les formules seront indiqués exactement comme vous devrez les taper. Quand il vous faudra spécifier un bloc de cases, il sera indiqué entre accolades à la fin de la ligne. Souvent, ce bloc sera celui proposé par QL Abacus et il vous suffira donc d'appuyer sur **ENTREE**. Parfois, vous devrez saisir vous-même le bloc. Si le curseur doit être positionné dans une case donnée, sa référence sera signalée entre crochets au début de la ligne, mais ne le tapez pas. Par exemple, la ligne :

```
[A4] rang=mois(col()-1) [colonnes B à M]
```

signifie : amenez le curseur dans la case A4 et tapez :

```
rang=mois(col()-1)
```

et, si besoin est, modifiez le bloc suggéré par QL Abacus s'il ne correspond pas aux colonnes B à M.

Si vous devez taper une référence de bloc donnée, soit p.ex_ b3:e15, elle vous sera donnée sous ce format.

Quand des commandes sont indiquées intégralement, elles sont rapportées exactement comme elles doivent apparaître sur l'écran. N'oubliez pas qu'il vous suffit de taper la première lettre de chaque option; QL Abacus se charge de compléter son nom. En outre, si vous choisissez l'options cases pleines, il vous suffit d'appuyer sur **ENTREE**.

Vous commencerez chaque exemple avec une grille vierge. Si besoin est, vous l'effacerez avec la commande **Abandon** avant d'enregistrer de nouvelles données.

MODELISATION DE CASH FLOW

Cet exemple approfondit celui donné au chapitre 4. Quand vous l'aurez terminé, votre grille devra se présenter comme celle de la Figure 5.1.

	A	B	C	D	E	F	G	
1	CASH FLOW							
2	=====							
3								
4		Janvier	Février	Mars	Avril	Mai		
5	-----							
6	VENTES	F40000.00	F40800.00	F41616.00	F42448.32	F43297.29	F4416	
7	CPU	F27500.00	F27900.00	F28308.00	F28724.16	F29148.64	F2958	
8	-----							
9	BRUT	F12500.00	F12900.00	F13308.00	F13724.16	F14148.64	F1458	
10	-----							
11	CHARGES							
12	salaires	F7000.00	F7000.00	F7000.00	F7000.00	F7000.00	F700	
13	publicité	F1000.00	F1000.00	F1000.00	F1000.00	F1000.00	F100	
14	loyer	F2000.00	F2000.00	F2000.00	F2000.00	F2000.00	F200	
15	électricité	F500.00	F500.00	F500.00	F500.00	F500.00	F50	
16	amortissement	F900.00	F900.00	F900.00	F900.00	F900.00	F90	
17	-----							
18	TOTAL CHARGES	F11400.00	F11400.00	F11400.00	F11400.00	F11400.00	F1140	
19	-----							
20	NET	F1100.00	F1500.00	F1908.00	F2324.16	F2748.64	F318	
21	-----							
22	?							
	CASE A1	LIMITES	A1:M21	MEMOIRE	14K			
	CONTENU	VIDE						

Figure 5.1 Tableau de calcul du cash flow (5 premières colonnes)

Les deux premières saisies produisent un titre souligné.

```
[C1] "CASH FLOW
[C2] repro ("=",long(c1))
```

La seconde saisie fait appel à la fonction **repro()** qui exige deux arguments. Le premier est un texte ou la référence d'une case indiquant ce texte et le second est numérique. La fonction répète alors le premier caractère du texte. Notre texte étant le signe "=" la fonction le répète sous le titre. Même si vous décidez de changer le titre, vous n'aurez pas à modifier la formule dans la case C2 puisqu'elle fait intervenir la fonction **long()** pour reconnaître la longueur du texte dans la case C1.

Comme la première colonne contiendra vos intitulés, elle doit être plus large que les autres. Le curseur étant dans A1, nous utilisons la commande **Grille** comme suit pour modifier la largeur de 10 à 15 caractères :

```
[A1] F3 G L 15 { de A à A}
```

Ensuite,

```
[A4] rang=mois(col()-1) { colonnes A à M}
[A5] rang=repro("-", larg()+1) { colonnes B à M}
```

Ces deux saisies inscrivent les mois sur la rangée 4 et tirent une ligne en-dessous sur toute la section utile de la grille. La fonction **larg()** précise la largeur de chaque colonne par pas horizontal de caractères. Elle permet donc de tirer des lignes sur un tableau avec des colonnes de différentes largeurs. Du fait qu'un espace supplémentaire sépare chaque colonne, la formule prévoit +1.

```
[A6] "VENTES
[B6] 40000
[C6] rang=ven.jan*1.02 { colonnes C à M}
```

Ces saisies inscrivent les chiffres des ventes pour l'année d'après un chiffre d'affaires de 40 000 en janvier et un essor hypothétique de 2% par mois.

```
[A7] "CPV
[B7] cpv=ven*0.5+7500 {colonnes B à M}
```

(Le coût des produits vendus est supposé être égal à la moitié du prix de vente augmenté d'un coût fixe de F7500.)

```
[A8] rang=a5      {colonnes A à M}
[A9] "BRUT
[B9] brut=ven-cpv  {colonnes B à M}
```

Le système tire alors une ligne sous le coût des produits vendus et calcule le bénéfice brut mensuel.

```
[A11] "CHARGES
[A12] "salaires
[B12] rang=7000 {colonnes B à M}

[A13] "publicité
[B13] rang=1000 {colonnes B à M}

[A14] "loyer
[B14] rang=2000 {colonnes B à M}

[A15] "électricité
[B15] rang=500  {colonnes B à M}

[A16] "amortissement
[B16] rang=900  {colonnes B à M}
```

Ces saisies enregistrent tous les frais qui sont ici censés être constants tout du long de l'année. Vous pouvez naturellement changer les rubriques et les montants à votre guise. Votre tableau peut compter plus ou moins de paramètres ; il vous suffit de modifier en conséquence les références des cases. Il est possible que dans, votre cas, les valeurs diffèrent selon les mois ; il est cependant plus rapide d'établir en premier lieu un tableau avec des valeurs fixes et de les modifier par la suite.

```
[A17] rang=a5      {colonnes A à M}
[A18] "TOTAL CHARGES
[B18] rang=somme(col) {rangées 12 à 16, colonnes B à M}
[A19] rang=a5      {colonnes A à M}
```

Vous obtenez ainsi le total de vos charges mensuelles.

La fonction **somme()** additionne le contenu de toutes les valeurs numériques présentes dans le bloc spécifié par l'argument. Toutes les cases vides, ainsi que celles contenant du texte, sont ignorées. Une référence explicite peut être attribuée au bloc, soit par exemple B12:B16. Comme dans notre cas le bloc ne concerne qu'une colonne, nous nous limitons à l'identificateur col. Répondez simplement aux questions posées par QL Abacus et appuyez sur **ENTREE** si ses propositions vous conviennent.

Vous remarquerez que cette formule fait intervenir les identificateurs **rang** et **col** de deux manières différentes. En premier lieu, **rang** signifie que la formule devra s'inscrire dans plusieurs cases de la rangée en cours. Intervient ensuite **col** qui spécifie sur quel bloc l'addition doit être effectuée. Ces deux identificateurs vous demandent de confirmer, ou de modifier les points de départ et d'arrêt suggérés. Dans notre exemple, QL Abacus se charge tout d'abord du bloc concerné par la fonction **somme**.

```
[A20] "NET
[B20] net=brut-tot  {colonnes B à M}
[A21] rang=repro("=", larg()+1) {colonnes A à M}
```


Les calculs sont maintenant terminés. Les bénéfices nets sont donnés par la différence entre les bénéfices bruts et les charges. Il vous suffit maintenant de faire intervenir quelques commandes pour modifier la présentation du tableau. Nous avons déjà vu comment changer la largeur d'une colonne. Essayons maintenant d'utiliser la commande **justif**. (toujours après avoir appuyé sur **F3**) :

```
justifier,pleines,texte,droite,bloc a4:m4
justifier,pleines,texte,droite,bloc a12:a16
notation,pleines,décimale,décimales 2,bloc a1:m21
```

Nous avons ainsi choisi d'afficher les nombres en format décimal avec deux chiffres après la virgule. Si vous voulez faire apparaître l'abréviation du Franc, vous rappelez notation comme suit :

```
notation,pleines,monétaire,signe moins,bloc a1:m21
```

Vous voulez changer un nombre ? Rien de plus simple. Modifiez par exemple le budget publicitaire de février. Appuyez sur **F5** (accès à) puis tapez la référence de la case :

```
c13
```

Le curseur se déplace alors à cette case et vous pouvez inscrire un nouveau chiffre.

N'oubliez pas que le chiffre d'affaires a été calculé par une formule supposant une augmentation mensuelle de 2%. Si vous donnez à l'une des cases concernées une valeur numérique, vous détruisez sa formule. Les formules des autres cases restent cependant identiques. Les montants inscrits dans les cases suivants augmenteront donc de 2% par mois à partir du changement enregistré.

TABLES DE MULTIPLICATION

Ce simple exemple vous indique comment obtenir une table de multiplication. Le système vous demande laquelle puis l'affiche. Quand vous aurez obtenu votre première table, vous demanderez à QL Abacus d'en calculer automatiquement une autre en lançant la commande **Exec**.

```
F3 E
```

QL Abacus vous demandera alors quelle table vous désirez; vous taperez un chiffre et il calculera automatiquement la nouvelle table.

Un exemple de l'affichage ainsi obtenu vous est présenté sur la Figure 5.2.

Commencez comme d'habitude par le titre :

```
[B1] "TABLES DE MULTIPLICATION
[B2] repro( "=",long(b1))
```

Les trois lignes suivantes donnent le titre même de la table.

```
[B3] "La table
[C3] "de
[D3] saisisn(c3)
```

A la question "de? "; tapez par exemple 7

AIDE F1	CURSEUR ↑↓←→	VALEURS et FORMULES	TEXTE Guillemets pour commencer	COMMANDES F3					
GUIDE F2	ACCES CASE F5	Saisie directe ENT pour finir		ANNULER ESC					
	A	B	C	D	E	F	G	H	I
1	TABLES DE MULTIPLICATION								
2	=====								
3	La table de 7								
4		1 fois	7	=	7				
5		2 fois	7	=	14				
6		3 fois	7	=	21				
7		4 fois	7	=	28				
8		5 fois	7	=	35				
9		6 fois	7	=	42				
10		7 fois	7	=	49				
11		8 fois	7	=	56				
12		9 fois	7	=	63				
13		10 fois	7	=	70				
14		11 fois	7	=	77				
15		12 fois	7	=	84				
16									
?									
CASE I1 LIMITES A1:E15 MEMOIRE 18K									
CONTENU VIDE									

Figure 5.2 Une table de multiplication

Nous avons fait appel à la fonction **saisisn()** pour demander une donnée, ce qui nous permet de choisir un facteur de multiplication.

Cette fonction prend un texte pour argument et le reporte sur la ligne de saisie sous forme de question. Vous tapez alors votre multiplicateur puis appuyez sur **ENTREE**. Ce chiffre s'inscrit dans la case contenant la fonction **saisisn()**.

Nota : en auto-calcul, **saisisn()** n'attend pas de données de votre part. Seul le message est affiché et une saisie n'est sollicitée que quand la formule est inscrite dans la case ou quand vous obligez le système à recalculer le tableau avec la commande **Execute**. Quand une valeur est enregistrée dans une case, elle y reste jusqu'à ce que vous fassiez intervenir la commande **Execute**.

Les opérations suivantes rempliront les colonnes pour produire la table de multiplication.

```
[B4] col=chn(rang()-3,2,0)+" fois"    {rangées 4 à 15}
```

Cette formule est la plus complexe de notre exemple. Elle affiche le chiffre à multiplier à chaque rangée. Chaque chiffre est converti en une chaîne alphanumérique afin que nous puissions l'associer au signe de multiplication et les afficher tous deux dans une case.

La fonction **chn()** convertit un nombre en une chaîne de chiffres. Elle prend trois valeurs : le nombre à convertir, un code pour le format d'affichage du nombre (0 = décimal, 1 = scientifique, 2 = entier, 3 = général) et le nombre de décimales à indiquer.

Dans le cas présent, la valeur est obtenue de l'expression **rang()-3**, soit 1 à la quatrième rangée, 2 à la cinquième et ainsi de suite jusqu'à 12 à la quinzième. La valeur suivante précise le format entier. La troisième sert normalement à indiquer le nombre de décimales. Elle est ici ignorée puisque nous voulons obtenir des nombres entiers. Nous tapons donc un zéro.

Enfin, le résultat est concaténé avec la chaîne "*" ; de sorte que le chiffre à multiplier et le signe de multiplication soient affichés dans une même case.

```
[C4] col=$d3    {rangées 4 à 15}
```

La colonne C contient alors le chiffre tapé en réponse à la fonction **saisisn()** soit ici 7. La référence de case est précédée du signe \$ pour en faire une *référence absolue*. Faites maintenant monter et descendre le curseur le long de la colonne C ; vous remarquerez que la zone d'état affiche toujours la référence \$C3 qui n'est donc pas modifiée à chaque changement de rangée. Une référence absolue se rattache toujours à une case donnée à partir de toute position dans la grille. Pour rendre une référence absolue, il suffit de la préfixer du signe \$.

```
[D4] col="=" {rangées 4 à 15}
```

```
[E4] col=$d3*(rang()-3) {rangées 4 à 15}
```

Ces deux dernières saisies se passent presque d'explications. Elles produisent à chaque rangée le signe égal et le résultat de la multiplication. La dernière formule multiplie le 7 (une seconde référence absolue donnée en réponse à la question de la fonction **saisisn()**) par l'expression **rang()-3** qui, comme nous l'avons vu précédemment, donne une valeur de 1 à la quatrième rangée, de 2 à la cinquième et ainsi de suite jusqu' à 12 à la quinzième.

