

P. GUEULLE

PILOTEZ VOTRE ZX81

10 REM
15 PRINT
20 LET N=3
22 INPUT A\$
25 CLS
30 PRINT "TEXTE?"
30 INPUT B\$



OUVRAGES DU MEME AUTEUR :

- Réalisez vos récepteurs en circuits intégrés.
- Interphone, téléphone, montages périphériques.

Dans la collection *Technique Poche* :

- N° 17 – Réalisez vos circuits imprimés et décors de panneaux.
- N° 27 – Réduisez votre consommation d'électricité.
- N° 29 – Montages économiseurs d'essence.
- N° 32 – Antennes pour CiBistes.
- N° 34 – DéTECTeurs de trésors.

Dans la collection *Micro-Systèmes E.T.S.F.* :

- Maîtrisez votre ZX-81

Dans la collection *POCHE informatique* :

- Montages périphériques pour ZX-81.

En langue allemande :

- Energiesparen (Franzis Verlag, Munich).

En langue italienne :

- Realizzazioni di radioricevitori a circuiti integrati (Edizioni CELI, Bologne).

Illustration de couverture : Isabelle MUNIER.

La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'article 41, d'une part, que « les copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective » et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause, est illicite » (alinéa 1^{er} de l'Art. 40). Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les Art. 425 et suivants du Code pénal ».

© 1983 - E.T.S.F.

ISBN 2-85535-056-5

Sommaire

Avant-Propos.....	5
Introduction.....	9
Chapitre 1 : Prise de contact avec le ZX-81.....	11
Chapitre 2 : Le ZX-81 en vedette	21
1 ^o programme du « dessinateur paresseux »	22
2 ^o programme du « voyageur pressé ».....	23
3 ^o programme de « l'étudiant inquiet ».....	25
4 ^o programme du « tireur embusqué ».....	26
5 ^o programmes de « divination » (2 programmes)	28
6 ^o programme de calendrier perpétuel	30
Chapitre 3 : Des programmes de calcul	33
1 ^o programmes pour le calcul des circuits RC	36
2 ^o programmes pour le calcul des circuits LC.....	44
3 ^o programmes pour le calcul des groupements série / parallèle	49
4 ^o programme pour le calcul des bobinages sur ferrite	55
5 ^o programme pour le calcul des bobinages sur air	56
Chapitre 4 : Un peu de détente !	59
1 ^o faisons dessiner notre ZX-81	59
2 ^o une bataille navale arbitrée par ordinateur	64
3 ^o l'ordinateur et les messages secrets.....	68
4 ^o perçons les mystères du courrier	70
Chapitre 5 : le ZX-81 traceur de courbes	73
1 ^o histogrammes et statistiques	73
2 ^o étalonnons notre écran	75
3 ^o traçons une fonction	76
4 ^o toujours plus fort : une intégrale définie !	82

Chapitre 6 : L'ordinateur et vos économies	85
1 ^o calcul du rendement d'un placement	85
2 ^o calcul de la plus-value d'un placement	87
Chapitre 7 : Le ZX-81 et la comptabilité	89
1 ^o programmes de facturation	89
2 ^o programmes d'enregistrement des recettes et dépenses...	93
Chapitre 8 : Constitution de fichiers de données	99
1 ^o programme répertoire universel	99
2 ^o programme d'annuaire électronique	101
Chapitre 9 : le ZX-81 ordinateur de bord automobile	103
1 ^o programme de calcul de la consommation.....	103
2 ^o programme d'aide à la conduite.....	105
Chapitre 10 : « Manutention » des programmes	109
1 ^o comment effectuer de bons enregistrements sur cassette ..	109
2 ^o comment recopier des cassettes de programmes.....	110
3 ^o comment « téléphoner » ses programmes	111
Chapitre 11 : Utilisation de l'imprimante ZX-81	115
Appendice : Les branchements du ZX-81	123

choses plus sérieuses que les jeux, et que de nombreux professionnels pourront l'utiliser avec succès. Nul doute qu'après la lecture de cet ouvrage, nombreux seront ceux qui se laisseront tenter et achèteront leur ZX-81. C'est tout le bien que nous leur souhaitons.

Un mot sur l'imprimante. L'auteur l'a utilisée pour « authentifier » ses listings, mais aussi pour éviter toute erreur de transcription des programmes. Car les erreurs pouvant se produire feraient le désespoir des néophytes. Par contre, il est évident que, à moins de vouloir archiver ses programmes sous une autre forme que la cassette magnétique, et leurs résultats, l'utilisateur moyen n'a aucun besoin de cette imprimante pour faire fonctionner son matériel.

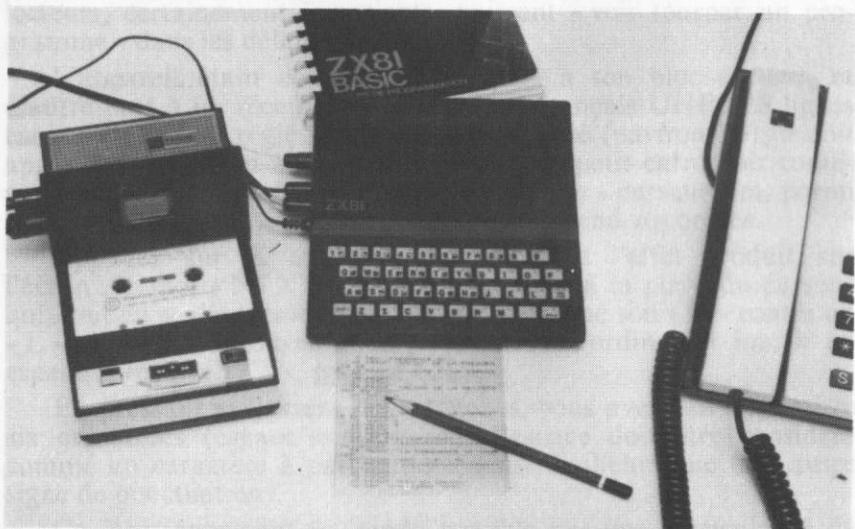
Enfin, un petit mot sur les programmes. L'auteur est un débutant, et le dit d'ailleurs dans les lignes qui suivent. Ses programmes sont tels qu'il les a expérimentés. Il est vraisemblable que certains feront sourire les programmeurs chevronnés... Mais qu'ils se souviennent de leurs débuts, et admettent avec nous que... pour un coup d'essai, ce n'est pas si mal.

L'éditeur

Introduction

Le micro-ordinateur ZX-81 de Sinclair semble devoir se tailler la part du lion parmi ce qu'il est convenu d'appeler les « ordinateurs individuels ».

Son prix extrêmement abordable allié à son extraordinaire facilité d'adaptation à des « périphériques » des plus courants, tels que récepteur TV et magnétophone à cassettes, justifient pleinement son acquisition par quiconque souhaite ne pas rester définitivement exclu de « ceux qui parlent aux machines ».



L'auteur de ces lignes, électronicien convaincu et jusqu'à présent assez méfiant vis-à-vis de l'informatique individuelle, s'est finalement laissé tenter, et ne regrette en rien son expérience passionnante, ce livre en étant le témoin !

Une petite semaine de loisirs suffit largement pour acquérir une pratique honnête du langage BASIC qui, loin de n'être qu'un jouet, permet de faire effectuer à la machine toute une gamme de travaux très évolués rendant de fiers services tant au niveau professionnel que familial.

Parti du niveau zéro, l'auteur retrace ici, sous la forme des programmes qu'il a lui-même mis au point pour ses besoins les plus variés, les étapes successives de son apprentissage des différentes instructions BASIC du ZX-81.

Certes, ces programmes peuvent être directement chargés en machine et utilisés, mais nous sommes certain que nos lecteurs, après avoir lancé les premiers d'entre eux, ne résisteront pas à ce « virus » qui devrait très vite les pousser de façon irrésistible à composer leurs programmes personnels !

Chapitre 1

Prise de contact avec le ZX-81

Ce chapitre ne saurait dispenser le lecteur de tous ses devoirs vis-à-vis du manuel d'emploi du ZX-81, ou tout au moins de la lecture de ses premières pages.

Nous allons simplement tenter ici de rendre aussi rapide que possible la première prise en main de l'appareil, de façon que nos lecteurs, certainement impatients, puissent « voir tourner un programme » dans les délais les plus brefs.

L'appareil étant connecté d'une part à son bloc secteur, et d'autre part à un récepteur TV standard français UHF 625 lignes (noir ou couleur) réglé sur le canal approprié (environ 36), il doit apparaître en bas et à gauche de l'écran un petit carré noir contenant un « K » blanc. Ce petit carré se nomme « curseur » et, parmi ses multiples usages, signale que le ZX-81 attend vos ordres.

Appuyez sur la touche N, et regardez l'effet produit sur l'écran : le mot « NEXT » vient de s'inscrire à la place du curseur qui, tout en se déplaçant vers la droite, a troqué son « K » contre un « L ». Appuyez à nouveau sur la touche N, l'ordinateur inscrit un espace après « NEXT », puis un « N ».

En pressant seulement deux touches, vous avez fait apparaître six caractères (espace compris, car l'espace doit être considéré comme un caractère à part entière, ainsi d'ailleurs que tout autre signe de ponctuation).

Ce fonctionnement est rendu possible par une particularité de conception de l'ordinateur capable de vous faire gagner un temps précieux : chaque fois que le curseur contient la lettre « K » (comme « KEYWORD » ou « mot-clé »), la machine interprétera la pression sur une touche en inscrivant sur l'écran le mot qui figure, sur le clavier, *au-dessus* de cette touche, en blanc sur fond noir.

A première vue, il semblerait que ceci réduise les possibilités de l'ordinateur, puisqu'il n'est pas possible, par exemple, d'écrire « NI-COLE » en essayant d'utiliser le clavier comme celui d'une machine à écrire.

Seulement voilà, un ordinateur n'est pas une machine à écrire, et une ligne commençant par un « N » isolé serait interprétée comme une erreur de programmation. Les seules possibilités autorisées de lignes débutant par « N » sont les mots « NEXT » et « NEW » (touche A), ce qui signifie que la fonction « entrée de mots-clé » gagne non seulement du temps de dactylographie, mais évite également certaines erreurs de programmation, en mettant l'utilisateur dans l'impossibilité de les commettre. C'est l'ordinateur lui-même qui décide comment il interprétera votre action sur la prochaine touche, et il vous informe de cette décision par la lettre figurant dans le curseur. Soyez assuré que son choix est toujours le bon et que, de toutes façons, le langage BASIC n'en autoriserait pas d'autre !

Les nostalgiques de la machine à écrire trouveront dans le manuel Sinclair divers programmes permettant, lorsqu'ils auront été chargés *et lancés*, d'inscrire sur l'écran n'importe quel caractère à n'importe quelle place.

Pour l'instant, l'ordinateur est vide de tout programme BASIC : il se contente, comme une calculatrice, d'exécuter immédiatement les ordres qu'on lui donne, pour en perdre aussitôt toute trace, sauf dans certains cas comme nous le verrons plus loin.

Essayons de faire exécuter au ZX-81 l'addition $2 + 3$: pas besoin de programme, la puissance de calcul du « mode commande » dans lequel nous nous trouvons suffisant largement.

Appuyons éventuellement sur la touche NEWLINE (parfois nommée « retour chariot » même en l'absence de cette mécanique) afin d'effacer ce qui pourrait subsister de nos essais précédents.

Résistons à l'envie de frapper « $2 + 3 =$ » puis NEWLINE (pour indiquer que c'est à l'ordinateur de travailler), car la machine, ne comprenant pas ce que nous lui demandons, répondrait par une « erreur de syntaxe », c'est-à-dire un curseur contenant la lettre « S » après le signe plus. En effet, la machine a besoin de savoir ce qu'elle doit faire du résultat de l'opération ! Va-t-elle le mettre en mémoire, l'imprimer sur papier, le faire apparaître sur l'écran, allez-vous lui demander de l'additionner à une autre valeur ?

Pour le moment, contentons-nous de faire apparaître le résultat sur l'écran, au moyen d'un ordre « print » : il faut pour cela appuyer

sur la touche « P » (qui est interprétée comme PRINT puisque le curseur est un « K », puis entrer $2 + 3$. Il apparaît sur l'écran :

PRINT 2 + 3

et il suffit alors de « donner la parole » à la machine en pressant NEWLINE pour voir apparaître le résultat seul sur l'écran : 5.

A partir de ce moment, la machine oublie tout de cette opération, ce qui fait que si vous pressez de nouveau NEWLINE, tout redevient comme « au premier jour », c'est-à-dire lorsque vous avez branché la machine.

Il est possible de cette façon de faire exécuter au ZX-81 des calculs très compliqués, rien qu'en les faisant précédé du mot-clé PRINT, quitte à utiliser des parenthèses tout comme on le ferait normalement sur le papier. Les quatre opérations sont bien sûr disponibles, la multiplication étant repérée * et la division /, mais le ZX-81 dispose aussi de fonctions mathématiques beaucoup plus évoluées (puissances, racines, trigonométrie, logarithmes, exponentielles, valeur absolue, partie entière), et peut même délivrer des nombres « aléatoires ». L'utilisation de ces fonctions ressemble à ce qu'il est habituel de faire sur une calculatrice. Le manuel de Sinclair donne le détail des abréviations désignant ces diverses fonctions, mais il nous faut faire deux remarques :

- l'unité dans laquelle travaillent les fonctions trigonométriques du ZX-81 est le *radian*. La conversion en degrés est facile grâce à la touche π (PI sur l'écran) ;
- les fonctions qui sont inscrites *en dessous* d'une touche doivent être appelées par la procédure suivante :
 - appuyer sur SHIFT (majuscules), ce qui met en service les fonctions repérées *en rouge* et seulement celles-ci ;
 - sans lâcher SHIFT, appuyer sur la touche NEWLINE, qui « devient » FUNCTION tant que SHIFT est enfoncée ;
 - lâcher ces deux touches et observer l'écran : le curseur contient la lettre F, signifiant que la prochaine touche pressée commandera la *fonction* indiquée *en dessous* d'elle (par exemple SIN pour la touche Q).
 - presser la touche de fonction choisie, ce qui doit ramener le curseur dans l'état L, puisque le mode « fonction » a disparu.

Si l'on veut faire exécuter à l'ordinateur le calcul suivant :

arc cos cos π (pour calculer π)

la succession des diverses actions sur les touches est la suivante :

ou	PRINT	SHIFT	NEWLINE	ARCCOS	SHIFT	NEWLINE
	P	FUNCTION		A	FUNCTION	
ou	COS	SHIFT	NEWLINE	π	NEWLINE	
	W	FUNCTION		M	NEWLINE	

soit onze touches, alors que trois auraient suffi sur une calculatrice. Ceci prouve que ce n'est certes pas dans ce mode « *commande* » que l'ordinateur donne le meilleur de lui-même, mais bien en mode « *programmation* ».

Un programme est une liste d'ordres appelés *instructions*, numérotés dans un ordre chronologique afin que la machine sache comment les exécuter. La rédaction de ces instructions doit obéir à des règles très précises qui constituent le *langage* de l'ordinateur.

Le ZX-81 travaille dans une version assez simple du langage BASIC, dont l'étude est suffisamment aisée pour permettre au programmeur néophyte d'obtenir rapidement des résultats très encourageants.

Ceux-ci ne doivent cependant pas faire perdre de vue que la programmation est un métier, souvent ingrat, et que la parfaite maîtrise d'un ou de plusieurs langages exige des mois ou des années d'expérience quotidienne.

Lorsque vous sentirez la fierté vous envahir et que vous estimerez mériter le titre d'informaticien, reportez-vous donc, dans le manuel BASIC, aux instructions PEEK, POKE, et surtout USR. Tenter de les utiliser (correctement, s'entend) est une excellente leçon de modestie !

Dans le cadre de l'*informatique individuelle*, à vocation essentiellement utilitaire et récréative, on peut cependant arriver à des résultats extrêmement satisfaisants en ne faisant appel qu'à une quinzaine d'instructions dont le fonctionnement peut être assimilé en quelques heures. Pour aller plus loin dans la découverte du BASIC, il faut étudier petit à petit toutes les autres instructions figurant dans le manuel Sinclair (le ZX-81 n'en connaît pas d'autres), et les mettre en œuvre en écrivant et en mettant au point des programmes. C'est en cela que cet ouvrage peut faciliter le travail

de ceux qui souhaitent tirer le meilleur parti de leur ZX-81, puisque les programmes qu'il contient ont été élaborés par l'auteur tout au long de son apprentissage personnel du BASIC, et ce, en partant d'un niveau zéro.

Insistons sur le fait que l'auteur n'est nullement un informaticien de métier, et que ses programmes sont bien évidemment perfectibles. **Il est fort probable que certains de nos lecteurs trouveront des moyens pour diminuer leur encombrement mémoire, ou pour en accélérer le déroulement.** Loin d'en être jaloux, nous le souhaitons, car nous aurons ainsi atteint notre but, en donnant à ces lecteurs les moyens de perfectionner leurs connaissances.

Ceci étant précisé, il est temps de passer à l'élaboration de notre premier programme.

Bien souvent, le futur programmeur se retrouvant pour la première fois devant un clavier d'ordinateur souhaite commencer par faire écrire son nom à la machine. Pourquoi pas, puisque nous avons vu que le ZX-81 ne peut pas être utilisé simplement comme une machine à écrire, et qu'il serait de mauvais goût de renoncer si vite. Essayons !

Il faut commencer par donner un numéro à notre première instruction, numéro qui est souvent 10 (les instructions sont fréquemment numérotées de dix en dix afin de faciliter les éventuelles adjonctions ultérieures).

Frappons donc :

10 PRINT PRENOM

(Vous pourrez remplacer PRENOM par... votre propre prénom, même s'il est long.)

Appuyez sur NEWLINE, la ligne frappée monte en haut de l'écran, ce qui signifie que, ne comportant pas d'erreur de *syntaxe*, elle a été acceptée et mise en mémoire par l'ordinateur.

Lançons ce programme par la procédure classique, qui consiste à presser RUN (touche R avec le curseur indiquant « K ») puis NEWLINE.

Consternation, le prénom n'apparaît pas, mais en bas de l'écran s'affiche le message 2/10.

Ce message se nomme « *compte rendu* » et indique, si le premier caractère qui le compose n'est pas un zéro, qu'un incident est survenu dans le déroulement du programme. Le chiffre 2 indique la

nature de l'incident, et 10 rend compte que le défaut réside dans la ligne 10, ce qui n'a rien d'étonnant puisqu'il n'en existe pas d'autre !

Le code des comptes rendus figurant dans le manuel Sinclair nous apprend que le programme s'est « planté » (terme vulgaire mais universellement employé) parce que « la variable qui a été utilisée est indéterminée ». Une variable, notre prénom ? Hélas oui, car nous n'avons pas pris la précaution de le faire figurer entre guillemets (touche P en position SHIFT).

Dans une instruction PRINT, tout assemblage de caractères commençant par une lettre et ne contenant que des lettres ou des chiffres est identifié comme une variable, c'est-à-dire une grandeur pouvant prendre différentes valeurs. En mathématiques, les variables se nomment x, y, a, b, c, etc., mais le ZX-81 peut travailler sur des variables qui s'appelleraient « PRIxDUTICKETDEMETRO », ou « POIDSDULITREDELIQUIDE », etc. (il est cependant recommandé d'éviter les noms de variables trop longs afin de ne pas encombrer inutilement la mémoire).

Donc, PRENOM a été reconnu comme une variable, et comme aucune valeur numérique ne lui a été attribuée par une instruction précédente, l'ordinateur nous informe qu'il ne peut pas imprimer cette valeur, ce qui est son droit le plus strict.

Prenons-nous au jeu, et donnons-lui satisfaction : frappons sur LIST, puis sur NEWLINE pour faire réapparaître notre programme (après un compte rendu d'erreur, NEWLINE seule aurait d'ailleurs pu suffire), puis entrons :

5 LET PRENOM = 255

suivi de NEWLINE, et lançons ce nouveau programme par RUN et NEWLINE.

La machine répond cette fois par 255, car l'instruction 5 a affecté (LET) la valeur 255 à la variable PRENOM, valeur qui a donc pu être imprimée sans équivoque par l'instruction 10. L'ordinateur nous donne, d'ailleurs, le compte rendu 0/10 qui signifie « tout va bien à bord, je suis arrêté à la ligne 10, la fin de votre programme ».

Tout ceci ne nous permet toujours pas d'imprimer ce fameux prénom !

Listons notre programme et cherchons ce qu'il convient de faire : il faut bien sûr supprimer la ligne 5 qui ne nous est d'aucune

utilité pour ce faire. Remplaçons-la donc par une ligne « vide » en faisant :

5 NEWLINE

La ligne 5 disparue, il nous reste à rectifier la ligne 10. Pour ce faire, deux voies peuvent être suivies :

- la refaire entièrement en tapant :

10 PRINT "PRENOM" puis NEWLINE,

qui déclenchera « l'échange standard » ;

- la corriger signe par signe, en la sortant du programme. Après avoir vérifié que le curseur en forme de signe « supérieur » (>) est bien situé après 10, faire :

EDIT (touche 1 en position SHIFT)

(Si le curseur est absent, on le fera apparaître par LIST 10 NEWLINE.)

La ligne 10 se trouve ainsi recopiée en bas de l'écran, avec un curseur « K ».

Appuyer sur SHIFT et, en maintenant cet appui, agir sur les touches 5 et 8, qui portent de petites flèches rouges horizontales. Chaque appui provoque un saut d'un cran du curseur, dans un sens ou dans l'autre. Après l'avoir amené juste devant le P de PRENOM, appuyer sur la touche P sans lâcher SHIFT, ce qui inscrit un guillemet. Faire la même chose à la fin du mot et rentrer la ligne dans le programme au moyen de NEWLINE, puis lancer le programme et vérifier que le but recherché est atteint.

L'intérêt d'un tel programme n'est pas énorme, mais une petite modification va permettre de faire apparaître à quel point les ordinateurs sont doués pour les *tâches répétitives*.

Listons le programme, et ajoutons-lui la ligne suivante :

20 GOTO 10

Launchons ce nouveau programme qui nous donne une colonne composée de 22 lignes identiques, puis le compte rendu 5/10 indiquant que le programme s'est arrêté à la ligne 10, faute de place sur l'écran. Pour le faire continuer, il suffit de faire CONT NEWLINE. Sur une imprimante, ce programme « tournerait en rond » jusqu'à épuisement du rouleau de papier. Nous verrons plus loin

que des instructions bien choisies permettent d'arrêter où l'on veut tout programme qui « boucle ».

Listons à nouveau le programme (en faisant **LIST 10 NEW-LINE** car LIST seul ne nous donnerait pas le curseur nécessaire à la suite), et modifions ainsi la ligne 10 :

10 PRINT "PRENOM";

Launchons le programme et constatons que le point-virgule, sous des dehors assez insignifiants, est capable de bouleverser complètement le déroulement du programme, en supprimant le retour à la ligne après chaque instruction PRINT.

Pour rendre l'affichage moins confus, nous pouvons ajouter un ou deux espaces avant le guillemet de fermeture, en faisant :

10 PRINT "PRENOM ";

(Attention, la touche d'espace doit être manœuvrée après relâchement de la touche SHIFT. Dans le cas contraire, on insère des caractères £, qu'il est possible d'effacer un à un au moyen de RUBOUT, en position majuscules cette fois.)

Noter la différence d'affichage selon que l'on a introduit un ou deux espaces (lorsque le ZX-81 éprouve le besoin d'aller à la ligne, il le fait sans le plus petit souci des règles de coupure des mots !)

Dernière expérience de « mise en page », le remplacement du point-virgule par une simple virgule :

10 PRINT "PRENOM",

La machine saute le retour à la ligne une fois sur deux, mais la position de la deuxième colonne est indépendante de la longueur du mot à imprimer (ou du nombre d'espaces que l'on introduit), tant que la première colonne ne rejoint pas la seconde (essayez de rentrer de plus en plus d'espaces !).

Ajoutons maintenant la ligne suivante :

15 PRINT

qui nous apprend deux choses :

– il est bien commode de disposer de « lignes libres » entre les instructions ;

– une instruction PRINT sans « *argument* » (indications à la suite) est interprétée comme « sauter une ligne », c'est-à-dire comme PRINT "", tout en occupant moins de place en mémoire.

Enfin, pourachever cette prise de contact avec les diverses possibilités de l'instruction PRINT, qui est à la base de tous les programmes devant afficher un résultat sur l'écran, vous pouvez ajouter une instruction permettant d'imprimer votre nom de famille dans diverses positions par rapport à votre prénom, en utilisant les possibilités de « mise en page » précédemment introduites.

Vous en savez assez, à présent, pour charger et faire démarrer les programmes qui vont être présentés dans les pages suivantes, et au fil desquels seront étudiées les autres instructions du BASIC ZX-81, ainsi que les astuces de programmation permettant leur mise en œuvre satisfaisante.

Vous êtes ainsi averti, preuves visuelles à l'appui, que la moindre différence entre le listing publié et les instructions que vous frappez, même au niveau d'un espace ou d'un signe de ponctuation, peut rendre le programme inutilisable ! Dans une telle éventualité, bien sûr, les commandes LIST, EDIT, RUBOUT, et le curseur mobile vous permettent d'effectuer les corrections voulues de la façon la plus commode possible. Vous voudrez bien noter que les listings présentés dans la suite de l'ouvrage ont été imprimés par le ZX-81 sur lequel ils ont été essayés, et ne peuvent donc comporter aucune erreur de transcription.

Le ZX-81 en vedette

Les quelques notions de base indispensables à un « pilotage » satisfaisant du ZX-81 étant supposées acquises par la lecture du chapitre précédent (dans le cas contraire, on se reportera sagement quelques pages en arrière...), il est possible dès maintenant d'organiser une sorte de « parade » destinée à mettre en relief les possibilités de la machine.

Les différents programmes présentés dans ce chapitre ne seront accompagnés que d'un simple « mode d'emploi ».

C'est dire qu'ils ne contribueront pas à l'apprentissage du langage BASIC par nos lecteurs mais, en revanche, ils devraient leur « mettre l'eau à la bouche » en les poussant à étudier le fonctionnement des diverses instructions du BASIC ZX-81, qui seront introduites progressivement au fil des chapitres suivants.

Egalement, ce n'est qu'à partir du prochain chapitre que nos programmes prendront un côté vraiment « universel », permettant de résoudre vite et bien toute une variété de problèmes qui se posent dans les domaines les plus divers.

Insistons une fois encore sur l'*absolue nécessité* de frapper un programme de façon que chacune de ses lignes apparaisse sur l'écran, après l'action sur NEWLINE, très exactement semblable à sa réplique imprimée dans le listing publié. Toute différence, au niveau d'un signe de ponctuation, d'un guillemet simple ou double, d'un O confondu avec un Ø, ou d'un \$ troqué contre une £, et même d'un espace absent ou en trop, peut entraîner un fonctionnement défectueux (quoique heureusement sans danger matériel) de la machine. Si après cette première démonstration, vous faites l'effort de suivre en profondeur le contenu des chapitres suivants, vous pourrez passer du « grade » d'*utilisateur* à celui de *programmeur* et, de ce fait, « descendre » dans les programmes afin de modifier leur fonctionnement conformément à vos désirs ou à vos besoins.

1° Programme du « dessinateur paresseux »

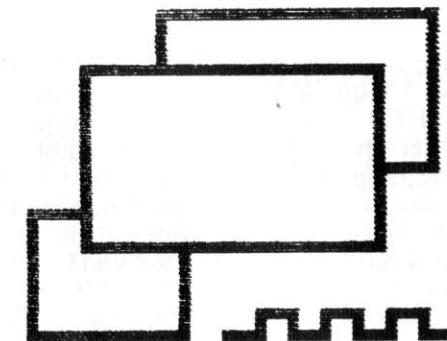
Il nous a semblé intéressant de faire débuter cette démonstration par un exemple frappant des immenses possibilités graphiques du ZX-81.

```
5 REM "TRACE"
6 LET L=22
7 LET N=9
8 LET C=32
10 IF INKEY$="5" THEN LET L=L-
11 IF INKEY$="8" THEN LET L=L+
12 IF INKEY$="7" THEN LET C=C+
13 IF INKEY$="6" THEN LET C=C-
14 IF INKEY$="9" THEN LET N=9
15 IF INKEY$="0" THEN LET N=0
16 IF N=9 THEN PLOT L,C
17 IF N=0 THEN PLOT L,C
18 IF N=0 THEN UNPLOT L,C
19 GOTO 10
20 REM "COPYRIGHT 1982"
```

Après avoir frappé soigneusement ce programme, on le lancera en pressant RUN, puis NEWLINE. Un tout petit carré noir apparaîtra alors seul sur l'écran, si le programme a été convenablement entré.

A partir de ce moment, presque toutes les touches du clavier deviennent inopérantes, afin de rendre toute fausse manœuvre impossible. La touche SPACE permet d'arrêter le programme (on reçoit alors le compte rendu D/32 en bas d'écran), mais les véritables touches de « travail » sont les touches 5, 6, 7, 8, 9, Ø. Les quatre premières, qui comportent de petites flèches rouges en vue d'applications qui ont été vues plus haut, permettent de déplacer horizontalement ou verticalement (mais pas en oblique) le véritable « stylo électronique » qu'est finalement le petit carré. Essayez, et vous parviendrez rapidement à créer de jolis graphismes. Vous regretterez cependant de ne pouvoir déplacer le « stylo » sans qu'il laisse de trace, lorsque vous éprouverez le besoin de dessiner des motifs séparés sur la même image. Le programme permet pourtant de « lever le stylo », en pressant la touche Ø (marquée RUBOUT). Si

vous essayez, vous constaterez que le carré se met à clignoter, et que ses déplacements s'opèrent alors sans laisser la moindre trace (notez que sa dernière position se trouve effacée). Lorsque vous aurez amené le « stylo » là où vous souhaitez reprendre le tracé, vous n'aurez qu'à presser la touche 9 (marquée GRAPHICS) pour le « poser » à nouveau.



2° Programme du « voyageur pressé »

Ce programme est un exemple de calcul de niveau « certificat d'études », que le ZX-81 effectue bien sûr avec brio, après avoir posé, en français, les questions nécessaires à son information.

