



EXERCICES EN BASIC POUR AMSTRAD



MAURICE CHARBIT

***EXERCICES EN BASIC
POUR AMSTRAD***

Autres ouvrages relatifs à l'Amstrad CPC 464, 664 ou 6128 :

- La découverte de l'Amstrad - Daniel-Jean David
- Amstrad en famille - Jean-François Sehan
- Super jeux Amstrad - Jean-François Sehan
- 102 programmes pour Amstrad - Jacques Deconchat, adapté par Alain Lavenir
- Basic Amstrad, 1. Méthodes pratiques - Jacques Boisgontier et Bruno Césard
- Basic Amstrad, 2. Programmes et fichiers - Jacques Boisgontier
- Basic plus, 80 routines sur Amstrad - Michel Martin
- Assembleur de l'Amstrad - Marcel Henrot
- Clefs pour Amstrad, 1. Système de base - Daniel Martin
- Clefs pour Amstrad, 2. Système disque - Daniel Martin et Philippe Jadoul
- Périphériques et gestion de fichiers sur Amstrad - Daniel-Jean David
- Photographie sur Amstrad et Apple II - Pierrick Moigneau et Xavier de la Tullaye
- Super générateur de caractères sur Amstrad - Jean-François Sehan
- I.A. sur Amstrad CPC - Thierry et Eric Lévy-Abégnoli
- Amstrad 3-D - Jean-Pierre Petit

Ouvrages d'initiation :

- Visa pour l'informatique — Jean-Michel Jégo
- Visa pour le Basic — Jean-Michel Jégo
- Mon ordinateur - Jean-Claude Barbance
- Le logotron informatique - Jean-Pierre Petit
- Les mots de la micro - Alan Freedman, adapté par Bernard Sauteur

A paraître :

- Clefs pour Amstrad PCW - Didier Roy et J.-J. Weyer

Pour tout problème rencontré dans les ouvrages P.S.I.
vous pouvez nous contacter au numéro ci-dessous :

Numéro Vert/Appel Gratuit en France

05 21 22 01

(Composer tous les chiffres, même en région parisienne)

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'article 40).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles 425 et suivants du Code Pénal.

© Éditions du P.S.I. - B.P. 86 - 77402 Lagny/Marne cedex
1986

ISBN 2-86595-271-1

EXERCICES EN BASIC POUR AMSTRAD

MAURICE CHARBIT



**ÉDITIONS DU P.S.I.
1986**

Maurice CHARBIT est ingénieur et enseignant à l'Ecole Nationale Supérieure des Télécommunications.

Il est l'auteur de :

- La découverte de l'Alice
- Exercices en Basic pour Alice et Alice 90

et, en collaboration avec Dominique SCHRAEN, de :

- La découverte des MO5 et TO7-70
- Exercices pour MO5 et TO7-70

SOMMAIRE

<i>INTRODUCTION</i>		6
<i>PRÉSENTATION DES EXERCICES</i>		7
<i>PREMIÈRE PARTIE : ÉNONCÉS ET ANALYSES</i>		9
<hr/>		
CHAPITRE I	Affichage, calculs simples, tirage de nombres aléatoires	11
<hr/>		
CHAPITRE II	Calculs, manipulations de tableaux numériques (recherche, classement, insertion)	25
<hr/>		
CHAPITRE III	Traitements de chaînes de caractères (recherche, insertion, suppression de caractères), traitements de fichiers	45
<hr/>		
CHAPITRE IV	Dessins, histogrammes, graphiques, tracés de fonctions, programmes musicaux	63
<hr/>		
CHAPITRE V	Jeux graphiques (tir à la cible, télécran, glouton, loto sportif, loto, machine à sous, pendu, calcul mental, mastermind, leçon de géographie, tours d'Hanoi, répertoire téléphonique)	77
<hr/>		
<i>DEUXIÈME PARTIE : SOLUTION DES EXERCICES</i>		93

INTRODUCTION

CE LIVRE S'ADRESSE A TOUS CEUX QUI ONT UNE CONNAISSANCE ÉLÉMENTAIRE DU BASIC ET SONT DÉSIREUX DE LA METTRE EN PRATIQUE SUR DES EXEMPLES CONCRETS. IL PERMETTRA AINSI DE MAITRISER LE JEU D'INSTRUCTIONS PAR LA RÉOLUTION D'EXERCICES CLASSIQUES DE PROGRAMMATION.

CE LIVRE COMPREND UNE PREMIÈRE PARTIE DONNANT LES ÉNONCÉS DES EXERCICES ET UNE SECONDE PARTIE PROPOSANT DES SOLUTIONS COMMENTÉES.

LA PREMIÈRE PARTIE SE COMPOSE DE CINQ CHAPITRES. LE PREMIER CHAPITRE EST COMPOSÉ D'EXERCICES SIMPLES QUI METTENT PLUS PARTICULIÈREMENT L'ACCENT SUR UNE INSTRUCTION DE FAÇON A EN COMPRENDRE PLEINEMENT L'USAGE. LE SECOND CHAPITRE PRÉSENTE UN RECUEIL D'EXERCICES SUR L'UTILISATION DU BASIC POUR RÉSOUDRE DES PROBLÈMES DE TYPES NUMÉRIQUE ET ARITHMÉTIQUE, TANDIS QUE LE TROISIÈME CHAPITRE EST CONSACRÉ A LA MANIPULATION DES CHAINES DE CARACTÈRES. DANS CES DEUX CHAPITRES, UN SOIN TOUT PARTICULIER EST MIS DANS LA PRÉSENTATION DE L'AFFICHAGE DES RÉSULTATS A L'ÉCRAN.