Modifions maintenant la table pour lui donner un format plus commode avec les commandes **justif. et grille**.

```
justifier,pleines,texte,droite,bloc b3:b15
```

```
justifier,pleines,texte,droite,bloc d4:d15
```

```
justifier,pleines,texte,centre,bloc c3
```

```
justifier,pleines,nombres,gauche,bloc d3
```

```
grille> largeur 3 DE c JUSQU' A c
```

```
grille>largeur 2 DE d JUSQU'A d
```

```
grille>largeur 4 DE e JUSQU'A e
```

Vous pouvez maintenant changer de table de multiplication avec la commande **Execute** qui se chargera de faire le calcul. Appuyez donc sur **F3** puis sur **E** et vous retrouvez la question précédente "de ?"; tapez alors un autre multiplicateur (de 1 à 12), 9 par exemple, et vous obtenez la table de 9.

RAPPROCHEMENT DE COMPTES

Cet exemple vous indique comment vérifier la concordance entre votre chéquier et votre compte en banque. Vous inscrivez les montants des chèques que vous avez tirés puis, à la fin du mois, d'autres éléments tels que votre salaire, les virements automatiques et autres. Le bilan ainsi obtenu est comparé avec votre relevé bancaire.

Le résultat doit être semblable à l'exemple fourni à la Figure 5.3.

	A	B	C	D	E	F
1		RAPPROCHEMENT DE COMPTES				
2		=====				
3						
4			Mois	Janvier		
5						
6	Solde d'ouverture		2000.00			
7	Salaires		5273.50			
8	Autres revenus		0.00			
9						
10		CREDIT	7273.50			
11			=====			
12						
13	Virements			1300.00		
14	Agios			0.00		
15						
16	Chèques	Date	Numéro	Montant		
17		3/01/85	123456	500.00		
18		10/01/85	123457	500.00		
19		14/01/85	123458	322.10		
20		17/01/85	123459	500.00		
21		24/01/85	123460	250.00		
22		31/01/85	123461	350.00		
23		---	---	---		
24		---	---	---		
25		---	---	---		
26		---	---	---		
27						
28		DEBIT		3722.10		
29				=====		
30	Solde de fermeture		3551.40			
31			=====			
32						
33						
34						
35						
36						
37						
38						
39						
?						
	CASE A39	LIMITES A1:D31	MEMOIRE 18K			
	CONTENU VIDE					

Figure 5.3 Rapprochement de comptes

```
[B1] "RAPPROCHEMENT DE COMPTES
[B2] repro ("=",long(b1))

[C4] "Mois
[D4] saisist(c4)
```

et, à la question "Mois?, indiquez le mois en question, disons ici Janvier

La fonction **saisist()** fonctionne exactement comme la fonction **saisisn()**, à cette simple différence près que le système vous demande de saisir un texte et non plus un nombre. Si vous faites appel à la commande **Exécute**, QL Abacus affiche le message sur la ligne de saisie et attend que vous tapiez votre texte.

Elargissons maintenant la première colonne, celle définissant vos rubriques. Faites appel à la commande **Grille**, sélectionnez L et, au lieu de 10, précisez 20 de A à A.

```
[A6] "Solde d'ouverture
[A7] "Salaire
[A8] "Autres revenus
[C6] saisissn (a6+" pour "+$d4)
```

A la question "Solde d'ouverture pour janvier", vous inscrivez le montant du solde. La fonction **saisisn()** reprend le texte saisi dans d'autres cases d'après des références absolues et relatives.

La commande **Repro** (disponible après avoir appuyé sur **F3**) va maintenant nous permettre d'appliquer la formule de la case C6 aux cases C7 et C8. Au lieu de taper la référence de bloc C7:C8, nous utilisons l'identificateur **col**.

```
Reproduire la case C6,dans le bloc col DE 7 JUSQU'A 8
```

Choisissez alors la commande **Exécute** puis répondez aux questions "Salaire pour janvier" et "Autres revenus pour janvier."

```
[B10] "CREDIT
[C10] somme(col)      {rangées de 6 à 8}
```

La case C10 donne le total de tous les postes du mois à votre crédit. Elle est identifiée par la référence "crédit.mois".

Le contenu de cette case est calculé par la fonction **somme()** que nous avons déjà étudiée. Elle additionne le contenu numérique de toutes les cases du bloc spécifié par son argument. N'oubliez pas qu'elle ignore toute case vide ou contenant du texte.

Nous avons ici repris la définition somme(col) pour indiquer que les cases à additionner se situent dans la colonne en cours. Comme toujours, QL Abacus vous demande de préciser le bloc exact en vous proposant un bloc possible d'après vos calculs précédents.

```
[C11] repro("=",long(chn(crédit.mois,0,2)))
```

Le total est alors souligné dans la case C11 grâce, ici encore, aux fonctions **repro()** et **long**. Puisque nous ne connaissons pas à l'avance le nombre de chiffres à souligner, nous prévoyons une chaîne avec la fonction **chn()** en supposant que les valeurs seront présentées en format numérique avec deux chiffres après la virgule. Le nombre exact de caractères sera ainsi souligné.

```
[A13] "Virements
[A14] "Agios
[D13] saisissn(a13+" pour "+$d4)
[D14] saisissn(a14+" pour "+$d4)
```

Ces formules nous permettent d'enregistrer les débits du mois en réponse aux questions posées par la fonction **saisisn()**, comme nous l'avons vu auparavant.

```
[A16] "Chèques
[B16] "Date
[C16] "Numéro
[D16] "Montant
[B17] rang="---"      {colonnes B à D}
```

Cette dernière formule prépare une section du tableau à recevoir les précisions nécessaires sur les chèques.

```
[B28] "DEBIT
[D28] somme(col)      {rangées 13 à 26}
```

Cette formule calcule le total des postes débiteurs. N'oubliez pas que la fonction **somme()** n'additionne que les valeurs numériques saisies dans les cases du bloc nommé et qu'elle ignore toute case vide ou contenant du texte (et donc les postes inutilisés de la liste de chèques ainsi que les titres prévus dans la colonne D).

```
[A30] "Solde de fermeture
[C30] crédit.mois-débit.montant
```

Avec le solde de fermeture, nous arrivons à la fin de la saisie des postes. Complétons maintenant notre travail en faisant appel à quelques autres commandes.

Utilisons tout d'abord la commande **Repro** pour remplir le tableau des chèques et souligner les totaux. Cette commande reproduit le contenu d'une case dans le bloc de cases donné. La première des commandes que vous exécuterez reproduira le contenu de la case B17 dans toutes les cases renfermées dans le triangle formé par B18 en haut à droite et D26 en bas à gauche.

```
reproduire la case b17,dans le bloc b18:d26
reproduire la case c11,dans le bloc d29:d29
reproduire la case c11,dans le bloc c31 : c31
```

Nous demandons ensuite que le tableau soit affiché en format décimal, avec deux décimales après la virgule, mais en format entier pour les numéros des chèques.

```
notation,pleines,décimale,décimales 2,bloc a1:d30
notation,pleines,entière,signe moins,bloc c17:c26
```

Nous savons déjà que, pour QL Abacus, une case vide n'existe pas; un changement de format n'affectera donc que des cases pleines. Nous pouvons maintenant remplir le tableau du chéquier et ainsi vérifier que nos commandes de formats entier et décimal sont bien observées. Il serait également possible d'employer le format par défaut.

Terminons par une justification du texte que nous soulignerons pour améliorer la présentation de l'ensemble.

```
justifier,pleines,texte,droite,bloc b16:d26
justifier,pleines,texte,droite,bloc c11
justifier,pleines,texte,droite,bloc d29
justifier,pleines,texte,droite,bloc c31
```

Le tableau est trop grand pour être vu en entier sur l'écran. Pour voir les résultats obtenus parallèlement aux valeurs saisies en réponse aux fonctions **saisisn()** et **saisist()**, vous pouvez faire appel à la commande **Hublot** qui scinde l'écran, horizontalement ou verticalement, en deux fenêtres. La division a lieu là où vous placez le curseur.

Pour notre tableau, il est préférable de prévoir une division verticale. Vous placez donc le curseur au centre de la fenêtre puis vous lancez la commande **Hublot** :

```
hublot,vert.,fenêtres indépendantes
```

Pour déplacer le curseur d'une fenêtre à l'autre, appuyez sur **F4**. Le curseur étant dans la fenêtre gauche, vous pouvez afficher la case A1 puis appuyer sur **F4** pour l'amener dans la fenêtre droite afin d'y opposer la case C15.

ECARTS

Dans cet exemple, nous calculerons l'écart-type et l'écart moyen d'une série de chiffres. Grâce au système d'étiquettes de QL Abacus, les formules n'exigeront pratiquement aucune explication.

	A	B	C	D	E	F	G
1		ECART TYPE					
2		=====					
3		Valeur	Ecart	Coefficient			
4		5	-4.5	20.25			
5		6	-3.5	12.25			
6		7	-2.5	6.25			
7		8	-1.5	2.25			
8		9	-0.5	0.25			
9		10	0.5	0.25			
10		11	1.5	2.25			
11		12	2.5	6.25			
12		13	3.5	12.25			
13		14	4.5	20.25			
14							
15							
16	Moyenne	9.5	Variance	8.25			
17			Type	2.87228132			
18			-----				
19							
20							
21							
22							
23							
24							
25							
26							
27							
28							
29							
30							
31							
32							
33							
34							
35							
36							
37							
38							
39							
40							
41							
42							
43							
44							
45							
46							
47							
48							
49							
50							
51							
52							
53							
54							
55							
56							
57							
58							
59							
60							
61							
62							
63							
64							
65							
66							
67							
68							
69							
70							
71							
72							
73							
74							
75							
76							
77							
78							
79							
80							
81							
82							
83							
84							
85							
86							
87							
88							
89							
90							
91							
92							
93							
94							
95							
96							
97							
98							
99							
100							

Figure 5.4 Calcul de l'écart-type

De plus, nous utiliserons un format de grille impliquant un calcul par colonne et non plus par rangée.

De manière générale, une formule ne se rattache qu'à des cases situées au-dessus et à gauche de la case qui la renferme, son rang et sa colonne compris.

Si vous ne respectez pas cette règle, les résultats risquent d'être faussés. Vous pouvez généralement les corriger en obligeant un second calcul de la grille au moyen de la commande **Exécute**. Nous n'allons pas la respecter mais procéder à un calcul par colonne.

```
[B1] "ECART-TYPE
[B2] repro ("=",long(b1))
[B4] "Valeur
[C4] "Ecart
[D4] "Coefficient
[B5] col=rang()          {rangées de 5 à 14}
```

Cette dernière formule introduit des valeurs fictives dans les cases de la colonne B afin de vérifier l'application. Quand vous aurez terminé de remplir la grille, vous leur donnerez d'autres valeurs. Notre exemple ne prévoit que dix valeurs, mais vous pouvez bien sûr en saisir plus si vous le désirez.

```
[A16] "Moyenne
[B16] moy(valeur)          {rangées de 5 à 14}
[C5] col=valeur-$valeur.moyenne {rangées de 5 à 14}
[D5] col=c5*c5            {rangées de 5 à 14}
```

```
[C16] "Variance
[D16] moy(coef)                {rangées de 5 à 14}
```

Ces formules définissent la variance comme la moyenne des carrés des écarts.

```
[C17] "Type
[D17] rac(variance)           {colonnes D à D}
```

L'écart type est alors la racine carrée de la variance.

```
[D18] repro("-",long(chn(type.coef,3,0)))
```

Le format choisi est le format général afin que toutes valeurs puissent être traitées. Pour souligner, nous avons utilisé la fonction **chn** qui tire ainsi un trait sous la valeur inscrite dans la case supérieure (ici identifiée par "type.coef").

Vous pouvez améliorer la présentation en centrant le texte dans le bloc B4:D4 et en justifiant à gauche les nombres saisis dans le bloc B16:D17.

Si vous essayez maintenant de reprendre cet exemple avec différentes valeurs dans la colonne B, les résultats seront faux du fait que les calculs seront effectués rangée par rangée et de haut en bas. Tout changement que vous introduirez sera calculé d'après une moyenne incorrecte (puisque la nouvelle moyenne ne sera calculée qu'après les écarts). Les calculs doivent donc être effectués par colonne, de gauche à droite ; c'est ce que permet la commande **Format**.

Cette commande vous propose plusieurs options. Vous choisirez dans ce cas la troisième, ORDRE de calcul, en tapant O, pour obtenir "COL" à la fin de la ligne avant d'appuyer sur **ENTREE**. Vous pouvez maintenant modifier vos valeurs dans la colonne B ; les résultats seront corrects puisque la nouvelle moyenne sera calculée avant les écarts. Si ce système de calcul par colonne est parfois très utile, il est néanmoins préférable de ne pas y avoir recours trop fréquemment du fait qu'il est beaucoup plus lent que le calcul par rangée.