Il permet de calculer la vitesse d'un train, d'une voiture, d'un escargot, etc., en chronométrant ses performances sur un parcours de longueur connue.

```
5 REM "VITESSE"
9 CLS
10 PRINT "DISTANCE D""ESSAI "
11 PRINT "EN METRES ?"
12 INPUT D
13 CLS
14 PRINT "TEMPS MIS POUR COUVR
IR"
15 INPUT T
16 CLS
17 PRINT AT 11,0;"VITESSE : ";
18 (Suite au verso)
```

```

3.6*D/T; " KM/H"
26 PRINT AT 15,0; "MEME DISTANCE : NEXT"
27 PRINT "AUTRE DISTANCE : CLEAR"
28 IF INKEY$="N" THEN GOTO 18
29 IF INKEY$="X" THEN GOTO 9
30 GOTO 28
40 REM "COPYRIGHT 1982"

```

Le programme se lance par RUN NEWLINE, et pose une première question, à laquelle il est de bon ton de répondre par un nombre (souvent 1000 lorsque le chronométrage s'effectue entre deux repères kilométriques). Cela fait, la machine réclame le résultat du chronométrage, par exemple 13.8 pour 13"8/10. Sitôt cette « entrée » validée par NEWLINE, la machine calcule la vitesse, ici un bon 260 km/h, ce qui indique sans nul doute que nous nous livrons à nos essais à bord du TGV Paris-Lyon !

DISTANCE D'ESSAI
EN METRES ?

1000

TEMPS MIS POUR COUVRIR
1000 METRES ? (EN SEC)

13.8

VITESSE : 260.86957 KM/H

MEME DISTANCE : NEXT
AUTRE DISTANCE : CLEAR

Quelques lignes plus bas, la machine nous indique que si nous souhaitons calculer une autre vitesse par chronométrage sur la même distance, nous serions aimable de lui notifier ce choix en pressant la touche N (marquée NEXT), mais qu'elle ne refusera pas de travailler sur une autre distance, à charge pour nous de

presser la touche X (marquée CLEAR). A ce stade, le reste du clavier est complètement neutralisé, sauf la touche SPACE (marquée BREAK) qui permet d'arrêter le programme sur un compte rendu D/30, libérant ainsi le clavier en vue de l'entrée de commandes telles que LIST, RUN, STOP, NEW, SAVE, LOAD, etc. En particulier, ce n'est qu'en procédant ainsi que l'on pourra stocker ce programme sur cassette.

3^e Programme de « l'étudiant inquiet »

Dans cet exemple, le ZX-81 montre son aptitude à cumuler pratiquement à l'infini des données qui lui sont fournies petit à petit, et à en extraire un résultat final. Ici, la machine calcule la moyenne générale d'un élève à partir de toutes ses notes et des coefficients des différentes matières. Aucun ordre n'est à respecter lors de l'entrée des notes, ce qui permet de calculer des moyennes provisoires tout au long d'une année scolaire, exercice tantôt rassurant, tantôt inquiétant !

```

5 REM "MOYENNE"
6 LET K=0
8 LET T=0
10 PRINT "COEFFICIENT ? "
15 INPUT C$
20 IF C$="" THEN GOTO 50
21 PRINT VAL C$
22 PRINT
25 PRINT "NOTE ? "
28 INPUT N
30 PRINT N
35 PAUSE 100
38 CLS
40 LET T=T+(VAL C$*N)
42 LET K=K+VAL C$
45 GOTO 10
50 CLS
52 PRINT "MOYENNE GENERALE : "
55 PRINT
60 PRINT " ; T/K"
70 REM "COPYRIGHT 1982"

```

Sitôt lancé par RUN NEWLINE, le programme fait demander à la machine le coefficient de la première note. Dès l'appui sur NEWLINE, celle-ci accuse réception de cette information et ré-

clame la note. Une courte pause, et le coefficient de la note suivante est demandé. Ce cycle pourrait se répéter à l'infini si l'opérateur n'avait la possibilité de *ne pas répondre* à la question « coefficient ? », lorsque toutes les notes ont été données. En effet, si on se contente à ce moment de presser NEWLINE sans avoir entré aucun nombre, la machine calcule immédiatement la moyenne générale, et arrête le déroulement du programme. On pourra alors le relancer à loisir en faisant à nouveau RUN NEWLINE. La

```

COEFFICIENT ?
2
NOTE ?
18
COEFFICIENT ?
4
NOTE ?
12
COEFFICIENT ?
MOYENNE GENERALE :
14

```

différence est d'importance par rapport aux précédents programmes, qui ne pouvaient s'arrêter que par une action sur la touche SPACE/BREAK. Dans cet ouvrage, le lecteur rencontrera indifféremment les deux types de programmes, et il lui appartiendra d'exécuter les manœuvres adéquates lorsqu'il souhaitera en interrompre le déroulement. On peut cependant préciser que l'apparition d'un « compte rendu » en bas d'écran permet d'affirmer que le programme s'est arrêté seul. Par contre, l'absence de compte rendu ne prouve rien !

4^o Programme du « tireur embusqué »

Ce jeu fait appel à la fois aux possibilités graphiques et mathématiques de la machine, sans oublier la fonction de génération de nombres « aléatoires ».

```

5 REM "TIR"
6 LET T=0
8 LET P=0
10 LET L=INT (RND*40)
11 LET K=INT (RND*50)
12 PLOT 10+K,L
13 LET P=P+1
15 FOR F=1 TO 40
18 PLOT 0,F
19 UNPLOT 0,F-1
20 IF INKEY$="F" THEN GOTO 30
22 NEXT F
24 CLS
25 GOTO 10
31 FOR G=0 TO 60
32 PLOT G,F
35 IF G=10+K AND F=L THEN GOTO
50
40 NEXT G
42 PRINT TAB 0;"MANQUE"
43 GOSUB 53
44 CLS
45 GOTO 10
50 PRINT TAB 0;"TOUCHE"
51 LET T=T+1
52 GOTO 43
53 LET A$="S"
55 PRINT
56 PRINT
57 IF T=1 OR T=0 THEN LET A$=""
58 PRINT T;" TIR";A$;" AU BUT
SUR ";P
59 PAUSE 100
60 RETURN
75 REM "COPYRIGHT 1982"

```

Dès le lancement de ce programme par RUN NEWLINE, un petit carré noir (la cible) apparaît quelque part dans les deux tiers de droite de l'écran. Simultanément, un autre carré évolue de bas en haut à l'extrême gauche de l'image. Ce second carré représentant le « tireur », le jeu consiste à appuyer sur la touche F (comme FEU) dès que l'on estime que les deux carrés sont sur une même horizontale. Un trait horizontal se trace alors, figurant la trajectoire de la balle, et le coup est soit réussi, soit manqué suivant la qualité des réflexes du joueur. La machine arbitre le coup, et donne un score global après chaque essai. On notera que toute possibilité de tir manquée (c'est-à-dire toute position de la cible n'ayant pas fait l'objet d'un tir) est décomptée comme un tir manqué. Les

TOUCHE

1 TIR AU BUT SUR 7

MANQUE

1 TIR AU BUT SUR 8

performances de débutants sont généralement très modestes, laissant place à d'importants progrès au cours des premières parties. La longueur des parties n'étant pas limitée au niveau de la machine, il est prévu des « repos » automatiques et aléatoires, se manifestant par l'arrêt des mouvements des points mobiles, et par l'apparition d'un compte rendu en bas de l'écran. Après quelques instants de répit, on peut relancer le programme par CONT NEWLINE, sans pour autant remettre les compteurs à zéro. Si l'on désire provoquer manuellement un tel arrêt, on appuiera, bien sûr, sur BREAK. On remarquera l'emploi de la fonction RND qui génère des nombres aléatoires entre 0 et 1, et qu'on multiplie pour avoir des nombres plus grands que 1.

5^e Programmes de « divination »

Ce programme permet à l'utilisateur de faire deviner un nombre compris entre 0 et 50 à son ZX-81. La machine compte très honnêtement ses tentatives (elle les affiche d'ailleurs sur l'écran) et donne le résultat de ce comptage sitôt le nombre trouvé.

```
5 REM "DEVIN"
6 LET M=1
10 PRINT "ENTREZ UN NOMBRE"
12 PRINT "COMPRIS ENTRE 0 ET 5
@"
15 INPUT N
16 CLS
18 PRINT "LE ZX 81 DOIT DEVINE
R ";"N
```

```
19 PRINT
20 LET R=INT (RND*50)
25 PRINT R; " "
28 IF R=N THEN CLS
30 IF R=N THEN PRINT "TROUVE E
N ";M;" ESSAIS"
35 IF R=N THEN STOP
40 IF R>N THEN LET R=R-1
45 IF R<N THEN LET R=R+1
48 LET M=M+1
50 GOTO 25
60 REM "COPYRIGHT 1982"
```

Le lancement s'effectue au moyen de RUN NEWLINE, et l'ordinateur demande alors le nombre à deviner, nombre qu'il ne regardera pas, c'est promis !

Sitôt l'entrée validée par NEWLINE, la recherche commence à partir d'un nombre librement choisi par la machine.

```
ENTREZ UN NOMBRE
COMPRIS ENTRE 0 ET 50
20
LE ZX 81 DOIT DEVINER 20
25 24 23 22 21 20
TROUVE EN 6 ESSAIS
```

La procédure de recherche est assez naïve, puisque le ZX-81 progresse d'une unité à chaque tentative. Il appartiendra au lecteur d'essayer de surpasser la machine, en appliquant des progressions plus étudiées, au moyen de la variante ci-dessous, qui renverse purement et simplement les rôles.

```
5 REM "MEDIUM"
6 LET M=1
7 LET R=INT (RND*50)
10 PRINT "DEVINEZ UN NOMBRE"
12 PRINT "COMPRIS ENTRE 0 ET 5
@"
15 INPUT N
16 CLS
25 PRINT N; " "
```

(Suite au verso)

```

28 IF R=N THEN CLS
30 IF R=N THEN PRINT "TROUVE EN"
N ";M;" "ESSAIS"
35 IF R=N THEN STOP
40 IF R>N THEN PRINT "PLUS QUE
CELA..."
45 IF R<N THEN PRINT "MOINS QU
E CELA..."
48 LET M=M+1
50 GOTO 15
60 REM "COPYRIGHT 1982"

```

C'est alors la machine qui choisit un nombre compris entre zéro et cinquante, et qui indique au « médium » amateur dans quelle direction doivent porter ses recherches. Bien évidemment, le score est calculé et affiché en fin de programme.

```

DEVINEZ UN NOMBRE
COMPRIS ENTRE 0 ET 50

20 PLUS QUE CELA...
30 MOINS QUE CELA...
28 MOINS QUE CELA...
25
TROUVE EN 4 ESSAIS

```

6^e Programme du « calendrier perpétuel »

Ce genre de programme, qui fait toujours une forte impression sur les néophytes, se retrouve, sous des variantes diverses, dans la plupart des manuels accompagnant les calculatrices programmables, car il n'occupe que peu de place dans leur mémoire, malgré l'apparente complexité de la tâche qu'il doit accomplir.

L'adaptation d'un tel programme en BASIC ZX-81 présente l'avantage d'autoriser un dialogue en langage clair entre l'utilisateur et la machine, ce qui n'est généralement pas le cas avec les calculatrices, même évoluées.

Sitôt lancé par RUN NEWLINE, le programme fait poser à la machine les questions lui permettant d'apprendre quelle est la date

(jour, mois, année) de laquelle on se propose de déterminer le jour de la semaine correspondant.

```

5 REM "CALENDRIER"
10 PRINT "QUEL JOUR ?"
11 INPUT D
12 CLS
14 PRINT "QUEL MOIS ?"
15 INPUT M
18 CLS
20 PRINT "QUELLE ANNEE ?"
21 INPUT J
22 CLS
24 PRINT "LE ";D;" / ";M;" / ";J;" "
EST UN "
25 IF M<=2 THEN LET J=J-1
28 IF M<=2 THEN LET M=M+12
30 LET E=INT ((M+1)*(13/5))
33 LET F=INT (5*J/4)
34 LET G=INT (J/100)
35 LET H=INT (J/400)
36 LET T=D+E+F-G+H
37 LET DA=INT ((T/7-INT (T/7))
*7+.1)
40 IF DA=0 THEN PRINT "SAMEDI"
41 IF DA=1 THEN PRINT "DIMANCHE"
42 IF DA=2 THEN PRINT "LUNDI"
43 IF DA=3 THEN PRINT "MARDI"
44 IF DA=4 THEN PRINT "MERCREDI"
45 IF DA=5 THEN PRINT "JEUDI"
46 IF DA=6 THEN PRINT "VENDREDI"
50 REM "COPYRIGHT 1982"

```

Chaque réponse se fera sous forme de chiffres, l'année étant bien sûr entrée en entier. (Entrer 82 à la place de 1982 désignerait tout simplement... l'an 82 !)

Essayons d'entrer la date du 14 juillet 1789 en faisant :

14 NEWLINE, 7 NEWLINE et 1789 NEWLINE,

et nous aurons le privilège de savoir que la prise de la Bastille s'est déroulée un mardi !

Un test rapide consiste à essayer quelques dates figurant sur un calendrier de l'année en cours, mais il est beaucoup plus attrayant de traiter les dates de naissance de tous les membres de la famille, ou encore des dates à venir : on découvrira alors que nous dispose-

```
QUEL JOUR ?  
14  
QUEL MOIS ?  
7  
QUELLE ANNEE ?  
1789  
LE 14/7/1789 EST UN MARDI
```

rons d'un dimanche pour nous remettre des festivités du 1^{er} janvier de l'an 2000 ! Précisons enfin qu'après chaque détermination, l'ordinateur s'arrête sur un $\emptyset/5\emptyset$, et peut être relancé par RUN NEWLINE.

Ces quelques programmes nous ont permis de brosser un rapide tableau des principales possibilités du ZX-81. Leur mise en service est à la portée de quiconque aura fait l'effort de consacrer un bon quart d'heure à étudier les quelques particularités d'exploitation du clavier de la machine, en s'aidant pour cela du chapitre premier de cet ouvrage.

Les autres programmes qui vont être présentés dans les chapitres suivants peuvent tout aussi bien être chargés et utilisés sans notions supplémentaires de programmation. Cependant, nous avons voulu introduire petit à petit les principales instructions du BASIC ZX-81 afin que ceux de nos lecteurs qui feront l'effort de suivre nos explications puissent parvenir, au terme de leur lecture, à une pratique suffisante leur permettant d'écrire des programmes personnels.

Notre démarche ne prétend pas concurrencer le manuel fourni par le constructeur, mais plutôt éclairer sous un jour plus pratique les explications assez succinctes qu'il donne sur les règles d'écriture de programmes à vocation utilitaire.

Le lecteur de cet ouvrage devra donc se référer de temps à autre à ce manuel, lorsqu'il ressentira le besoin de compléments d'information et, notamment, lorsqu'il souhaitera, en fin de lecture, revenir sur les programmes de ce chapitre afin d'en comprendre le mécanisme.

Nous avons volontairement utilisé ici des instructions non décrites par la suite (par exemple INKEY\$ ou RND), de façon à donner à nos lecteurs la possibilité de se « dégager » du cadre nécessairement limité de cet ouvrage, et de découvrir par eux-mêmes les dernières instructions du BASIC ZX-81 avec l'aide du manuel Sinclair.

Des programmes de calcul

Un micro-ordinateur peut faciliter les calculs répétitifs. Etant électronicien, l'auteur a tenu à dédier ses premiers programmes « utilitaires » à l'électronique, afin de soulager une calculatrice un peu trop sollicitée depuis quelque temps déjà ! Mais les non-électroniciens pourront considérer qu'il s'agit simplement de programmes de calculs répétitifs, faciles à adapter à d'autres domaines.

L'idée de commencer par de tels programmes, qui ne se classent pas parmi les plus simples, est venue d'une tentative malheureuse, mais excusable de la part d'un néophyte, de charger sans autre forme de procès, sur le ZX-81, deux programmes destinés à un autre « ordinateur personnel » travaillant en BASIC, le PC-100 Siemens.

L'échec d'une telle tentative, dont l'analyse des causes s'avère riche d'enseignements, est dû à deux raisons principales :

- l'encombrement mémoire de ces programmes est très supérieur à la capacité de 1 024 octets (1Ko) du ZX-81 de base ;
- le langage BASIC utilisé par le PC-100 possède des instructions qui n'existent pas dans celui du ZX-81. En conséquence, celui-ci ne peut en aucun cas les reconnaître, et lance un curseur d'erreur de syntaxe dès leur introduction.

Ceci signifie que tout programme écrit pour un ordinateur BASIC autre que le ZX-81 nécessitera de sérieuses modifications pour pouvoir fonctionner sur celui-ci, et réciproquement. Dans la plupart des cas d'ailleurs, on gagne du temps en repartant à zéro, et en écrivant de nouveaux programmes, d'où ce livre...

```

1 REM**B AT PELKA
2 REM**15.1.80
3 REM**SCHWK
4 REM**PROGRAMM ZUR BERECHNUNG VON SCHWINGKREISEN**
10 PRINT "SCHWINGKREIS":GOSUB 200
20 INPUT " C=? ,L=? ,F=? ;A$"
30 IF A$="C" THEN 270
40 IF A$="L" THEN 140
50 IF A$="F" THEN 80
60 PRINT "FALSCH EINGABE":GOSUB 200:GOTO 20
80 INPUT "C IN PF ";C
90 INPUT "L IN UH ";L
100 F=1E9/(2*3.1415926*SQR(C*L))
110 IF F>1E6 THEN F=F/1E6:PRINT "F=? ;F;"MHZ":GOTO 220
120 IF F>1E3 THEN F=F/1E3:PRINT "F=? ;F;"KHZ":GOTO 220
130 PRINT "F=? ;F;"HZ":GOTO 220
140 INPUT "C IN PF ";C
150 INPUT "F IN HZ ";F
160 L=1E15/((2*3.1415926*F)↑2*C)
170 IF L<1E-4 THEN L=L*1E6:PRINT "L=? ;L;"NH":GOTO 220
180 IF L<1 THEN L=L*1E3:PRINT "L=? ;L;"UH":GOTO 220
190 PRINT "L=? ;L;"MH":GOTO 220
200 FOR I=1 TO 1000:NEXT
210 RETURN
220 GET F$
230 IF LEN(F$)=0 THEN 220
240 F=ASC(F$)
250 IF F=78 GOTO 20
260 GOTO 220
270 INPUT "F IN HZ ";F
280 INPUT "L IN UH ";L
290 C=1E12/((2*3.1415926*F)↑2*L)
300 IF C<1E-3 THEN C=C*1E6:PRINT "C=? ;C;"PF":GOTO 220
310 IF C<1 THEN C=C*1E3:PRINT "C=? ;C;"NF":GOTO 220
320 PRINT "C=? ;C;"UF":GOTO 220
330 END

```

```

1 REM**B AT PELKA
2 REM**20.5.80
3 REM**RC-GL
4 REM**PROGRAMM ZUR BERECHNUNG VON RC-GLIEDERN
10 PRINT "GRENZFREQUENZ VON":GOSUB 200
20 PRINT "RC-GLIEDERN":GOSUB 200
30 DIM A$(10)
40 INPUT "R=? ,C=? ,FG=? ;A$"
50 IF A$="R" THEN 160
60 IF A$="C" THEN 120
70 IF A$="FG" GOTO 80
75 PRINT "FALSCH EINGABE":GOSUB 200:GOTO 40
80 INPUT "R IN OHM ";R
90 INPUT "C IN FARAD ";C
100 F=1/(2*3.1416*R*C)
105 IF F>1E6 THEN F=F/1E6:PRINT "FG=? ;F;"MHZ":GOTO 220
110 IF F>1E3 THEN F=F/1E3:PRINT "FG=? ;F;"KHZ":GOTO 220
115 PRINT "FG=? ;F;"HZ":GOTO 220
120 INPUT "R IN OHM ";R
130 INPUT "FG IN HZ ";F
140 C=1E6/(2*3.1416*R*F)
145 IF C<1E-3 THEN C=C*1E6:PRINT "C=? ;C;"PF":GOTO 220
150 IF C<1 THEN C=C*1E3:PRINT "C=? ;C;"NF":GOTO 220
155 PRINT "C=? ;C;"UF":GOTO 220
160 INPUT "C IN FARAD ";C
170 INPUT "FG IN HZ ";F
180 R=1/(2*3.1416*C*F)
185 IF R>1E6 THEN R=R/1E6:PRINT "R=? ;R;"MOHM":GOTO 220
190 IF R>1E3 THEN R=R/1E3:PRINT "R=? ;R;"KOHM":GOTO 220
195 PRINT "R=? ;R;"OHM":GOTO 220
200 FOR I=1 TO 1000:NEXT
210 RETURN
220 GET F$
225 IF LEN(F$)=0 GOTO 220
230 F=ASC(F$)
240 IF F=78 GOTO 40
250 GOTO 220
260 END

```

1° Programmes pour le calcul des circuits RC

Chacun sait que la formule donnant la « fréquence de coupure » d'un circuit RC s'écrit :

$$f_c = \frac{1}{2\pi RC}$$

et donne naissance à deux formules complémentaires :

$$C = \frac{1}{2\pi f_c R} \quad \text{et} \quad R = \frac{1}{2\pi f_c C}$$

Ces formules sont faciles à utiliser sur n'importe quelle calculatrice, mais deviennent vite fastidieuses lorsqu'elles doivent être utilisées de façon répétée, notamment au cours de l'ajustement fin de tels circuits.

A titre de comparaison, entrez sur votre ZX-81 le programme suivant :

```
5 REM "FREQUENCE RC"
10 PRINT "CALCUL DE LA FREQUEN
CE DE"
12 PRINT "COUPURE D'UN CIRCU
IT RC"
15 PRINT
17 PRINT
20 PRINT "VALEUR DE LA RESISTA
NCE EN OHM?"
25 INPUT R
30 PRINT R
33 PRINT
35 PRINT
37 PRINT "VALEUR DE LA CAPACIT
E EN FARAD?"
39 INPUT C
40 PRINT C
43 PRINT
45 PRINT
50 LET F=1/(2*PI*R*C)
53 PRINT "=====
55 PRINT "LA FREQUENCE DE COUP
URE"
60 PRINT "EST :";F;" HZ"
63 PRINT "=====
65 PRINT
70 PRINT
80 PRINT "FIN DU CALCUL"
90 REM "COPYRIGHT 1982"
```

□ Attention : ne pas confondre les deux guillemets obtenus en pressant la touche Q avec ceux obtenus en pressant deux fois la touche P (voir page 51).

et lancez-le en faisant RUN NEWLINE.

Vous constatez que ce programme permet à l'ordinateur de « dialoguer » avec vous, en français, au moyen de questions parfaitement claires.

Répondez-lui donc en fixant, par exemple, la valeur de la résistance à 2 700 Ω .

CALCUL DE LA FREQUENCE DE
COUPURE D'UN CIRCUIT RC

VALEUR DE LA RESISTANCE EN OHM?
2700

VALEUR DE LA CAPACITE EN FARAD?
1E-7

=====
LA FREQUENCE DE COUPURE
EST : 589.46275 HZ
=====

FIN DU CALCUL

Pour ce faire, vous avez le choix entre deux modes d'entrée, comme sur une calculatrice :

- frapper directement 2700 ;
- employer la notation scientifique en frappant 2.7E3 (soit $2,7 \cdot 10^3$ ou 2,7 k Ω). On remarque que la virgule doit être transformée en point (notation anglo-saxonne), et que « l'exposant » doit être séparé de la « mantisse » par la lettre E, éventuellement suivie du signe – dans le cas de puissances négatives de 10.

Appuyez sur NEWLINE pour « valider » votre entrée (jusqu'à présent, vous pouviez la corriger à loisir au moyen du curseur mobile et de la touche RUBOUT : essayez par exemple de transformer 2 700 Ω en 27 k Ω , puis de revenir à 2 700 Ω).

La valeur entrée s'inscrit à la suite de la question à des fins de contrôle (l'ordinateur choisit lui-même le mode d'affichage le mieux adapté, normal ou scientifique), et la valeur de la capacité vous est demandée.

Entrez par exemple $0,1 \mu\text{F}$, c'est-à-dire :

.1E - 6 (on n'entre pas le zéro précédent la virgule des nombres inférieurs à 1).

La machine mémorise cette donnée sous la forme équivalente :

1E - 7

et indique immédiatement la fréquence de coupure correspondant aux valeurs R et C précédemment introduites. L'affichage persistera jusqu'à ce que vous relanciez le programme par RUN NEWLINE.

Si vous souhaitez effacer ce programme, il vous suffit de faire : NEW NEWLINE ou simplement de débrancher la machine, mais vous souhaiterez peut-être le conserver sur cassette.

Pour ce faire, après avoir connecté et réglé votre magnétophone comme indiqué dans le manuel Sinclair (une modification des cordons peut éventuellement s'imposer, voir appendice), vous frappez la commande suivante :

SAVE "FREQUENCE RC"

(Voir également le chapitre 10 de cet ouvrage.)

Après avoir enclenché votre magnétophone en enregistrement, vous presserez NEWLINE, et attendrez le compte rendu \emptyset/\emptyset sur l'écran pour l'arrêter.

Par la suite, pour recharger ce même programme, il vous suffira de ramener la bande au début, de frapper

LOAD "FREQUENCE RC"

puis, après avoir enclenché le magnétophone en lecture, de presser NEWLINE. Dès l'apparition du compte rendu \emptyset/\emptyset , le programme est chargé. Toute autre réaction de l'ordinateur indique un incident de transmission (neuf fois sur dix un mauvais réglage de la commande de volume du magnétophone), auquel cas il faut mettre le ZX-81 hors tension puis le réalimenter et refaire une tentative. Dès l'obtention de \emptyset/\emptyset , vous pouvez soit lister le programme en faisant :

LIST NEWLINE

soit le lancer directement par :

RUN NEWLINE

Lorsque votre cassette contiendra de nombreux programmes, l'instruction LOAD permettra de retrouver celui dont vous donnez le nom. Sachez cependant qu'une parfaite rigueur doit être respectée dans l'énoncé du nom, et que "FREQUENCE RC" n'est pas identique à "FREQUENCERC" ou "FREQUENCE", par exemple. Si toutefois il vous arrive de vouloir charger un programme dont vous ignorez le nom, frappez :

LOAD"" (sans espace entre les deux guillemets)

puis NEWLINE, et la machine chargera le premier programme qui se présentera.

Essayons à présent de comprendre le fonctionnement de ce programme :

L'instruction 5, du type « REM », ne sert strictement à rien au niveau de l'exécution du programme, mais permet d'en faire apparaître le titre sur les listings à des fins d'identification. De même pour l'instruction 90 qui ne sert qu'à rappeler aux lecteurs de cet ouvrage que l'utilisation de ces programmes doit être limitée à des usages personnels, et qu'aucune exploitation commerciale ne peut être envisagée sans une autorisation expresse de l'auteur.

Dans tous les cas, les instructions REM peuvent être supprimées dès que la place mémoire vient à manquer, de même que leur contenu (tout ce qui se trouve entre les guillemets) peut être modifié à loisir.

Les instructions 10 et 12 impriment « l'objet du calcul » : deux lignes ont été utilisées au lieu d'une afin de permettre une présentation agréable, évitant toute coupure intempestive d'un mot grâce à un retour à la ligne anticipé.

Les instructions 15 et 17 font sauter deux lignes afin d'aérer la mise en page. Là aussi, de telles instructions peuvent disparaître pour dégager de la place mémoire.

L'instruction 20 imprime la première question, et est immédiatement suivie de l'instruction 25 qui place la machine en attente tant que la réponse n'aura pas été donnée. Dès que cette dernière aura été validée par NEWLINE, la variable R prendra la valeur entrée.

Si vous appuyez sur NEWLINE sans avoir indiqué de valeur, il apparaît un double curseur « LS » trahissant l'erreur de syntaxe. Vous pouvez alors « rectifier le tir » en entrant sans autre précaution une valeur vraisemblable.

Dès acceptation de la valeur de R, la machine en imprime la valeur sous la question grâce à l'instruction 30.

Après le saut de deux lignes par les instructions 33 et 35, le même scénario se déroule pour l'acquisition de la valeur de C (instructions 37 à 45).

A la ligne 50 s'opère le calcul proprement dit, selon la formule indiquée plus haut, et qui a été transcrise selon les règles du langage BASIC. Les parenthèses sont indispensables, car en leur absence, la machine exécuterait l'opération suivante :

$$F = 0,5 \times \pi \times R \times C \text{ (car } 1/2 = 0,5\text{)}$$

ce qui ne conduit pas précisément au même résultat.

Vous pouvez provoquer l'erreur (elle est riche d'enseignements) en réécrivant la ligne 50 sans parenthèses.

Notez à cette occasion que la commande EDIT fonctionne toujours sur un programme entré au clavier mais pas nécessairement lorsque le chargement a été effectué à partir de la cassette.

Les instructions 53 et 63 ne servent qu'à des fins de mise en page, afin de faire ressortir le résultat.

La ligne 60 est typique des multiples possibilités de l'instruction PRINT :

Elle commence par imprimer le mot "EST : ", à la ligne par rapport au texte dû à l'instruction 55. On notera la présence d'un espace après les deux points, et il pourra être intéressant de le supprimer pour mettre en relief son rôle de présentation.

Le point-virgule situé après le guillemet de fermeture permet à la valeur de F résultant du calcul d'être imprimée à la suite de "EST : " et non à la ligne. De même pour le point virgule précédent le texte " HZ" (noter l'espace précédent ces deux lettres).

Enfin, les instructions 65 à 80 mettent en page l'impression de « FIN DU CALCUL », texte dont on pourrait fort bien se dispenser en cas de pénurie de mémoire.

On notera qu'il ne faut pas abuser des artifices de mise en page tels que les sauts de lignes ou les bandeaux décoratifs, au risque de dépasser la capacité de l'écran, qui n'est que de 22 lignes de

32 caractères. En cas de dépassement, on obtient un compte rendu de la forme 5/N° de ligne, indiquant qu'il est possible de visualiser les lignes non affichées par manque de place en faisant :

CONT NEWLINE

étant entendu, bien sûr, que l'image précédente sera alors effacée et perdue.

L'essai est facile à faire en ajoutant des instructions de saut de ligne dans le programme.

Dernière remarque importante : le BASIC du ZX-81 possédant une fonction automatique appelée « évaluateur d'expressions », il est possible de modifier ainsi le programme afin de gagner une ligne :

- supprimer la ligne 50 (faire 50 NEWLINE) ;
- remplacer la ligne 60 par :

60 PRINT "EST : " ; 1/(2*Pi*R*C) ; " HZ"

Toutefois, cette façon de procéder n'est rentable qu'à la condition que la valeur de F ne soit pas utilisée plus loin dans le programme.

Dans le même ordre d'idées, nous vous proposons deux programmes permettant de calculer la valeur du condensateur à partir de celles de la résistance et de la fréquence de coupure, et inversement, de la résistance à partir de la capacité et de la fréquence.

```

5 REM "CAPACITE RC"
10 PRINT "CALCUL DE LA CAPACIT
E"
15 PRINT "D'UN CIRCUIT RC"
17 PRINT
18 PRINT
20 PRINT "FREQUENCE DE COUPURE
"
21 PRINT "DESIREE EN HERTZ?"
30 INPUT F
35 PRINT F
37 PRINT
40 PRINT "VALEUR DE LA RESISTA
NCE EN OHM?"
45 INPUT R
50 PRINT R
52 PRINT
53 PRINT
54 PRINT "=====

```

(Suite au verso)

```

=====
55 PRINT
60 LET C=1/(2*PI*F*R)
65 PRINT "LA VALEUR DE LA CAPACITE"
66 PRINT "EST: ";C;" FARADS"
67 PRINT
70 PRINT "===== "
75 PRINT
80 PRINT "FIN DU CALCUL"
90 REM "COPYRIGHT 1982"

```

```

5 REM "RESISTANCE RC"
10 PRINT "CALCUL DE LA RESISTANCE"
20 PRINT "D'UN CIRCUIT RC"
25 PRINT
30 PRINT
40 PRINT "FREQUENCE DE COUPURE"
45 PRINT "SOUHAITEE EN HERTZ?"
50 INPUT F
55 PRINT F
57 PRINT
58 PRINT
60 PRINT "VALEUR DE LA CAPACITE EN FARAD?"
65 INPUT C
70 PRINT C
71 PRINT
72 PRINT
75 LET R=1/(2*PI*F*C)
80 PRINT "===== "
90 PRINT "LA VALEUR DE LA RESISTANCE"
95 PRINT "EST : ";R;" OHMS"
100 PRINT "===== "
110 PRINT ...
120 PRINT "FIN DU CALCUL"
130 REM "COPYRIGHT 1982"

```

Le déroulement de ces programmes est tout à fait similaire à celui du précédent, au point même qu'il est possible de les entrer en machine à partir d'un enregistrement sur cassette du premier des trois, dans lequel on corrigera les lignes qui diffèrent. Le gain de temps est minime, mais l'exercice est profitable.

CALCUL DE LA CAPACITE
D'UN CIRCUIT RC

FREQUENCE DE COUPURE
DESIREE EN HERTZ?
3000

VALEUR DE LA RESISTANCE EN OHM?
22000

=====
LA VALEUR DE LA CAPACITE
EST: 2.4114385E-9 FARADS

=====
FIN DU CALCUL

CALCUL DE LA RESISTANCE
D'UN CIRCUIT RC

FREQUENCE DE COUPURE
SOUHAITEE EN HERTZ?
3000

VALEUR DE LA CAPACITE EN FARAD?
2.2E-9

=====
LA VALEUR DE LA RESISTANCE
EST : 24114.385 OHMS

=====
FIN DU CALCUL

On remarquera deux instructions PRINT encadrant le résultat du calcul, et qui servent à aérer la mise en page en écartant d'une ligne chacun des deux bandeaux décoratifs réalisés au moyen de signes =. A des fins de comparaison, cette variante est incorporée à l'un seulement des deux programmes.

2^o Programmes pour le calcul des circuits LC

Le calcul des circuits résonnantes LC fait appel à la célèbre *formule de Thomson* :

$$f = \frac{1}{2\pi\sqrt{LC}}$$

qui peut se transformer en :

$$L = \frac{1}{4\pi^2 f^2 C}$$

et

$$C = \frac{1}{4\pi^2 f^2 L}$$

Ces trois formules sont de première importance pour tout pratiquant de la radio, mais restent d'un emploi fastidieux même sur une calculatrice, dès lors que de nombreuses déterminations successives sont à prévoir. Le programme ci-dessous permet une exploitation bien plus confortable de la formule de base.

```

2 REM "FREQUENCE LC"
5 PRINT "      CALCUL DE LA FRE
QUENCE"
6 PRINT "DE RESONANCE D""UN C
IRCUIT LC"
10 PRINT
12 PRINT "CAPACITE EN FARAD?"
15 INPUT C
16 PRINT C
17 PRINT
18 PRINT
20 PRINT "SELF EN HENRY?"
25 INPUT L
26 PRINT L
30 LET F=1/(2*PI*SQR (L*C))
31 PRINT
32 PRINT "=====
35 PRINT "LA FREQUENCE DE RESO
NANCE EST: "F;"HERTZ"
36 PRINT "=====
40 PRINT
45 PRINT
50 PRINT "FIN DU CALCUL"
60 REM "COPYRIGHT 1982"

```

Le début de ce programme, consacré au dialogue permettant l'acquisition des données nécessaires, est tout à fait semblable à ce qui a été vu auparavant.

Au niveau de la ligne 30, réservée au calcul de la fréquence (raison d'être de ce programme), il faut noter le recours à la fonction « racine carrée », nommée SQR (de l'anglais SQuare Root).

Comme toutes les autres fonctions mathématiques du ZX-81, la fonction SQR peut s'appliquer directement à un nombre ou à une variable (exemple : SQR 2, SQR A, etc.), mais également, comme c'est le cas ici, à une expression, aussi complexe soit-elle.

Dans notre programme, l'expression est (L*C), que nous avons placée entre parenthèses pour éviter tout risque de confusion entre :

SQR (L*C) soit \sqrt{LC}

et

SQR L*C soit $C\sqrt{L}$

On notera qu'il est indispensable que l'expression à laquelle s'applique la fonction SQR ne devienne jamais négative, sous peine de lancement immédiat d'un compte rendu d'erreur A. Egalement, on remarquera la constitution particulière de la ligne 35, effectuant l'impression du résultat : les différentes parties du « texte » à éditer sont séparées par des virgules au lieu de point-virgules, et aucun espace n'est prévu pour encadrer les textes placés entre guillemets. Malgré tout, la mise en page reste claire.

Ceci s'explique par le fait que la virgule n'autorise que deux positions d'impression : au début d'une ligne (colonne zéro) ou bien en milieu de ligne (colonne 16). Comme la longueur du texte « la fréquence de résonance est : » dépasse 16 caractères mais reste inférieure à 32 caractères, la valeur de F ne peut qu'être inscrite après un retour à la ligne. Egalement, comme la longueur de la valeur de la fréquence ne dépassera pas 16 caractères, le texte « hertz » reste sur la même ligne, à la 16^e colonne, donc après un blanc de longueur suffisante.

Cette façon de programmer la sortie d'un résultat encadré de textes économise de la place mémoire, car elle n'exige qu'une ligne de programme. En revanche, elle réclame davantage de réflexion lors de l'écriture du programme : constatez ce qui se produit si vous remplacez le texte :

"la fréquence de résonance est :"
par "la fréquence de résonance du circuit est :"

**CALCUL DE LA FREQUENCE
DE RESONANCE D'UN CIRCUIT LC**

CAPACITE EN FARAD?
2.2E-11

SELF EN HENRY?
.000015

=====
LA FREQUENCE DE RESONANCE EST:
8761191.3 **HERTZ**
=====

FIN DU CALCUL

C'est pourquoi il est souvent préférable, lorsque la pénurie de mémoire ne se fait pas encore sentir, de prévoir plusieurs lignes de programme utilisant point-virgules et espaces pour faire exécuter à la machine une mise en page impeccable et sans surprises.

```
5 REM "CAPACITE LC"
10 PRINT " CALCUL DE LA CAPA
CITE"
12 PRINT "D""ACCORD D""UN CIRC
UIT LC"
14 PRINT
15 PRINT
20 PRINT "FREQUENCE D""ACCORD
DESIREE"
22 PRINT "EN HERTZ"
25 INPUT F
30 PRINT F
32 PRINT
33 PRINT
35 PRINT "VALEUR DE LA SELF EN
HENRY?"
40 INPUT L
45 PRINT L
47 PRINT
48 PRINT
49 PRINT
```

```
50 LET C=1/(4*PI*PI*F*L)
53 PRINT "=====
55 PRINT "LA CAPACITE EST:",C,
"FARAD"
57 PRINT "=====
60 PRINT
62 PRINT
64 PRINT
65 PRINT "FIN DU CALCUL"
70 REM "COPYRIGHT 1982"
```

```
5 REM "SELF LC"
10 PRINT "CALCUL DE LA SELF"
12 PRINT "D""UN CIRCUIT LC"
14 PRINT
15 PRINT
16 PRINT
20 PRINT "FREQUENCE D""ACCORD
DESIREE"
22 PRINT "EN HERTZ?"
25 INPUT F
30 PRINT F
32 PRINT
33 PRINT
35 PRINT "VALEUR DE LA CAPACIT
E EN FARAD?"
37 INPUT C
40 PRINT C
42 PRINT
43 PRINT
44 PRINT
45 PRINT "=====
50 LET L=1/(4*PI*PI*F*L)
55 PRINT "LA VALEUR DE LA SELF
EST:"
57 PRINT L,"HENRY"
62 PRINT "=====
63 PRINT
65 PRINT "FIN DU CALCUL"
70 REM "COPYRIGHT 1982"
```

Les programmes ci-dessus nécessitent, pour le calcul des éléments L ou C à partir de la fréquence de résonance désirée, l'élévation au carré de certaines quantités, conformément aux formules qui ont été données.

Pour ce faire, il existe bien la fonction « puissance », repérée par le signe ** sur le clavier du ZX-81.

En fait, cette fonction est surtout utile pour les puissances entières supérieures à 2, et encore plus pour les puissances fractionnaires, donc pour les racines nièmes.

Par exemple, on peut citer :

$$2^{16} = 65\,536 \text{ obtenu par } 2**16$$

$$\sqrt[4]{2} = 2^{1/4} = 1,19 \text{ obtenu par } 2**.25$$

Dans le cas qui nous intéresse, il est tout aussi commode d'utiliser une double multiplication pour nos élévations au carré, comme on le voit à la ligne 50 du programme de la page 46, et à la ligne 50, également, du programme de la page 47.

La suite de ces programmes ne présente pas de différence notable par rapport aux précédents, et n'appelle donc pas de commentaire particulier.

```

CALCUL DE LA CAPACITE
D'ACCORD D'UN CIRCUIT LC
FREQUENCE D'ACCORD DESIREE
EN HERTZ
455000

VALEUR DE LA SELF EN HENRY?
.002
=====
LA CAPACITE EST:
6.1176901E-11 FARAD
=====

FIN DU CALCUL

CALCUL DE LA SELF
D'UN CIRCUIT LC

FREQUENCE D'ACCORD DESIREE
EN HERTZ?
455000

VALEUR DE LA CAPACITE EN FARAD?
1E-9
=====
LA VALEUR DE LA SELF EST:
.0001223538 HENRY
=====
FIN DU CALCUL

```

3° Programmes pour le calcul des groupements série/parallèle

On sait que la mise en série de condensateurs, ou en parallèle de résistances, donne un résultat global qui peut être déterminé au moyen des deux formules suivantes :

$$\frac{1}{C} = \frac{1}{C_1} + \frac{1}{C_2} + \dots + \frac{1}{C_n} \text{ (en série)}$$

$$\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2} + \dots + \frac{1}{R_n} \text{ (en parallèle)}$$

Les deux programmes suivants, qui ne diffèrent l'un de l'autre que par les « textes » des instructions PRINT, permettent la mise en œuvre de ces formules pour un nombre quelconque de composants mis en série ou en parallèle.

```

10 PRINT "RESISTANCES EN //"
12 PRINT
15 LET C=0
16 LET N=1
20 PRINT "RESISTANCE ";N;" EN
OHMS ?"
22 INPUT A$
24 IF A$="" THEN GOTO 35
25 LET C=C+1/VAL A$
26 LET N=N+1
27 CLS
30 GOTO 20
35 LET R=1/C
36 CLS
37 PRINT
38 PRINT "===== "
39 PRINT
40 PRINT "RESISTANCE EQUIVALEN
TE"
42 PRINT "AUX ";N-1;" EN // :"
43 PRINT
45 PRINT R;" OHMS"
47 PRINT
50 PRINT "===== "
52 REM "COPYRIGHT 1982"

```

```

10 PRINT "CONDENSATEURS EN SER
IE"
12 PRINT
15 LET C=0
16 LET N=1
20 PRINT "CAPACITE ";N;" EN FA
RAD ?"
22 INPUT A$
24 IF A$="" THEN GOTO 35
25 LET C=C+1/VAL A$
26 LET N=N+1
27 CLS
30 GOTO 20
35 LET R=1/C
36 CLS
37 PRINT
38 PRINT "====="
39 PRINT
40 PRINT "CAPACITE EQUIVALENTE"
42 PRINT "AUX ";N-1;" EN SERIE
43 PRINT
45 PRINT R;" FARAD"
47 PRINT
50 PRINT "====="
51 REM "COPYRIGHT 1982"

```

Essayons de « faire tourner » le programme, en calculant pour commencer la résistance équivalente à deux fois 100 Ω en parallèle.

```

RESISTANCES EN //
RESISTANCE 1 EN OHMS ?

100
RESISTANCE 2 EN OHMS ?

100
RESISTANCE 3 EN OHMS ?

=====
RESISTANCE EQUIVALENTE
AUX 2 EN // :
50 OHMS
=====
```

Faisons RUN NEWLINE pour démarrer, la machine nous demande aussitôt la valeur de la première résistance.

Répondons par 100 NEWLINE, elle réclame la seconde. Entrons encore 100 NEWLINE, et... elle en réclame une troisième ! Ce programme est visiblement insatiable et, s'il n'était prévu un moyen de l'informer que nous n'avons plus d'autre valeur à entrer, il en réclamerait d'autres jusqu'à la fin des temps, ou peu s'en faut ! Répondons à la troisième sollicitation par NEWLINE, et constatons que la machine calcule bien la résistance équivalente à la mise en parallèle des deux valeurs que nous lui avons indiquées, soit 50 Ω .

Comment cela se peut-il puisque nous avons vu qu'il n'était pas possible d'entrer une variable indéterminée, par NEWLINE, sans affoler la machine, et recevoir un curseur d'erreur de syntaxe ? La réponse se situe sur l'écran lorsque la machine demande une valeur : le curseur L se trouve placé entre deux guillemets, ce qui signifie que l'ordinateur n'attend pas un « nombre » (suite de chiffres exclusivement, ou « mot numérique »), ou à la rigueur une expression du genre 2 + 2, mais une « chaîne ». Une chaîne est une suite de caractères quelconques, chiffres, lettres, signes, espaces, symboles, ou même mots clef, autrement dit de « mots alphanumériques ». Le rôle principal des chaînes est de permettre à la machine de traiter du texte (lorsque nous avons fait PRINT "PRENOM", "PRENOM" était une chaîne), mais nous nous en servons ici pour « véhiculer » des valeurs numériques.

L'avantage de ce procédé est que, lorsque la machine attend une « chaîne », elle ne se « plante » pas si l'on se contente de presser NEWLINE. Elle comprend simplement que l'entrée consistait en une « chaîne vide », ne comportant aucun caractère ni espace.

La « chaîne vide » se symbolise par "", symbole obtenu en pressant deux fois la touche de guillemet (P en majuscule). Rappelons encore que la touche de double guillemet (Q en majuscule) ne peut en aucun cas remplacer cette double action sur la touche de simple guillemet. Le double guillemet sert uniquement lorsque l'on veut, à l'intérieur d'une chaîne qui sera imprimée, prévoir un guillemet ou une apostrophe. Il n'est pas question d'utiliser le simple guillemet "P", car il serait interprété par la machine comme la fin de la chaîne. Aussi a-t-on prévu le double guillemet "Q" qui s'imprime sous la forme d'un guillemet simple, lorsqu'il apparaît dans une chaîne.

Le contenu d'une chaîne se présente donc toujours encadré de guillemets, mais il faut également pouvoir donner des noms aux

chaînes sur lesquelles la machine va travailler. Le choix de ces noms est plus réduit qu'en ce qui concerne les variables, car les 26 seules possibilités existantes sont : toute lettre de l'alphabet suivie du caractère \$ (touche U en majuscule). Les 26 chaînes possibles sont donc nommées A\$ à Z\$. Notons que, sur le ZX-81, A£ n'est pas une chaîne, comme sur certains autres ordinateurs BASIC.

Dans notre programme, la chaîne que nous entrons est appelée A\$, et son acquisition se fait au niveau de l'instruction 22.

L'instruction 24, qui la suit immédiatement, regroupe deux fonctions très puissantes du langage BASIC :

- un « branchement conditionnel » IF THEN
- un renvoi GOTO

Ces deux fonctions sont à peu près indispensables à l'écriture de tout programme tant soit peu élaboré. Les deux mots anglais IF et THEN correspondent ici à SI et ALORS en français. De ce fait, il est facile de comprendre qu'une instruction contenant ces deux mots (ils sont d'ailleurs indissociables sous peine d'erreur de syntaxe) déclenchera l'exécution de l'ordre placé après THEN, si, et seulement si, la condition placée après IF est remplie. Dans le cas contraire, l'instruction IF-THEN est purement et simplement ignorée.

En ce qui nous concerne, l'instruction 24 n'entre en jeu que si la chaîne A\$ qui vient d'être entrée est une chaîne vide, autrement dit, que si une pression sur NEWLINE a été effectuée sans introduction d'une valeur de résistance. Si tel est bien le cas, donc, la machine exécute l'ordre prévu à la suite de THEN, et qui s'écrit GOTO 35. GOTO signifie tout simplement « aller à », ce qui fait que, si la machine exécute le GOTO 35 de l'instruction 24, elle sautera les étapes du programme matérialisées par les instructions 25, 26, 27, 30, pour exécuter directement l'instruction 35.

Mais nous n'en sommes pas encore là, puisqu'il nous faut d'abord rentrer un certain nombre de valeurs de résistances.

Dès le début du programme, l'instruction 16 a affecté à la variable N la valeur 1 (l'instruction LET= est d'ailleurs appelée instruction d'affectation).

L'instruction 20 va donc utiliser cette valeur (provisoire) de N pour imprimer "RESISTANCE 1 EN OHMS?". Aussitôt après, la machine passe en attente jusqu'à ce qu'une pression sur NEWLINE lui signale que vous avez terminé de frapper ladite valeur.

Si les conditions précédemment définies pour la mise en branle

de l'instruction 24 ne sont pas réunies (donc si nous avons frappé une valeur, quelle qu'elle soit), l'instruction 25 est exécutée à la suite de l'instruction 22.

La variable C, qui valait 0 depuis l'instruction 15, se voit affecter une valeur augmentée de la quantité :

$$\frac{1}{\text{VAL A\$}}$$

La fonction VAL, appliquée à une chaîne telle que A\$, extrait la valeur numérique de cette chaîne. Cette fonction ne peut bien évidemment remplir son office que si la chaîne ne contient que des chiffres. Essayez, par exemple, de frapper CENT au lieu de 100 : vous obtiendrez le compte rendu 2/25, car la variable VAL A\$ est indéterminée dans ce cas précis. En fait, la fonction VAL sert à transformer en véritable variable numérique une chaîne qui contient cette valeur numérique, car une chaîne ne peut faire l'objet d'aucune manipulation arithmétique.

Donc, si A\$ contient une valeur de résistance R_1 , $\frac{1}{\text{VAL A\$}}$ devient $\frac{1}{R_1}$. Puisque, pour le moment, C = 0, la variable C prend la valeur $\frac{1}{R_1}$.

Cela étant fait, on incrémente d'une unité la variable N, qui passe ainsi de 1 à 2, servant en fait de « compteur de valeurs demandées ».

Juste après cette mise à jour de N au moyen de l'instruction 26, nous effaçons l'écran au moyen de l'instruction 27 CLS (Clear Screen en anglais). Cet effacement n'affecte en rien les valeurs contenues dans les mémoires de la machine.

Ensuite, l'instruction 30 renvoie le programme à la ligne 20, pour qu'il réclame la valeur de la seconde résistance, puisque N est passé à 2 peu de temps auparavant. Après le processus habituel d'acquisition, l'opération $\frac{1}{\text{VAL A\$}}$ est effectuée par l'instruction 25,

et la valeur de $\frac{1}{R_2}$ ainsi obtenue est ajoutée à C, qui valait $\frac{1}{R_1}$ et qui devient $\frac{1}{R_1} + \frac{1}{R_2}$.

L'instruction 15, qui affecte la valeur 0 à C, ne sera plus jamais exécutée (pas plus que l'instruction 16), puisque le retour en arrière

se fait au niveau de la ligne 20. On peut s'interroger sur le rôle de cette instruction, puisque la ligne 25 semble suffire pour incrémenter C en fonction des valeurs de résistance amenées par A\$.

Essayons de supprimer cette ligne 15, et nous constatons que le programme s'arrête sur l'instruction 25, car il ne sait pas *à quoi* ajouter $\frac{1}{\text{VAL A\$}}$ pour obtenir C.

Bref, la variable C est indéterminée lors de ce premier passage à l'instruction 25, d'où le compte rendu 2/25. Ce cycle se poursuit jusqu'à ce que A\$ devienne une chaîne vide, auquel cas, comme nous l'avons vu, le branchement de la ligne 24 dévie le programme vers la ligne 35, par l'instruction GOTO 35.

Cette instruction 35 effectue l'inverse de C, c'est-à-dire :

$$\frac{1}{\frac{1}{R_1} + \frac{1}{R_2} + \dots + \frac{1}{R_n}}$$

La variable R, à laquelle l'instruction LET affecte cet inverse, représente tout simplement la résistance résultante.

Il ne reste plus qu'à effacer l'écran (instruction 36 CLS), et à mettre en page ce résultat au moyen des instructions 37, 38, 39, 40, 42, 43, 45, 47 et 50.

Cette débauche d'instructions n'est possible que parce que le bouclage réalisé par le branchement IF THEN de la ligne 24 économise de très nombreuses lignes. Elle présente l'avantage d'une présentation claire, puisque faisant ressortir, à des fins de vérification, le nombre de résistances composant le montage duquel on vient de calculer la résistance équivalente. Ce nombre est bien sûr égal à N-1 puisque la N^{ième} tentative d'acquisition d'une valeur a tourné court au niveau de la ligne 24, mais avait déjà été comptabilisée.

Ce programme assez court peut rendre de fiers services à qui-conque souhaite réaliser rapidement une résistance de valeur non normalisée, mais présente surtout l'avantage de mettre en œuvre les instructions essentielles du BASIC ZX-81. On peut affirmer que quiconque fera l'effort de bien comprendre tous les détails de ce programme, puis, après l'avoir effacé, de le reconstituer de mémoire, sera armé pour rédiger seul des programmes personnels d'un niveau déjà intéressant.

Terminons en signalant que le listing de la page 50 peut être très facilement frappé par quelques modifications de lignes sur le programme le précédent immédiatement et précédemment entré.

4^o Programme pour le calcul des bobinages sur ferrite

L'utilisation de noyaux ferrite simplifie énormément le calcul des bobinages, puisque le fabricant du noyau fournit un coefficient appelé A_L, à partir duquel le nombre de spires à bobiner peut être calculé par la formule :

$$n = \sqrt{\frac{L}{A_L}}$$

(A_L est exprimée en nH/sp²) et L en nH.

Un très court programme permet d'automatiser ce calcul :

```

5 REM "POT FERRITE"
10 PRINT "VALEUR DE LA SELF"
12 PRINT "A REALISER EN HENRY
??
15 INPUT L
17 CLS
20 PRINT "COEFFICIENT DU POT"
22 PRINT "EN NH/SPIRE 2 ?"
25 INPUT A
27 CLS
30 LET N=INT SQR (L/(A*1E-9))
35 PRINT AT 10,0;"POUR FAIRE U
NE SELF DE ";L;" H"
37 PRINT
40 PRINT "IL FAUT BOBINER ";N;
" SPIRES"
42 PRINT
50 PRINT "DANS UN POT DE ";A;" "
NH/SP 2"
60 REM "COPYRIGHT 1982"
```

Toute sa première partie est très classique, l'originalité se situant au niveau de l'impression du résultat, qui reprend les données fournies, de façon à faire une synthèse commode de tous les paramètres de fabrication, et au niveau du centrage au milieu de l'écran de ce résultat.

Ce centrage est obtenu au moyen de la partie AT de l'instruction 35. En effet, si l'on insère, après le mot PRINT d'une instruction d'écriture, les indications :

AT L,C ;

avec L = numéro de ligne et C = numéro de colonne, l'impression débutera à la ligne et à la colonne indiquées, pour se poursuivre normalement ensuite. Rappelons simplement que la partie utile de

l'écran comporte 22 lignes numérotées de 0 à 21 en partant du haut, et 32 colonnes, numérotées de 0 à 31 à partir de la gauche.

Pour notre part, PRINT AT 10,0 ; permet de sauter les dix premières lignes, sans encombrer le programme de dix instructions PRINT de saut de ligne.

VALEUR DE LA SELF
A REALISER EN HENRY ?

50E-3

COEFFICIENT DU POT
EN NH/SPIRE 2 ?

250

POUR FAIRE UNE SELF DE .05 H

IL FAUT BOBINER 447 SPIRES
DANS UN POT DE 250 NH/SP 2

5^e Programme pour l'exploitation de la formule de Nagaoka (calcul des bobinages sur air)

Nous voudrions terminer cette série de programmes dédiés à nos amis électroniciens en rendant à ceux-ci un signalé service : celui d'automatiser à leur intention la mise en œuvre, ô combien fastidieuse, de la formule de Nagaoka qui seule, permet de calculer avec une certaine précision les bobinages sur air.

Rappelons que cette formule s'écrit :

$$L = \frac{0,08 n^2 d^2}{3d + 91 + 10e}$$

avec :

L = self en microhenrys

d = diamètre moyen du bobinage en centimètres

l = longueur (ou hauteur) en centimètres

e = épaisseur de fil en centimètres

Notre programme fonctionne « à l'envers » du précédent, puisqu'il calcule la self à partir des différentes données géométriques qui concernent la réalisation du bobinage. Cette façon de procéder s'avère indispensable, car la recherche des bonnes valeurs de cons-

truction ne peut se faire que par approches successives, chaque paramètre réagissant sur les autres.

```

5 REM "NAGAOKA"
10 PRINT "NOMBRE DE SPIRES PREU
UU ?"
12 INPUT N
14 CLS
15 PRINT "Ø MOYEN ""D"" EN CM
?""
17 INPUT D
18 CLS
20 PRINT "LONGUEUR ""L"" EN CM
?""
22 INPUT L
24 CLS
25 PRINT "EPAISSEUR ""E"" EN C
M ?"
27 INPUT E
28 CLS
30 LET L=.08*N*N*D*D/((3*D)+(9
*L)+(10*E))
35 PRINT "LA VALEUR DE LA SELF
"
37 PRINT
40 PRINT "AINSI OBTENUE EST :
"
45 PRINT
50 PRINT L; " MICROHENRYS"
60 REM "COPYRIGHT 1982"

```

Ce programme, surtout destiné à soulager les usagers de cette formule fort utile, mais bien lourde à manier, ne comporte pas d'astuce particulière de conception. Les plus courageux de nos lecteurs pourront le modifier de façon à obtenir le centrage du résultat sur l'écran !

```

NOMBRE DE SPIRES PREU ?
```

50

Ø MOYEN "D" EN CM ?

1

LONGUEUR "L" EN CM ?

EPAISSEUR "E" EN CM ?

.1

LA VALEUR DE LA SELF

AINSI OBTENUE EST :

15.384615 MICROHENRYS

Un peu de détente !

Entre deux vagues d'applications sérieuses, et surtout en début d'apprentissage de la programmation, il semble souhaitable de marquer une pause en étudiant quelques programmes récréatifs, non dépourvus toutefois de caractère instructif.

1^o Faisons dessiner notre ZX-81

Le ZX-81 possède une large gamme de fonctions graphiques, qui représentent une mine inépuisable de sujets de distraction.

Egalement, la réalisation de graphiques complexes exige le recours à des *tâches répétitives*, domaine dans lequel le ZX-81 se sent tout particulièrement à l'aise grâce à sa possibilité d'exécuter des *boucles "FOR-NEXT"*.

```
5 REM "GRILLE"
10 FOR N=0 TO 40 STEP 5
20 FOR M=0 TO 40
30 PLOT N,M
40 PLOT M,N
50 NEXT M
60 NEXT N
70 REM "COPYRIGHT 1982"
```

La fonction graphique la plus puissante du ZX-81 est sans nul doute la fonction PLOT. Il suffit, en effet, d'indiquer à la machine :

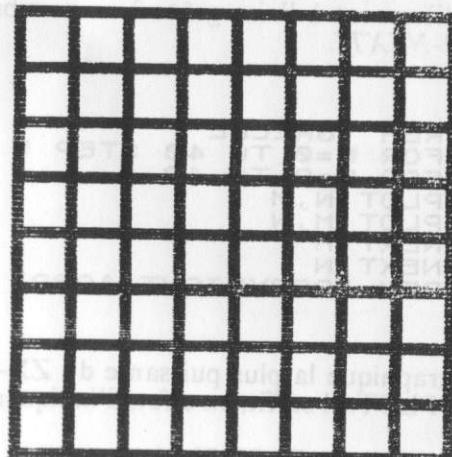
PLOT X, Y

avec X compris entre 0 et 63, et Y compris entre 0 et 43, pour que s'inscrive sur l'écran un petit carré noir dont l'emplacement correspond à ces coordonnées « cartésiennes ». Selon les conventions habituelles en la matière, les coordonnées X croissent de gauche à droite, alors que les Y croissent de bas en haut. On ne confondra pas ces conventions avec celles régissant le fonctionnement de l'instruction PRINT AT L,C; qui sont, nous l'avons vu, tout à fait différentes.

Cette instruction PLOT permet « d'adresser » séparément chacun des $44 \times 64 = 2816$ petits carrés de l'écran. Ceci signifie, qu'à condition d'écrire le programme correspondant, on peut faire effectuer à l'ordinateur tout dessin qui serait obtenu manuellement en noircissant à volonté n'importe quel nombre de carreaux d'une feuille quadrillée 5×5 mm de 22×32 cm ! Il faut reconnaître qu'une telle possibilité est inestimable, comme le prouveront les applications proposées plus loin. Signalons également la fonction UNPLOT, capable de supprimer les inscriptions exécutées par PLOT, et dont le fonctionnement est similaire.

La seule contrainte de ces instructions est qu'il faut veiller à ce que les variables X et Y ne sortent jamais de l'intervalle qui leur est imparti, sous peine d'arrêt immédiat du programme en cours, sur un compte rendu de type B.