Quand vous sauvegardez une grille sur un fichier Microdrive, vous y incorporez également toutes les options sélectionnées avec la commande **Format** et vous les retrouverez donc quand vous chargerez de nouveau votre fichier.

BUDGET FAMILIAL

Cet exemple vous permet de prévoir vos dépenses de l'année. Vous enregistrez les dépenses que vous envisagez de faire selon un certain nombre de rubriques par trimestre. Vous obtenez les totaux trimestriels ainsi que les dépenses annuelles et la moyenne mensuelle.

Attendez d'avoir établi le tableau avant d'y inscrire des chiffres. Vous pourrez ainsi modifier l'affichage numérique au moyen de l'option par défaut de la commande **Notation** que nous expliquerons par la suite.

	A	B	C	D	E	F	G	H	I	J	K	
1	BUDGET FAMILIAL											
2	=====											
3	-----											
4	-----											
5			DEPENSES ESTIMATIVES									
6	Rubriques		jan-mars	avr-juin	juil-sep	oct-déc						
7	-----											
8	Prêt/Loyer		F4000.00	F4000.00	F4000.00	F4000.00						
9	Impôts locaux			F4500.00								
10	Gaz		F1500.00	F800.00	F600.00	F1500.00						
11	Electricité		F400.00	F300.00	F300.00	F400.00						
12	Eau			F350.00		F350.00						
13	Téléphone		F1500.00	F1500.00	F1500.00	F1500.00						
14	Assurances											
15	Vêtements											
16	Crédits											
17	Vignette											
18	Essence											
19	Redev. Télé											
20	Epargnes											
21	-----											
22	Total		F7400.00	F11450.00	F6400.00	F7750.00						
23	-----											
24												
25			annuels	mensuels								
26												
27	Frais		F33000.00	F2750.00								
28			-----	-----								
29												
30												
31												
32												
33												
34												
?												
	CASE A34 LIMITES A1:K28 MEMOIRE 16K											
	CONTENU VIDE											

Figure 5.5 Exemple de budget familial

```
[D1] "BUDGET FAMILIAL
[D2] repro("=",long(dl))
```

Nous allons maintenant dessiner le tableau

```
[A4] rang=repro("-",larg()+1) {colonnes A à K}
[A5] col ="!" {colonnes 5 à 20}
```

Et, pour terminer :

```
Grille>largeur 16 DE b JUSQU'A b
Grille>largeur 9 DE d JUSQU'A j
Grille>largeur 1 DE a JUSQU'A a
Grille>largeur 1 DE c JUSQU'A c
Grille>largeur 1 DE e JUSQU'A e
Grille>largeur 1 DE g JUSQU'A g
Grille>largeur 1 DE i JUSQU'A i
Grille>largeur 1 DE k JUSQU'A k

Reproduire la case a5,dans le bloc c5:c22
Reproduire la case a5,dans le bloc e6:e22
Reproduire la case a5,dans le bloc g6:g22
Reproduire la case a5,dans le bloc i6:i22
Reproduire la case a5,dans le bloc k5:k22
Reproduire la case a4,dans le bloc b7:j7
Reproduire la case a4,dans le bloc b21:k21
Reproduire la case a4,dans le bloc c23:k23

[A7] " !-
[F5] "DEPENSES ESTIMATIVES
[B6] "Rubriques
[D6] "jan-mars
[F6] "avr-juin
[H6] "juil-sep
[J6] "oct-déc

[B8] "Prêt/Loyer
[B9] "Impôts loc.
[B10] "Gaz
[B11] "Electricité
[B12] "Eau
[B13] "Téléphone
[B14] "Assurances
[B15] "Vêtements
[B16] "Crédits
[B17] "Vignette
[B18] "Essence
[B19] "Redev.télé
[B20] "Epargnes

[B22] "Total

[D22] somme(col) {rangées 8 à 20}
[F22] somme(col) {rangées 8 à 20}
[H22] somme(col) {rangées 8 à 20}
[J22] somme(col) {rangées 8 à 20}

[D25] "annuels
[F25] "mensuels
[B27] "Frais
```

```
[D27] somme(d22:j22)
[F27] ann.frai/12
[D28] repro("_",long(chn(ann.frai,0,2))+1)
[F28] d28
```

Vous remarquerez que le soulignement des deux derniers montants suppose un format monétaire. Sa longueur est celle du format décimal avec deux décimales, plus un (pour le symbole monétaire).

Procédons maintenant à la justification ; plaçons le texte du bloc B22:B27 à droite (les totaux trimestriels et les frais) et alignons les chiffres au-dessus des cases contenant les frais annuels et mensuels.

Vous devez également modifier le format d'affichage numérique. Du fait que de nombreuses cases sont toujours vides, il convient ici d'adopter le format par défaut (cases vides).

La commande suivante donnera à l'ensemble du tableau le format monétaire :

```
notation,vides,monétaire,signe moins
```

La Figure 5.5 présente un affichage en format décimal, avec deux décimales; seuls les frais mensuels et annuels sont donnés en format monétaire. Pour obtenir ce résultat, poursuivez avec la commande **Notation** :

```
notation,vides,décimale,décimales 2
notation,pleines,monétaire,signe moins,bloc d27:f27
```

Cette dernière commande peut faire intervenir l'option cases pleines du fait que les cases concernées existent.

Maintenant, pour inscrire vos prévisions de dépenses sur ce tableau, vous amenez le curseur aux cases correspondantes et tapez les montants. Pour aller plus vite, appuyez sur F5 (accès à) et indiquez la référence de la case recherchée, par exemple :

```
avr.gaz
```

GRAPHIQUE A BARRES (avec cadrage automatique)

Le graphique affiche douze valeurs affectées aux mois. Les valeurs sont lues de douze cases prévues au sommet du tableau. L'échelle verticale est réglée automatiquement pour que toutes les valeurs apparaissent à l'écran. Les données ne doivent compter que des chiffres positifs.

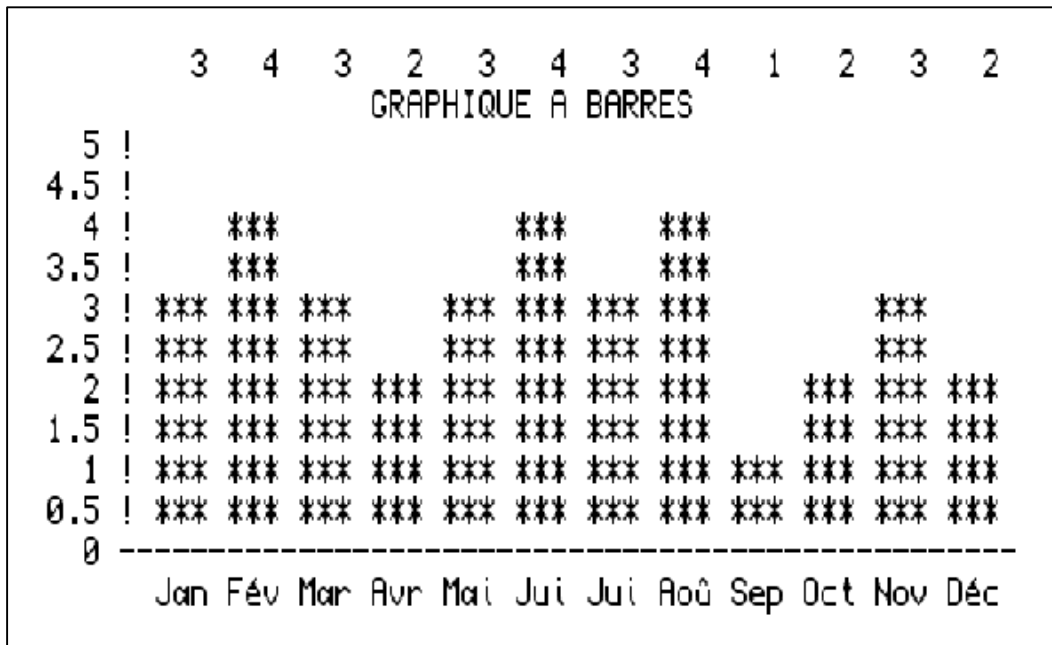


Figure 5.6 Graphique à barres cadré

Définissez en premier lieu la largeur des colonnes : 5 pour A, 1 pour B et trois de C à N (option L de la commande **Grille**).

```
[C2] rang=0          {colonnes C à N}
```

La seconde rangée contiendra les valeurs à afficher qui restent pour l'instant des zéros fictifs.

```
[F3] "GRAPHIQUE A BARRES CADRE
```

```
[P2] ent(maxi(c2:n2)/5+1,)*5
```

```
[Q2] ent(mini(c2:n2)/5)*5
```

Les cases P2 et Q2 contiennent les minimas et les maximas de l'échelle verticale. Elles sont choisies à bonne distance du graphique même afin de ne pas apparaître sur l'affichage final. Leurs valeurs initiales sont respectivement de cinq et zéro.

La fonction **maxi()** relève le maxima, soit la valeur numérique la plus élevée, du bloc spécifié par l'argument. De même, la fonction **mini()** relève le minima, soit la plus faible valeur du bloc.

Etudions la formule introduite dans la case Q2. La fonction **mini()**, comme nous l'avons indiqué, relève le minima de bloc spécifié; cette valeur est alors divisée par cinq. La fonction **ent()** élimine ensuite les décimales du résultat de la division. Si, par exemple, le minima est de 13, sa division par 5 donne 2.6 que la fonction **ent()** simplifie pour donner 2. Quand ce résultat est multiplié par 5, nous obtenons 10, soit le plus grand multiple de 5 inférieur au minima.

La formule introduite dans la case P2 fonctionne selon le même principe, à cette différence près qu'elle relève le maxima, y ajoute 1 et multiplie le tout par 5. Si donc nous supposons un maxima de 21, la formule donne 25, soit le plus petit multiple de 5 supérieur au maxima.

Les deux valeurs de ces cases encadreront donc toujours les valeurs inscrites dans les cases C2 à N2 et leur différence sera toujours un multiple de 5.

La formule suivante affichera l'échelle verticale du graphique dans la colonne A

```
[A4] col=$q2+(14-rang())*( $p2-$q2)/10          {rangées 4 à 14}
```

L'intervalle entre chaque valeur de l'échelle verticale est défini par $(P2-Q2) / 10$. Vous remarquerez que nous avons choisi un multiple de cinq comme différence entre P2 et Q2 afin de donner une valeur simple à cet intervalle.

L'intervalle est multiplié par un nombre, **(14-rang())**, commençant à zéro à la quatorzième rangée et augmentant de un pour arriver à dix à la quatrième rangée. Le résultat est ajouté au minima défini par la case Q2 pour produire le nombre de chaque case.

En conséquence, la valeur dans la case Q2 est affichée dans A14, celle de P2 dans A4 et les cases intermédiaires présentent une série de valeurs progressives entre ces deux limites.

```
[B4] col="!"                               {rangées 4 à 14}
[B14] rang=repro("-",larg()+1)            {colonnes B à N}
[C15] rang=mois(col()-2) (jusq 3)        {colonnes C à N}
```

Ces formules tracent les axes du graphique et ajoutent sous l'axe horizontal les mois de l'année. De plus, nous complétons la troisième formule avec l'opérateur de rupture de chaîne ("jusq"), utilisé en SuperBASIC, pour abrégé à trois lettres le nom de chaque mois.

```
[C4] si(indice(1,rang()) > indice(col(),2)," ","****")
```

Toutes les barres seront établies grâce à cette formule qui doit maintenant être appliquée à toutes les cases du graphique :

Reproduire la case c4,dans le bloc c4:n13

Quelques explications s'imposent. La fonction **si()** décide s'il s'agit d'afficher une section d'une barre. Elle prend trois arguments, le premier étant une expression devant donner un résultat numérique. Si ce résultat est différent de zéro, la case affiche le second argument qui, lui, peut être numérique ou alphabétique. Par contre, si le résultat est égal à zéro, la case affiche le troisième argument. Ici encore, l'argument peut être numérique ou alphabétique.

Dans chaque case, la formule compare le nombre dans la première colonne de cette rangée (la valeur identifiant l'axe vertical) au nombre dans la seconde rangée de la même colonne (la valeur à afficher sur le graphique). Si l'étiquette de l'axe a une valeur supérieure à la valeur à afficher, la condition est vérifiée (évaluée à 1) et rien n'apparaît dans la case. Si l'étiquette de la case est inférieure ou égale à la valeur à afficher, la condition donne zéro et trois astérisques apparaissent. Le résultat est le tracé d'une barre à la hauteur voulue dans chaque colonne.

Du fait qu'une seule formule est appliquée à toutes les cases, leurs références ne peuvent être ni absolues, ni relatives. La référence aux valeurs d'affichage doit varier quand nous changeons de colonne (autrement dit, elle doit être relative dans une colonne); nous devons néanmoins toujours faire référence à la seconde rangée quand nous descendons d'une rangée à une autre. Notre référence de case doit être relative aux colonnes et absolue par rapport aux rangées.

Cette exigence est satisfaite par la fonction **indice()**. Elle prend deux paramètres, un numéro de colonne et un numéro de rangée, renvoyant ensuite le contenu de la case spécifiée. Nous sommes ainsi en mesure de définir toutes références à la fois absolues et relatives. Par exemple :

Fonction	réf.col	réf.rang
indice(5,5)	absolue	absolue
indice(col(),5)	relative	absolue
indice(5,rang())	absolue	relative
indice(col(),rang())	relative	relative

La fonction **indice(col(),2)** renvoie donc le contenu de la case à l'intersection de la rangée deux et de la colonne en cours et **indice(1,rang())** renvoie le contenu de la case à l'intersection de la colonne un (A) et de la rangée en cours.