Le programme fourni ci-dessus se lance par RUN NEWLINE, et s'arrête seul, sitôt son travail achevé, environ vingt secondes plus tard.



Si l'on est pressé, on peut réduire ce délai à cinq secondes en envoyant l'ordre FAST avant de lancer le programme. Cependant, l'écran étant neutralisé pendant les calculs, on se prive du plaisir de voir le dessin se construire peu à peu.

Cette grille peut servir de mire pour contrôler les éventuels défauts de géométrie d'un récepteur TV, ou d'un magnétoscope. Elle comporte 40 éléments d'image par côté, ce qui explique que les instructions FOR fixent une limite de 40 à l'ardeur du programme.

En effet, une boucle FOR NEXT (constituée de deux instructions indissociables, par exemple, ici, les lignes 10 et 60, ou bien les lignes 20 et 50) ordonne à la machine d'effectuer le travail matérialisé par la partie de programme située entre ces deux instructions, pour toutes les valeurs de la variable citée après FOR, comprises entre les deux limites données de part et d'autre de TO. Dans notre cas, la boucle « M » exécutera quarante et une fois les deux instructions 30 et 40, pour M variant de 0 à 40.

Seulement, la boucle « M » est elle-même contenue dans une seconde boucle, appelée « N ». Celle-ci présente une particularité, à savoir la mention STEP 5.

Ceci signifie que la boucle « N » ne fera varier la variable N que de 5 en 5, et que celle-ci ne prendra donc que les 9 valeurs : 0, 5, 10, 15, 20, 25, 30, 35, 40. Essayons de supprimer l'indication STEP 5, et nous obtiendrons un listing analogue à celui-ci :

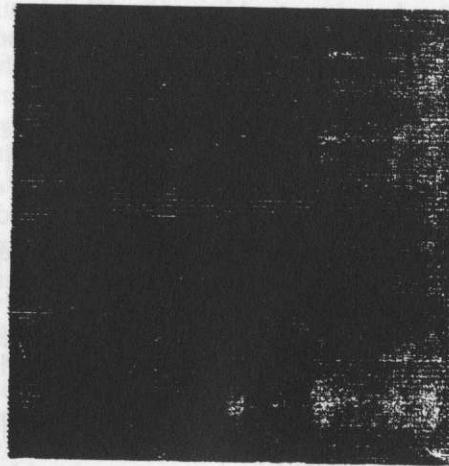
```

5 REM "CARRE"
10 FOR N=0 TO 40
20 FOR M=0 TO 40
30 PLOT N,M
40 PLOT M,N
50 NEXT M
60 NEXT N
70 REM "COPYRIGHT 1982"

```

Nous constatons que ce programme conduit à un résultat graphique très différent, un carré entièrement noir, au lieu de la grille dont les lignes étaient précisément espacées de cinq « cases » ou « éléments d'image ».

Egalement, nous remarquerons que la construction de ce carré prend 1 mn 35 s, ce qui est normal puisque les instructions à exécuter sont cinq fois plus nombreuses. On peut faire varier le « pas » de la grille en introduisant à nouveau STEP 8, ou STEP 2, ou même STEP 10 ou STEP 20, l'important étant de conserver, ici, un sous-



multiple de 40. (Essayons STEP 15, par exemple, pour juger de l'effet produit dans le cas contraire !)

Nous ne fournirons pas de longues explications quant aux détails d'écriture de ce programme. Il sera beaucoup plus profitable à nos lecteurs les plus curieux de poser directement leurs questions à la machine, par exemple en intervertissant les instructions des lignes 50 et 60, afin de découvrir les règles « d'imbrication des boucles FOR-NEXT », quelques « expériences malheureuses » judicieusement provoquées et analysées se révélant être le meilleur professeur.

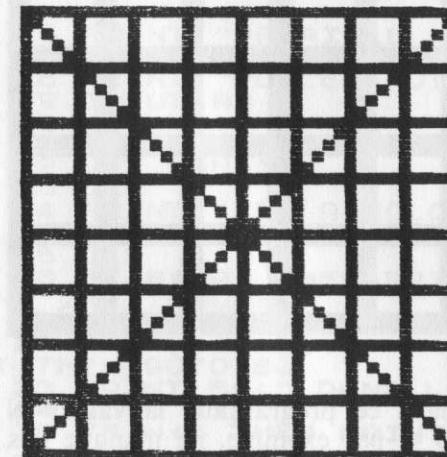
Le programme suivant agrémentera la grille de deux diagonales du plus bel effet, par le simple jeu de deux instructions supplémentaires (lignes 35 et 45), dont la compréhension constitue également un très bon exercice. En particulier, on remarquera que, si plusieurs

```

5 REM "GRILLE"
10 FOR N=0 TO 40 STEP 5
20 FOR M=0 TO 40
30 PLOT N,M
35 PLOT M,M
40 PLOT M,N
45 PLOT M,40-M
50 NEXT M
60 NEXT N
70 REM "COPYRIGHT 1982"

```

instructions PLOT successives commandent le noircissement d'un même élément d'image, celui-ci reste invariablement noir, sauf instruction contraire du type UNPLOT, ou CLS, par exemple.



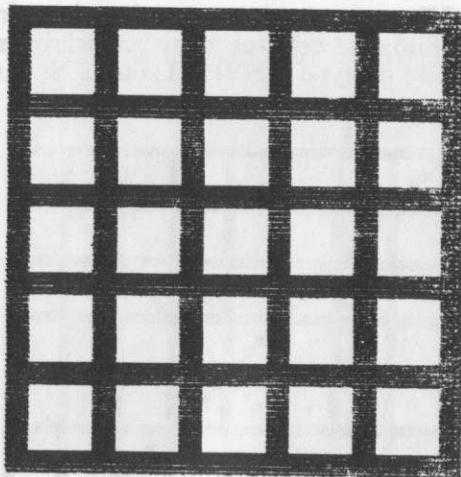
Avec le programme que voici, il est possible d'obtenir un résultat quasi semblable à celui du programme de la page 59, mais sans faire appel à la fonction PLOT. On utilise en effet la fonction PRINT AT, complétée par l'indication CHR\$ N.

```

5 REM "GRILLE 2"
10 FOR N=0 TO 20 STEP 4
20 FOR M=0 TO 20
30 PRINT AT N,M;CHR$ 128
40 PRINT AT M,N;CHR$ 128
50 NEXT M
60 NEXT N
70 REM "COPYRIGHT 1982"

```

Si on insère cette indication dans un ordre PRINT, et que N prenne une des valeurs suivantes : 0 à 10 ou 128 à 138, la machine imprime non plus un caractère, mais l'un des 22 symboles graphiques de son répertoire (voir détails dans le manuel Sinclair et dans le chapitre 11 de cet ouvrage). CHR\$ 128, par exemple, représente un carré entièrement noir. Il est très amusant (et instructif) de faire



varier à loisir, dans ce programme, la valeur N parmi les 22 autorisées. CHR\$ 4, par exemple, ne manque pas d'intérêt, et il pourrait être profitable au lecteur de tenter d'obtenir le même résultat par une modification des boucles FOR-NEXT du programme antérieur. Egalelement, tenter de faire construire un *damier* au ZX-81 n'est pas une mauvaise leçon de modestie...

On remarquera pour finir que, à résultat identique, la solution "PRINT AT... CHR\$" peut s'avérer deux fois plus rapide que la solution "PLOT" puisque les « caractères graphiques » ont des dimensions doubles de celles des « éléments d'image » (ou une surface quadruple). Dans certains cas, il peut être très intéressant de combiner les deux techniques dans un même programme.

2° Une bataille navale arbitrée par ordinateur

Le déroulement classique des parties de bataille navale laisse place à toutes sortes d'indiscrétions visant à découvrir par des moyens inavouables la position des bâtiments de l'adversaire.

Le programme proposé ci-dessous fournit une amorce de solution, puisque les données de position ne sont inscrites que dans la mémoire de la machine, et que chaque joueur dialogue sans relâche avec le ZX-81 jusqu'à l'issue fatale. Cependant, la machine tient

une comptabilité inflexible de toutes les tentatives du joueur, et ne mâche pas ses mots lorsque, le navire adverse ayant été finalement coulé, elle estime que le gaspillage de munitions est un peu trop évident.

```

5 PRINT "BATAILLE NAVALE"
6 LET N=0
8 PRINT "PLACEZ VOTRE NAVIRE"
9 INPUT A$
10 CLS
11 PAUSE 300
12 LET N=N+1
13 CLS
14 PRINT "FEU A VOLONTE"
15 INPUT B$
16 CLS
18 IF B$=A$ THEN PRINT "COULE
EN ",N," COUPS"
19 IF B$=A$ THEN GOTO 70
20 IF A$(1)=B$(1) OR A$(2)=B$(2)
2) THEN GOTO 80
23 PRINT B$;" DANS L""EAU"
24 GOTO 11
70 IF N<5 THEN PRINT "EXCELLEN
T"
71 IF N<8 AND N>=5 THEN PRINT
" PAS MAL"
72 IF N<12 AND N>=8 THEN PRINT
" PAS FAMEUX"
73 IF N>=12 THEN PRINT "LAMENT
ABLE"
74 PAUSE 300
75 CLS
76 GOTO 6
80 PRINT "EN VUE PAR ";B$
82 GOTO 11
90 REM "COPYRIGHT 1982"

```

Ce programme remplissant déjà presque complètement la mémoire du ZX-81 de base (apparition du compte rendu 4 lors des opérations de listage), il a fallu limiter très sérieusement les fonctions demandées à la machine :

- un seul navire peut être placé par le joueur, sur une grille graduée de A à I et de 1 à 9 (exemple C7) ;
- la machine répondra « en vue » lorsque le coup tombera soit dans la ligne, soit dans la colonne où se trouve le navire (un coup double entraîne bien sûr la réponse « coulé » alors qu'un coup nul est signalé par « dans l'eau »).

Imaginons que nous placions notre bâtiment en C7. Nous frappons C7 NEWLINE, et l'écran devient blanc un moment, avant que ne s'inscrive « feu à volonté ». Diverses occasions ont été ainsi prévues, de façon à créer un certain « suspense » favorable à l'ambiance de jeu. Pour obtenir un tel effet, il suffit d'introduire une instruction PAUSE suivie d'un nombre indiquant la durée souhaitée de l'arrêt du programme, en *cinquantièmes de seconde*.

Après l'ordre « feu à volonté », frappons par exemple A1 puis NEWLINE. La machine répond « A1 dans l'eau », et nous fait attendre un peu avant de nous autoriser à tirer à nouveau. Faisons cette fois C2, nous sommes « en vue par C2 ».

Nous pouvons alors couler par C7, auquel cas la machine nous informe qu'un tel résultat au bout de trois coups est excellent, ou bien faire traîner un peu les choses afin de recevoir des félicitations de moins en moins chaudes !

Un renvoi est prévu (GOTO 6 après un CLS à la ligne 75) de façon à permettre à l'adversaire de vous céder sa place pour disposer votre navire.

BATAILLE NAUCALE
PLACEZ VOTRE NAVIRE

B2
FEU A VOLONTE
A1
A1 DANS L"EAU
FEU A VOLONTE
A2
EN VUE PAR A2
FEU A VOLONTE
B1
EN VUE PAR B1
FEU A VOLONTE
B2

COULE EN 4 COUPS
EXCELLENT

Il est bien évident que les groupes de coordonnées alphanumériques doivent être entrés sous forme de chaînes. On appelle ainsi A\$ la position vraie, et B\$ les positions supposées successives. Dès que B\$ = A\$, le navire est déclaré coulé, et on « relève le compteur »

des salves, matérialisé par la variable N, régulièrement incrémentée par l'instruction 12.

Plus délicate est la détection de la situation « en vue ». Elle est effectuée au niveau de l'instruction 20 et fait appel à de nouvelles possibilités du BASIC ZX-81 :

Cette instruction est de la forme IF-THEN à deux niveaux, séparés par OR. Ceci signifie que l'ordre précédé de THEN sera exécuté si *l'une ou l'autre* des conditions introduites par IF est remplie.

Ces conditions sont l'égalité de deux *sous-chaînes* A\$ (1) et B\$ (1) d'une part, A\$ (2) et B\$ (2) d'autre part. Sachez que A\$ (1) est tout simplement le *premier caractère* de A\$ (donc la lettre) et que B\$ (2) est le *second caractère* de B\$ (le chiffre).

On détecte ainsi, en une seule ligne de programme, l'éventuelle égalité soit des deux lettres (ligne), soit des deux chiffres (colonne). Lorsqu'une telle situation « en vue » est détectée, le programme se déroute vers l'instruction 80 qui rend compte de cette situation, tout en rappelant les coordonnées B\$ du tir effectué, afin de faciliter le choix de l'objectif suivant.

Dans le cas où les conditions prévues dans les lignes 18 (et 19 afin de pouvoir exécuter deux tâches à partir d'une même comparaison IF) ou 20 ne sont pas remplies, le programme parvient seul à la ligne 23, qui lui fait imprimer « dans l'eau » avec rappel des coordonnées du tir malheureux. Enfin, la ligne 24 renvoie le programme vers « feu à volonté », après un effacement d'écran et une pause. En cas de coup au but, par contre, les opérations se compliquent : la ligne 18 imprime le compte rendu du fait d'armes, et la ligne 19 envoie le programme vers la ligne 70, début d'une longue suite de comparaisons destinées à commenter le résultat obtenu.

Les choix suivants ont été faits, qui peuvent, bien sûr, être modifiés de façon à ménager les susceptibilités de chacun :

- coulé en moins de 5 coups : « excellent »
- coulé en 5 à 8 coups : « pas mal »
- coulé en 8 à 12 coups : « pas fameux »
- coulé en plus de 12 coups : « lamentable »

Dès que la machine a ainsi donné son avis, elle passe en mode « attente » pour les six secondes que dure la pause de la ligne 74, avant que l'écran ne s'efface, et que le joueur soit invité à placer ailleurs son navire ! Sitôt les coordonnées correspondantes indiquées et NEWLINE actionnée, l'écran est effacé, et l'adversaire peut être appelé : feu à volonté !

3° L'ordinateur et les messages secrets

Le programme que nous donnons ici se situe à mi chemin entre le jeu et l'application « sérieuse » : il permet de coder des messages, puis de les décoder, simplement en les frappant au clavier, ce qui est considérablement plus rapide que l'utilisation de tout autre procédé « manuel ». Une application professionnelle peut être trouvée dans le domaine de la protection des communications télex confidentielles.

```
1 REM "MESSAGE"
10 INPUT B$
12 LET A$=B$
20 IF B$="E" THEN LET A$="S"
22 IF B$="S" THEN LET A$="E"
24 IF B$="A" THEN LET A$="R"
26 IF B$="R" THEN LET A$="A"
28 IF B$="I" THEN LET A$="N"
30 IF B$="N" THEN LET A$="I"
32 IF B$="T" THEN LET A$="O"
34 IF B$="O" THEN LET A$="T"
36 IF B$="U" THEN LET A$="L"
37 IF B$="L" THEN LET A$="C"
38 IF B$="C" THEN LET A$="U"
39 IF B$="U" THEN LET A$="C"
40 PRINT A$;
50 GOTO 10
60 REM "COPYRIGHT 1982"
```

Le programme étant chargé (on aura d'ailleurs intérêt à le stocker sur cassette au moyen d'une commande **SAVE "MESSAGE"**), il faut le lancer en appuyant successivement sur **RUN**, puis **NEWLINE**.

Il suffit alors de frapper les lettres du message à coder (respectivement à décoder), *en les séparant par l'appui sur NEWLINE*, pour voir se composer lettre à lettre le message codé (respectivement décodé). Il est important de ne pas oublier les espaces entre les mots, que l'on entrera exactement comme tout autre caractère, et de ne jamais omettre d'enfoncer **NEWLINE** après chaque caractère ou espace.

Si vous essayez d'entrer "PATRICK GUEULLE", la machine doit vous composer "PROAN KCGLSLUUS", ce qui peut faire office de pseudonyme à consonnance exotique ! Rien ne vous empêche de vous livrer ainsi à quelques expérimentations sur les noms de vos amis et connaissances ! Sachez que, ce programme étant en

PATRICK GUEULLE
PROAN KCGLSLUUS

interrogation permanente, il n'est pas simple de l'arrêter lorsqu'il est en route. Le manuel du ZX-81 reste très discret sur de tels cas, et vous propose de couper l'alimentation, donc d'effacer le programme. Vous pouvez utiliser la procédure, moins radicale, suivante :

RUBOUT (pour effacer le guillemet de gauche)
STOP
NEWLINE

L'ordinateur vous répond D/10, et si à ce moment vous faites :

RUN
NEWLINE, il reprend le programme après avoir effacé l'écran, et toute trace des opérations précédentes.

Si vous voulez changer de programme, vous pouvez utiliser la séquence suivante, lorsque le programme est en action :

RUBOUT
STOP
NEWLINE
NEW
NEWLINE... et tout est effacé.

Par contre, pour charger le programme sur cassette, vous devez faire :

RUBOUT
STOP
NEWLINE
SAVE « MESSAGE »
Puis, après avoir démarré l'enregistreur : NEWLINE.

Etudions maintenant plus en détail le programme lui-même : l'instruction 10 vient chercher sur le clavier la lettre que vous voulez entrer. Cette lettre constitue ce que l'on appelle une « chaîne » (par opposition à une variable numérique), et on la

baptise B\$. La suite du programme ne devrait pas nous faire perdre B\$, on « duplique » B\$ en A\$ par l'instruction 12.

Le codage (ou le décodage) proprement dit s'opère de l'instruction 20 à l'instruction 39 : si la lettre entrée est l'une des dix plus fréquentes de la langue française (E, S, A, R, I, N, T, U, L, O, par ordre de fréquence décroissante), ou s'il s'agit d'un espace, on procède aux échanges suivants :

Si B\$ est :	alors, on transforme A\$ en
E	S
S	E
A	R
R	A
I	N
N	I
T	O
U	L
L	U
O	T
espace	C
C	espace

Ce tableau n'est autre que la traduction en langage usuel des douze instructions "IF-THEN".

En arrivant à l'instruction 40, de deux choses l'une : ou bien la lettre entrée a été transformée au cours des douze instructions précédentes, ou elle ne l'a pas été. Dans les deux cas, elle est imprimée à la suite de la précédente, grâce au point-virgule qui termine l'instruction PRINT A\$. L'instruction 50 ramène le programme au début afin qu'il puisse traiter le caractère suivant.

Bien qu'il ne code pas toutes les lettres de l'alphabet (pour « tromper l'ennemi » et économiser la place mémoire), ce programme exécute un codage déjà sérieux, qui fera la joie des enfants, et... des moins jeunes aussi.

4^o Perçons les mystères du courrier

Le principe du code postal est désormais bien connu de chacun. Ce qui l'est moins est le détail des opérations de tri automatisé rendues possibles par ces cinq chiffres. On remarque bien souvent

au bas des enveloppes, une série de petits bâtonnets fluorescents. Ceux-ci ne sont que la traduction, dans un code accessible aux machines de tri, des cinq chiffres, précédemment reconnus par une première machine, ou par un agent disposant d'un clavier.

Le programme suivant permet de reconstituer ces « marques d'indexation », telles qu'on les lirait en retournant l'enveloppe sens dessus dessous, c'est-à-dire avec le timbre en bas à gauche. Le numéro de code est entré sous la forme d'une chaîne nommée Z\$, qui est imprimée à des fins de vérification.

```

5 REM "INDEXATION"
10 PRINT "DONNER UN NUMERO DE
CODE POSTAL"
11 INPUT Z$
12 PRINT
13 PRINT
31 PRINT Z$
32 PRINT
33 IF LEN Z$<>5 THEN GOTO 10
34 PRINT "LES MARQUES D""INDEX
ATION SONT :"
35 PRINT
36 PRINT
70 FOR I=1 TO 5
80 LET Y$=Z$(I)
85 LET A=VAL Y$
90 IF A=0 THEN PRINT "IIII "
91 IF A=1 THEN PRINT "III I "
92 IF A=2 THEN PRINT "II II "
93 IF A=3 THEN PRINT "I III "
94 IF A=4 THEN PRINT "III I "
95 IF A=5 THEN PRINT "II I I "
96 IF A=6 THEN PRINT "I II I "
97 IF A=7 THEN PRINT "II I II "
98 IF A=8 THEN PRINT "I I II "
99 IF A=9 THEN PRINT "I III "
500 NEXT I
505 PRINT "
510 GOTO 10
600 REM "COPYRIGHT 1982"

```

La ligne 33 teste la longueur de cette chaîne (LEN Z\$ représente le nombre de caractères de Z\$) et la refuse si elle ne comporte pas exactement cinq chiffres. Essayez 75, puis un véritable code (par exemple 75019).

Lorsque le code est accepté, la machine imprime un texte explicatif, puis se lance dans une boucle FOR-NEXT à cinq étapes :

Chaque caractère (Z\$(I) pour I variant de 1 à 5) est provisoirement nommé Y\$, et sa valeur numérique est affectée à la variable A – notons que les lignes 80 et 85 pourraient fusionner en :

80 LET A = VAL Z\$(I).

Suit une longue liste d'instructions conditionnelles IF-THEN, chargées de la conversion de code, comme dans notre programme précédent. Chaque chiffre de 0 à 9 est remplacé par un groupe de six caractères dont deux espaces et quatre I (c'est le I qui ressemble le plus aux bâtonnets d'indexation).

Lorsque l'on frappe ce programme, on veillera bien à n'oublier aucun espace, même lorsqu'il y en a deux à la suite. Inutile de préciser que l'auteur de ces lignes a passé des moments fort réjouissants au cours de ses recherches visant à percer la clé de ce code !

Sitôt sorti de cette boucle, l'ordinateur imprime un large blanc destiné à annuler l'effet du point-virgule placé après le dernier groupe de bâtonnets, afin que la ligne imprimée lors du retour à l'instruction 10 ne se trouve pas coupée en deux.

DONNER UN NUMERO DE CODE POSTAL

75940

LES MARQUES D'INDEXATION SONT :

II IIII I II IIIIII IIIII

DONNER UN NUMERO DE CODE POSTAL

Z

DONNER UN NUMERO DE CODE POSTAL

Ce programme permet à son utilisateur de faire montre de singuliers talents divinatoires, mais peut également servir à découvrir la cause de certains retards apparemment inexplicables dans l'acheminement du courrier : il suffit qu'un seul bâtonnet n'occupe pas la place qui lui revient pour que la lettre qui le porte soit invariablement aiguillée dans la mauvaise direction dès qu'elle entre dans une machine automatique. Pour une telle lettre ainsi marquée par le destin, en dehors du bon vieux tri manuel, point de salut !

Le ZX-81 traceur de courbes

Nous avons vu au chapitre des jeux comment l'ordinateur peut tracer toutes sortes de graphismes distrayants ou utilitaires.

Ici, nous exploiterons les larges possibilités graphiques du ZX-81 pour lui faire tracer en un temps record des diagrammes ou des courbes qui exigerait parfois pas loin d'une heure pour leur construction manuelle.

1^o Histogrammes et statistiques

C'est tous les jours que l'on voit, dans la presse ou ailleurs, l'évolution d'une donnée économique ou autre, présentée sous la forme d'un « diagramme en marches d'escalier », ou *histogramme*. Dans « histogramme », il y a « histoire » et, en effet, une telle représentation graphique fait apparaître très clairement l'évolution historique de la donnée statistique à laquelle on s'intéresse : popularité d'un homme politique, nombre de voitures vendues par Renault, nombre d'accidents de la route causés par des alcooliques, et, pourquoi pas, la progression des notes de mathématiques du petit dernier !

L'une des premières précautions prévues par ce programme est de faire « déclarer » à l'utilisateur la valeur maximale que pourra atteindre la grandeur dont on se propose de retracer graphiquement l'évolution. On évitera ainsi toute tentative d'inscription d'un point hors de l'écran, et on profitera au mieux de la surface disponible. La précision de la représentation pourra atteindre 1/60, c'est-à-dire mieux que 1,7 %, ce qui est très satisfaisant.

```

5 REM "HISTOGRAMME"
10 LET L=40
15 PRINT "HISTOGRAMME"
16 PRINT
20 PRINT "VALEUR MAX ?"
22 INPUT M
23 CLS
25 PRINT "VALEURS ?"
30 INPUT A
35 LET A=60*A/M
40 LET L=L-2
45 FOR N=1 TO A
50 PLOT N,L
55 NEXT N
60 GOTO 30
70 REM "COPYRIGHT 1982"

```

On aurait certes pu faire déterminer ce maximum à la machine, et nous avons bien sûr effectué diverses tentatives dans ce sens. (Il suffit, notamment, de mémoriser dans une variable, que l'on pourrait appeler M, la dernière valeur entrée si elle est supérieure à celle précédemment contenue dans M).

Cependant, ceci exige que toutes les valeurs soient stockées en mémoire jusqu'à l'introduction de la dernière, avant que le programme ne puisse procéder à leur *mise à l'échelle*, conformément à la ligne 35.

Ceci exige une capacité mémoire importante, et ne peut raisonnablement s'envisager autrement qu'au moyen d'un bloc d'extension mémoire (16 Ko Sinclair).

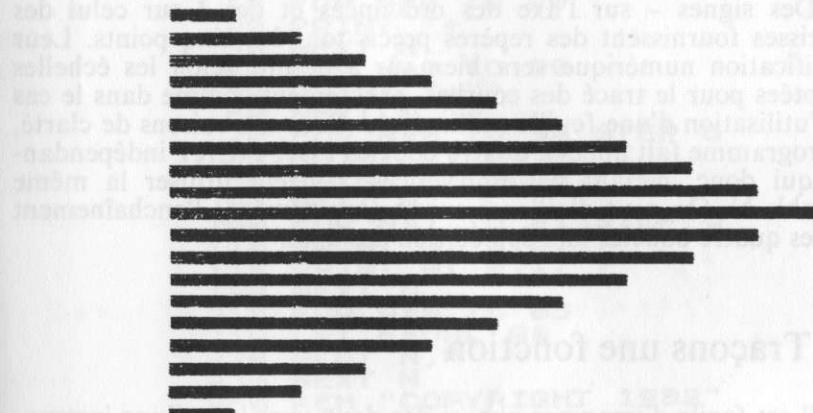
Notre programme, qui dispose, dès le début, de la valeur de M, procède à la mise à l'échelle des valeurs au fur et à mesure de leur acquisition, en les multipliant par 60/M. Ainsi, la valeur maximale couvrira 60 « carreaux » sur l'écran, soit pratiquement le maximum autorisé. Avantage supplémentaire du procédé, chaque barre de l'histogramme est tracée immédiatement, alors que, dans le précédent système, il aurait fallu attendre la fin des entrées.

On remarquera la ligne 40 qui permet, par incrémentation de deux unités de L après chaque acquisition, de ménager une ligne blanche entre chaque barre de l'histogramme, en vue d'un gain en lisibilité. Ceci permet encore de tracer 20 barres sur un seul écran, ce qui suffit très largement pour symboliser, par exemple, les douze mois de l'année.

On notera que l'« amplitude de variation » de la boucle FOR-NEXT, occupant les lignes 45 à 55, est fonction de la valeur de la grandeur « mise à l'échelle » A. C'est précisément ce qui permet d'interrompre le tracé de la barre dès que la longueur correspondant à A est atteinte.

La présentation « couchée » de l'histogramme est un peu inhabituelle, mais simplifie nettement l'écriture du programme. Rien n'empêche d'ailleurs de tourner la tête de 90° !

VALEURS ?



2^o Etalonnons notre écran

Dans tout ce qui va suivre, la précision des tracés exécutés par le ZX-81 sera telle qu'il apparaît souhaitable de disposer d'axes gradués sur l'écran, capables de fournir tous les points de repère nécessaires lors de l'exploitation des graphiques.

```

5 REM "AXES"
10 FOR N=0 TO 20
20 PRINT "+"
30 NEXT N
40 FOR N=0 TO 30
50 PRINT "+";
60 NEXT N
70 FOR N=0 TO 30 STEP 5
80 PRINT AT 21,N;"|"
90 NEXT N
100 FOR N=1 TO 21 STEP 5
110 PRINT AT N,0;"-"
120 NEXT N
130 REM "COPYRIGHT 1982"

```

Ce programme, assez court, pourra être incorporé dans tout autre programme visant à tracer une courbe, mais peut être, bien évidemment, lancé seul.

Chaque axe est matérialisé par une rangée de signes +, bien adaptés à cette fonction de repérage précis d'un point.

Des signes – sur l'axe des ordonnées et des I sur celui des abscisses fournissent des repères précis tous les cinq points. Leur signification numérique sera bien sûr à définir selon les échelles adoptées pour le tracé des courbes, exactement comme dans le cas de l'utilisation d'une feuille millimétrée. Pour des raisons de clarté, le programme fait appel à quatre boucles FOR-NEXT indépendantes qui donc, n'étant pas imbriquées, peuvent utiliser la même variable N. On peut d'ailleurs suivre (rapidement) l'enchaînement de ces quatre boucles lors de l'édition des axes.

3° Traçons une fonction

Il est facile d'imaginer que, si des deux variables d'une instruction PLOT nous en faisons varier une de façon continue, et que nous imposons à l'autre une loi de variation dépendant de la première, nous obtiendrons le tracé de la courbe représentative de la fonction liant ces deux variables l'une à l'autre.

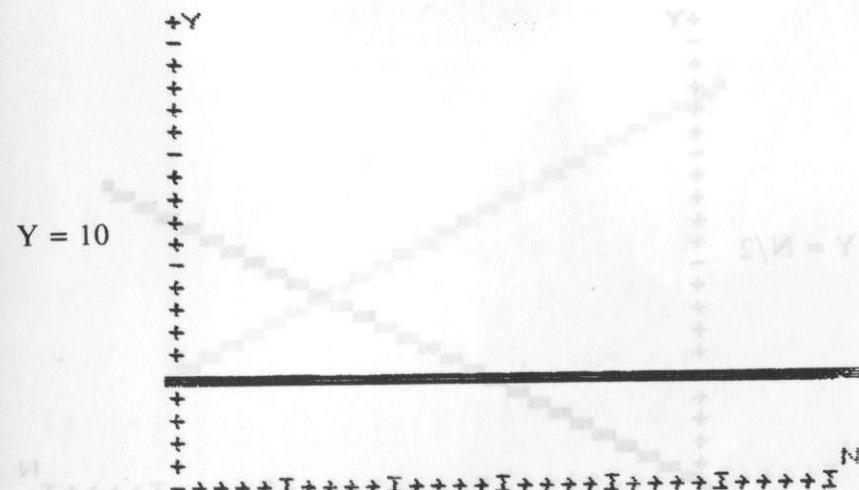
L'intérêt d'une telle possibilité dans le cadre de l'étude mathématique des fonctions, tellement abstraite, ne laissera pas indifférents nos lecteurs postulants bacheliers !

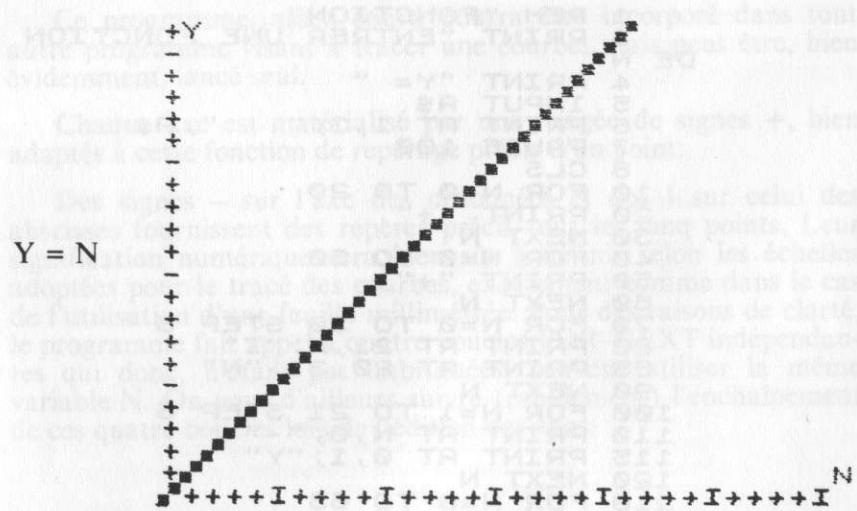
Le ZX-81 affiche cependant certaines limites quant à ce genre de travail, à cause de la place disponible sur l'écran. Il faut très bien choisir sa fonction, ou la « trafiquer » au moyen de deux coefficients judicieusement placés, afin, d'une part, de parvenir à une représentation digne d'intérêt pour une excursion de la variable entre 0 et 60, et, d'autre part, que la valeur de la fonction dans cet intervalle reste comprise entre 0 et 40.

Tout dépassement de ces limites entraînerait l'arrêt immédiat du programme.

On peut facilement voir ce programme à l'œuvre en lui proposant, en respectant les symbolisations du BASIC, les fonctions suivantes : (N remplaçant l'habituel X).

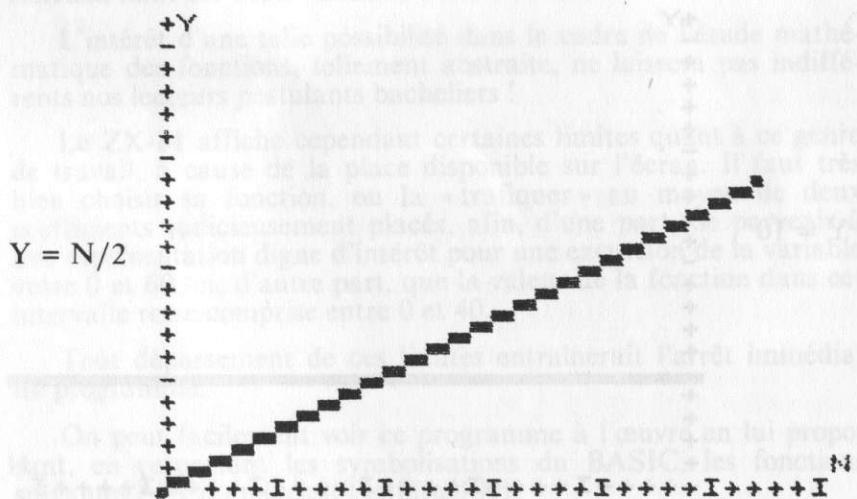
```
1 REM "FONCTION"
3 PRINT "ENTRER UNE FONCTION
DE N"
4 PRINT "Y= "
5 INPUT A$
6 PRINT AT 1,3;" ";A$
7 PAUSE 100
8 CLS
10 FOR N=0 TO 20
20 PRINT "+"
30 NEXT N
40 FOR N=0 TO 30
50 PRINT "+"
60 NEXT N
70 FOR N=0 TO 30 STEP 5
80 PRINT AT 21,N;"I"
85 PRINT AT 20,31;"N"
90 NEXT N
100 FOR N=1 TO 21 STEP 5
110 PRINT AT N,0;"-"
115 PRINT AT 0,1;"Y"
120 NEXT N
210 FOR N=0 TO 60
219 LET F=VAL A$
220 PLOT N,F
230 NEXT N
250 REM "COPYRIGHT 1982"
```



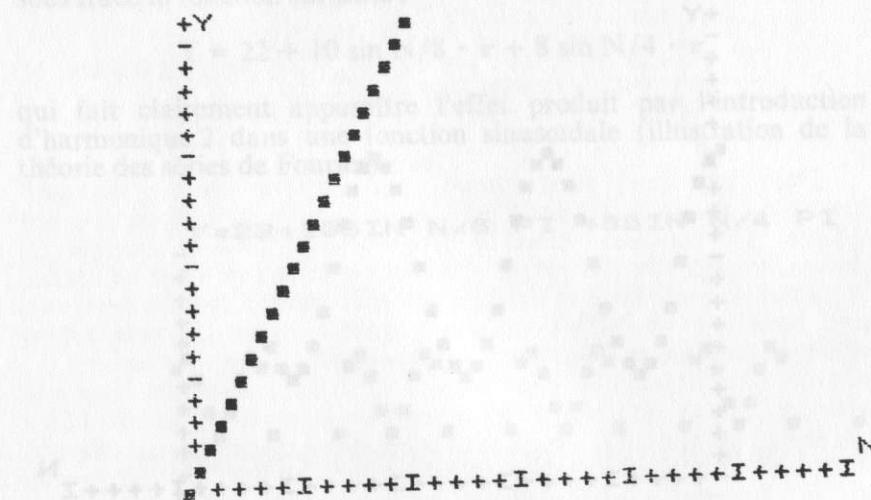


3° Traçons une fonction de N

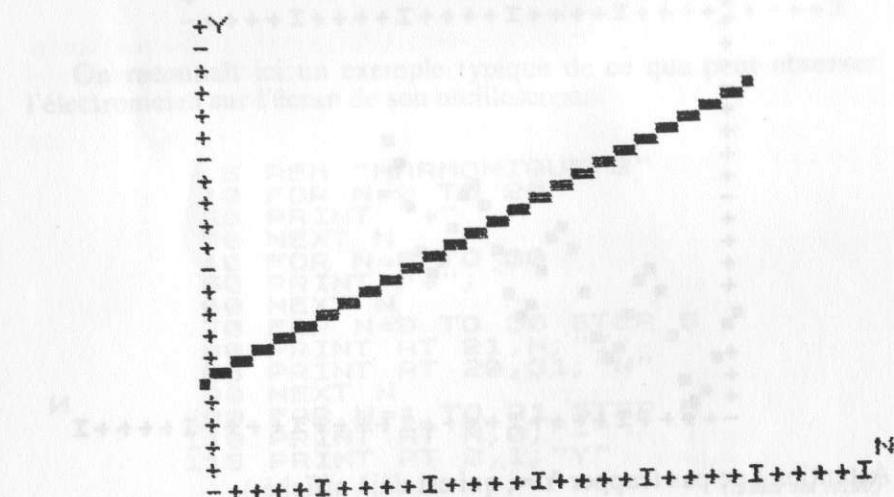
Il est facile d'imaginer que pour une fonction FLOT nous faisons varier une variable continue et que nous imposons à l'autre une loi de variation. Si l'on fait de la première, nous obtiendrons le tracé de la fonction représentative de la fonction dans les deux variables.



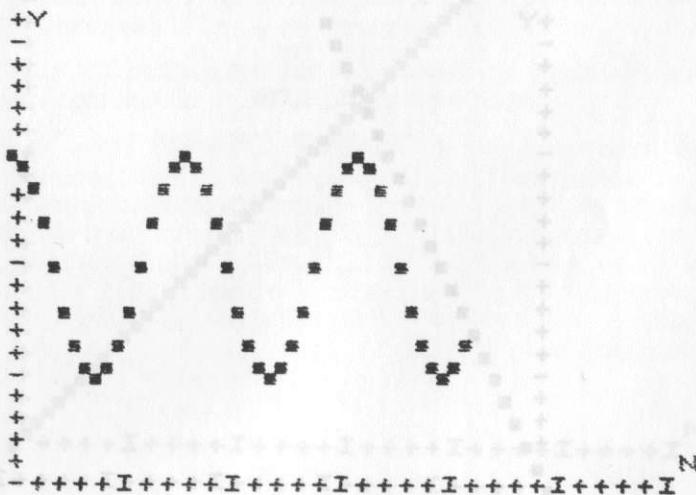
$$Y = 2N \quad (2 \cdot N)$$



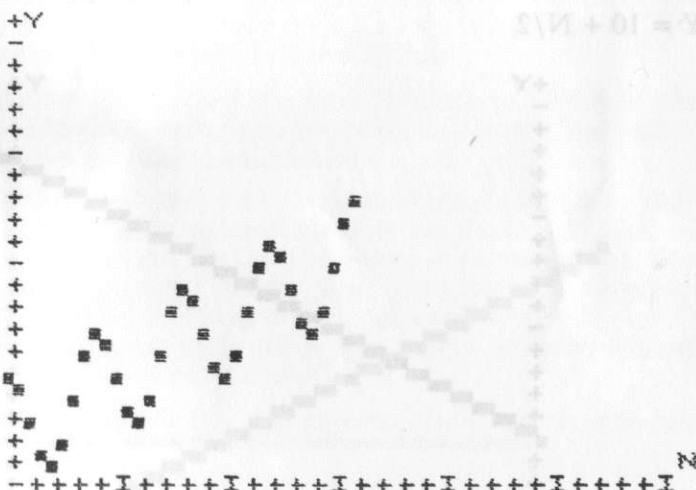
$$Y = 10 + N/2$$



$$Y = 20 + 10 \times \cos(N/8 \times \pi)$$



$$Y = 5 + (N/2) + 5 \times \cos(N/4 \times \pi)$$

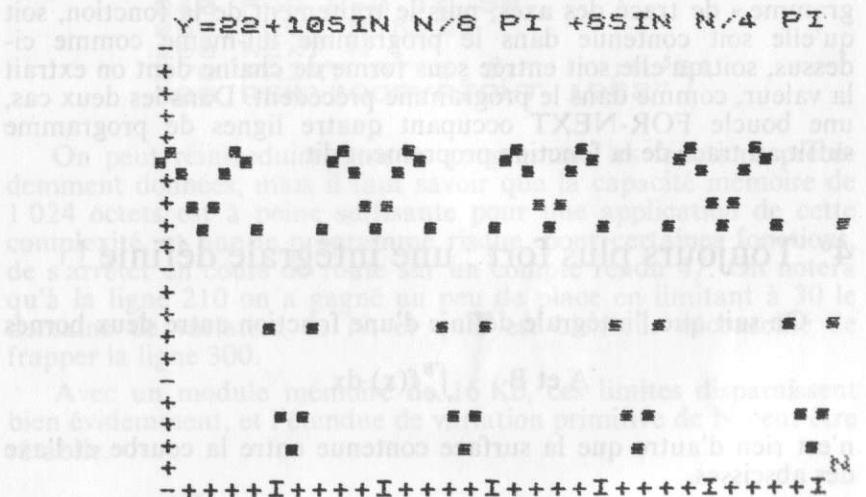


(Attention ! ne pas frapper Y=, qui est déjà affiché.)

A des fins de démonstrations, d'ailleurs, le programme ci-dessous trace la fonction suivante :

$$Y = 22 + 10 \sin N/8 \cdot \pi + 8 \sin N/4 \cdot \pi$$

qui fait clairement apparaître l'effet produit par l'introduction d'harmonique 2 dans une fonction sinusoïdale (illustration de la théorie des séries de Fourier).



On reconnaît ici un exemple typique de ce que peut observer l'électronicien sur l'écran de son oscilloscope.

```

5 REM "HARMONIQUE 2"
10 FOR N=0 TO 20
20 PRINT "+"
30 NEXT N
40 FOR N=0 TO 30
50 PRINT "+";
60 NEXT N
70 FOR N=0 TO 30 STEP 5
80 PRINT AT 21,N;"I"
85 PRINT AT 20,31;"N"
90 NEXT N
100 FOR N=1 TO 21 STEP 5
110 PRINT AT N,0;"-"
115 PRINT AT 0,1;"Y"

```

(Suite au verso)

```

120 NEXT N
130 FOR N=0 TO 63
140 PLOT N,22+10*SIN (N/8*PI)+8
*10*SIN (N/4*PI)
150 NEXT N
160 PRINT AT 0,0;" Y=22+10SIN N
/8 PI +8SIN N/4 PI"
170 REM "COPYRIGHT 1982"

```

Dans ces deux programmes, on retrouve notre « sous-programme » de tracé des axes, puis le traitement de la fonction, soit qu'elle soit contenue dans le programme lui-même comme ci-dessus, soit qu'elle soit entrée sous forme de chaîne dont on extrait la valeur, comme dans le programme précédent. Dans les deux cas, une boucle FOR-NEXT occupant quatre lignes de programme suffit au tracé de la fonction proprement dit.

4° Toujours plus fort : une intégrale définie !

On sait que l'intégrale définie d'une fonction entre deux bornes

$$A \text{ et } B : \int_A^B f(x) dx$$

n'est rien d'autre que la surface contenue entre la courbe et l'axe des abscisses.

Lorsque l'évaluation d'intégrales définies par la voie mathématique s'avère trop pénible, on trace souvent la courbe sur une feuille quadrillée ou millimétrée, et... on compte les carreaux !

Notre ZX-81 sait très bien compter les « carreaux » de l'écran TV, moyennant quelques instructions adéquates.

```

4 PRINT "FONCTION ?"
5 INPUT A$
6 CLS
10 FOR N=0 TO 20
20 PRINT "+"
30 NEXT N
40 FOR N=0 TO 30
50 PRINT "+";
60 NEXT N
70 FOR N=0 TO 30 STEP 5
80 PRINT AT 21,N;"I"
85 PRINT AT 20,31;"N"

```

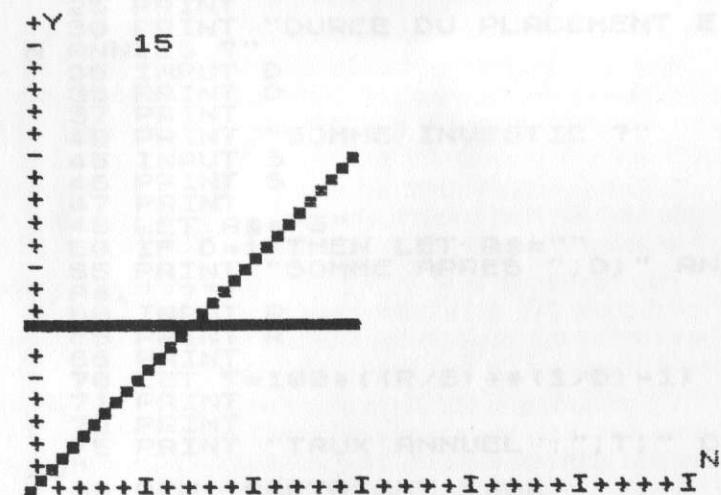
```

90 NEXT N
100 FOR N=1 TO 21 STEP 5
110 PRINT AT N,0;"-"
115 PRINT AT 0,1;"Y"
120 NEXT N
130 LET K=0
140 FOR N=0 TO 30
150 LET F=VAL A$
160 PLOT N,F
170 LET K=K+F
180 NEXT N
190 FOR N=0 TO 30
200 PLOT N,K/31
210 NEXT N
220 PRINT AT 1,3;" ";K/31
300 REM "COPYRIGHT 1982"

```

On peut réintroduire dans ce programme les fonctions précédemment données, mais il faut savoir que la capacité mémoire de 1 024 octets est à peine suffisante pour une application de cette complexité, et que le programme risque, pour certaines fonctions, de s'arrêter en cours de route sur un compte rendu 4/. On notera qu'à la ligne 210 on a gagné un peu de place en limitant à 30 le domaine de variation de N, et qu'il est bien sûr déconseillé de frapper la ligne 300.

Avec un module mémoire de 16 Ko, ces limites disparaissent bien évidemment, et l'étendue de variation primitive de N peut être rétablie.



Lorsque le programme va jusqu'au bout, une horizontale se trace sur l'écran, matérialisant donc la *valeur moyenne* de la portion de fonction qui a été tracée. Puis, en haut de l'écran, s'imprime l'ordonnée de cette droite, qui n'est autre que la valeur de l'intégrale définie entre 0 et 30 de cette fonction.

En électronique, ce programme peut rendre de grands services pour tout ce qui touche aux calculs d'alimentations. C'est au niveau de la ligne 225 que s'effectue, colonne par colonne, et tout au long du tracé de la fonction, le comptage des « carreaux ». Dès le tracé terminé, une division par 31 (pour tenir compte de la colonne origine) calcule l'intégrale définie $K/31$, qui peut dès lors être tracée (lignes 240 à 260) et imprimée (ligne 270).

Les gros ordinateurs sont devenus absolument indispensables aux professionnels de la finance, mais ceci n'empêche pas le ZX-81 de faire bonne figure au niveau des calculs financiers individuels.

Lorsque l'évaluation d'intégrales définies par la théorie mathématique s'avère trop pénible, on trace souvent la courbe sur une feuille quadrillée au millimètre, etc. On compte les « carreaux ».