Inscrivez maintenant différentes valeurs dans les cases C2 à N2 et observez votre graphique se construire.

CALCUL DES REMBOURSEMENTS D'UN PRET

Cet exemple vous indique comment calculer les remboursements mensuels d'un prêt hypothécaire. Vous préciserez le montant du prêt, le taux d'intérêt, l'échéance en années et le mois du premier remboursement. Les versements mensuels seront alors calculés et affichés et le système établira le tableau complet des remboursements jusqu'à l'échéance du prêt. Le tableau signale également le montant toujours dû au début de chaque mois.

Plusieurs calculs feront appel à des valeurs saisies en réponse aux questions de la fonction **saisisn()**.

Calcul des remboursements d'un prêt

Nous travaillerons tout d'abord sur la section de la grille qui acceptera vos saisies et calculera les remboursements mensuels. Quand vous aurez tapé les formules et ajouté quelques chiffres en réponse aux fonctions **saisisn()**, vous obtiendrez le résultat présenté à la Figure 5.7.

AIDE F1	CURSEUR ↑↓←→	VALEURS et FORMULES	TEXTE Guillemets pour commencer	COMMANDES F3
GUIDE F2	ACCES CASE F5	Saisie directe ENT pour finir		ANNULER ESC
A	B	C	D	E
1		CALCUL D'UN REMBOURSEMENT DE PRET		
2		=====		
3				
4	Prêt	F25000.00		
5	Taux	12.00%	Mois	
6	Terme	20	1	
7			(Janvier)	
8			REMBOURSEMENTS	
9			-----	
10		annuels	F3346.97	
11		mensuels	F278.91	
12			-----	
13				
14				
15				
16				
?				
CASE A1 LIMITE A1:H12 MEMOIRE 19K				
CONTENU VIDE				

Figure 5.7 Calcul des remboursements

```
[C1] "CALCUL DES REMBOURSEMENTS D'UN PRET
[C2] repro("=",long(c1))
[B4] "Prêt
[C4] saisissn("Montant du prêt")
```

Les trois saisies suivantes concernent le taux d'intérêt. La première, le taux indiciaire, s'inscrira à bonne distance de l'affichage (dans la case H4) afin de ne pas apparaître en même temps à l'écran. Omettez le signe "%" quand vous enregistrez ce taux (soit par exemple 12 et non pas 12%). La valeur requise par les autres formules est une fraction de ce taux (12 devient ainsi 0.12) dont le calcul est effectué par la formule inscrite dans C5.

```
[H4] saisissn("pourcentage taux")
[B5] "Taux
[C5] h4/100
```

```
[B6] "Terme
[C6] saisisn("Echéance en années (maximum 35)")

[E5] "Mois
[D6] "premier
[E6] saisisn("Mois premier paiement [Jan=1, Fév=2, etc]")
[E7] '(' + mois(e6) + ")"
```

Cette dernière formule est précédée d'une apostrophe et non pas de guillemets afin que QL Abacus interprète la saisie comme une formule et non pas comme un texte.

```
[D8] "REMBOURSEMENTS
[D9] repro("-",long(d8))
[C10] "annuels
[D10] c4*c5/(1-(1+c5)^(-c6))
```

Ce total annuel suppose que les intérêts sont calculés pour l'année puis ajoutés au prêt avant le commencement des versements.

```
[C11] "mensuels
[D11] d10/12
[D12] d9
```

QL Abacus dispose maintenant de suffisamment de données pour établir le plan de remboursement. Pour vous rendre mieux compte du fonctionnement du système, lancez la commande **Execute** et répondez aux questions posées.

Pour des raisons d'esthétique, nous pouvons modifier l'emplacement de certains chiffres au moyen de la commande **Notation**. Il est ici inutile de modifier le format numérique par défaut puisque, quand votre plan sera terminé, vous n'aurez pas d'autres chiffres à y inscrire.

```
notation,pleines,en p.cent,décimales 2,bloc c5
notation,pleines,monetaire,signe moins,bloc c4
notation,pleines,monetaire,signe moins,bloc d10:d11
```

La présentation sera encore plus claire si nous disposons les chiffres inscrits dans les rangées 4, 5 et 6 à gauche :

```
justifier,pleines,nombres,gauche,bloc c4:e6
```


Si vous avez contracté un emprunt, inscrivez vos propres chiffres. Mais ne vous attardez pas sur les remboursements des quelques premières années; ils peuvent être déprimants !

```
[C15] "PLAN DE REMBOURSEMENT
[C16] repro("_",long(c15))
[B18] "Année
[C18] rang=col()-2           {colonnes C à AK}
[B19] rang=repro("-",larg()+1) {colonnes B à AK}
[B20] col=mois(rang()-20+$e6) {rangées 20 à 31}
```

Ces saisies produisent les intitulés du plan ; il nous faut maintenant ajouter les formules qui calculeront les valeurs. Nous commençons par la première rubrique, le versement initial, qui est calculée en ajoutant les intérêts de la première année au montant du prêt.

```
[C20] c4*(1+c5)
```

Le reste de la première rangée est ensuite calculé en soustrayant le règlement annuel et en y ajoutant les intérêts de l'année en question. Du fait que le calcul doit s'arrêter au terme de 25 années, nous faisons appel à la fonction **si()**. Si le chiffre de l'année (stipulé par **col()-2**) est supérieur au terme de l'emprunt, un zéro vient s'inscrire dans la case.

```
[D20] rang=si((col()-2)>$c6,0,(c20-$d10)*(1+$c5))
{colonnes D à AK}
```

Il suffit d'une seule formule pour remplir le reste du plan. Nous inscrivons dans la première case une formule qui soustrait le remboursement mensuel du montant saisi dans la case située au-dessus. Nous utilisons ici encore la fonction **si()** pour que les calculs ne portent pas sur plus de 25 ans.

```
[C21] si((col()-2)>$c6,0,c20-$d11)
```

Vous pouvez alors lancer la commande **repro** pour reproduire la formule inscrite dans C21 dans le bloc D21:AK31.

```
reproduire la case c21,dans le bloc c21:ak31
```

Terminez enfin en ajoutant une dernière rangée qui indiquera le solde au terme de chaque année. Pour faciliter la lecture, il est pratique d'inscrire de nouveau le numéro des années.

```
[B33] rang=b18           {colonnes B à AK}
[A35] "Solde fin d'année
[C35] rang=si((col()-2)>$c6,0,c31-$d11)   {colonnes C à AK}
```

L'ensemble du plan ainsi que les soldes de fin d'année peuvent être établis en format monétaire ou en format par défaut avec deux décimales. Vous terminerez ainsi votre travail avec la commande **Notation** en spécifiant les blocs C20:AK31 et C35:AK35.

ANALYSE DE FOURIER

Le savant Fourier a démontré qu'une onde progressive peut être formée à partir d'une série d'ondes sinusoïdales ou cosinusoidales d'amplitudes et de fréquences données. La création d'ondes complexes assurée de cette manière s'intitule la synthèse de Fourier; de nos jours, elle est fréquemment employée dans des synthétiseurs de musique.

Le processus inverse, c'est à dire la décomposition d'une onde de forme complexe en un certain nombre d'ondes sinusoïdales et cosinusoidales pures, est ce qu'on appelle une analyse de Fourier. L'exemple suivant vous permettra de procéder à une analyse de Fourier sur n'importe quelle onde. Il vous suffira d'indiquer son amplitude à seize intervalles équidistants et laisser les formules se charger du reste. Les formules supposent que l'onde se reproduira de manière identique après la seizième valeur ; autrement dit, la dix-septième sera identique à la première, la dix-huitième à la seconde, etc.

Calcul de la conversion de Fourier

Du fait que les calculs prennent un certain temps, il est utile d'annuler le système d'auto-calcul (commande **Format**), avant de taper l'exemple.

```
[C1] "ANALYSE DE FOURIER
[C2] repro("=",long (c1))
[B3] "Fonction:
[A7] "Valeurs
[A8] "à traiter
```

Les valeurs saisies seront enregistrées dans les seizièmes cases, de B9 à B24.

```
col=rang()-9      {rangées 9 à 24}
```

Les composantes cosinusoidales

Nous établissons ensuite les en-têtes du tableau qui calculeront les composantes cosinusoidales de l'onde. Le résultat comprend les valeurs de toutes les ondes de ce type incorporées dans les paramètres donnés en entrée.

```
[E3] "convertir :
[E4] "cosinus
[D6] "cyclesrang=col()-5    { colonnes E à T}
[D8] "Exemple
```

Etonnamment, une seule formule convertit tous les cosinus. Dans chaque rangée, le paramètre sera multiplié par le cosinus d'un angle (en radians) calculé comme suit :

```
angle=2* pi()*numérorang*numérocol/16
```

Les numéros colonne et rangée sont, respectivement, ceux donnés dans la rangée portant l'étiquette "cycle" et ceux inscrits dans la colonne portant "exemple" Chaque série s'étend de zéro à quinze. Le dernier diviseur est simplement le nombre de points dans l'entrée (ou la sortie).

```
[E9] indice(2,rang()*cos(pi()*(rang()-9)*(col()-5)/8)
```

Lançons maintenant la commande **Repro** pour recopier le contenu de la case E9 dans les cases du bloc E14 à T24.

Le résultat final, les seize valeurs de sortie, sera obtenu en additionnant le contenu de chaque colonne.

```
[A26] "Composantes
[E26] rang=somme(col)      {rangées 9 à 24, colonnes E à T}
```

Les composantes sinusoïdales

Le calcul des composantes sinusoïdales est exactement identique à celui des composantes cosinusoidales. Les valeurs obtenues seront celles de toutes les ondes de ce type présentes dans les paramètres donnés.

```
[X4] "Sinus
[X6] rang=col()-24 {colonnes X à AM}
[X9] indice(2,rang())*sin(pi()*(rang()-9)*(col()-24)/8)
```

Avec la commande **Repro**, nous reproduisons le contenu de la case X9 dans le bloc X10 à AM24 pour remplir le reste du tableau, ainsi que celui de la case C9 dans la colonne V, de V9 à V24 (pour recopier les "exemples").

```
[X26] rang=somme(col) {rangées 9 à 24, colonnes X à AM}
```

Le spectre des fréquences

Une onde qui n'est pas purement sinusoïdale ou cosinusoidale produit généralement des composantes des deux types, ainsi que parfois des éléments négatifs, notamment quand vous procédez à la conversion de nombreux types d'ondes. Un calcul supplémentaire s'avère nécessaire pour associer ces deux types de composantes et éliminer les négatives. Ce calcul additionnera les carrés des composantes sinusoïdales et cosinusoidales. On obtiendra ainsi ce qu'on appelle généralement un spectre de fréquences. Pour éviter d'avoir trop de valeurs dans notre simple affichage graphique, nous calculerons ici la racine carrée de ce spectre.

```
[C28] "Spectre
[E28] rang=rac(comp.cos*comp.cos + comp.sin*comp.sin)
{colonnes E à T}
```

Affichage graphique de la conversion de Fourier

Pour plus de clarté, les résultats peuvent être visualisés sous forme d'un graphique. Le meilleur moyen de produire un graphique de haute qualité est de faire appel à la commande **EXPort** ; cette commande crée des fichiers qui seront interprétés par le programme Easel d'après les paramètres d'entrée et de sortie des calculs. Cependant, les commandes suivantes vous permettront déjà d'obtenir une présentation graphique très simple sur la grille.