Notre ZX-81 sait très bien compter les « carreaux » de l'écran TV, moyennant quelques instructions adéquates.

```
4 PRINT "FONCTION ?"
5 INPUT R$
```



```
6 CLS
7 FOR N=0 TO 20
8 PRINT N
9 FOR N=0 TO 30
10 PRINT N
11 NEXT N
12 FOR N=0 TO 100 STEP 10
13 PRINT N
14 NEXT N
15 FOR N=0 TO 100 STEP 10
16 PRINT N
17 NEXT N
18 FOR N=0 TO 100 STEP 10
19 PRINT N
20 NEXT N
21 FOR N=0 TO 100 STEP 10
22 PRINT N
23 NEXT N
24 FOR N=0 TO 100 STEP 10
25 PRINT N
26 NEXT N
27 FOR N=0 TO 100 STEP 10
28 PRINT N
29 NEXT N
30 FOR N=0 TO 100 STEP 10
31 PRINT N
32 NEXT N
33 FOR N=0 TO 100 STEP 10
34 PRINT N
35 NEXT N
36 FOR N=0 TO 100 STEP 10
37 PRINT N
38 NEXT N
39 FOR N=0 TO 100 STEP 10
40 PRINT N
41 NEXT N
42 FOR N=0 TO 100 STEP 10
43 PRINT N
44 NEXT N
45 FOR N=0 TO 100 STEP 10
46 PRINT N
47 NEXT N
48 FOR N=0 TO 100 STEP 10
49 PRINT N
50 NEXT N
51 FOR N=0 TO 100 STEP 10
52 PRINT N
53 NEXT N
54 FOR N=0 TO 100 STEP 10
55 PRINT N
56 NEXT N
57 FOR N=0 TO 100 STEP 10
58 PRINT N
59 NEXT N
60 FOR N=0 TO 100 STEP 10
61 PRINT N
62 NEXT N
63 FOR N=0 TO 100 STEP 10
64 PRINT N
65 NEXT N
66 FOR N=0 TO 100 STEP 10
67 PRINT N
68 NEXT N
69 FOR N=0 TO 100 STEP 10
70 PRINT N
71 NEXT N
72 FOR N=0 TO 100 STEP 10
73 PRINT N
74 NEXT N
75 FOR N=0 TO 100 STEP 10
76 PRINT N
77 NEXT N
78 FOR N=0 TO 100 STEP 10
79 PRINT N
80 NEXT N
81 FOR N=0 TO 100 STEP 10
82 PRINT N
83 NEXT N
84 FOR N=0 TO 100 STEP 10
85 PRINT N
86 NEXT N
87 FOR N=0 TO 100 STEP 10
88 PRINT N
89 NEXT N
90 FOR N=0 TO 100 STEP 10
91 PRINT N
92 NEXT N
93 FOR N=0 TO 100 STEP 10
94 PRINT N
95 NEXT N
96 FOR N=0 TO 100 STEP 10
97 PRINT N
98 NEXT N
99 FOR N=0 TO 100 STEP 10
100 PRINT N
101 NEXT N
102 FOR N=0 TO 100 STEP 10
103 PRINT N
104 NEXT N
105 FOR N=0 TO 100 STEP 10
106 PRINT N
107 NEXT N
108 FOR N=0 TO 100 STEP 10
109 PRINT N
110 NEXT N
111 FOR N=0 TO 100 STEP 10
112 PRINT N
113 NEXT N
114 FOR N=0 TO 100 STEP 10
115 PRINT N
116 NEXT N
117 FOR N=0 TO 100 STEP 10
118 PRINT N
119 NEXT N
120 FOR N=0 TO 100 STEP 10
121 PRINT N
122 NEXT N
123 FOR N=0 TO 100 STEP 10
124 PRINT N
125 NEXT N
126 FOR N=0 TO 100 STEP 10
127 PRINT N
128 NEXT N
129 FOR N=0 TO 100 STEP 10
130 PRINT N
131 NEXT N
132 FOR N=0 TO 100 STEP 10
133 PRINT N
134 NEXT N
135 FOR N=0 TO 100 STEP 10
136 PRINT N
137 NEXT N
138 FOR N=0 TO 100 STEP 10
139 PRINT N
140 NEXT N
141 FOR N=0 TO 100 STEP 10
142 PRINT N
143 NEXT N
144 FOR N=0 TO 100 STEP 10
145 PRINT N
146 NEXT N
147 FOR N=0 TO 100 STEP 10
148 PRINT N
149 NEXT N
150 FOR N=0 TO 100 STEP 10
151 PRINT N
152 NEXT N
153 FOR N=0 TO 100 STEP 10
154 PRINT N
155 NEXT N
156 FOR N=0 TO 100 STEP 10
157 PRINT N
158 NEXT N
159 FOR N=0 TO 100 STEP 10
160 PRINT N
161 NEXT N
162 FOR N=0 TO 100 STEP 10
163 PRINT N
164 NEXT N
165 FOR N=0 TO 100 STEP 10
166 PRINT N
167 NEXT N
168 FOR N=0 TO 100 STEP 10
169 PRINT N
170 NEXT N
171 FOR N=0 TO 100 STEP 10
172 PRINT N
173 NEXT N
174 FOR N=0 TO 100 STEP 10
175 PRINT N
176 NEXT N
177 FOR N=0 TO 100 STEP 10
178 PRINT N
179 NEXT N
180 FOR N=0 TO 100 STEP 10
181 PRINT N
182 NEXT N
183 FOR N=0 TO 100 STEP 10
184 PRINT N
185 NEXT N
186 FOR N=0 TO 100 STEP 10
187 PRINT N
188 NEXT N
189 FOR N=0 TO 100 STEP 10
190 PRINT N
191 NEXT N
192 FOR N=0 TO 100 STEP 10
193 PRINT N
194 NEXT N
195 FOR N=0 TO 100 STEP 10
196 PRINT N
197 NEXT N
198 FOR N=0 TO 100 STEP 10
199 PRINT N
200 NEXT N
201 FOR N=0 TO 100 STEP 10
202 PRINT N
203 NEXT N
204 FOR N=0 TO 100 STEP 10
205 PRINT N
206 NEXT N
207 FOR N=0 TO 100 STEP 10
208 PRINT N
209 NEXT N
210 FOR N=0 TO 100 STEP 10
211 PRINT N
212 NEXT N
213 FOR N=0 TO 100 STEP 10
214 PRINT N
215 NEXT N
216 FOR N=0 TO 100 STEP 10
217 PRINT N
218 NEXT N
219 FOR N=0 TO 100 STEP 10
220 PRINT N
221 NEXT N
222 FOR N=0 TO 100 STEP 10
223 PRINT N
224 NEXT N
225 FOR N=0 TO 100 STEP 10
226 PRINT N
227 NEXT N
228 FOR N=0 TO 100 STEP 10
229 PRINT N
230 NEXT N
231 FOR N=0 TO 100 STEP 10
232 PRINT N
233 NEXT N
234 FOR N=0 TO 100 STEP 10
235 PRINT N
236 NEXT N
237 FOR N=0 TO 100 STEP 10
238 PRINT N
239 NEXT N
240 FOR N=0 TO 100 STEP 10
241 PRINT N
242 NEXT N
243 FOR N=0 TO 100 STEP 10
244 PRINT N
245 NEXT N
246 FOR N=0 TO 100 STEP 10
247 PRINT N
248 NEXT N
249 FOR N=0 TO 100 STEP 10
250 PRINT N
251 NEXT N
252 FOR N=0 TO 100 STEP 10
253 PRINT N
254 NEXT N
255 FOR N=0 TO 100 STEP 10
256 PRINT N
257 NEXT N
258 FOR N=0 TO 100 STEP 10
259 PRINT N
260 NEXT N
261 FOR N=0 TO 100 STEP 10
262 PRINT N
263 NEXT N
264 FOR N=0 TO 100 STEP 10
265 PRINT N
266 NEXT N
267 FOR N=0 TO 100 STEP 10
268 PRINT N
269 NEXT N
270 FOR N=0 TO 100 STEP 10
271 PRINT N
272 NEXT N
273 FOR N=0 TO 100 STEP 10
274 PRINT N
275 NEXT N
276 FOR N=0 TO 100 STEP 10
277 PRINT N
278 NEXT N
279 FOR N=0 TO 100 STEP 10
280 PRINT N
281 NEXT N
282 FOR N=0 TO 100 STEP 10
283 PRINT N
284 NEXT N
285 FOR N=0 TO 100 STEP 10
286 PRINT N
287 NEXT N
288 FOR N=0 TO 100 STEP 10
289 PRINT N
290 NEXT N
291 FOR N=0 TO 100 STEP 10
292 PRINT N
293 NEXT N
294 FOR N=0 TO 100 STEP 10
295 PRINT N
296 NEXT N
297 FOR N=0 TO 100 STEP 10
298 PRINT N
299 NEXT N
300 FOR N=0 TO 100 STEP 10
301 PRINT N
302 NEXT N
303 FOR N=0 TO 100 STEP 10
304 PRINT N
305 NEXT N
306 FOR N=0 TO 100 STEP 10
307 PRINT N
308 NEXT N
309 FOR N=0 TO 100 STEP 10
310 PRINT N
311 NEXT N
312 FOR N=0 TO 100 STEP 10
313 PRINT N
314 NEXT N
315 FOR N=0 TO 100 STEP 10
316 PRINT N
317 NEXT N
318 FOR N=0 TO 100 STEP 10
319 PRINT N
320 NEXT N
321 FOR N=0 TO 100 STEP 10
322 PRINT N
323 NEXT N
324 FOR N=0 TO 100 STEP 10
325 PRINT N
326 NEXT N
327 FOR N=0 TO 100 STEP 10
328 PRINT N
329 NEXT N
330 FOR N=0 TO 100 STEP 10
331 PRINT N
332 NEXT N
333 FOR N=0 TO 100 STEP 10
334 PRINT N
335 NEXT N
336 FOR N=0 TO 100 STEP 10
337 PRINT N
338 NEXT N
339 FOR N=0 TO 100 STEP 10
340 PRINT N
341 NEXT N
342 FOR N=0 TO 100 STEP 10
343 PRINT N
344 NEXT N
345 FOR N=0 TO 100 STEP 10
346 PRINT N
347 NEXT N
348 FOR N=0 TO 100 STEP 10
349 PRINT N
350 NEXT N
351 FOR N=0 TO 100 STEP 10
352 PRINT N
353 NEXT N
354 FOR N=0 TO 100 STEP 10
355 PRINT N
356 NEXT N
357 FOR N=0 TO 100 STEP 10
358 PRINT N
359 NEXT N
360 FOR N=0 TO 100 STEP 10
361 PRINT N
362 NEXT N
363 FOR N=0 TO 100 STEP 10
364 PRINT N
365 NEXT N
366 FOR N=0 TO 100 STEP 10
367 PRINT N
368 NEXT N
369 FOR N=0 TO 100 STEP 10
370 PRINT N
371 NEXT N
372 FOR N=0 TO 100 STEP 10
373 PRINT N
374 NEXT N
375 FOR N=0 TO 100 STEP 10
376 PRINT N
377 NEXT N
378 FOR N=0 TO 100 STEP 10
379 PRINT N
380 NEXT N
381 FOR N=0 TO 100 STEP 10
382 PRINT N
383 NEXT N
384 FOR N=0 TO 100 STEP 10
385 PRINT N
386 NEXT N
387 FOR N=0 TO 100 STEP 10
388 PRINT N
389 NEXT N
390 FOR N=0 TO 100 STEP 10
391 PRINT N
392 NEXT N
393 FOR N=0 TO 100 STEP 10
394 PRINT N
395 NEXT N
396 FOR N=0 TO 100 STEP 10
397 PRINT N
398 NEXT N
399 FOR N=0 TO 100 STEP 10
400 PRINT N
401 NEXT N
402 FOR N=0 TO 100 STEP 10
403 PRINT N
404 NEXT N
405 FOR N=0 TO 100 STEP 10
406 PRINT N
407 NEXT N
408 FOR N=0 TO 100 STEP 10
409 PRINT N
410 NEXT N
411 FOR N=0 TO 100 STEP 10
412 PRINT N
413 NEXT N
414 FOR N=0 TO 100 STEP 10
415 PRINT N
416 NEXT N
417 FOR N=0 TO 100 STEP 10
418 PRINT N
419 NEXT N
420 FOR N=0 TO 100 STEP 10
421 PRINT N
422 NEXT N
423 FOR N=0 TO 100 STEP 10
424 PRINT N
425 NEXT N
426 FOR N=0 TO 100 STEP 10
427 PRINT N
428 NEXT N
429 FOR N=0 TO 100 STEP 10
430 PRINT N
431 NEXT N
432 FOR N=0 TO 100 STEP 10
433 PRINT N
434 NEXT N
435 FOR N=0 TO 100 STEP 10
436 PRINT N
437 NEXT N
438 FOR N=0 TO 100 STEP 10
439 PRINT N
440 NEXT N
441 FOR N=0 TO 100 STEP 10
442 PRINT N
443 NEXT N
444 FOR N=0 TO 100 STEP 10
445 PRINT N
446 NEXT N
447 FOR N=0 TO 100 STEP 10
448 PRINT N
449 NEXT N
450 FOR N=0 TO 100 STEP 10
451 PRINT N
452 NEXT N
453 FOR N=0 TO 100 STEP 10
454 PRINT N
455 NEXT N
456 FOR N=0 TO 100 STEP 10
457 PRINT N
458 NEXT N
459 FOR N=0 TO 100 STEP 10
460 PRINT N
461 NEXT N
462 FOR N=0 TO 100 STEP 10
463 PRINT N
464 NEXT N
465 FOR N=0 TO 100 STEP 10
466 PRINT N
467 NEXT N
468 FOR N=0 TO 100 STEP 10
469 PRINT N
470 NEXT N
471 FOR N=0 TO 100 STEP 10
472 PRINT N
473 NEXT N
474 FOR N=0 TO 100 STEP 10
475 PRINT N
476 NEXT N
477 FOR N=0 TO 100 STEP 10
478 PRINT N
479 NEXT N
480 FOR N=0 TO 100 STEP 10
481 PRINT N
482 NEXT N
483 FOR N=0 TO 100 STEP 10
484 PRINT N
485 NEXT N
486 FOR N=0 TO 100 STEP 10
487 PRINT N
488 NEXT N
489 FOR N=0 TO 100 STEP 10
490 PRINT N
491 NEXT N
492 FOR N=0 TO 100 STEP 10
493 PRINT N
494 NEXT N
495 FOR N=0 TO 100 STEP 10
496 PRINT N
497 NEXT N
498 FOR N=0 TO 100 STEP 10
499 PRINT N
500 NEXT N
501 FOR N=0 TO 100 STEP 10
502 PRINT N
503 NEXT N
504 FOR N=0 TO 100 STEP 10
505 PRINT N
506 NEXT N
507 FOR N=0 TO 100 STEP 10
508 PRINT N
509 NEXT N
510 FOR N=0 TO 100 STEP 10
511 PRINT N
512 NEXT N
513 FOR N=0 TO 100 STEP 10
514 PRINT N
515 NEXT N
516 FOR N=0 TO 100 STEP 10
517 PRINT N
518 NEXT N
519 FOR N=0 TO 100 STEP 10
520 PRINT N
521 NEXT N
522 FOR N=0 TO 100 STEP 10
523 PRINT N
524 NEXT N
525 FOR N=0 TO 100 STEP 10
526 PRINT N
527 NEXT N
528 FOR N=0 TO 100 STEP 10
529 PRINT N
530 NEXT N
531 FOR N=0 TO 100 STEP 10
532 PRINT N
533 NEXT N
534 FOR N=0 TO 100 STEP 10
535 PRINT N
536 NEXT N
537 FOR N=0 TO 100 STEP 10
538 PRINT N
539 NEXT N
540 FOR N=0 TO 100 STEP 10
541 PRINT N
542 NEXT N
543 FOR N=0 TO 100 STEP 10
544 PRINT N
545 NEXT N
546 FOR N=0 TO 100 STEP 10
547 PRINT N
548 NEXT N
549 FOR N=0 TO 100 STEP 10
550 PRINT N
551 NEXT N
552 FOR N=0 TO 100 STEP 10
553 PRINT N
554 NEXT N
555 FOR N=0 TO 100 STEP 10
556 PRINT N
557 NEXT N
558 FOR N=0 TO 100 STEP 10
559 PRINT N
560 NEXT N
561 FOR N=0 TO 100 STEP 10
562 PRINT N
563 NEXT N
564 FOR N=0 TO 100 STEP 10
565 PRINT N
566 NEXT N
567 FOR N=0 TO 100 STEP 10
568 PRINT N
569 NEXT N
570 FOR N=0 TO 100 STEP 10
571 PRINT N
572 NEXT N
573 FOR N=0 TO 100 STEP 10
574 PRINT N
575 NEXT N
576 FOR N=0 TO 100 STEP 10
577 PRINT N
578 NEXT N
579 FOR N=0 TO 100 STEP 10
580 PRINT N
581 NEXT N
582 FOR N=0 TO 100 STEP 10
583 PRINT N
584 NEXT N
585 FOR N=0 TO 100 STEP 10
586 PRINT N
587 NEXT N
588 FOR N=0 TO 100 STEP 10
589 PRINT N
590 NEXT N
591 FOR N=0 TO 100 STEP 10
592 PRINT N
593 NEXT N
594 FOR N=0 TO 100 STEP 10
595 PRINT N
596 NEXT N
597 FOR N=0 TO 100 STEP 10
598 PRINT N
599 NEXT N
600 FOR N=0 TO 100 STEP 10
601 PRINT N
602 NEXT N
603 FOR N=0 TO 100 STEP 10
604 PRINT N
605 NEXT N
606 FOR N=0 TO 100 STEP 10
607 PRINT N
608 NEXT N
609 FOR N=0 TO 100 STEP 10
610 PRINT N
611 NEXT N
612 FOR N=0 TO 100 STEP 10
613 PRINT N
614 NEXT N
615 FOR N=0 TO 100 STEP 10
616 PRINT N
617 NEXT N
618 FOR N=0 TO 100 STEP 10
619 PRINT N
620 NEXT N
621 FOR N=0 TO 100 STEP 10
622 PRINT N
623 NEXT N
624 FOR N=0 TO 100 STEP 10
625 PRINT N
626 NEXT N
627 FOR N=0 TO 100 STEP 10
628 PRINT N
629 NEXT N
630 FOR N=0 TO 100 STEP 10
631 PRINT N
632 NEXT N
633 FOR N=0 TO 100 STEP 10
634 PRINT N
635 NEXT N
636 FOR N=0 TO 100 STEP 10
637 PRINT N
638 NEXT N
639 FOR N=0 TO 100 STEP 10
640 PRINT N
641 NEXT N
642 FOR N=0 TO 100 STEP 10
643 PRINT N
644 NEXT N
645 FOR N=0 TO 100 STEP 10
646 PRINT N
647 NEXT N
648 FOR N=0 TO 100 STEP 10
649 PRINT N
650 NEXT N
651 FOR N=0 TO 100 STEP 10
652 PRINT N
653 NEXT N
654 FOR N=0 TO 100 STEP 10
655 PRINT N
656 NEXT N
657 FOR N=0 TO 100 STEP 10
658 PRINT N
659 NEXT N
660 FOR N=0 TO 100 STEP 10
661 PRINT N
662 NEXT N
663 FOR N=0 TO 100 STEP 10
664 PRINT N
665 NEXT N
666 FOR N=0 TO 100 STEP 10
667 PRINT N
668 NEXT N
669 FOR N=0 TO 100 STEP 10
670 PRINT N
671 NEXT N
672 FOR N=0 TO 100 STEP 10
673 PRINT N
674 NEXT N
675 FOR N=0 TO 100 STEP 10
676 PRINT N
677 NEXT N
678 FOR N=0 TO 100 STEP 10
679 PRINT N
680 NEXT N
681 FOR N=0 TO 100 STEP 10
682 PRINT N
683 NEXT N
684 FOR N=0 TO 100 STEP 10
685 PRINT N
686 NEXT N
687 FOR N=0 TO 100 STEP 10
688 PRINT N
689 NEXT N
690 FOR N=0 TO 100 STEP 10
691 PRINT N
692 NEXT N
693 FOR N=0 TO 100 STEP 10
694 PRINT N
695 NEXT N
696 FOR N=0 TO 100 STEP 10
697 PRINT N
698 NEXT N
699 FOR N=0 TO 100 STEP 10
700 PRINT N
701 NEXT N
702 FOR N=0 TO 100 STEP 10
703 PRINT N
704 NEXT N
705 FOR N=0 TO 100 STEP 10
706 PRINT N
707 NEXT N
708 FOR N=0 TO 100 STEP 10
709 PRINT N
710 NEXT N
711 FOR N=0 TO 100 STEP 10
712 PRINT N
713 NEXT N
714 FOR N=0 TO 100 STEP 10
715 PRINT N
716 NEXT N
717 FOR N=0 TO 100 STEP 10
718 PRINT N
719 NEXT N
720 FOR N=0 TO 100 STEP 10
721 PRINT N
722 NEXT N
723 FOR N=0 TO 100 STEP 10
724 PRINT N
725 NEXT N
726 FOR N=0 TO 100 STEP 10
727 PRINT N
728 NEXT N
729 FOR N=0 TO 100 STEP 10
730 PRINT N
731 NEXT N
732 FOR N=0 TO 100 STEP 10
733 PRINT N
734 NEXT N
735 FOR N=0 TO 100 STEP 10
736 PRINT N
737 NEXT N
738 FOR N=0 TO 100 STEP 10
739 PRINT N
740 NEXT N
741 FOR N=0 TO 100 STEP 10
742 PRINT N
743 NEXT N
744 FOR N=0 TO 100 STEP 10
745 PRINT N
746 NEXT N
747 FOR N=0 TO 100 STEP 10
748 PRINT N
749 NEXT N
750 FOR N=0 TO 100 STEP 10
751 PRINT N
752 NEXT N
753 FOR N=0 TO 100 STEP 10
754 PRINT N
755 NEXT N
756 FOR N=0 TO 100 STEP 10
757 PRINT N
758 NEXT N
759 FOR N=0 TO 100 STEP 10
760 PRINT N
761 NEXT N
762 FOR N=0 TO 100 STEP 10
763 PRINT N
764 NEXT N
765 FOR N=0 TO 100 STEP 10
766 PRINT N
767 NEXT N
768 FOR N=0 TO 100 STEP 10
769 PRINT N
770 NEXT N
771 FOR N=0 TO 100 STEP 10
772 PRINT N
773 NEXT N
774 FOR N=0 TO 100 STEP 10
775 PRINT N
776 NEXT N
777 FOR N=0 TO 100 STEP 10
778 PRINT N
779 NEXT N
780 FOR N=0 TO 100 STEP 10
781 PRINT N
782 NEXT N
783 FOR N=0 TO 100 STEP 10
784 PRINT N
785 NEXT N
786 FOR N=0 TO 100 STEP 10
787 PRINT N
788 NEXT N
789 FOR N=0 TO 100 STEP 10
790 PRINT N
791 NEXT N
792 FOR N=0 TO 100 STEP 10
793 PRINT N
794 NEXT N
795 FOR N=0 TO 100 STEP 10
796 PRINT N
797 NEXT N
798 FOR N=0 TO 100 STEP 10
799 PRINT N
800 NEXT N
801 FOR N=0 TO 100 STEP 10
802 PRINT N
803 NEXT N
804 FOR N=0 TO 100 STEP 10
805 PRINT N
806 NEXT N
807 FOR N=0 TO 100 STEP 10
808 PRINT N
809 NEXT N
810 FOR N=0 TO 100 STEP 10
811 PRINT N
812 NEXT N
813 FOR N=0 TO 100 STEP 10
814 PRINT N
815 NEXT N
816 FOR N=0 TO 100 STEP 10
817 PRINT N
818 NEXT N
819 FOR N=0 TO 100 STEP 10
820 PRINT N
821 NEXT N
822 FOR N=0 TO 100 STEP 10
823 PRINT N
824 NEXT N
825 FOR N=0 TO 100 STEP 10
826 PRINT N
827 NEXT N
828 FOR N=0 TO 100 STEP 10
829 PRINT N
830 NEXT N
831 FOR N=0 TO 100 STEP 10
832 PRINT N
833 NEXT N
834 FOR N=0 TO 100 STEP 10
835 PRINT N
836 NEXT N
837 FOR N=0 TO 100 STEP 10
838 PRINT N
839 NEXT N
840 FOR N=0 TO 100 STEP 10
841 PRINT N
842 NEXT N
843 FOR N=0 TO 100 STEP 10
844 PRINT N
845 NEXT N
846 FOR N=0 TO 100 STEP 10
847 PRINT N
848 NEXT N
849 FOR N=0 TO 100 STEP 10
850 PRINT N
851 NEXT N
852 FOR N=0 TO 100 STEP 10
853 PRINT N
854 NEXT N
855 FOR N=0 TO 100 STEP 10
856 PRINT N
857 NEXT N
858 FOR N=0 TO 100 STEP 10
859 PRINT N
860 NEXT N
861 FOR N=0 TO 100 STEP 10
862 PRINT N
863 NEXT N
864 FOR N=0 TO 100 STEP 10
865 PRINT N
866 NEXT N
867 FOR N=0 TO 100 STEP 10
868 PRINT N
869 NEXT N
870 FOR N=0 TO 100 STEP 10
871 PRINT N
872 NEXT N
873 FOR N=0 TO 100 STEP 10
874 PRINT N
875 NEXT N
876 FOR N=0 TO 100 STEP 10
877 PRINT N
878 NEXT N
879 FOR N=0 TO 100 STEP 10
880 PRINT N
881 NEXT N
882 FOR N=0 TO 100 STEP 10
883 PRINT N
884 NEXT N
885 FOR N=0 TO 100 STEP 10
886 PRINT N
887 NEXT N
888 FOR N=0 TO 100 STEP 10
889 PRINT N
890 NEXT N
891 FOR N=0 TO 100 STEP 10
892 PRINT N
893 NEXT N
894 FOR N=0 TO 100 STEP 10
895 PRINT N
896 NEXT N
897 FOR N=0 TO 100 STEP 10
898 PRINT N
899 NEXT N
900 FOR N=0 TO 100 STEP 10
901 PRINT N
902 NEXT N
903 FOR N=0 TO 100 STEP 10
904 PRINT N
905 NEXT N
906 FOR N=0 TO 100 STEP 10
907 PRINT N
908 NEXT N
909 FOR N=0 TO 100 STEP 10
910 PRINT N
911 NEXT N
912 FOR N=0 TO 100 STEP 10
913 PRINT N
914 NEXT N
915 FOR N=0 TO 100 STEP 10
916 PRINT N
917 NEXT N
918 FOR N=0 TO 100 STEP 10
919 PRINT N
920 NEXT N
921 FOR N=0 TO 100 STEP 10
922 PRINT N
923 NEXT N
924 FOR N=0 TO 100 STEP 10
925 PRINT N
926 NEXT N
927 FOR N=0 TO 100 STEP 10
928 PRINT N
929 NEXT N
930 FOR N=0 TO 100 STEP 10
931 PRINT N
932 NEXT N
933 FOR N=0 TO 100 STEP 10
934 PRINT N
935 NEXT N
936 FOR N=0 TO 100 STEP 10
937 PRINT N
938 NEXT N
939 FOR N=0 TO 100 STEP 10
940 PRINT N
941 NEXT N
942 FOR N=0 TO 100 STEP 10
943 PRINT N
944 NEXT N
945 FOR N=0 TO 100 STEP 10
946 PRINT N
947 NEXT N
948 FOR N=0 TO 100 STEP 10
949 PRINT N
950 NEXT N
951 FOR N=0 TO 100 STEP 10
952 PRINT N
953 NEXT N
954 FOR N=0 TO 100 STEP 10
955 PRINT N
956 NEXT N
957 FOR N=0 TO 100 STEP 10
958 PRINT N
959 NEXT N
960 FOR N=0 TO 100 STEP 10
961 PRINT N
962 NEXT N
963 FOR N=0 TO 100 STEP 10
964 PRINT N
965 NEXT N
966 FOR N=0 TO 100 STEP 10
967 PRINT N
968 NEXT N
969 FOR N=0 TO 100 STEP 10
970 PRINT N
971 NEXT N
972 FOR N=0 TO 100 STEP 10
973 PRINT N
974 NEXT N
975 FOR N=0 TO 100 STEP 10
976 PRINT N
977 NEXT N
978 FOR N=0 TO 100 STEP 10
979 PRINT N
980 NEXT N
981 FOR N=0 TO 100 STEP 10
982 PRINT N
983 NEXT N
984 FOR N=0 TO 100 STEP 10
985 PRINT N
986 NEXT N
987 FOR N=0 TO 100 STEP 10
988 PRINT N
989 NEXT N
990 FOR N=0 TO 100 STEP 10
991 PRINT N
992 NEXT N
993 FOR N=0 TO 100 STEP 10
994 PRINT N
995 NEXT N
996 FOR N=0 TO 100 STEP 10
997 PRINT N
998 NEXT N
999 FOR N=0 TO 100 STEP 10
1000 PRINT N
1001 NEXT N
1002 FOR N=0 TO 100 STEP 10
1003 PRINT N
1004 NEXT N
1005 FOR N=0 TO 100 STEP 10
1006 PRINT N
1007 NEXT N
1008 FOR N=0 TO 100 STEP 10
1009 PRINT N
1010 NEXT N
1011 FOR N=0 TO 100 STEP 10
1012 PRINT N
1013 NEXT N
1014 FOR N=0 TO 100 STEP 10
1015 PRINT N
1016 NEXT N
1017 FOR N=0 TO 100 STEP 10
1018 PRINT N
1019 NEXT N
1020 FOR N=0 TO 100 STEP 10
1021 PRINT N
1022 NEXT N
1023 FOR N=0 TO 100 STEP 10
1024 PRINT N
1025 NEXT N
1026 FOR N=0 TO 100 STEP 10
1027 PRINT N
1028 NEXT N
1029 FOR N=0 TO 100 STEP 10
1030 PRINT N
1031 NEXT N
1032 FOR N=0 TO 100 STEP 10
1033 PRINT N
1034 NEXT N
1035 FOR N=0 TO 100 STEP 10
1036 PRINT N
1037 NEXT N
1038 FOR N=0 TO 100 STEP 10
1039 PRINT N
1040 NEXT N
1041 FOR N=0 TO 100 STEP 10
1042 PRINT N
1043 NEXT N
1044 FOR N=0 TO 100 STEP 10
1045 PRINT N
1046 NEXT N
1047 FOR N=0 TO 100 STEP 10
1048 PRINT N
1049 NEXT N
1050 FOR N=0 TO 100 STEP 10
1051 PRINT N
1052 NEXT N
1053 FOR N=0 TO 100 STEP 10
1054 PRINT N
1055 NEXT N
1056 FOR N=0 TO 100 STEP 10
1057 PRINT N
1058 NEXT N
1059 FOR N=0 TO 100 STEP 10
1060 PRINT N
1061 NEXT N
1062 FOR N=0 TO 100 STEP 10
1063 PRINT N
1064 NEXT N
1065 FOR N=0 TO 100 STEP 10
1066 PRINT N
1067 NEXT N
1068 FOR N=0 TO 100 STEP 10
1069 PRINT N
1070 NEXT N
1071 FOR N=0 TO 100 STEP 10
1072 PRINT N
1073 NEXT N
1074 FOR N=0 TO 100 STEP 10
1075 PRINT N
1076 NEXT N
1077 FOR N=0 TO 100 STEP 10
1078 PRINT N
1079 NEXT N
1080 FOR N=0 TO 100 STEP 10
1081 PRINT N
1082 NEXT N
1083 FOR N=0 TO 100 STEP 10
1084 PRINT N
1085 NEXT N
1086 FOR N=0 TO 100 STEP 10
1087 PRINT N
1088 NEXT N
1089 FOR N=0 TO 100 STEP 10
1090 PRINT N
1091 NEXT N
1092 FOR N=0 TO 100 STEP 10
1093 PRINT N
1094 NEXT N
1095 FOR N=0 TO 100 STEP 10
1096
```

Launchons le programme par RUN NEWLINE, et imaginons qu'on nous promette de nous rendre au bout de 15 ans la somme investie, multipliée par deux : coup double ! Répondons 15 à la première question de la machine, et supposons que nous placions 5 000 F. Les 10 000 F récupérés au terme des 15 ans sont livrés à l'ordinateur, qui répond par un *taux annuel* de 4,73 %. Loin de s'avérer exceptionnelle, une telle opération risque de se révéler catastrophique alors que d'aucuns jugent déjà bien maigres les 8,5 % de la Caisse d'Epargne ! Le calcul proprement dit s'effectue à la ligne 70, et ne doit rien de particulier à l'informatique : le principe des intérêts composés existait en effet bien avant les premiers ordinateurs.

CALCUL DE RENDEMENT

DUREE DU PLACEMENT EN ANNEES ?
15

SOMME INVESTIE ?
5000

SOMME APRES 15 ANS ?
10000

TAUX ANNUEL : 4.7294123 0/0

Remarquons par contre la présence des lignes 48 et 50, qui n'ont pu être prévues que parce que le reste du programme n'est pas bien encombrant, mais qui permettent à la machine de ne pas commettre de fautes d'orthographe !

L'instruction 48 construit une chaîne A\$, qui ne contient qu'un caractère : S. Pour toute autre durée de placement que un an, ce S est imprimé à la suite du texte « AN » de l'instruction 55.

Par contre, si D = 1, A\$ est transformée en chaîne vide par l'instruction 50, et n'interviendra pas dans la ligne 55. Enfin, l'impression du résultat s'effectue de façon très classique au moyen des instructions 71 à 75.

Ce programme est particulièrement utile pour évaluer les performances des placements proposés sous forme de « bons ». Il convient cependant de déduire de la somme investie les éventuelles économies fiscales pouvant résulter du placement, et d'ajouter à la

somme récupérée à terme les éventuels excédents d'actif que certains organismes s'engagent à distribuer. On obtiendra ainsi la valeur réelle du rendement.

2° Calcul de la plus-value d'un placement

Ce programme fonctionne « à l'envers » du précédent, en ce sens qu'il permet, à partir du taux du placement, de calculer ce que devient en un nombre donné d'années un capital donné.

```

10 REM "PLACEMENT"
20 PRINT "CALCUL D'INTERETS"
30 PRINT "-----"
40 PRINT
50 PRINT "TAUX EN 0/0 ?"
60 INPUT T
61 PRINT T; " 0/0"
62 PRINT
70 PRINT "DUREE EN ANNEES ?"
80 INPUT D
81 PRINT D
82 PRINT
90 PRINT "SOMME INVESTIE ?"
100 INPUT S
101 PRINT S; " F"
110 LET R=S*((1+T/100)^*D)
112 PRINT
115 PRINT "====="
116 PRINT
117 LET A$="S"
118 IF D=1 THEN LET A$=""
120 PRINT "CAPITAL APRES ";D;" AN";A$; " : ";R; " F"
121 PRINT
122 PRINT "====="
130 REM "COPYRIGHT 1982"

```

Il est par exemple amusant (et très classique) de calculer l'avoir du livret de Caisse d'Epargne d'un centenaire, à qui on aurait ouvert un compte de 1 F le jour de sa naissance, et qui n'aurait jamais effectué de versement ni de retrait.

En supposant un taux uniforme de 8,5 % sur 100 ans, le mécanisme des intérêts capitalisés mène à un avoir frisant les 3 500 F !

CALCUL D'INTERETS

TAUX EN % ?

6.5 %

DUREE EN ANNEES ?

100

SOMME INVESTIE ?

1 F

CAPITAL APRES 100 ANS : 3491.1927

F

Bien évidemment, votre banquier habituel vous démontrera, « érosion monétaire » aidant, que le franc de l'époque représentait beaucoup plus, en pouvoir d'achat, que les 3 500 F d'aujourd'hui.

Si vous essayez de calculer le solde du livret de Caisse d'Epargne de Jésus-Christ à la date de publication de cet ouvrage, même en supposant le premier et unique versement limité à un centime (0,01 F), la machine refuse pudiquement de répondre autre chose que 6/110. En effet, le résultat du calcul de la ligne 110 dépasse la capacité arithmétique de la machine, soit 10^{38} environ : un nombre comportant plus de trente-huit zéros... joli magot !

Le déroulement du programme est similaire à celui du précédent, mise à part la formule dans laquelle la racine $n^{\text{ème}}$ a cédé la place à une puissance $n^{\text{ème}}$, et à l'exception de la mise en pages, plus luxueuse, mais toujours expurgée de toute velléité de faute d'orthographe.

Des modifications mineures de ces deux programmes permettraient facilement le calcul des taux réels des crédits, ou l'optimisation de certains placements spéciaux, tels que les plans d'épargne-logement.

Chapitre 7**Le ZX-81 et la comptabilité**

Il ne saurait évidemment être question ici d'envisager le passage sur ZX-81 de toute la comptabilité d'une entreprise, encore que la puissance de calcul de cet « ordinateur individuel » puisse se compara favorablement à celle de certains « monstres » des années passées.

1^o Facturation

Nous donnerons simplement ici quelques programmes permettant l'édition automatique de factures simples. Bien sûr, il est nécessaire de prévoir la transcription du contenu de l'écran sur imprimante, au moyen de la commande COPY.

```

2 PRINT "FACTURE DU "
4 INPUT A$
10 PRINT A$ -----
11 PRINT -----
12 PRINT
15 PRINT "MONTANT HORS TAXES"
20 INPUT H
25 PRINT H
30 PRINT "ESCOMpte    %"
35 INPUT E
40 PRINT AT 5,9;E
41 LET R=E*H/100
46 PRINT "-";R
48 PRINT
50 PRINT "MONTANT NET H.T."
55 LET N=H-R
60 PRINT N
65 PRINT

```

(Suite au verso)

```

70 PRINT "T.U.A. 17,60 0/0: "
75 LET T=17.6*N/100
76 PRINT T
78 PRINT
79 PRINT "=====
80 PRINT "NET A PAYER TTC: "
85 LET C=N+T
90 PRINT C; " F"
110 PRINT "=====
120 REM "COPYRIGHT 1982"

```

Ce programme est le plus complet des trois qui vont être regroupés ici : il permet en effet d'établir une facture avec escompte à taux quelconque, et TVA au taux de 17,60 % (la modification serait extrêmement simple pour tout autre taux).

Tous les montants intermédiaires sont indiqués, et le net à payer TTC est mis en valeur par deux bandeaux décoratifs. De même, il est prévu de faire ressortir la date au moyen d'une ligne décorative.

La machine pose les questions en langage clair, ce qui fait que ce programme peut être utilisé sans connaissances particulières en informatique.

```

FACTURE DU 13/01/82
-----
MONTANT HORS TAXES
100
ESCOMPTE 2 0/0
-2
MONTANT NET H.T.
98
T.U.A. 17,60 0/0:
17.248
=====
NET A PAYER TTC:
115.248 F
=====
```

La date est acquise et imprimée sous forme de chaîne, ce qui permet une grande liberté quant à son libellé. Il est même possible de prévoir une indication du mois en toutes lettres, à condition de rallonger la ligne de pointillés de l'instruction 11.

Le reste des calculs ne fait pas appel à d'autres mécanismes que ceux étudiés jusqu'à présent.

```

2 PRINT "FACTURE DU ";
4 INPUT A$
10 PRINT A$ -----
11 PRINT "-----"
12 PRINT
15 PRINT "MONTANT HORS TAXES"
20 INPUT H
25 PRINT H
48 PRINT
55 LET N=H
65 PRINT
70 PRINT "T.U.A. 17,60 0/0: "
75 LET T=17.6*N/100
76 PRINT T
77 PRINT
78 PRINT
79 PRINT "=====
80 PRINT "NET A PAYER TTC: "
85 LET C=N+T
90 PRINT C; " F"
110 PRINT "=====
120 REM "COPYRIGHT 1982"

```

```

FACTURE DU 14/01/82
-----
MONTANT HORS TAXES
1000
T.U.A. 17,60 0/0:
176
=====
NET A PAYER TTC:
1176 F
=====
```

Ce programme est une version simplifiée du précédent, en ce sens qu'il ne prévoit pas d'escompte. Il est en effet inutile d'imprimer : « escompte 0 % », et de frustrer ainsi les clients qui ne bénéficient pas de cet avantage.

```

2 PRINT "FACTURE DU "
4 INPUT A$
10 PRINT A$
11 PRINT "-----"
12 PRINT
15 PRINT "MONTANT HORS TAXES"
20 INPUT H
25 PRINT H
48 PRINT
65 PRINT
77 PRINT
78 PRINT
79 PRINT "====="
80 PRINT "NET A PAYER: "
82 PRINT
85 PRINT H; " F"
90 PRINT
110 PRINT "====="
112 PRINT
113 PRINT
115 PRINT " PRESTATIONS EXONEREES DE TVA"
120 PRINT "-----"
130 REM "COPYRIGHT 1982"

FACTURE DU 15/01/82
-----
MONTANT HORS TAXES
100
=====
NET A PAYER: 100 F
=====
PRESTATIONS EXONEREES DE TVA
-----
```

Le plus simple des trois, ce programme couvre les cas de vente hors taxes. Il s'emploie exactement comme le précédent, en ce sens qu'il suffit d'entrer la date et le montant de base pour que la facture s'édite d'elle-même.

Ce jeu de trois programmes permet d'économiser du temps de dactylographie toujours précieux, ou peut servir de base pour une initiation aux mécanismes de la facturation.

2° Recettes dépenses

Les services rendus par le ZX-81 en matière comptable ne s'arrêtent pas là : voici maintenant deux programmes assez spéciaux, qu'il convient d'utiliser avec deux cassettes sans fin de durée une minute (TDK ENDLESS EC1).

Ils permettent de cumuler recettes et dépenses durant toute une année, et de fournir à la demande, et n'importe quand, un état récapitulatif avec ventilation selon le régime de TVA.

```

1 REM "RECETTES"
2 LET RC=0
3 LET RH=0
4 LET RX=0
5 LET RV=0
6 LET RT=0
16 CLS
18 PRINT "RECETTES DU JOUR ?"
25 PRINT "TTC: "
30 INPUT TTC
35 PRINT AT 1,8;TTC
40 PRINT " EXO: "
45 INPUT EXO
50 PRINT AT 2,8;EXO
55 LET RC=RC+TTC
56 LET HT=TTC/1.176
57 LET TVA=TTC-HT
60 LET RH=RH+HT+EXO
65 LET RX=RX+EXO
70 LET RV=RV+TVA
72 LET RT=RT+TTC
75 PRINT "RECETTES TOTALES: "
80 PRINT "HORS TAXES: ";RH; " F
85 PRINT "DONT EXONERE: ";RX; " F"
90 PRINT "TVA: ";RV; " F"
93 PRINT "TTC: ";RT; " F"
94 PRINT "TOTALES: ";RT+RX; " F
95 PRINT "ENREGISTRER"
96 PAUSE 300
97 SAVE "R"
98 CLS
99 SAVE "B"
100 GOTO 10
110 REM "COPYRIGHT 1982"
```

Ce programme doit faire l'objet de précautions d'emploi particulières, en raison de son principe : il se stocke en effet de lui-même *avec toutes les données relatives à l'état de ses variables internes*, sur une boucle de bande d'une durée de une minute, et ce, à deux reprises, de façon à éviter les problèmes dus au passage de la collure (perte d'informations).

De ce fait, *il n'est pas question* de le lancer par RUN NEWLINE, ce qui *effacerait à jamais* ces précieuses données, mais uniquement par GOTO 16 NEWLINE. Egalement, il ne faut pas oublier de commuter le magnétophone en enregistrement dès que cet ordre apparaît sur l'écran, ni de l'arrêter immédiatement lorsqu'au bout d'une minute apparaît à nouveau la question « *recettes du jour ?* ». Dans l'intervalle, on ne se souciera pas des divers aspects présentés par l'écran TV.

```

RECETTES DU JOUR ?
TTC: 0
EXO: 150
RECETTES TOTALES:
HORS TAXES: 450 F
DONT EXONERE: 350 F
TVA: 17.6 F
TTC: 117.6 F
TOTALES: 467.6 F
ENREGISTRER

```

Ce programme suppose que les recettes possibles sont de deux types : TTC avec une TVA de 17,60 %, ou exonérées de TVA. Comme précédemment, il serait facile de prévoir d'autres cas, dans les limites, toutefois, des 1 024 octets de mémoire, en l'absence de l'extension de 16 Ko.

Sitôt ces deux données entrées, la machine calcule toute une variété de totaux, *tenant compte des opérations antérieures*, conservées sur la cassette, totaux qui se révèlent très utiles lors de l'établissement des diverses déclarations fiscales.

Le déroulement des calculs est très classique, excepté le fait que les noms des variables font appel à plusieurs lettres, afin d'en faciliter l'identification.

La fin du programme, gérant les opérations d'enregistrement, appelle par contre quelques commentaires.

L'instruction 95 imprime l'ordre de service « *enregistrer* », mais avec des caractères présentés en « *vidéo inversée* » pour attirer l'attention de l'opérateur.

On obtient cet effet en frappant la commande « GRAPHICS » (touche 0 en position shift) *avant et après* le groupe de caractères à imprimer (dans notre cas, avant E et après R). Une pause prévue à la ligne 96 laisse quelques secondes pour démarrer l'enregistreur. Cette pause, ajoutée à la durée des deux « *sauvegardes* » des lignes 97 et 99, permet de renouveler entièrement le contenu de la cassette en boucle. Entre les deux, l'écran est effacé afin que le retour à l'instruction 16 se fasse sans perte de place.

On notera que la présentation en vidéo inversée des « R » contenus dans les instructions SAVE n'est qu'une anomalie de fonctionnement de l'ordinateur (d'ailleurs avouée par Sinclair) et qu'il n'est pas nécessaire de frapper ainsi cette lettre lors de l'entrée du programme.

L'appel du programme sur la cassette se fera par :

```

LOAD ""
ou par
LOAD " R"

```

Mais on ne s'inquiétera pas d'éventuelles réactions imprévues de la machine lors du passage de la collure de la bande : il suffit de désalimenter l'ordinateur quelques secondes et d'effectuer une nouvelle tentative.

Dans tous les cas, l'un au moins des deux enregistrements est utilisable, et il peut arriver que les deux le soient.

A la condition, donc, qu'aucune fausse manœuvre ne soit commise, ou qu'aucun incident ne survienne, la cassette contient à tout instant des données comptables à jour, permettant une surveillance précise des recettes, ainsi que des prévisions fiscales.

Bien entendu, ces données ne doivent être considérées que comme indicatives, du fait même de leur fragilité, et doivent être doublées par un enregistrement « *papier* » des recettes, effectué selon les règles habituelles en la matière.

Le programme suivant traite les dépenses selon la même procédure :

```

2 LET T=0
4 LET V=0
6 LET H=0
8 LET X=0
9 PRINT "FRAIS"
10 PRINT
11 PRINT "TTC 17"

```

(Suite au verso)

```

12 INPUT A
14 PRINT A
16 PRINT "TTC 33"
18 INPUT B
20 PRINT B
22 PRINT "EXO"
24 INPUT E
26 PRINT E
28 PRINT
29 PRINT
30 PRINT "TOTAUX"
31 PRINT
32 LET T=T+A+B
33 LET Y=(A/1.176)+(B/1.3333)
34 LET U=U+A+B-Y
36 LET H=H+Y+E
38 LET X=X+E
40 PRINT "HT: ";(H); " F"
42 PRINT "DONT EXO: ";(X); " F"
44 PRINT "TVA: ";(U); " F"
46 PRINT "TTC: ";(T); " F"
47 PRINT "TOTAL: ";(T+X); " F"
48 PRINT
50 PRINT "ENREGISTRER"
53 PAUSE 500
54 CLS
55 SAVE "F"
57 SAVE "F"
60 GOTO 9
70 REM "COPYRIGHT 1982"

```

Il est cependant plus compliqué, car il doit envisager les trois types suivants de frais : taxés à 17,60 %, taxés à 33 %, et exonérés. (L'introduction d'autres taux de TVA n'est guère possible avec 1 024 octets.)

Il s'utilise de la même façon, excepté que son lancement se fait par GOTO 9 au lieu de GOTO 16, et qu'il peut se lancer seul lors de sa lecture sur magnétophone, et selon les humeurs (imprévisibles) de la machine.

FRAIS

```

TTC 17
117.6
TTC 33
133.33
EXO
200

```

TOTAUX

HT:	400 F
DONT EXO:	200 F
TVA:	50.93 F
TTC:	250.93 F
TOTAL:	450.93 F

ENREGISTRER

Pour nos lecteurs qui ne souhaiteraient pas utiliser ces programmes en liaison avec une cassette mémoire de masse, signalons la possibilité de supprimer les instructions SAVE et PAUSE, et de les utiliser pour effectuer des totalisations à partir d'une comptabilité « papier ».

De toutes façons, ces programmes ne sont publiés ici qu'à titre documentaire, et nous ne saurions engager en aucune façon notre responsabilité quant à leur utilisation à des fins professionnelles, utilisation pour laquelle notre accord serait d'ailleurs nécessaire.

Constitution de fichiers de données

Dans l'esprit du grand public, l'ordinateur est souvent assimilé à un gigantesque fichier.

En effet, ce n'est pas le moindre mérite de ces machines que de pouvoir retrouver, parmi une énorme masse de données, celle-là même dont on a précisément besoin. Au niveau individuel, ce type de fonctionnement peut surtout servir à faire fonctionner le ZX-81 en répertoire d'adresses, de renseignements divers, ou de numéros de téléphone.

1^o Répertoire

Le premier programme de ce chapitre permet de stocker sur cassette, puis de retrouver n'importe quel texte préalablement frappé au clavier (autrement dit une chaîne), et dont la longueur peut être assez importante. On pourrait envisager ainsi le stockage de poèmes, de formules mathématiques, d'adresses, de recettes de cuisine, de résumés de livres, etc.

Chaque texte ainsi stocké doit être affublé d'un nom, sous lequel on l'enregistrera, et sous lequel (très exactement le même) on ira le rechercher.

Lorsqu'un grand nombre d'enregistrements sont ainsi effectués sur une même cassette, il peut être utile de prévoir un index lié aux indications du compteur du magnétophone. Autrement, le *temps d'accès* à l'information risque d'être voisin de la durée de lecture de la cassette !

```

5 REM "FICHIER"
10 PRINT "TITRE ?"
15 LET N=0
20 INPUT A$
22 CLS
25 PRINT "TEXTE ?"
30 INPUT B$
32 CLS
34 PRINT B$
35 IF N=1 THEN STOP
36 LET N=1
37 SAVE A$
40 LET N=1
44 CLS
45 GOTO 34
50 REM "COPYRIGHT 1982"

```

Ce programme est composé de deux parties théoriquement distinctes :

- une partie « enregistrement » comprenant les instructions 10 à 37 pour demander à l'opérateur titre et contenu du texte à stocker ;
- une partie « lecture », composée d'une partie des instructions précédentes, mais exécutées avec $N = 1$ au lieu de $N = 0$ précédemment, ce qui change tout !

Le programme se lance par RUN NEWLINE lorsque l'on veut s'en servir pour un enregistrement, mais *se lance seul* lorsqu'il est identifié lors d'une recherche sur la cassette, à partir d'un nom préalablement indiqué.

Le mode d'emploi est simple :

- Charger le programme en machine et le lancer par RUN NEWLINE.
- Donner un nom au texte à stocker, par exemple : **ETSF** et presser NEWLINE.
- Frapper le texte, par exemple : **EDITIONS TECHNIQUES ET SCIENTIFIQUES FRANÇAISES.** (Noter qu'il pourra être utile de terminer chaque ligne par quelques espaces, afin d'éviter des coupures de mots malencontreuses.)

Egalement, on ne dépassera pas sans précautions la capacité de l'écran (il faudrait utiliser les commandes **CLS** et **CONT** ou même la fonction **SCROLL**).

- En fin de frappe, presser NEWLINE après avoir démarré l'enregistreur. L'arrêter dès la fin du transfert.

Pour retrouver ce texte parmi d'autres, il suffit d'amener la bande un peu avant l'emplacement supposé de l'enregistrement, et de frapper :

```

LOAD "ETSF"
NEWLINE

```

avant de démarrer le magnétophone en lecture. Le texte s'affiche sur l'écran dès qu'il est retrouvé.

2^e Annuaire

Pour des textes très courts (par exemple des numéros de téléphone), ce programme est assez lourd et occupe beaucoup trop de place sur la cassette. On pourra alors lui préférer celui-ci, qui regroupe plusieurs « textes » par programme, par exemple tous les numéros de téléphone des personnes dont le nom commence par la même lettre, dans la limite bien sûr de la capacité mémoire disponible. On peut déjà arriver à un résultat intéressant avec 1 Ko, mais le module 16 Ko donne accès à un véritable service « d'annuaire électronique » personnel.

L'exemple ci-dessous est rédigé avec quelques numéros que nous avons choisis, et il est bien évident que chacun écrira pour son propre compte les instructions IF-THEN contenant les renseignements à stocker.

Les programmes ainsi obtenus pourront alors être sauvegardés manuellement sous un titre rappelant, par exemple, l'initiale des noms qu'ils contiennent.

Une fois chargés en machine à partir de la cassette, ils doivent être lancés par RUN NEWLINE avant de pouvoir commencer le dialogue permettant de trouver le numéro de l'abonné dont on connaît le nom.

Dans notre exemple, si l'on frappe **POMPIERS**, la machine répond, après NEWLINE : texte d'introduction, 18. Essayez donc un nom qui n'est pas prévu dans le programme !

```

5 REM "TELEPHONE"
10 PRINT "NOM DE L'ABONNE ?"
12 LET B$=""
15 INPUT A$
18 CLS
20 IF A$="ETSF" THEN LET B$="2
00 33 05"

```

(Suite au verso)

```

30 IF A$="LIBRAIRIE" THEN LET
B$="878 09 94"
40 IF A$="HORLOGE" THEN LET B$=
="463 84 00"
50 IF A$="POMPIERS" THEN LET B
$="18"
200 IF B$="" THEN PRINT "INCONN
U AU BATAILLON"
210 IF B$="" THEN GOTO 10
220 PRINT "LE NUMERO DE TELEPHO
NE DE :"
230 PRINT A$;" EST : ";B$
250 REM "COPYRIGHT 1982"

```

LE NUMERO DE TELEPHONE DE
ETSF EST : 200 33 05

INCONNU AU BATAILLON
NOM DE L'ABONNE ?

Le fonctionnement du programme est facile à comprendre, car il n'utilise pas de subtilité particulière par rapport aux précédents.

Par contre, il est intéressant de noter que ce programme pourrait permettre de transformer un ZX-81 muni de la mémoire 16 Ko et... programmé comme il convient, en un dictionnaire électronique analogue aux « traductrices de poche » que l'on trouve actuellement sur le marché.

Le ZX-81 ordinateur de bord automobile

De nombreuses voitures « haut de gamme » sont désormais équipées munies d'un « ordinateur de bord » destiné à fournir au conducteur des informations d'aide à la conduite.

Le ZX-81 peut remplir de telles fonctions, à condition de l'associer à l'un de ces combinés TV/cassette/radio miniatures, ou encore à l'imprimante Sinclair et à un mégaphone ordinaire (par exemple le lecteur autoradio), pour le chargement des programmes.

Un adaptateur ramenant à 9 V la tension de la batterie sera bien sûr à prévoir.

1^o Calculateur de consommation

```

5 PRINT "      CONSOmmATION"
10 PRINT
15 PRINT "KILOMETRAGE A CE PLE
IN?"
16 INPUT K2
17 CLS
18 PRINT "KILOMETRAGE AU"
20 PRINT "PLEIN PRECEDENT ?"
22 INPUT K1
24 CLS
27 PRINT "LITRES CONSUMMÉS DEP
UIS"
28 PRINT "LE DERNIER PLEIN ?"
30 INPUT L
35 LET K=(K2-K1)/100
37 LET C=(INT (100*L/K))/100
38 CLS
40 PRINT "CONSOmmATION: "
41 PRINT
42 PRINT C;" LITRES AUX 100 KM
"45 REM "COPYRIGHT 1982"

```

Ce programme est une application de la méthode de calcul dite « anti-gaspi » ou « des pleins successifs ».

Celle-ci permet une évaluation de la consommation moyenne d'une voiture, plein après plein, et donc un dépistage rapide de toute augmentation anormale due à un incident mécanique ou à une évolution fâcheuse du style de conduite de l'utilisateur. Une autre application peut être trouvée dans la vérification des consommations annoncées par les constructeurs dans les trois situations « standard ».

Le dialogue en langage clair permet à tout conducteur, même « fâché avec l'informatique », de se servir du programme. Il est, bien sûr, nécessaire de noter les kilométrages des pleins précédents, afin de permettre à l'usager de répondre aux questions de la machine. Cependant, un stockage sur cassette, plein après plein, pourrait être envisagé, comme dans le cas de nos programmes comptables du chapitre 7, avec si nécessaire une comparaison entre la consommation moyenne sur tous les pleins et celle venant d'être calculée.

Nous laissons aux plus courageux de nos lecteurs le soin de mettre en œuvre cette idée, qui nous semble par trop lourde pour l'application envisagée.

Il nous faut, par contre, expliquer le fonctionnement de la ligne 37, laquelle calcule la consommation arrondie à deux chiffres après la « virgule ».

Pour ce faire, elle multiplie par cent la consommation exacte L/K, lui applique la fonction INT (partie entière) qui ne conserve que les chiffres précédant la virgule (ici le « point décimal »), puis divise à nouveau par cent cette partie entière pour retrouver deux chiffres significatifs.

Les lignes 35 et 37 auraient pu, bien sûr, être fusionnées en une seule, mais l'artifice décrit ci-dessus serait devenu beaucoup moins visible.

Notons également que, dans le but de ne pas distraire inutilement l'attention du conducteur, chaque étape du dialogue ne se fait qu'après effacement de l'écran qui ne contient ainsi que le strict minimum d'informations.

2° Programme d'aide à la conduite

```
5 REM "ROUTE"
12 PRINT
15 PRINT "KILOMETRAGE AU DEPAR
T ?"
20 INPUT D
22 PRINT D
25 GOSUB 123
28 LET MD=M+(60*H)
29 PRINT
30 PRINT "DISTANCE A COUVRIR ?"
32 INPUT B
34 PRINT B
36 PRINT
38 PRINT "BONNE ROUTE . . ."
39 PAUSE 600
40 CLS
42 PRINT "KILOMETRAGE ?"
43 INPUT K
44 CLS
45 GOSUB 123
46 LET MC=M+(60*H)
47 PAUSE 200
48 CLS
49 LET Q=(K-D)*60/(MC-MD)
50 PRINT "MOYENNE: ";Q;" KM/H"
51 PRINT "ENCORE ";B-K+D;" KM"
52 PRINT "SOIT ";60*(B-K+D)/Q;" MINUTES"
53 PAUSE 600
55 CLS
60 GOTO 42
123 PRINT
125 PRINT "HEURE ?"
126 PRINT AT 6,3;"HEURES ";
128 INPUT H
130 PRINT AT 6,0;H
132 PRINT AT 6,13;" MINUTES"
133 PRINT
134 INPUT M
136 PRINT AT 6,11;M
137 PRINT
138 RETURN
150 REM "COPYRIGHT 1982"
```

Ce programme permet une surveillance constante et, partant, une optimisation des conditions de conduite (vitesse, repos, etc.) en fonction des impératifs du trajet en cours (distance à couvrir en un temps donné). Il pourra rendre service à tous les « grands rouleurs », du représentant au routier, et, au niveau du particulier, à

tous ceux qui cherchent désespérément un moyen de distraire leurs jeunes passagers lors des longs trajets.

Sitôt lancé par RUN NEWLINE, le programme fait poser à la machine les questions nécessaires à son « initialisation » :

- kilométrage au départ
- heure exacte de départ en deux étapes (heures et minutes)
- distance à couvrir.

Sitôt satisfait, l'ordinateur vous souhaite bonne route, puis attend que, après quelques kilomètres tout de même, vous commeniez à l'interroger.

Pour ce faire, vous devez lui indiquer le kilométrage au compteur, puis l'heure, également en deux étapes.

Après un court instant de réflexion, il vous indique :

- votre moyenne horaire depuis le départ
- le kilométrage restant à parcourir
- le temps nécessaire pour le parcourir à la même moyenne.

KILOMETRAGE AU DEPART ?
90000

HEURE ?
8 HEURES 55 MINUTES

DISTANCE A COUVRIR ?
220

BONNE ROUTE !!!

MOYENNE: 140 KM/H
ENCORE 165 KM
SOIT 79.265714 MINUTES

Par manque de capacité mémoire, il n'est pas possible de faire calculer à la machine l'heure probable d'arrivée, ni, par exemple, la moyenne nécessaire pour arriver à une heure donnée, ni même d'aérer la mise en page au moyen de quelques ordres PRINT bien placés. C'est dire à quel point les 1 024 octets sont utilisés jusqu'au bout !

Ici encore, l'acquisition du module 16 Ko peut ouvrir la porte à une sérieuse extension des services rendus.

Pour les mêmes raisons de pénurie de mémoire, il a fallu mettre à contribution la technique du *sous-programme*, outil puissant dont nous avions pu nous passer jusqu'à présent. En effet, la procédure d'acquisition de l'heure, assez compliquée puisqu'une conversion en minutes est nécessaire, intervient deux fois dans ce programme.

Cette partie de programme, cette « routine », a été reléguée tout à fait en fin de listing, à partir de l'instruction 123, et a été terminée par une instruction RETURN (ligne 138).

Dans ces conditions, il est possible d'« appeler » ce « sous-programme » autant de fois que nécessaire, à l'aide d'une instruction GOSUB 123. Ceci est effectué aux lignes 25 et 45. Sitôt l'exécution du sous-programme terminée, l'instruction RETURN fait reprendre l'exécution du programme principal là où il en était resté, c'est-à-dire, dans notre cas, à la ligne 28 ou, respectivement, à la ligne 46.

À ce niveau, il est bien sûr nécessaire de « recopier », tout en leur faisant éventuellement subir un « prétraitement », les données H et M élaborées par le sous-programme, faute de quoi elles se trouveraient « surchargées » lors de sa seconde intervention. C'est ainsi qu'apparaissent les variables « refuge » MD (minutes départ) et MC (minutes courantes), nécessaires pour les calculs de moyenne à la ligne 49 (la place mémoire ne permet pas de procéder sans risque à un arrondi).

Un aperçu de cet encombrement de la mémoire peut être obtenu en listant le programme : seules quelques lignes apparaissent, suivies très vite par un compte rendu de la forme 4/. Cela signifie qu'il ne reste même pas assez de place mémoire pour le « fichier d'affichage », et que des catastrophes sont imminentes si on tente d'entrer de nouvelles lignes de programme (par exemple, « panne de mémoire » dans le courant des calculs, comme cela peut se produire avec notre programme de calcul des intégrales définies au chapitre 5).

Nous atteignons ici l'extrême limite des possibilités du ZX-81 de base, dépourvu de module d'extension mémoire.

Nous espérons avoir démontré à nos lecteurs, au moyen des programmes qui leur ont été proposés, que même en se fixant cette limite de 1 024 octets de RAM, des applications évoluées pouvaient être trouvées.

Il est bien évident que l'adjonction du module 16 Ko qui multiplie par seize la capacité de cette mémoire centrale, ouvre la porte à des applications très sophistiquées, qu'il ne saurait être question de traiter dans le cadre d'un seul volume tel que celui-ci.

En contrepartie, évidemment, l'écriture de programmes atteignant ce niveau de complexité rend souhaitable une maîtrise du langage BASIC qui ne peut, et encore, être considérée comme acquise que dans le cas où tous les programmes présentés ici auront été compris dans leurs moindres détails, et même améliorés par le lecteur.

Il est évident que l'écriture de programmes BASIC nécessite une connaissance approfondie de ce langage. Cependant, il existe de nombreux outils qui facilitent considérablement ce travail. Les principaux sont les éditeurs de code, qui permettent de saisir et de modifier le code source directement sur l'écran du terminal. Ces éditeurs sont généralement intégrés à l'environnement de développement (IDE) de l'éditeur, ce qui facilite la gestion des fichiers et des projets.

Il existe également des outils de débogage qui permettent d'exécuter le programme en temps réel et de visualiser les états de la machine à chaque étape. Ces outils sont utiles pour identifier les erreurs et les bugs dans le code. Ils peuvent également aider à optimiser le programme en identifiant les sections qui consomment le plus de temps ou de mémoire.

Il est également recommandé d'utiliser des bibliothèques et des frameworks pour gérer les tâches courantes. Par exemple, si vous souhaitez développer une application web, vous pouvez utiliser des frameworks comme Node.js ou Django. Ces frameworks fournissent des fonctionnalités préimplémentées pour gérer les requêtes, la base de données et la sécurité, ce qui vous permet de développer rapidement et efficacement.

Enfin, il est important de tester régulièrement le programme pour détecter les erreurs et les bugs. Vous pouvez utiliser des outils de test unitaire comme Jest ou Mocha pour vérifier que les fonctionnalités sont correctement implémentées. Il est également recommandé de faire des tests de performance pour vérifier que le programme fonctionne rapidement et efficacement.

Manutention des programmes

La possibilité de traduire en signaux audibles, pouvant être enregistrés sur cassette, les programmes et données qu'il contient, est l'un des points forts du ZX-81. Il convient donc de tirer le meilleur parti de cette puissante fonction.

1° Comment effectuer de bons enregistrements

Le procédé d'enregistrement retenu par Sinclair fait appel à des trains d'onde, d'une fréquence voisine de 3 kHz. C'est dire que la qualité Hi-Fi ne s'impose pas, loin de là, et que l'on n'aura pas à attacher d'importance à la courbe de réponse du magnétophone utilisé. Par contre, on aura intérêt à vérifier de près la qualité mécanique de l'appareil, un pleurage excessif ou un glissement de vitesse pouvant entraîner des conséquences fâcheuses.

Il faut noter, également, que la « vitesse de transmission » entre l'ordinateur et le magnétophone est de 250 informations par seconde. A la vitesse de défilement de la bande de 4,75 cm/s, ceci correspond à une information tous les deux dixièmes de millimètre !

Ainsi, des défauts légers dans la couche d'oxyde, ou des collures parfaitement tolérables en enregistrement sonore, sont presque toujours fatals à l'enregistrement d'un programme. On évitera donc pour ces applications les cassettes « tout venant », au profit de bonnes cassettes de marques connues, à l'oxyde de fer cependant, car les qualités des cassettes au chrome ou au métal ne serviraient ici strictement à rien.

Autre calcul édifiant : sachant que les plus longs programmes « tenant » dans 1 024 octets ne représentent, une fois enregistrés,

qu'environ trente secondes de bande, on choisira de préférence des cassettes C 60, ou même plus courtes si l'on parvient à en trouver.

Une C 60 peut ainsi accepter environ soixante programmes par piste, ce qui est déjà trop pour une recherche rapide.

Il est de toutes façons conseillé de recopier les programmes importants sur au moins une autre cassette, et de ne pas hésiter à utiliser les ergots défonçables de protection contre l'effacement. Il faut peut-être 30 secondes pour effacer entièrement un programme, mais un geste malheureux, de deux dixièmes de seconde, lui, suffit à le rendre définitivement inutilisable !

2^o Comment recopier des programmes

La copie de cassettes de programmes peut être utilisée à deux fins principales :

- protéger des programmes existants contre les risques de détérioration de la cassette qui les supporte ;
- communication à d'autres personnes de programmes ou de données de nature personnelle.

Nous devons mettre en garde nos lecteurs contre la tentation de « piratage » de cassettes de programmes, similaire à celui qui sévit dans le domaine des disques.

La loi est très sévère sur tout ce qui concerne les reproductions sans autorisation, et quiconque se risquerait à diffuser des programmes qui ne lui seraient pas personnels, par voie d'enregistrement magnétique (par exemple ceux publiés dans cet ouvrage) doit savoir qu'il s'exposerait à des sanctions sans commune mesure avec le service rendu à ses « correspondants » (loi du 11 mars 1957).

Ceci étant précisé, il faut savoir que les programmes traduits sous forme acoustique supportent assez mal la copie de magnétophone à magnétophone.

Une bonne copie se fait par l'intermédiaire du ZX-81, en chargeant le programme en machine, puis en le « sauvegardant » sur une autre cassette, sans oublier de rappeler son nom dans la commande SAVE.

L'ordinateur « régénère » le message sonore et ainsi, il n'existe aucune différence de qualité entre l'original et la copie. Les opérations sont bien sûr plus longues, mais en revanche, il ne faut qu'un seul magnétophone.

3^o Comment « téléphoner » ses programmes

Bien qu'une cassette puisse facilement voyager par la poste, il peut être intéressant de tenter de les « faire passer » par le réseau téléphonique, dont la bande passante de 300 à 3 400 Hz est exactement suffisante.

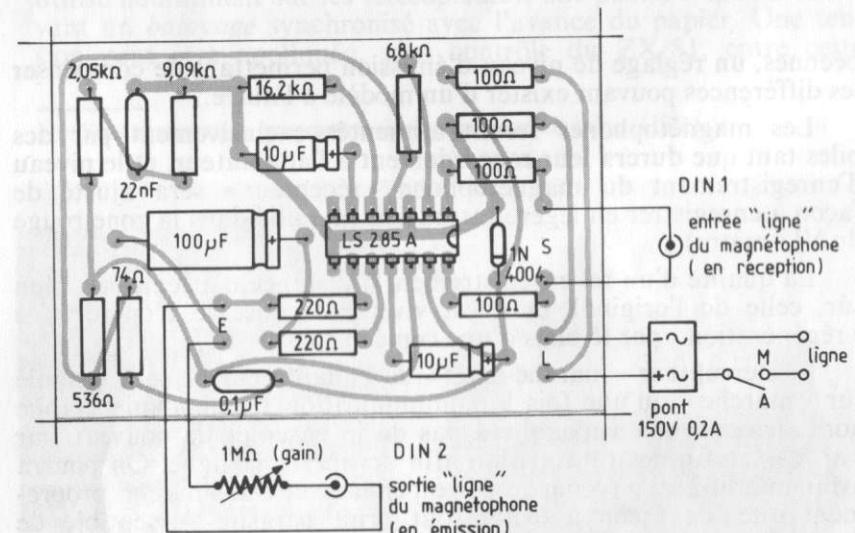
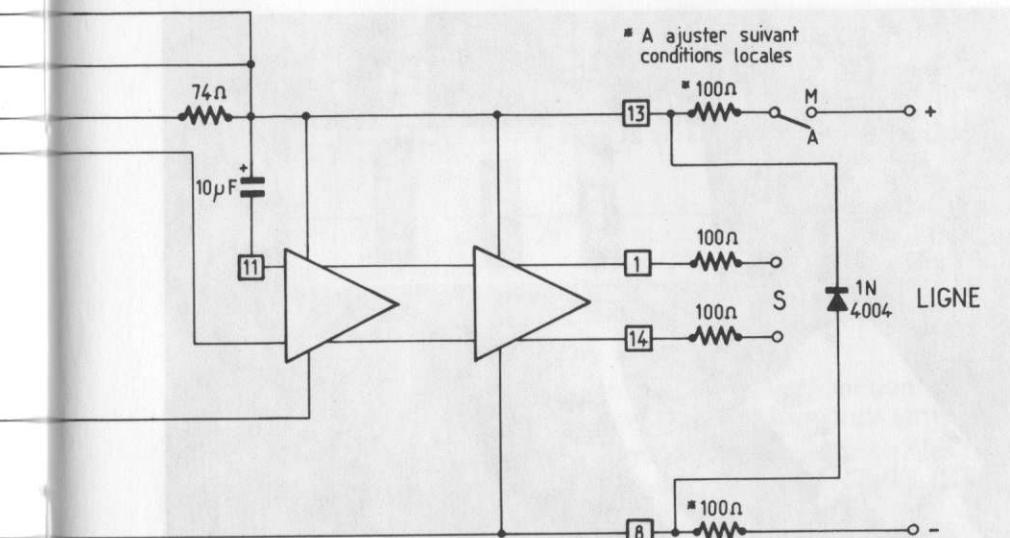
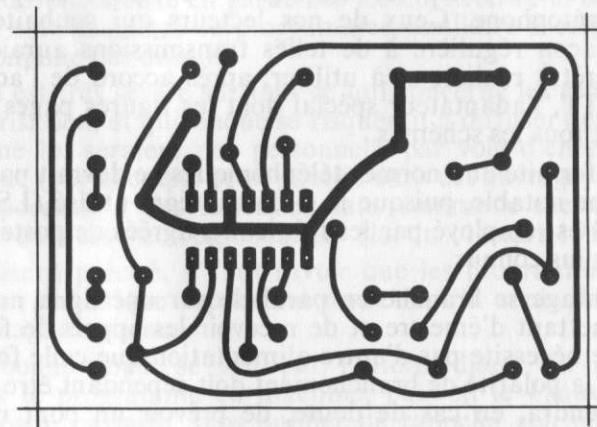
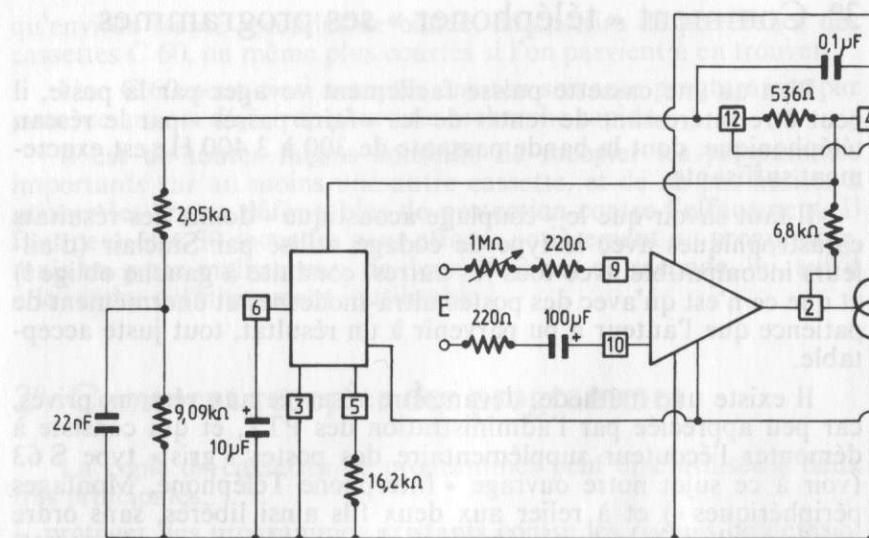
Il faut savoir que le « couplage acoustique » donne des résultats catastrophiques avec le type de codage utilisé par Sinclair (d'ailleurs incompatible avec tous les autres, conduite à gauche oblige !) et que ce n'est qu'avec des postes ultra-modernes et énormément de patience que l'auteur a pu parvenir à un résultat, tout juste acceptable.