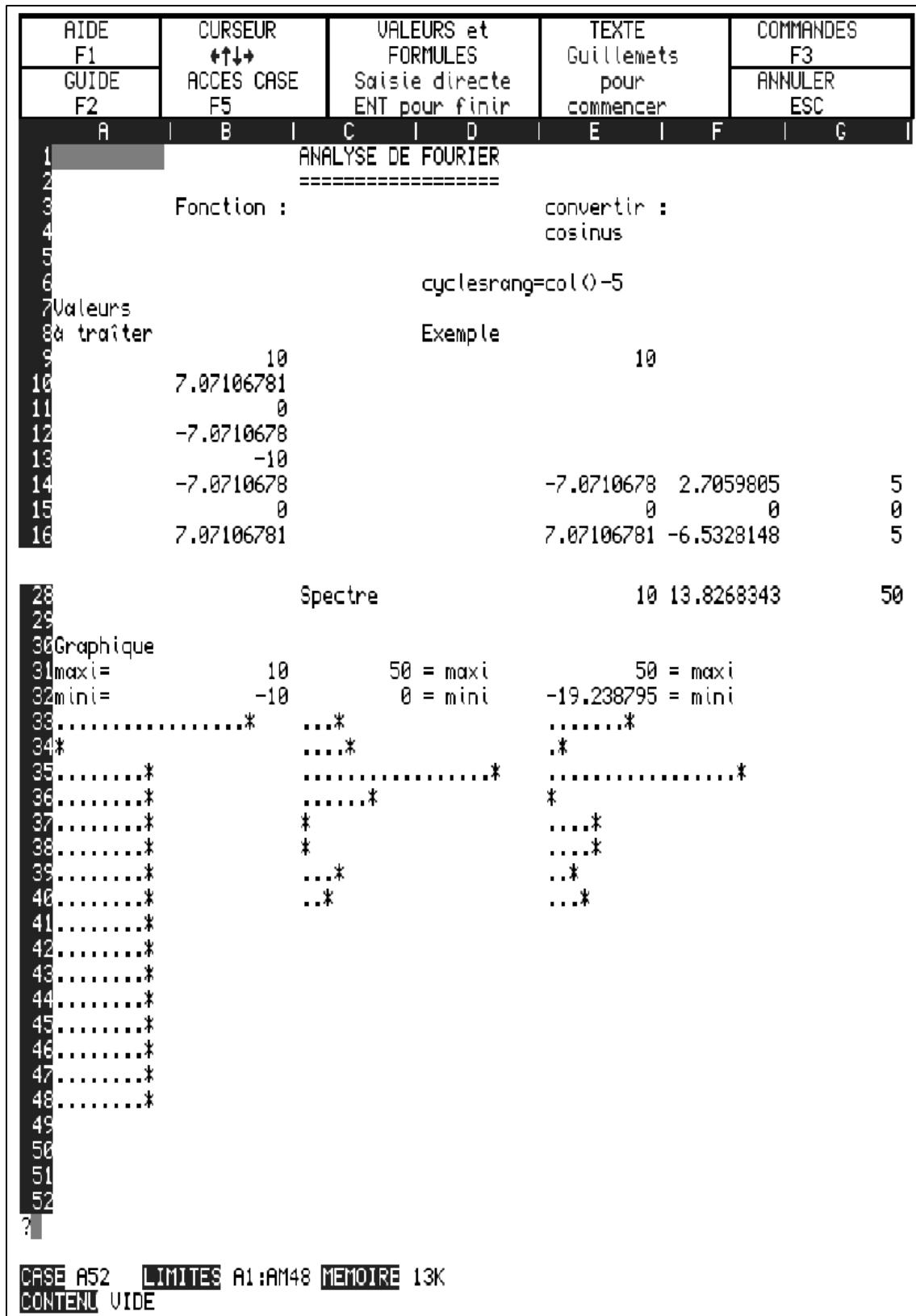


Figure 5.9 Simple affichage graphique

Les graphiques obtenus seront réduits à la moitié du graphique initial puisque la fréquence discernable la plus élevée est numériquement égale à la moitié du nombre de points d'entrée. Toutes les informations sont présentes dans la première partie des résultats.

La première partie produit un graphique à barres des paramètres donnés.

```
[A30] "Graphique
[A31] "maxi=
[B31] maxi(col) {rangées 9 à 24}
[A32] "mini=
[B32] mini(col) {rangées 9 à 24}
[A33] col=repro(".",(fonc.maxi-$fonc.mini)*18/($fonc.maxi-
$fonc.mini+1))+"" {rangées 33 à 48}
```

La seconde partie produit le graphique du spectre des fréquences.

```
[D31] "= maxi
[C31] maxi(e28:t28)
[D32] "= mini
[C32] 0
[C33] col=repro(".",(indice(rang()-28,28)-$mini.spe)
*18/($maxi.spe-$mini.spe+1))+"" {rangées 33 à 40}
```

La troisième partie dessine les graphiques des composantes cosinusoidales.

```
[F31] "= maxi
[E31] maxi(e26:t26)
[F32] "= mini
[E32] mini(e26:t26)
[E33] col=repro(".",(indice(rang()-28,26)-$mini.cos)
*18/($maxi.cos-$mini.cos+1))+"" {rangées 33 à 40}
```

Nous terminons enfin avec le tracé des composantes sinusoidales.

```
[Y31] "= maxi
[X31] maxi(x26:am26)
[Y32] "= mini
[X32] mini(x26:am26)
[X33] col=repro(".",(indice(rang()-9,26)-$mini.sin)
*18/($maxi.sin-$mini.sin+1))+"" {rangées 33 à 40}
```

Passons à la pratique

Vous n'avez plus maintenant qu'à inscrire vos paramètres dans les cases B9 à B24.

Peu importe ces paramètres, mais nous vous proposons les suivants pour un premier essai.

```
[B9] col=10*cos(pi()*(rang()-9)/8) {rangées 9 à 24}
[B9] col=10*cos(pi()*(rang()-9)/4) {rangées 9 à 24}
[B9] col=10*sin(pi()*(rang()-9)/8) {rangées 9 à 24}
[B9] col=10*sin(cos(pi()*(rang 0-9)/8)) {rangées 9 à 24}
[B9] col=10 {rangées 9 à 24}
```

N'oubliez pas enfin que, du fait que le système d'auto-calcul est annulé, vous devez lancer la commande **Execute** pour calculer chaque résultat.

La prévision de nombreuses étiquettes présente un autre avantage : vous pouvez amener la fenêtre aux sections les plus intéressantes de la grille avec la touche **F5**; à sa question "accès à", vous indiquerez alors simplement l'étiquette de la case à titre de référence.

CHAPITRE 6

GUIDE DE REFERENCE

TOUCHES DE FONCTION

En plus des touches F1, F2 et F3 communes aux quatre programmes, vous disposez avec QL Abacus de deux autres touches de fonctions :

- F4** qui déplace le curseur d'une fenêtre à une autre
- F5** qui amène le curseur dans une case donnée.

REFERENCES DE CASES

Pour faire référence à une case, des rangées ou des blocs, vous avez deux options : indiquer leur référence avec des lettres et des chiffres ou leur affecter des étiquettes alphabétiques.

Références de cases

La référence d'une case se compose de celle de sa rangée et de celle de sa colonne. La grille compte 64 colonnes de A à BL et 255 rangées de 1 à 255. Exemples :

A1 AC13 BD200

Références de blocs

La référence d'un bloc se compose de deux références de cases séparées par deux points. Sans ces deux points, le système ne comprendra pas votre référence. La première référence se rattache à la case supérieure gauche du bloc et la seconde à la case inférieure droite. Exemples :

B5:D9
AZ23:BA155

Référence d'une rangée ou d'une colonne

Une section de colonne ou de rangée peut être considérée comme un bloc d'une colonne de largeur (ou d'une rangée de hauteur). Vous avez donc recours au système de référence de bloc, soit par exemple :

A3 : L3 {cases A à L de la rangée 3}
D7:D11 {cases 7 à 11 de la colonne D}

Identificateurs de blocs

Deux identificateurs sont prévus : rang et col. Ils se rattachent respectivement aux cases de la rangée ou de la colonne en cours (à l'intersection de la case contenant l'identificateur).

Quand vous employez un tel identificateur dans une formule, QL Abacus vous demande de préciser le bloc exact de cases et vous suggère toujours à l'avance des points raisonnables de départ et d'arrêt que vous pouvez accepter ou modifier.

Ces identificateurs assument deux rôles. Remplir la colonne ou la rangée en cours, comme dans l'exemple suivant :

`rang=(formule) ou col=(formule)`

Ou bien servir d'argument à toute fonction demandant la précision d'un bloc, soit par exemple : `nombcase(rang)`. Vous pouvez naturellement ne les utiliser de cette manière que quand vous voulez simplement faire référence aux cases d'une seule rangée ou d'une seule colonne.

Leurs deux rôles s'associent également librement. Par exemple :

`col = moy(rang)`

Chaque fois que vous prévoyez un identificateur dans une formule, QL Abacus vous demande de spécifier un bloc.

Références de cases relatives et absolues

QL Abacus suppose normalement que toutes les références de cases sont relatives, l'important est donc l'écart entre la case comportant la référence et la case que vous désignez. Quand vous reproduisez une telle référence dans une autre case, les références sont modifiées afin de maintenir cet écart relatif. Supposons par exemple qu'une formule dans la case B2 contienne une référence à la case A1 (une colonne à gauche et une rangée au-dessus). Quand la formule dans B2 est reproduite dans D4, elle se rattache alors à la case C3 (ici encore une colonne à gauche et une rangée au-dessus).

Cet exemple est illustré à la Figure 6.1. Une formule dans la case X contient une référence à la case grise. Quand cette formule est reproduite dans la case Y, elle se rattache alors à la case noire. Les deux cases de chaque paire ont des positions relatives identiques.

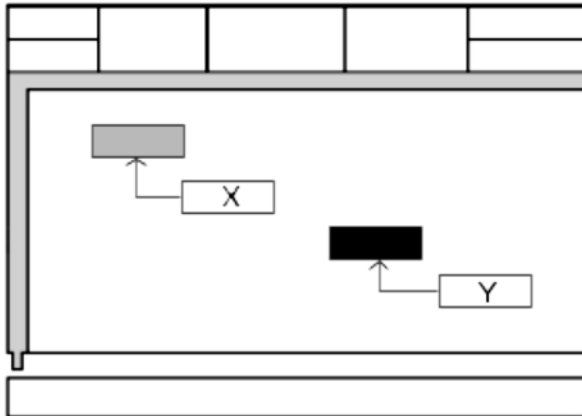


Figure 6.1 Références de cases relatives

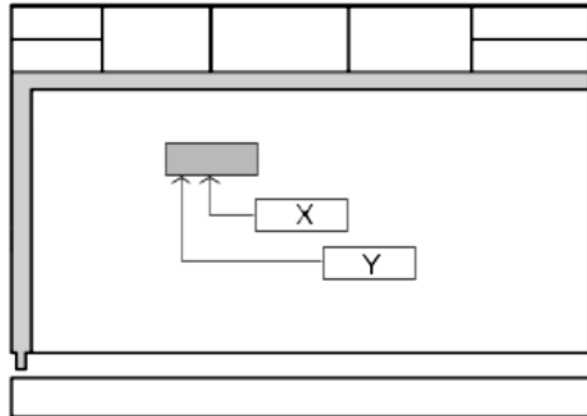


Figure 6.2 Références de cases absolues

Introduisons maintenant la formule $A1*2$ dans la case A2 avant de lancer la commande **Repro** pour la reproduire dans les cases du bloc B2:G2. Vous obtenez alors :

Case :	A2	B2	C2	D2	E2	F2	G2
Contenu :	$A1*2$	$B1*2$	$C1*2$	$D1*2$	$E1*2$	$F1*2$	$G1*2$

Pour convertir une référence relative en une référence absolue, vous la préfixez du signe \$. La référence ne sera alors plus modifiée quand vous reproduirez la formule dans d'autres cases. Par exemple, si B2 contient la référence $\$A1$, toute reproduction de la formule contiendra également $\$A1$. Une étiquette peut également être une référence absolue, p. ex. $\$mars,coûts$.

La Figure 6.2 montre l'effet d'une référence absolue. Une formule dans la case X contient une référence absolue à la case grise. Sa reproduction dans la case Y se rattache à cette même case grise.

Reprenons notre exemple mais avec une référence absolue. Introduisez la formule $\$A1*2$ dans A2 et, avec la commande **Repro**, reproduisez-la dans les cases B2 à G2. Vous obtenez dans ce cas :

Case :	A2	B2	C2	D2	E2	F2	G2
Contenu :	$\$A1*2$	$\$A1*2$	$\$A1*2$	$\$A1*2$	$\$A1*2$	$\$A1*2$	$\$A1*2$

Vous trouverez d'autres précisions à ce sujet dans la description de la fonction **indice**. Tout bloc de cases (avec identificateurs, rang et col) est toujours relatif.

ETIQUETTES

Étiquettes de rangées et de colonnes

Une étiquette est une case contenant un texte. Ce texte ne doit comprendre que des lettres et des chiffres. La case ainsi libellée identifie une rangée ou une colonne de la grille. Une étiquette peut également se rattacher à une seule case mais elle ne peut jamais remplacer une référence de bloc.

Quand vous désignez une étiquette dans une expression ou une formule, QL Abacus adopte une série de règles pour déterminer s'il s'agit d'une rangée, d'une colonne ou d'une case :

1. La rangée et la colonne se recoupant au niveau de l'étiquette sont analysées (à droite et en-dessous) pour relever des données numériques.
 - a. Si une seule donnée de rangée est relevée, l'étiquette se rattache à la rangée. à partir de la donnée relevée.
 - b. Si une seule donnée de colonne est relevée, l'étiquette se rattache à la colonne, à partir de la donnée trouvée.
 - c. Si l'analyse relève des données dans la rangée et dans la colonne, le choix est établi d'après la donnée la plus proche de la case libellée.
2. Si aucune décision ne peut être prise à l'étape 1) et que l'étiquette est à gauche d'une expression, elle est affectée d'après toute étiquette située à droite. Par exemple, si "coûts" est l'étiquette d'une rangée :

$$\text{Ventes} = \text{Coûts} * 0.5$$

et "Ventes" devient également une étiquette de rangée.

Si l'application de ces deux règles n'aboutit pas, QL Abacus vous signale qu'il ne comprend pas le sens de l'étiquette.