Il existe une méthode, devant être réservée aux réseaux privés, car peu appréciée par l'administration des PTT, et qui consiste à démonter l'écouteur supplémentaire des postes « gris » type S 63 (voir à ce sujet notre ouvrage « Interphone Téléphone, Montages périphériques ») et à relier aux deux fils ainsi libérés, sans ordre préférentiel, soit la sortie écouteur du magnétophone « transmetteur », soit l'entrée ligne (prise DIN) du magnétophone « récepteur ». Moyennant un certain soin dans les réglages de niveau, on peut obtenir ainsi des résultats convenables. On peut également utiliser, côté réception, les classiques « capteurs téléphoniques » pour magnétophone. Ceux de nos lecteurs qui souhaiteraient se livrer de façon régulière à de telles transmissions auraient néanmoins intérêt à réaliser et à utiliser, après accord de l'administration des PTT, l'adaptateur spécial dont les figures pages suivantes fournissent tous les schémas.

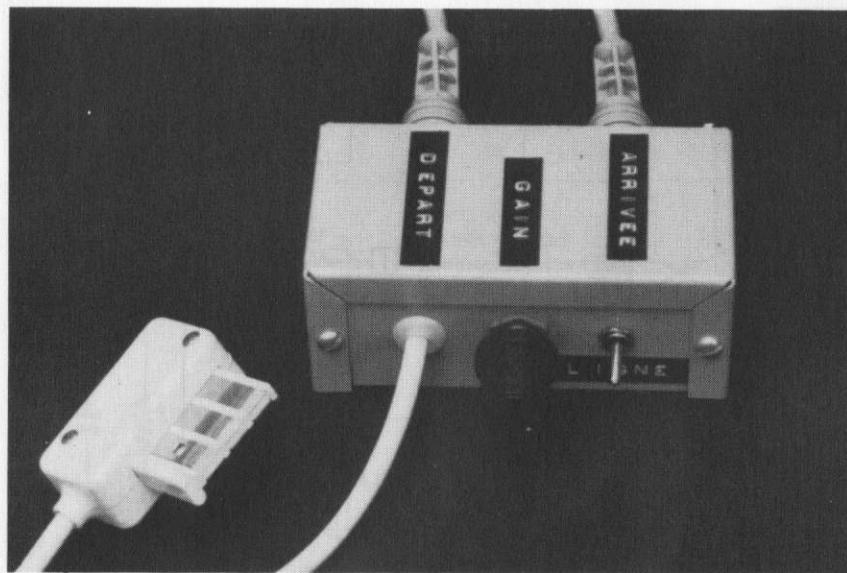
La conformité aux normes téléphoniques ne devrait pas soulever de problème notable, puisque le circuit intégré utilisé (LS 285A de SGS-Atès) est employé par les fabricants agréés de postes téléphoniques les plus connus.

Le montage se branche en parallèle sur une ligne munie d'un poste permettant d'émettre et de recevoir les appels de façon normale, et ne nécessite pas d'autre alimentation que celle fournie par le réseau. La polarité de branchement doit cependant être respectée et il conviendra, en cas de doute, de prévoir un pont redresseur entre la ligne et le montage. Là encore, nos lecteurs peu familiarisés avec les branchements téléphoniques auront intérêt à consulter notre ouvrage spécialisé, chez le même éditeur.

L'entrée comme la sortie sont prévues pour rejoindre la prise DIN « ligne » de tout magnétophone conforme aux normes euro-



Les valeurs de composants indiquées sont les valeurs exactes conduisant aux résultats les meilleurs. Il est cependant possible de se rabattre, sans inconvénient notable, sur les valeurs normalisées à 10 % les plus proches.



péennes, un réglage de niveau d'émission permettant de compenser les différences pouvant exister d'un modèle à l'autre.

Les magnétophones seront alimentés exclusivement par des piles tant que durera leur raccordement à l'adaptateur, et le niveau d'enregistrement du magnétophone « récepteur » sera ajusté de façon à enregistrer en légère saturation (aiguille dans la zone rouge du VU-mètre).

La qualité d'un tel enregistrement à distance n'atteint pas, bien sûr, celle de l'original, et il est vivement conseillé d'opérer une « régénération » par le biais d'une copie via le ZX-81.

L'interrupteur « marche-arrêt » de l'adaptateur ne sera basculé sur « marche » qu'une fois la communication téléphonique établie normalement, et on n'oubliera pas de le basculer de nouveau sur « arrêt » en fin de transmission afin de libérer la ligne. On pourra avoir intérêt à raccrocher le poste pendant la transmission proprement dite, de façon à éviter tout bruit parasite susceptible de fausser le programme.

Des transmissions par radio seraient également envisageables, mais l'ampleur des précautions à prendre dépasse largement le cadre de cette annexe à un recueil de programmes.

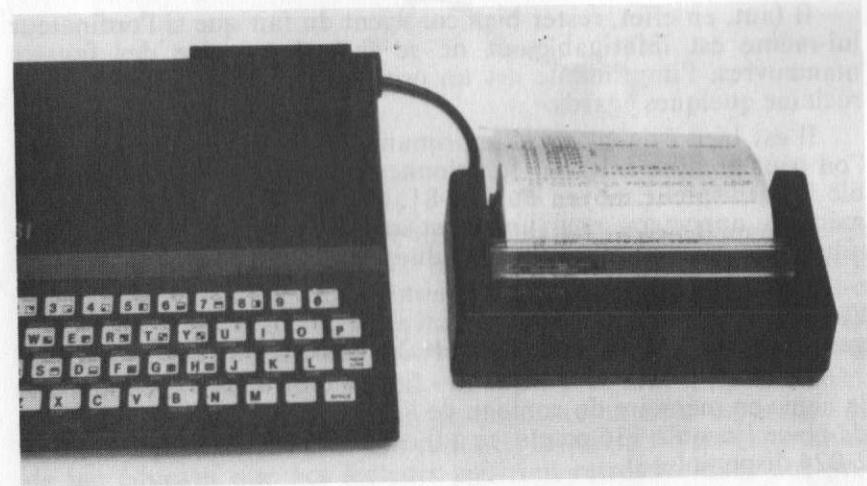
Utilisation de l'imprimante ZX-81

Lorsque nous avons entrepris la rédaction de cet ouvrage, seuls de très rares privilégiés disposaient de l'imprimante compatible avec leur ZX-81. Nos programmes exploitent donc uniquement les possibilités offertes par l'écran TV.

Ceux de nos lecteurs qui souhaiteraient faire fonctionner leur imprimante avec ces programmes devront noter les points suivants :

- L'imprimante ZX-81 ne peut être mise en service que par les trois instructions qui lui sont réservées, à savoir : LPRINT, LLIST, et COPY (1).
- Cette imprimante fonctionne sur du *papier à étincelage* (principe utilisé notamment sur les télécopieurs), une pointe très fine décrivant un *balayage* synchronisé avec l'avance du papier. Une tension peut être appliquée, sous contrôle du ZX-81, entre cette

(1) Sauf recours à la programmation en langage machine (USR).



pointe et le papier, créant ainsi une étincelle qui rend noir le papier initialement gris aluminium.

Ce fonctionnement est analogue à celui d'un écran TV, à ceci près que le retour sur une ligne déjà imprimée n'est pas possible.

- En conséquence de ceci, il n'est pas possible d'utiliser sans précaution l'instruction LPRINT en remplacement direct de PRINT. En particulier, les instructions PRINT AT, PRINT TAB, et les effets autorisés par l'emploi des point-virgules dans les instructions PRINT, et notamment à la fin de celles-ci, posent de réels problèmes d'adaptation.
- Chaque fois qu'on le pourra, on aura donc intérêt à laisser le programme travailler sur l'écran TV, même de façon fictive (c'est-à-dire, en l'absence de récepteur TV, dans le seul *fichier d'affichage* de la machine), puis de demander une copie sur papier du contenu de l'écran grâce à une instruction COPY.
- On pourra faire éditer à la machine des listings de programmes au moyen de l'instruction LLIST, mais celle-ci met à rude épreuve la mécanique délicate de l'imprimante. Le moteur, en effet, ralentit à la fin de chaque ligne pour repartir aussitôt à pleine vitesse. Si donc, on souhaite conserver un certain temps cette jolie pièce d'horlogerie, il est conseillé d'agir plutôt par le biais d'une instruction LIST, puis d'une commande COPY. Pour les programmes longs, on pourra procéder en plusieurs étapes en assignant à chaque instruction LIST le numéro de la ligne suivant immédiatement celle venant d'être imprimée.

Il faut, en effet, rester bien conscient du fait que si l'ordinateur lui-même est infatigable, et ne se formalise guère des fausses manœuvres, l'imprimante est un organe mécanique complexe qui réclame quelques égards.

Il est bien évident que l'imprimante, malgré tout le plaisir que l'on peut prendre à la voir fonctionner, n'est nullement indispensable à l'utilisateur moyen du ZX-81, auquel le module d'extension mémoire apportera, pour un investissement comparable, des possibilités considérablement plus étendues.

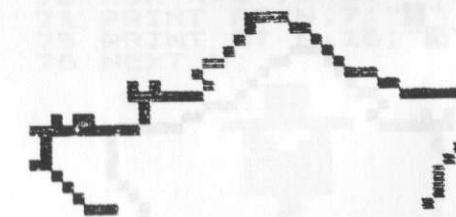
Cependant, ceux de nos lecteurs qui auraient préféré acquérir l'imprimante avant le module 16 K, pourront l'utiliser pour souager la mémoire du ZX-81 dans certains cas de pénurie.

En effet, il faut savoir que le « *fichier d'affichage* », c'est-à-dire la copie en mémoire du contenu de l'écran TV, occupe énormément de place (jusqu'à 736 octets, ce qui est beaucoup en comparaison de 1 024 disponibles !).

De ce fait, dès que l'on cherche à faire tracer au ZX-81 de base des graphismes relativement complexes, la panne de mémoire est bien souvent au rendez-vous. Il est cependant possible, dans certains cas, de laisser le fichier d'affichage vide, grâce à une sortie sur imprimante. Celle-ci travaillant ligne par ligne « oublie » toutes les lignes déjà imprimées, et n'exige ainsi qu'un « *fichier* » ou « *register* » de 32 octets.

Vous pourrez constater par vous-même les avantages du procédé en entrant le programme ci-dessous :

```
1 LET A$="778065 055 3 08 5  
5 05 7 02 2 77 81  
8 01 99 02 83 03 77 85 87 85 86  
86 56 06 66 06 66 06 67 06 67 06  
67 06 18 2 06 63 07 2 0  
66  
2 FOR N=1 TO LEN A$  
3 IF CODE A$(N) >=29 AND CODE  
A$(N) <=37 THEN GOTO 10  
4 IF A$(N) = " " THEN PRINT ""  
5 IF A$(N) = "0" THEN GOTO 8  
6 PRINT A$(N);  
8 NEXT N  
10 FOR F=1 TO VAL A$(N)  
11 PRINT " ";  
12 NEXT F  
13 LET N=N+1  
14 GOTO 3  
15 REM COPYRIGHT 1982
```



Attention, il est nécessaire de placer auparavant la machine en mode rapide en faisant FAST NEWLINE, et de frapper avec le plus grand soin le contenu de la chaîne A\$ (instruction 1). Celle-ci comprend à la fois des chiffres et des symboles graphiques, et exige une action sur la touche GRAPHICS à chaque fois que l'on passe de l'un des modes à l'autre.

Correctement entré, ce programme est capable de tracer un contour stylisé de la carte de France. Il est extrait d'un programme de jeu éducatif que nos lecteurs pourront retrouver dans une pro-

chaine publication, mettant à contribution les possibilités du module 16 K-octets.

Notre illustration montre que la capacité mémoire de 1 K-octet ne permet pas au programme d'aller jusqu'à son terme. Par suite d'un « engorgement » du fichier d'affichage, la machine est obligée de lancer un compte-rendu 4/10 avant même d'avoir tracé la moitié de la carte !

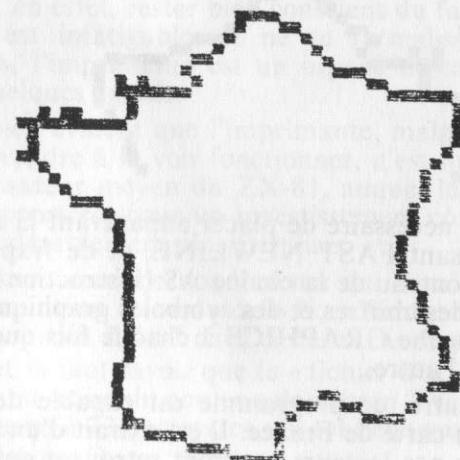
Modifions maintenant le programme au moyen d'instructions LPRINT, afin d'arriver à la forme suivante :

```

1 LET A$="778065 055 3 06 "
5 05 7 02 2 77 01 7
6 01 99 62 88 05 77 05 67 05 66
7 06 56 06 66 06 65 06 67 06 67 0
5 77 05 6 2 06 6 3 07 2 0
65

2 FOR N=1 TO LEN A$
3 IF CODE A$(N) >=29 AND CODE
A$(N) <=37 THEN GOTO 10
4 IF A$(N) = "0" THEN LPRINT ""
5 IF A$(N) = "0" THEN GOTO 6
6 LPRINT A$(N);
8 NEXT N
10 FOR F=1 TO VAL A$(N)
11 LPRINT " ";
12 NEXT F
13 LET N=N+1
14 GOTO 3
15 REM COPYRIGHT 1982

```



Lançons le programme par RUN NEWLINE, et maintenant, c'est sur l'imprimante que se dessine petit à petit notre hexagone, alors que l'écran reste vide.

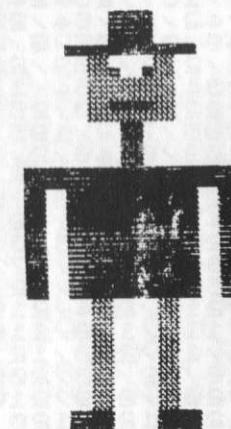
Grâce à cette économie de place mémoire, la carte peut maintenant être imprimée en totalité.

Le programme suivant est un autre exemple des possibilités graphiques étendues du ZX-81. Il nous a servi de base pour la mise au point d'un autre jeu éducatif que nos lecteurs pourront découvrir dans notre prochain ouvrage.

```

COPYRIGHT 1982
10 FOR N=4 TO 6
11 PRINT AT N,10;""
12 NEXT N
40 PRINT AT 7,10;""
50 PRINT AT 8,10;""
51 FOR L=10 TO 14
52 PRINT AT L,9;""
53 NEXT L
54 FOR L=15 TO 19
55 PRINT AT L,10;""
56 NEXT L
57 PRINT AT 20,9;""
58 PRINT AT 9,6;""
59 PRINT AT 9,9;""
60 PRINT AT 2,11;""
61 PRINT AT 4,11;""
62 PRINT AT 6,11;""
70 FOR N=9 TO 14
71 PRINT AT N,7;""
75 PRINT AT N,16;""
76 NEXT N

```



Ce programme occupe vraiment les tout derniers octets de la mémoire de base du ZX-81, au point que, si on tente de prévoir une instruction du type :

80 COPY

en supplément, notre sympathique petit bonhomme se retrouve avec des bras atrophiés, car le programme est obligé de s'arrêter à la ligne 76 pour cause de débordement du fichier d'affichage.

Il n'est cependant pas question ici de remplacer les instructions PRINT par des LPRINT, car, le chapeau étant tracé après les pieds, il faudrait que l'imprimante revienne sur des lignes déjà tracées, ce qui ne lui est pas possible. Une édition directe sur imprimante exigerait donc une refonte du programme.

Là encore, on ressent l'impérieuse nécessité de disposer d'une plus grande place mémoire, grâce au module 16 K.

Nous terminerons notre prise de contact avec l'imprimante en lui confiant un travail « d'espionnage », lequel nous fera apprécier à sa juste valeur le tour de force technique et économique que représente le ZX-81. Le programme que voici :

```

5 FAST
10 FOR N=0 TO 8191
20 LPRINT PEEK N;"";
30 NEXT N
40 REM COPYRIGHT 1982
211/253/1/255/127/195/203/3/42/2
2/64/34/24/64/24/70/167/194/241/
7/195/245/7/255/42/22/64/126/167
/192/0/0/205/73/0/24/247/255/255
/255/195/157/25/241/217/227/217/
201/197/42/20/64/229/195/136/28/
13/194/69/0/225/5/200/203/217/28
7/79/251/233/209/200/24/248/42/0
2/64/36/34/22/64/126/254/127/190
/24/246/225/110/253/117/0/237/190
3/2/64/205/7/2/195/196/28/255/0/
50/250/109/0/40/2/8/201/0/246/190
7/213/229/42/12/64/203/252/118/
11/253/221/233/63/61/48/59/38/0
/41/43/44/54/60/42/55/57/29/38/0
1/32/33/28/37/36/35/34/53/52/46/
58/62/118/49/48/47/45/0/27/58/0
/39/14/25/15/24/227/225/228/209/
226/192/217/224/219/221/117/218/
222/223/114/119/118/116/112/113/
11/17/16/13/220/121/28/21/22/210/
/12/26/18/19/23/205/206/193/128/
202/203/204/209/210/199/208/201/
207/64/128/128/128/128/128/1

```

```

20/120/120/120/194/211/196/214/0
13/120/212/198/197/208/128/128/0
5/215/65/8/10/9/138/137/129/130/
7/132/6/1/2/135/4/5/119/120/130/
3/131/139/145/144/141/134/120/14
6/149/150/136/143/11/139/38/185/
57/38/167/143/40/52/41/178/59/38
/177/49/42/179/56/48/179/40/52/1
84/57/38/179/38/56/179/38/40/184
/38/57/179/49/179/42/61/181/46/5

```

se charge de faire imprimer par la machine, séparés par des barres de fraction, les 8 192 octets que contient la mémoire ROM du ZX-81. Rappelons que ces 8 K-octets représentent le programme ineffaçable qui, rédigé en « langage machine », permet au ZX-81 de travailler en « langage évolué » ou BASIC. Sans lui, le ZX-81 ne serait qu'un assemblage de composants inutilisable. Si vous tenez à votre rouleau de papier, nous vous conseillons de modérer les ardeurs de ce programme en appuyant sur BREAK au bout de quelques instants, car, à raison de 9 octets en moyenne par ligne, le listing complet occuperait environ 2,73 mètres de papier imprimé en rangs serrés, et exigerait presque une heure pour s'édition, même en mode rapide, ce qui serait une bien rude épreuve pour l'imprimante, et une opération sans grand intérêt pour vous !

Dans le même ordre d'idées, il est facile de faire éditer sur imprimante la liste complète des caractères dont dispose la machine, afin de bien cerner les possibilités offertes. Il suffit pour cela de lancer ce dernier programme.

```

10 FOR N=0 TO 255
20 PRINT CHR$ N;
30 NEXT N
35 COPY
40 REM COPYRIGHT 1982
      "£$: ? () < = + - # / , . @ 123
456789ABCDEFGHIJKLMNOPQRSTUVWXYZ
RND INKEY$ PI?????????????????????????
?????????????????????????????????????
?????????????????????????????????????
?????????????????????????????????????
??. Q 123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ
STUUVWXYZ" AT TAB ?CODE VAL LEN SI
N COS TAN ASN ACS ATN LN EXP INT
SQR SGN ABS PEEK USA STA$ CHR$
NOT /* OR AND <=>= >< > THEN TO STE
P LPRINT LLIST STOP SLOW FAST NE
W SCROLL CONT DIM REM FOR GOTO G
OSUB INPUT LOAD LIST LET PAUSE N
EXT POKE PRINT PLOT RUN SAVE RAN
D IF CLS UNPLOT CLEAR RETURN COP
Y

```

Les branchements du ZX-81

Le micro-ordinateur ZX-81 est muni d'un jeu très complet de connecteurs permettant toute une variété de branchements, du plus simple au plus compliqué.

Il nous faut cependant signaler, à ceux de nos lecteurs qui seraient tentés « d'interfacer » leur ZX-81 avec d'autres équipements micro-informatiques, que pratiquement aucune compatibilité n'existe, si ce n'est à l'intérieur même de la gamme Sinclair. Peut-être est-ce voulu, d'ailleurs...

Par exemple, le format d'enregistrement des programmes sur cassette est exclusif à Sinclair, ce qui signifie qu'une cassette enregistrée par un ZX ne peut être exploitée que par un ZX et vice versa. Il n'existe aucun point commun entre le « standard » Sinclair et, par exemple, les diverses variantes du « Kansas City » largement utilisé par ailleurs.

Egalement, le jeu de caractères du ZX-81 est entièrement original, ce qui est en partie dû à l'existence de nombreux symboles graphiques inédits. Seulement, cette originalité se traduit par le fait qu'il n'existe pas la plus petite correspondance entre les octets affectés aux caractères Sinclair, et le code ASCII universellement utilisé par les autres constructeurs. Un exemple :

En ASCII, les lettres majuscules correspondent aux codes suivants :

A	B	C	D	...		X	Y	Z
Ø65	Ø66	Ø67	Ø68	...		Ø88	Ø89	Ø90

Alors que le code Sinclair est le suivant :

A	B	C	D	...	X	Y	Z
Ø38	Ø39	Ø40	Ø41	...	Ø61	Ø62	Ø63

De même, pour les chiffres, la situation est la suivante :

ASCII	0	1	2	...	7	8	9
	Ø48	Ø49	Ø50	...	Ø55	Ø56	Ø57

SINCLAIR	0	1	2	...	7	8	9
	Ø28	Ø29	Ø30	...	Ø35	Ø36	Ø37

Ceci, ajouté au fait que le jeu de caractères ZX contient beaucoup de signes qui lui sont propres, implique que de coûteux dispositifs d'interface seront à prévoir pour l'éventuel raccordement d'imprimantes autres que celle d'origine (qui travaille d'ailleurs, elle aussi, selon un principe des plus originaux).

Heureusement, d'autres possibilités s'offrent au bricoleur informaticien suffisamment averti :

1 – Le connecteur arrière

Le connecteur détrompé à 44 broches (pas de 2,54), prévu pour le branchement de l'imprimante et/ou du module 16K, contient tous les bus du microprocesseur Z80 A, autour duquel est bâti le ZX-81. Ceci ajouté au fait que le BASIC résident possède des possibilités de branchement vers le langage machine, ouvre des axes de recherche quasi illimités à quiconque maîtrise bien (et ce n'est pas si commode) le jeu d'instructions du Z80.

Sans en arriver là d'emblée, il est possible de se procurer une grande variété d'accessoires compatibles avec ce connecteur arrière, et notamment des *cartes d'entrée-sortie*, permettant de faire communiquer le ZX-81 avec le milieu extérieur, sans pour autant mettre sa vie en danger par des manipulations hésitantes sur les bus du microprocesseur.

Dans notre prochain ouvrage, nous montrerons ainsi comment le ZX-81 peut commander des appareils ménagers, jouer de la musique, et même... téléphoner ou faire office de centrale d'alarme !

2 – Les connecteurs latéraux

On trouve sur le côté du ZX-81 quatre douilles prévues respectivement pour le branchement de l'alimentation, du téléviseur et des entrées et sorties du magnétophone.

Au niveau de l'entrée d'alimentation, il est très facile de substituer à l'adaptateur secteur d'origine (dont on aura d'ailleurs intérêt à contrôler le câblage interne), toute autre alimentation continue fournit entre 7 et 11 V sous 1,2 A. On veillera à respecter la polarité, et on n'hésitera pas à prévoir un montage réducteur de tension lorsqu'un risque existe de dépasser 11 V (notamment en cas d'utilisation sur batterie de voiture, pour laquelle un fusible de 2 A est absolument indispensable).

On peut envisager une alimentation par cinq grosses piles « torche » alcalines, mais une batterie rechargeable de 7,5 V (six éléments CdNi) sera très vite rentabilisée.

Le modulateur UHF attaquant la sortie « antenne » peut très facilement être adapté à la plupart des téléviseurs UHF 625 lignes, même de provenance étrangère. On peut même le « contourner » pour attaquer directement un *moniteur* vidéo.

Les entrées/sorties pour magnétophone peuvent être utilisées à d'autres fins, mais presque uniquement par le biais de programmes écrits en langage machine.

Quelques commentaires s'imposent par ailleurs au sujet du branchement d'un magnétophone : il faut utiliser une entrée MICRO (et non radio, PU, ligne ou autre), et une sortie HAUT-PARLEUR ou, à la rigueur, écouteur (et non ligne, ampli, etc.). Certains magnétophones obligent à remplacer les jacks 3,5 mm

d'origine par des fiches DIN. En règle générale, il faut utiliser les broches 2 (masse) et 1 (signal) de la fiche « micro », tandis que le blindage du cordon HP rejoindra la broche plate (–) de la fiche correspondante. De telles transformations réclament du soin, certaines erreurs pouvant endommager le magnétophone.

Certains magnétophones se révèlent rebelles à l'association avec un ZX-81. Ce problème assez rare est souvent confondu avec celui, beaucoup plus fréquent, d'incompatibilité du magnétophone avec les cassettes d'édition ou d'emprunt.

Lorsque seules des cassettes de provenance extérieure posent des problèmes de chargement, dans neuf cas sur dix, le défaut se situe au niveau du réglage de la tête de lecture du magnétophone (petite vis parfois très bien cachée). Il faut rectifier ce réglage de façon à obtenir une sonorité aussi sèche que possible (maximum d'aiguës).

En pratique, lorsque l'on possède une « bibliothèque » de programmes de provenances diverses, il est monnaie courante d'avoir à refaire le réglage pour chaque cassette ! D'où notre conseil de « sacrifier » un magnétophone à un usage exclusivement informatique.

NOTE POUR LA TROISIÈME ÉDITION :

Lors de la mise sous presse de la première édition de ce livre, la totalité des connaissances informatiques de l'auteur se trouvaient réunies dans ces 128 pages.

Depuis, bien des progrès ont pu être accomplis vers une meilleure connaissance de cette merveilleuse petite machine qu'est le ZX-81.

Patrick GUEULLE peut ainsi proposer maintenant à ses lecteurs de se lancer avec lui à la conquête de la programmation 16 K, des interfaces (du commerce ou à construire soi-même), du LANGUAGE MACHINE et d'un foule d'autres choses, dans ses deux derniers ouvrages :

Maîtrisez votre ZX-81

Montages périphériques pour ZX-81

A très bientôt donc !

Gagnez du temps, EVITEZ LES ERREURS DE FRAPPE

*Procurez-vous la cassette
des programmes de cet ouvrage*

Les programmes dont les listings figurent dans cet ouvrage ont été enregistrés sur cassette.

Votre libraire habituel pourra vous la commander.

- *Cette cassette vous permet de charger sur le ZX-81 l'un ou l'autre de ces programmes, en quelques dizaines de secondes.*
- *De plus, cette façon de faire élimine toute possibilité d'erreur lors de l'entrée des programmes par le clavier.*

Si vous éprouvez des difficultés à vous procurer cette cassette, l'éditeur pourra vous transmettre l'adresse d'un revendeur assurant la vente par correspondance (joindre 1 enveloppe timbrée).

Il vous suffit de remplir et expédier le bon au dos.

- *Vous pouvez également utiliser ce bon pour nous faire connaître vos remarques et suggestions sur les domaines dans lesquels vous souhaiteriez voir développer des programmes pour ZX-81.*

BON

à découper ou à photocopier, et à retourner, rempli, à l'éditeur :
E.T.S.F., 2 à 12, rue de Bellevue, 75940 PARIS CEDEX 19

Nom : _____ Prénom : _____ Age : _____ Profession : _____

N° _____ Rue : _____ Ville : _____

Code postal : _____ Bureau distributeur : _____

Pour recevoir l'adresse d'un revendeur assurant la vente par correspondance de la cassette « PILOTEZ VOTRE ZX-81 », cocher cette case et joindre une enveloppe timbrée pour réponse.

• Je souhaiterais trouver, dans l'un de vos prochains ouvrages, des programmes se rapportant aux domaines suivants :

• Remarques et suggestions de toute nature :

(Si la place manque, joindre éventuellement un feuillet séparé).



PILOTEZ VOTRE ZX-81

Dans cet ouvrage l'auteur découvre avec vous le ZX-81 de SINCLAIR.

Des programmes originaux, qui peuvent être utilisés directement sans notions de programmation, mettent en œuvre de nombreuses **applications « domestiques »** de l'informatique. L'étude progressive des **instructions du Basic** et un tour complet des possibilités du ZX-81 dans sa version de base vous aideront à créer vos propres programmes.

- Prise de contact avec le ZX-81
- Jeux et divertissements
- Mathématiques
- Calculs pratiques
- Fonctions graphiques
- Fichiers et répertoires
- Annuaire électronique
- Ordinateur de bord automobile
- Echanges de programmes



*K7 N° 1 : P. GUEULLE
PILOTEZ VOTRE ZX-81*

Gagnez du temps, évitez les erreurs de frappe, découvrez de nouvelles possibilités à votre ZX-81.

Les programmes qui figurent dans cet ouvrage ont été enregistrés sur cassette. Commandez-la à votre revendeur.