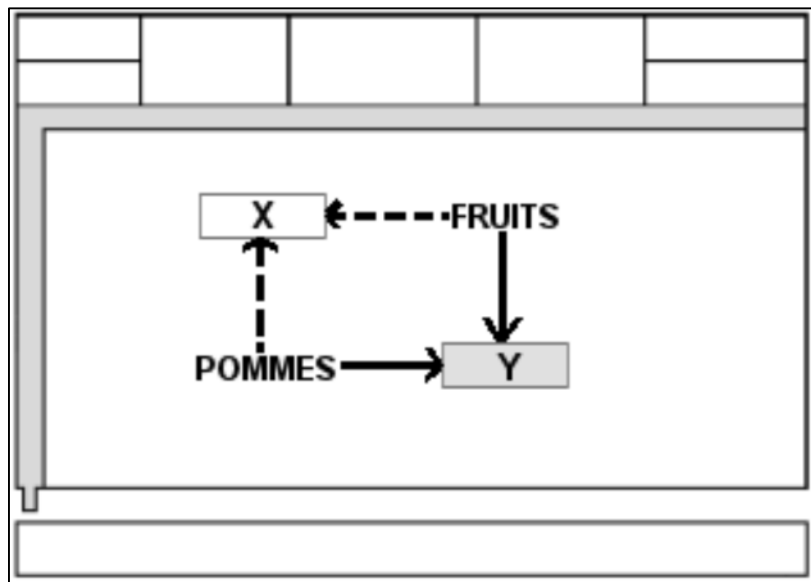


Figure 6.3 Affectation d'une étiquette à une case

Étiquettes de cases

La référence d'une case exige deux étiquettes séparées par un point. Si vos deux étiquettes sont "fruits" et "pommes"; votre référence sera :

`fruits.pommes`

(que vous pouvez d'ailleurs abrégé en fr.po). Peu importe l'ordre des deux étiquettes, pommes.fruits, po.fr, etc. auraient le même effet.

Une telle référence se rattache à une case située à l'intersection des rangées et des colonnes contenant les étiquettes mais, comme l'indique la Figure 6.3, il en existe deux (indiquées par X et Y).

La case choisie sera celle située dans la colonne la plus à droite et dans la plus inférieure des deux rangées. D'après notre exemple précédent, le choix portera sur la case Y. Placez donc toujours vos étiquettes au-dessus ou à gauche des cases auxquelles elles se rattachent.

FORMULES

Une formule est tout regroupement autorisé de fonctions, de références de cases, d'étiquettes et d'opérateurs arithmétiques. Exemples :

```
A1*B3
mois(col()-1)
si(souschn(B6,"est."),1,0)
repro("_",long(G23))+": "
```

Formules permanentes

Chaque nouvelle formule utilisée dans une ou plusieurs cases est mémorisée indépendamment dans une liste de formules permanentes. Une formule permanente peut ainsi apparaître dans une case ou dans plusieurs. Quand vous remplissez des cases par rangée ou colonne ou au moyen des commandes **Copie** ou **Repro**, toutes renferment alors une formule permanente commune. Si une formule permanente contient des références de cases relatives, celles-ci sont modifiées de manière à convenir à chaque case. Les formules peuvent donc sembler superficiellement différentes mais restent basées sur une même formule permanente.

Vous pouvez modifier toutes les copies d'une formule en éditant une seule d'entre elles. Si vous lancez la commande **Modifie** pour éditer une copie, vous modifiez sa formule permanente ainsi que toutes les copies correspondantes.

LES COMMANDES

Toutes les commandes dont vous disposez avec QL Abacus sont décrites dans cette section.

MODIFIE (M)

Cette commande vous permet de modifier le contenu d'une case. Le contenu où se trouve le curseur est reproduit sur la ligne de saisie ; vous pouvez alors le modifier comme indiqué dans l'introduction. Quand vous appuyez ensuite **ENTREE**, la case affiche la nouvelle version.

COPIE (C)

Cette commande recopie le contenu des cases d'un bloc spécifié dans un nouveau bloc de la grille. QL Abacus vous demande en premier lieu d'indiquer les références du bloc, soit par exemple A1:B3, après quoi vous appuyez sur **ENTREE**. Vous devez ensuite spécifier la référence de la case supérieure gauche du nouveau bloc et appuyer sur **ENTREE**. Le bloc est alors reproduit en cet endroit.

FORMAT (F)

La commande **Format** modifie la présentation générale du tableau, que l'affichage soit sur un téléviseur ou moniteur. Plusieurs possibilités vous sont proposées et, quand votre choix est fait, il reste en vigueur jusqu'à ce que vous l'annuliez. Quand vous sauvegardez votre travail, ces choix sont également sauvegardés (à l'exception de l'option écran), vous les retrouverez donc quand vous rechargerez votre fichier.

Cependant, une modification des options implicites n'affecte pas QL Abacus. Vous les définissez à votre gré chaque fois que vous chargez le programme à partir de SuperBASIC.

Quand vous avez choisi vos options, vous revenez à l'écran de travail en appuyant sur **ENTREE**. Vous disposez des options suivantes :

AUTO-CALCUL après chaque entrée

Met en œuvre ou annule l'auto-calcul. Chaque fois que vous appuyez sur la touche A, l'option passe de oui à non et vice-versa.

Sur OUI, tout le tableur est recalculé après la saisie de chaque donnée. Sur NON, le tableur n'est recalculé que quand vous lancez la commande **Exécute**. L'option est normalement sur OUI.

VIDE si zéro

Valeurs nulles représentées par un zéro ou par une case vide. Normalement, le système affiche un zéro dans une case mais, si vous préférez afficher une case vide, vous tapez **V** avant d'appuyer sur **ENTREE**.

Vous n'obtiendrez une case vide qu'en cas de valeur nulle. Si par exemple vous avez choisi un format avec deux décimales et que la valeur de la case est de 0.003, l'écran affichera 0.00 et non pas une case vide du fait que la valeur n'est pas véritablement nulle.

ORDRE de calcul

Calcul du tableur par RANGée ou par COLonne. L'option permute chaque fois que vous appuyez sur la touche **O** (comme dans le cas de l'auto-calcul). Le choix établi se répercutera également sur l'auto-calcul et sur la commande **Exécute**. L'option de base est le calcul par rangée.

ECRAN 80, 64, 40 colonnes

Sélectionne la largeur de l'affichage. Vous tapez **8, 6** ou **4** (puis **ENTREE**) pour choisir 80, 64 ou 40 caractères par ligne. Le système est normalement réglé sur 80 ou 40 (80 dans le cas d'un moniteur et 40 dans celui d'un téléviseur).

NOUVELLE page pour chaque tableau

Impression page par page ou non à la fin de chaque tableau. Cette option fonctionne comme l'auto-calcul et est normalement sur OUI.

INTERLIGNE d'impression

Détermine l'interligne sur la sortie imprimée. Pour le modifier, appuyez sur **I** puis choisissez entre **0, 1** et **2**. Inutile ici d'appuyer sur **ENTREE**. Pour obtenir un interligne double, vous spécifierez **1**. L'interligne de base est l'interligne simple, **0**.

LIGNES par page

Précise le nombre de lignes sur chaque page imprimée. Tapez un chiffre puis appuyez sur **ENTREE**. Le nombre initial de ligne est 66 et le maximum 255.

MONNAIE (p.ex. F, \$)

Précise le symbole monétaire à afficher. Indiquez simplement la première lettre du symbole sans appuyer sur **ENTREE**. Le symbole par défaut est le dollar (\$).

CARACTERES par ligne

Sélectionne la largeur d'impression. Indiquez le nombre de caractères par ligne désiré puis appuyez sur **ENTREE**. Le nombre initial est de 80 et le maximum de 255.

REPRO (R)

La commande **Repro** reproduit les données ou la formule prévues dans une case donnée dans toutes les cases du bloc spécifié.

Vous pouvez soit, accepter la proposition de QL Abacus en appuyant sur **ENTREE**, soit spécifier une case donnée, dans quel cas vous précisez le bloc où la case doit être reproduite avant d'appuyer sur **ENTREE**.

DOCU (D)

Propose un menu de sous-commandes ayant toutes trait à la gestion des fichiers sur Microdrive. Vous devez préciser des noms de fichiers, après quoi, vous tapez ? pour obtenir la liste de tous les fichiers enregistrés sur les Microdrive 1 ou 2.

Sous la commande **Docu**, plusieurs options vous sont proposées

Sauvegarde

Effectue une copie de sauvegarde d'un fichier QL Abacus. Vous devez préciser le nom du fichier à sauvegarder Il est vivement conseillé de sauvegarder tous les fichiers afin d'éviter tout risque de perte par accident ou autre.

Détruit

Efface un fichier donné sur Microdrive. **Attention : un fichier détruit est irrécupérable. Faites donc très attention avant d'utiliser cette option.**

Exporte

Exporte le fichier désigné. Le fichier en question est alors sauvegardé dans les autres programmes Archive, Easel ou Quill.

QL Abacus vous demande tout d'abord d'indiquer vers quel logiciel votre travail doit être exporté. S'il s'agit de Quill, appuyez simplement sur **ENTREE**; s'il s'agit d'Archive ou d'Easel, tapez **A** ou **E**.

Quel que soit le logiciel d'exportation, vous devez ensuite préciser le bloc concerné, avant d'appuyer sur **ENTREE**.

Si vous avez choisi de transmettre votre fichier à Archive ou Easel, vous pouvez le faire par rangée ou par colonne. QL Abacus vous propose d'abord de procéder par rangée, ce que vous pouvez accepter en appuyant sur **ENTREE**. Si le transfert doit être effectué par colonne, vous tapez **C**. Cette option est impossible dans le cas d'une exportation vers Quill quand les données sont toujours transférées par rangée.

QL Abacus vous demande enfin le nom du fichier à exporter. Si, à ce nom, vous ne précisez pas d' extension, le système ajoute -exp.

Formate

Formate la cartouche dans le Microdrive 1 ou 2. QL Abacus vous indique le spécificateur du Microdrive, mdv2_, quand vous devez préciser un nom de volume (vous pouvez choisir mdv1_).

Vérifiez que la cartouche dans le Microdrive 1 ou 2 ne contient aucun fichier à garder car, avec cette option, **TOUT** le contenu de la cartouche sera effacé.

Importe

Importe un fichier donné. QL Abacus peut ainsi lire des fichiers provenant d'Archive ou d'Easel. Tous renseignements à ce sujet dans l'Annexe.

L'importation peut avoir lieu par rangée ou par colonne. A vous de choisir et d'indiquer la référence de la case supérieure gauche de la zone de destination.

Si vous ne spécifiez pas d'extension au nom du fichier concerné, QL Abacus suppose l'extension _exp.

GRILLE (G)

La commande Grille propose un menu de sous-commandes qui affectent l'ensemble GRILLE du tableau en cours. Elle vous permet d'insérer ou d'effacer une rangée ou une colonne entière ou encore de changer le nombre de caractères affichés dans une ou plusieurs colonnes. Elle donne trois options :

Insertion

Insère une ou plusieurs rangées ou colonnes vides dans le tableau à l'endroit spécifié. Vous devez d'abord indiquer si vous voulez insérer des rangées (appuyez alors sur **ENTREE**) ou des colonnes (tapez **C**), avant de préciser une référence puis le nombre de rangées ou de colonnes. Dès que vous appuyez sur **ENTREE**, des rangées ou colonnes vides s'inscrivent avant la référence donnée. Les dernières rangées (ou colonnes) disparaissent de la grille. Si par exemple vous introduisez trois rangées, les trois dernières rangées de la grille disparaissent.

Les données enregistrées dans ces rangées ou colonnes seront irrémédiablement perdues.

Gomme

Efface des rangées ou colonnes entières du tableau. Vous devez d'abord préciser si vous effacerez des rangées (appuyez alors sur **ENTREE**) ou des colonnes (tapez **C**). Indiquez ensuite la référence de la rangée (ou colonne) de départ avant d'appuyer sur **ENTREE**. Enfin, indiquez la référence de la dernière rangée (ou colonne) à effacer.

Vous appuyez alors de nouveau sur **ENTREE** et le bloc spécifié est effacé. L'espace libéré est rempli par les rangées (ou colonnes) suivantes. Le bas de la grille se remplit de rangées vides (ou l'extrême droite de colonnes vides).

Quelle que soit l'option, toutes les formules prévues dans les rangées ou colonnes concernées sont ajustées selon leurs nouvelles positions.

Largeur

Change la largeur (le nombre de caractères) des colonnes du tableau. Vous devez tout d'abord préciser le nombre de caractères par colonne puis les références de la première et de la dernière colonne affectée.

JUSTIFIE (J)

AIDE F1	CURSEUR ↑↓←→	VALEURS et FORMULES	TEXTE Guillemets pour commencer	COMMANDES F3	
GUIDE F2	ACCES CASE F5	Saisie directe ENT pour finir		ANNULER ESC	
A	B	C	D	E	F
1					
2		GAUCHE	123.40		
3			DROITE	123.40	
4		PAR DEFAUT	123.40		
5		(texte à gauche, nombres à droite)			
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
?					
CASE A1 LIMITES A1:D5 MEMOIRE 20K					
CONTENU VIDE					

Figure 65 Justification

La commande **Justif** vous permet de modifier le cadrage de texte et de chiffres dans un bloc de cases. Deux premières options sont prévues : cases pleines, pour modifier des cases déjà utilisées, et cases vides, qui impose le cadrage par défaut adopté par QL Abacus dans des cases vides. Appuyez sur **ENTREE** pour l'option cases pleines et sur **V** pour l'option cases vides.

Vous devez ensuite préciser s'il s'agit de modifier le cadrage d'un texte (appuyez alors sur **ENTREE**) ou de nombres (tapez **N**). Trois possibilités vous sont alors proposées : cadrage au centre (**C**), à droite (**D**) ou à gauche (**G**).

Si vous avez choisi l'option cases pleines, vous devez enfin indiquer le bloc concerné par la modification.

Cette précision n'est pas nécessaire avec l'option cases vides. La justification implicite se rattache dans ce cas à toutes les nouvelles cases créées en tout lieu de la grille jusqu'à ce que vous changiez de nouveau la commande de cadrage.

La Figure 6.4 présente différents cadrages possibles et les options choisies (texte cadré à gauche et nombres cadrés à droite).

OUVRE(O)

Cette commande charge un fichier sur Microdrive et vous demande de préciser son nom. Pour obtenir le catalogue des fichiers sur Microdrive 1 ou 2, vous appuyez sur la touche **?**.

Si vous ne spécifiez pas d'extension après le nom du fichier, QL Abacus suppose l'extension **_aba**.

UNION (U)

Cette commande fusionne ou intègre les données d'un fichier sauvegardé à celles du tableau en cours. Après avoir lancé la commande, indiquez le nom du fichier à prélever du Microdrive puis précisez si les données doivent être ajoutées à celles de la grille (en appuyant sur **ENTREE**) ou si elles doivent être soustraites (tapez **S**).

Quand une case du fichier d'origine comportant un nombre ou une formule est rattachée à une case correspondante sur le tableau, sa valeur est ajoutée à celle du tableau (ou soustraite le cas échéant). Seules les valeurs numériques sont affectées par cette union et les cases renfermant du texte restent inchangées.

La nouvelle grille ne présente que des valeurs numériques dans les cases affectées par l'union. Les formules qui avaient établi les valeurs dans le fichier d'origine sont détruites puisqu'elles deviennent inutiles dans le tableau consolidé.

Cette commande permet d'associer rapidement et aisément les données de modèles du même ordre. Il est naturellement impératif que les deux tableaux aient une structure identique et que les cases correspondent pour que les résultats soient cohérents.

TRI (T)

Cette commande trie les rangées de la grille par ordre croissant, d'après le contenu d'une colonne donnée.

Vous indiquez en premier lieu la colonne sur laquelle le tri sera basé puis vous précisez la première et la dernière rangées concernées. Le tri procède comme suit :

- Case vides
- Valeurs numériques par ordre croissant
- Textes par ordre alphabétique

Ne triez que des rangées ou des colonnes qui contiennent des données. La commande risque en effet d'invalider les formules présentes dans la section concernée du tableau du fait qu'elles ne sont pas ajustées aux nouvelles positions.

IMPRIME (I)

Cette commande envoie le bloc délimité du tableau vers l'imprimante ou un fichier Microdrive. Vous indiquez tout d'abord si vous désirez que le tableau imprimé présente les valeurs ou les formules dans chaque case. Pour les valeurs, vous appuyez sur **ENTREE** et pour les formules, vous tapez **F**. Vous précisez ensuite le bloc à imprimer avant d'appuyer sur **ENTREE** pour que les bordures apparaissent ou de taper **G** pour les éliminer. Vous spécifiez enfin si le bloc doit être envoyé à l'imprimante (**ENTREE**) ou sur fichier Microdrive (touche **F**). Dans ce dernier cas, vous devez en plus préciser le nom du fichier avant d'appuyer sur **ENTREE**.

Le bloc délimité est alors envoyé à la destination indiquée. Pour arrêter l'impression à tout moment, il vous suffit d'appuyer sur **ESC**.

Si vous avez demandé que les formules soient affichées, QL Abacus imprime en premier lieu une liste numérotée de toutes les formules utilisées dans la grille. La grille est ensuite imprimée avec, dans chaque case contenant une formule, le numéro correspondant.

Si vous n'avez pas ajouté d'extension au nom du fichier, comme quand il s'agit d'envoyer le bloc à un Microdrive, QL Abacus suppose l'extension **_lis**.

QUITTE (Q)

Cette commande vous permet d'abandonner QL Abacus quand vous avez terminé votre travail.

Du fait que votre tableau disparaîtra alors, le système vous donne la possibilité de changer d'avis en demandant votre confirmation. Vous retrouverez votre tableau si vous appuyez sur **ESC**. Mais, en appuyant sur **ENTREE**, vous confirmez votre commande, vous quittez QL Abacus et vous revenez sous SuperBASIC.

VIDE (V)

Cette commande efface le contenu d'une ou de plusieurs cases de la grille. Quand vous la lancez, vous spécifiez un bloc dont toutes les cases seront effacées.

SAUVE (S)

Cette commande vous permet de sauvegarder un fichier sur Microdrive. Comme vous devez indiquer le nom du fichier, vous pouvez alors consulter la liste des fichiers sur Microdrive 1 ou 2 en tapant ?.

Si vous n'avez pas fait suivre le nom du fichier d'une extension, QL Abacus suppose l'extension **_aba**.

NOTATION (N)

La commande Notation modifie le format d'affichage de nombres dans une case ou dans un bloc. Elle ne modifie aucunement les valeurs numériques.

Vous indiquez tout d'abord si la commande concerne les cases pleines (en appuyant sur **ENTREE**) ou les cases vides (en tapant **V**), dans quel cas QL Abacus donnera le format spécifié à toutes les cases qui seront créées par la suite.

Dans les deux cas, vous choisissez ensuite l'un des six formats numériques suivants :

Décimal

Les nombres sont affichés en format décimal (un point remplaçant la virgule) avec un nombre de décimales fixe. Des nombres contenant plus de décimales que prévu sont arrondis à la décimale supérieure ou inférieure la plus proche selon les cas. Vous indiquerez sur la ligne de saisie le nombre de décimales choisi, nombre qui ne doit pas être supérieur à 14.

Si vous voulez arrondir les valeurs mêmes et non pas seulement leur affichage, vous devez le faire vous-même. Par exemple, pour arrondir une valeur à deux décimales :

1. multipliez par 100 (ou 1000 pour arrondir à trois décimales, etc.)
2. ajoutez 0.5
3. éliminez la fraction décimale avec la fonction **ent()**
4. divisez par 100 (ou 1000).

Avec la formule suivante, vous arrondissez la valeur dans la case C3 à deux décimales :

`ent(c3*100+0.5)/100`

Entier

Comme avec la fonction **ent()**, le système affiche des nombres entiers. Vous pouvez choisir de présenter des valeurs négatives entre parenthèses (tapez **P**) ou précédées du signe moins (appuyez sur **ENTREE**).

Si vous voulez que les valeurs soient calculées en nombres entiers (et non pas seulement affichées ainsi), utilisez la fonction **int()**.

Scientifique

Les nombres sont affichés en notation exponentielle ou scientifique. Vous devrez préciser le nombre de décimales (maximum 14). Les chiffres seront ici encore arrondis selon les cas à la décimale spécifiée la plus proche.

Pourcentage

Le tableau présente alors des pourcentages; par exemple, 0.55 devient 55%. Vous devez préciser le nombre décimales voulu (maximum 14).

Général

Cette option donne le format numérique général duquel les formats précédents sont choisis; il utilise au mieux l'espace disponible dans une case selon la valeur inscrite.

Monétaire

Les nombres sont affichés en format décimal avec un point au lieu d'une virgule suivi de deux décimales. Le tout est précédé d'un symbole monétaire. Vous pouvez choisir d'afficher les valeurs négatives entre parenthèses (tapez **P**) ou précédées du signe moins (appuyez sur **ENTREE**).

Si vous choisissez l'option cases pleines, QL Abacus vous demande enfin de préciser le bloc concerné par le changement. Vous tapez alors une référence de case ou de bloc (ou leurs étiquettes et identificateurs) avant d'appuyer sur **ENTREE** de nouveau.

Vous n'avez pas à indiquer de bloc avec l'option cases vides. Dans ce cas, le format monétaire est appliqué à toutes les nouvelles cases qui sont créées.

HUBLLOT (H)

Cette commande divise l'écran en deux fenêtres à l'emplacement du curseur ou joint en une seule deux fenêtres précédemment ouvertes.

Trois choix vous sont proposés après le lancement de la commande : jonction de 2 fenêtres (tapez **J**), division verticale (tapez **V**) et division horizontale (tapez **H**). Si la fenêtre est déjà divisée et que vous voulez la diviser horizontalement ou verticalement, vous devez joindre les deux sections avant de les diviser.

Quand vous divisez la fenêtre, la scission a lieu au niveau du curseur que vous disposerez donc sur l'axe choisi avant de taper **V** ou **H**. Les colonnes resteront affichées en entier et la largeur d'une fenêtre divisée ne sera jamais inférieure à dix caractères.

Vous pouvez alors choisir de faire défiler les fenêtres ensemble ou séparément. Vous taperez **I** pour les rendre indépendantes ou **ENTREE** pour les rendre solidaires. Si vous choisissez cette seconde option, tout changement de position d'une fenêtre (dans un sens parallèle à la division) se répercutera sur la seconde. Cette solidarité de translation n'a pas lieu en cas de mouvement à angle droit par rapport à la ligne de division. L'option sélectionnée en tapant **I** permet un déplacement indépendant de chaque fenêtre sur la grille.

EXECUTE (E)

La commande **Exec** oblige le système à recalculer toutes les formules de la grille. Un second calcul a normalement lieu chaque fois que vous introduisez de nouvelles données. Cette commande n'est donc lancée que quand vous avez désactivé l'option d'auto-calcul ou quand vous voulez mettre en œuvre des fonctions **saisin()** ou **saisist()** mémorisées dans des cases de la grille.

ABANDON (A)

Cette commande efface complètement le tableau en cours et ré-initialise QL Abacus. Comme son effet est irréversible, vous devez confirmer votre demande avant qu'elle soit exécutée. Si vous changez alors d'avis, il vous suffit d'appuyer sur **ESC** pour retrouver le menu de commandes et votre tableau tel qu'il était. Si, par contre, vous appuyez sur **ENTREE**, vous confirmez votre demande et la grille est totalement effacée.

FONCTIONS

Une fonction, c'est simplement un système de conversion de valeurs, dites arguments, en une valeur différente. Cette dernière est affichée par QL Abacus dans la case renfermant la formule qui l'a renvoyée.

Les fonctions assurées par QL Abacus peuvent prendre un, deux ou trois arguments, ou aucun, encadrés entre parenthèses après l'abréviation de la fonction. Aucun espace n'est autorisé entre l'abréviation et la première parenthèse mais crie espaces peuvent être prévus dans le texte entre parenthèses. Si la fonction demande plusieurs arguments, vous les séparerez par une virgule. Même si aucun argument n'est nécessaire, la fonction doit être suivie des parenthèses qui rappellent au système qu'il s'agit bien d'une fonction.

Dans les descriptions suivantes :

n est une expression numérique ou une référence à une case affichant une valeur numérique.

texte est un texte ou une référence à une case affichant un texte.

rang est une référence de rangée.

Une expression numérique est un nombre ou une expression donnant un résultat numérique.

Un texte est une chaîne alphabétique (encadrée de guillemets) ou une expression produisant un texte.

QL Abacus prévoit les fonctions suivantes.

abs(*n*)

renvoie à une valeur absolue de l'argument (ignorant tout signe moins).

Par exemple, `abs (3)` produit 3 et `abs (- 7)` renvoie 7.

saisisn(*texte*)

A la saisie de données numériques, affiche le texte spécifié (40 caractères maximum de longueur) sous forme de guide dans la ligne de saisie, suivi d'un "?" dans l'attente de votre réponse numérique. Votre réponse s'inscrit alors dans la case contenant la fonction. Des données ne vous sont demandées que quand vous avez introduit la fonction dans une case et quand vous demandez un second calcul de la grille en lançant la commande **Exéc**. Ces questions n'apparaissent donc pas en système de calcul automatique.

saisist(*texte*)

permet d'enregistrer un texte. Fonctionne comme **saisisn** mais demande que vous tapiez un texte et non pas un nombre.

atg(*n*)

donne l'angle, en radians, dont la tangente est *n*.

moy(*bloc*)

donne la moyenne de toutes les valeurs d'un bloc déterminé. Ce calcul ignore les cases vides et les cases contenant du texte. En l'absence de valeur numérique dans le bloc, le résultat est zéro.

car(*n*)

donne le caractère dont le code ASCII est *n*. Un caractère dont le code ASCII est inférieur à 32 n'apparaît pas à l'écran mais est transmis à l'imprimante (quand vous imprimez votre tableau) s'il est précédé d'un caractère nul ASCII. Exemple : **car(0)+car(13)** transmet à l'imprimante le

caractère signifiant retour chariot.

Autre exemple : vous obtiendrez un "A" sur l'écran en tapant **car(65)**.

code(*texte*)

donne le code ASCII du premier caractère du texte.

col()

donne le numéro de la colonne en cours.

cos(*n*)

donne le cosinus de l' angle *n* exprimé en radians.

nombcase(*bloc*)

donne le nombre de cases occupées dans le bloc. Le calcul ne porte que sur des cases contenant des valeurs numériques.

date(*n*)

donne la date du jour sous forme alphanumérique et selon l'un des trois formats suivants :

<i>n</i>	date
0	"aaaa/mm/jj" (format international)
1	"jj/mm/aaaa" (format européen)
2	"mm/jj/aaaa" (format américain)

L'horloge système doit avoir été préalablement mise à l'heure (explications à ce sujet dans la description des mots-clés de SuperBASIC).

jours(*texte*)

donne le nombre de jours écoulés depuis le premier janvier 1583 jusqu'à la date spécifiée dans le *texte* sous le format "aaaa/mm/jj" Le calcul étant basé sur le calendrier grégorien, la fonction ne peut être mise en œuvre que pour des dates ultérieures à 1582.

deg(*n*)

convertit en degrés un angle exprimé en radians.

exp(*n*)

donne la valeur de *e* (environ 2,718) à la puissance *n*. Le système affichera un message d'erreur si *n* n'est pas une valeur entre -87 et +88 du fait que le résultat dépasserait alors la portée numérique de QL Abacus.

si(*expression*,*vrai*,*faux*)

évalue l'expression et donne l'argument "vrai" si elle est non-nulle et l'argument "faux" si elle est nulle.

expression: = *n*
vrai: = *n* / *texte*
faux: = *n* / *texte*

Les arguments "vrai" et "faux" peuvent être des valeurs numériques ou alphanumériques. Les exemples suivants illustrent les cas d'emploi de cette fonction.

```
si(A1=B1,"égal","différent")  
si(A1,1,0)
```

Comme nous l'avons indiqué, les arguments peuvent être alphanumériques. Essayez de refaire l'exemple suivant :

```
[A1] 1  
[B1] 0  
[C1] si(A1 ou B1,"ou",0)
```

La conjonction "ou" apparaît alors dans la case C1 puisque le premier paramètre de si() donne une valeur non-nulle (vraie) si A1 ou B1 contient une valeur non-nulle. Si vous inscrivez ensuite un zéro dans A1, C1 affiche également un zéro.

indice (*colonne, rangée*)

colonne: = *n*

rangée: = *n*

donne le contenu d'une case à l'intersection de la colonne et de la rangée données.

souschn(*texte1, texte2*)

texte1: = *texte*

texte2: = *texte*

cherche le *texte2* dans le *texte1* et donne la position de son premier caractère. Si *texte2* ne figure pas dans *texte1*, la fonction retourne 0. La correspondance est basée sur les cases.

souschn("janvier", "jan") {donne 1}

souschn("janvier", "an") {donne 2}

souschn("janvier", "AN") {donne 0}

ent(*n*)

donne la partie entière du nombre par troncature au niveau de la virgule décimale.

Exemple :

ent(3.7) {donne 3}

ent(-4.8) {donne -4}

tir(*bloc, période*)

période: = *n*

calcule le taux interne de rentabilité des cash-flows dans le bloc (*colonne* ou *rangée*) sur une période de *n* mois.

Les valeurs négatives représentent des sorties et les positives des rentrées.

Cette fonction calcule le taux de rendement nécessaire pour que la valeur des rentrées de fonds soit égale au coût impliqué.

Supposons par exemple qu'on vous propose un rendement de 200 mille francs à une échéance régulière de sept années, pour une première mise de fonds d'un million. Est-ce une affaire intéressante ?

[A1] "Flux
[A2] -1000000
[A3] col=200000 {rangées 3 à 9}

Sous l' étiquette "flux", nous identifions les données et nous indiquons un intervalle de temps entre échéances successives de douze mois :

[C2] tir(flux,12) {rangées 2 à 9}

Votre tableau se présente alors comme à la Figure 6.5, le taux interne de rentabilité étant de 9,1%. si donc vous pouvez investir votre million à un taux supérieur mieux vaut oublier cette affaire.

	A	B	C	D
1	Flux			
2	-1000000.00		9.10	
3	200000.00			
4	200000.00			
5	200000.00			
6	200000.00			
7	200000.00			
8	200000.00			
9	200000.00			
10				
11				

Figure 6.5 Taux interne de rentabilité

Vous remarquerez que la première rubrique représente l'échéance zéro, la seconde la première, etc. La fonction suppose que le montant est payable intégralement au terme de chaque période en question.

long(*texte*)

donne le nombre de caractères du texte (un espace étant également compté comme un caractère).

ln(*n*)

donne le logarithme naturel (ou népérien) de *n*. Un message d'erreur est affiché si *n* est égal ou inférieur à zéro puisqu'il est impossible de calculer le logarithme de telles valeurs.

corresp(*bloc, décalage, valeur*)

décalage := *n*

valeur := *n*

Cette fonction procède à une recherche dans la grille qui est ici censée comprendre deux tableaux de valeurs. Le premier occupe le bloc spécifié (une rangée ou une colonne). Le second est parallèle au premier, dans la rangée ou colonne suivante. Par exemple, si le premier tableau se situe dans la colonne G de G10 à G25, le second est supposé être dans la colonne H, de H10 à H25. A chaque donnée inscrite dans le premier tableau doit correspondre une donnée dans le second. QL Abacus recherche dans le premier tableau la valeur la plus importante inférieure ou égale à la valeur donnée puis donne la donnée correspondante du second tableau. Pour que cette fonction puisse opérer correctement, il est supposé que les deux tableaux contiennent des valeurs numériques et que celles du premier sont classées par ordre croissant.

La première valeur du premier tableau est une valeur fictive. Elle doit être inférieure à la seconde qui représente la limite inférieure du processus de correspondance. La première valeur du second tableau est la valeur donnée quand la fonction **corresp()** est mise en œuvre avec un nombre inférieur à cette limite.

maxi(*bloc*)

donne la valeur maximum dans le bloc spécifié. En l'absence de cases numériques, la fonction indique le plus petit nombre possible (1.7-E+38).

mini(*bloc*)

donne la valeur minimum dans le bloc spécifié. En l'absence de cases numériques, la fonction indique le plus grand nombre possible (-1.7 E+38).

mois(*n*)

donne le nom du mois *n* modulo 12.

Exemple : **mois(3)** donne "Mars".

Si l'argument est supérieur à 12, il est remplacé par le résultat d'une division par 12. Ainsi, mois(13) et mois(1) donnent tous deux "**Janvier**".

van(*bloc, pourcentage, période*)

pourcentage := *n*

période := *n*

donne la valeur actuelle dans le bloc spécifié. Le pourcentage est le taux d'intérêt annuel (14 signifiant exemple 14%) et les données sont censées se rattacher à une série de périodes espacées par un nombre régulier de mois.

La valeur actuelle nette représente le montant nécessaire pour produire un certain cash-flow, avec taux d'intérêt. Supposons par exemple que vous puissiez acheter pour 700 000 francs le bail (sur 10 ans) d'un magasin donnant alors un revenu net annuel de 100 000 francs. Vous comptez sur une augmentation annuelle de ce revenu de 10%. Par contre, vos 700 000 francs vous rapportent actuellement 14% d'intérêts. Achetez-vous ou non ?

Vous calculez en premier lieu la valeur actuelle nette du revenu et vous la comparez avec le prix du bail.

```

[A1] "Flux
[A2] 0
[A3] 100000
[A4] col=a3*1.1           {rangées 4 à 12}
[A14] van(flux,14,12)    {rangées 2 à 12}

```

Le tableau obtenu est représenté sur la Figure 6.6.

	A	B
1	Flux	
2		0.00
3		100000.00
4		110000.00
5		121000.00
6		133100.00
7		146410.00
8		161051.00
9		177156.10
10		194871.71
11		214358.88
12		235794.77
13		
14		750885.07
15		

Figure 6.6 Valeur actuelle nette

La valeur actuelle nette (indiquée dans A14) du cash-flow de la boutique est supérieure au bail. L'affaire est donc intéressante.

La première rubrique de la liste correspond à la période zéro, la seconde à la période un, etc, ce qui est compatible avec la supposition de la fonction selon laquelle les revenus arrivent à la fin de chaque période. Vous devez donc attendre la fin d'une période pour bénéficier d'un rendement de votre investissement. Dans la réalité, vous procéderiez probablement à une analyse mensuelle plutôt qu'annuelle.

pi()

donne la valeur de la constante mathématique π

rd(*n*)

convertit en radians l'angle *n* exprimé en degrés.

repro(*texte*, *n*)

reproduit dans la case en cours en *n* exemplaires le premier caractère d'un texte donné.

Exemples :

```

repro( "*" , 5)      {inscrit cinq astérisques dans la case en cours}
repro( "abc" , 3)   {inscrit trois "a"}

```

rang()

donne le numéro de la rangée dans laquelle se trouve la case en cours.

sgn(*n*)

donne 1, -1 ou 0 suivant que *n* est positif, négatif ou nul,

sin(*n*)

donne le sinus de l'angle *n* exprimé en radians.

chn(nombre,format,nd)

nombre: = n

format: = n

nd:= n

convertit la première valeur numérique en une chaîne de caractères selon le format spécifié et, s'il y a lieu, avec le nombre de décimales nd. Quatre formats sont prévus

0	décimal (virgule flottante)
1	scientifique (ou exponentiel)
2	entier
3	général.

Le troisième paramètre, nd, précise le nombre de décimales à afficher après la chaîne. Il doit toujours être indiqué bien qu'aucune décimale n'apparaisse avec les formats entier, général et monétaire.

rac(n)

donne la racine carrée de n qui doit toujours être positif.

somme(bloc)

donne la somme des valeurs exactes dans le bloc spécifié. Cette fonction ignore toute valeur arrondie pouvant avoir été calculée au moyen de la commande **Notation**. Par exemple, si deux cases renferment les valeurs 3.44 et 9.73, la fonction **somme()** donnera 13.17 comme total. Si vous sélectionnez un format d'affichage avec une décimale, les deux valeurs sont alors arrondies à 3.4 et 9.7. Le total sera arrondi pour donner le résultat apparemment inexact de 13.2 au lieu de 13.17 (explications complémentaires au chapitre sur la commande **Notation**).

tg(n)

donne la tangente de l'angle n exprimé en radians.

heure()

donne l'heure de la journée sous forme alphanumérique et selon le format "hh:mm:ss". L'horloge système doit avoir été préalablement mise à l'heure comme indiqué dans la description des mots-clés du SuperBASIC.

val(texte)

convertit le texte en valeur numérique. La conversion ne peut avoir lieu que si le texte est composé de caractères numériques et elle s'arrête au premier caractère qui ne peut pas être interprété comme un chiffre. Par exemple, **val("2.2ABC")** donnera **2.2**, tandis que **val("ABC")** produira **0.0**.

larg()

donne la largeur de la colonne en cours en nombre de caractères. A noter qu'un espace est prévu pour séparer les colonnes.

ERREURS

Erreurs de saisie

Si vous faites une erreur de syntaxe en tapant une formule (nombre incorrect d'arguments dans une fonction, mélange de parenthèses et de crochets ou autres), QL Abacus vous envoie un message d'erreur. Votre formule reste dans la ligne de saisie et il vous suffit de la rectifier selon les indications du message.

Ces messages sont répertoriés ci-dessous :

Message	Exemple
Chaîne sans guillemets de	"abc" + "def" (fermeture)
Constante numérique incorrecte	1.5e (chiffre manquant après "e")
Nombre trop grand	1.5e99
Caractère inacceptable	12_5 (soulignement au lieu de moins)
Tous les noms doivent se référer à des colonnes	
Tous les noms doivent se référer à des rangées (Consulter à ce sujet la section sur les références des cases.)	
Référence de bloc incorrecte	a1:
Référence nominative incorrecte	c3.
Aucune rangée ou colonne de ce nom	(voir chapitre 3)
1ère référence nominative non définie	
2ème référence nominative non définie (le texte n'apparaît pas dans la grille au-dessus et à gauche de la case)	
Cette fonction exige une réf. de bloc	tir(1,2,3) (voir description de la fonction tir())
Bloc inacceptable	
Erreur de syntaxe	
Parenthèse manquante	
Valeurs incompatibles	1 + "abc"
Nombre incorrect d'arguments	rac (1,2)
Chaîne supérieure à 255 caractères	("*",256)
Division par zéro	
Arguments invalides	rac(-1)
Indice de chaîne hors limites (indice inférieur à zéro ou supérieur à 255, ou premier indice supérieur à la longueur du texte)	
Référence hors limites (à une case en dehors de la grille)	
Référence à une case erronée (une case contenant une formule produisant l'une des erreurs précédentes)	
Mémoire insuffisante VIDEr des cases	

Certaines erreurs (par exemple, une formule additionnant le contenu de deux cases dont l'un est un nombre et le second un texte) ne sont relevées qu'à la fin de la tentative de calcul, quand la formule est enregistrée dans la grille.

Si une formule prévoit une référence à une case vide, QL Abacus lui donne une valeur numérique nulle. Cette supposition risque donc de fausser les calculs.

Si QL Abacus relève une erreur pendant le calcul d'une formule, il transmet brièvement un message dans la case concernée. Vous pouvez alors amener le curseur dans cette case et rectifier l'erreur

Cas d'erreurs possibles :

# #TYPE	la formule contient une référence à une case dont les données ne conviennent pas (nombre au lieu d'un texte et vice-versa).
# # LONG	la formule contient une référence à une chaîne alphanumérique comptant plus de 255 caractères.
# #ZERO	la formule essaie d'effectuer une division par zéro.
# # ARG	la formule contient une fonction lancée avec un ou plusieurs arguments de valeur inacceptable (p.ex.: ln(-5).
# # SUB	la formule emploie un opérateur de division de chaîne comportant une erreur d'indice(s).
# REF	la formule contient une référence à une case extérieure à la grille. Le mot "ERREUR" apparaîtra alors dans la case en cours.
# # ERR	La formule contient une référence à une case comportant une erreur. Vous pouvez ignorer ces messages; ils disparaîtront quand l'erreur dans la case aura été corrigée.

Erreurs de classement

Les messages suivants n'apparaîtront qu'en cas d'erreur quand vous lancez une commande en rapport à un fichier, soit **Ouvre** ou **Documents**, par exemple :

Fichier inexistant : Le fichier désigné n'a pas été trouvé dans le Microdrive2.