LE QUATRIÈME CHAPITRE PROPOSE DES EXERCICES QUI UTILISENT DE FAÇON IMPORTANTE LES POSSIBILITÉS GRAPHIQUES ET SONORES DE VOTRE ORDINATEUR.

ENFIN LE CINQUIÈME CHAPITRE EST COMPOSÉ DE PROBLÈMES PLUS LONGS, NÉCESSITANT LA MAITRISE DES NOTIONS INTRODUITES PRÉCÉDEMMENT.

POUR TIRER LE MEILLEUR PROFIT DE CES EXERCICES, IL EST CONSEILLÉ D'ESSAYER DE RÉSOUDRE LE PROBLÈME UNIQUEMENT A PARTIR DE L'ÉNONCÉ, SANS SE RÉFÉRER, DANS UN PREMIER TEMPS, A L'ANALYSE PROPOSÉE.

LA SOLUTION DONNÉE POUR LES EXERCICES DANS LA SECONDE PARTIE DU LIVRE, SOUS FORME D'UN LISTING COMMENTÉ, N'EST ÉVIDEMMENT PAS LA SEULE POSSIBLE. IL EST D'AILLEURS RECOMMANDÉ AU LECTEUR DE LA COMPARER DE FAÇON CRITIQUE AUX SOLUTIONS QU'IL AURA TROUVÉES.

PRÉSENTATION DES EXERCICES

Les **énoncés** sont formés, en général, de trois rubriques :

- *Problème* : donnant l'objectif du programme,
- *Entrée* : décrivant les données que l'utilisateur devra fournir au cours de l'exécution du programme,
- *Sortie* : indiquant ce que le programme affichera sur l'écran.

L'**analyse** indique comment poser puis traduire le problème énoncé, en langage BASIC et est complétée, chaque fois qu'il est nécessaire, par un organigramme détaillé.

Les **solutions** comprennent le listing du programme, la liste des variables qu'il utilise et l'explication détaillée des lignes importantes du programme (lorsque l'explication concerne un groupe de lignes consécutives, seuls les numéros de la première et de la dernière lignes sont indiqués, séparés par un tiret).

Terminons cette présentation par quelques conseils pratiques :

- Avant de taper un programme ou d'en charger un à partir du lecteur de cassettes, initialisez votre AMSTRAD en appuyant simultanément sur les touches SHIFT CTRL ESC.
- De façon générale, n'hésitez pas en début de programme à faire une initialisation complète (encre, son, fenêtre).

EXERCICES EN BASIC POUR AMSTRAD

- *Si dans un programme, des tests sont effectués sur des chaînes de caractères entrées au clavier, vérifiez lors de l'exécution que le clavier est en mode majuscule ou minuscule comme le nécessitent ces comparaisons.*
- *Lors de l'utilisation de l'instruction SPEED KEY, vous risquez de rencontrer des difficultés au clavier, si la sortie du programme s'effectue sans rétablir une vitesse suffisamment lente. Pour éviter cela, tapez*

KEY 135, "SPEED KEY 10,3" + CHR\$(13)

La touche 7 du pavé numérique vous permet alors de revenir à un usage plus pratique du clavier.

- Pour des facilités d'affichage, quelques programmes bouclent : pour sortir, tapez deux fois sur la touche ESC.

*PREMIÈRE
PARTIE*

**ÉNONCÉS
ET
ANALYSES**

EXERCICES EN BASIC POUR AMSTRAD

Exercices	Enoncé et analyse	Solution	Exercices	Enoncé et analyse	Solution
1.1	11	94	3.6	48	152
1.2	11	95	3.7	49	153
1.3	12	96	3.8	51	155
1.4	12	97	3.9	52	157
1.5	13	98	3.10	53	159
1.6	13	98	3.11	53	160
1.7	14	99	3.12	55	162
1.8	16	100	3.13	56	164
1.9	16	101	3.14	57	165
1.10	16	102	3.15	58	167
1.11	17	103	3.16	59	168
1.12	18	104	3.17	59	170
1.13	18	105	3.18	60	172
1.14	19	106	3.19	61	173
1.15	19	107	3.20	62	174
1.16	20	108	4.1	63	177
1.17	20	109	4.2	65	178
1.18	21	110	4.3	65	180
1.19	21	112	4.4	66	182
1.20	21	114	4.5	67	185
1.21	22	115	4.6	67	187
1.22	23	116	4.7	68	189
1.23	23	118	4.8	69	191
1.24	23	118	4.9	69	193
1.25	24	119	4.10	70	195
2.1	25	121	4.11	70	196
2.2	27	122	4.12	71	198
2.3	28	123	4.13	72	200
2.4	29	124	4.14	72	201
2.5	29	126	4.15	73	205
2.6	30	128	4.16	74	206
2.7	31	131	4.17	75	209
2.8	33	132	4.18	76	211
2.9	36	135	5.1	78	214
2.10	36	136	5.2	78	216
2.11	37	139	5.3	79	219
2.12	39	140	5.4	81	221
2.13	41	142	5.5	82	225
2.14	42	144	5.6	83	228
2.15	42	145	5.7	84	231
2.16	43	146	5.8	85	232
3.1	46	148	5.9	86	235
3.2	46	149	5.10	87	238
3.3	46	150	5.11	89	242
3.4	47	151	5.12	90	246
3.5	47	151			

CHAPITRE I

AFFICHAGE, CALCULS SIMPLES, TIRAGE DE NOMBRES ALÉATOIRES

EXERCICE 1.1 (PRINT)

PROBLÈME : affichage des initiales et de la longueur (nombre de lettres) du nom d'une personne.

ENTRÉE : le nom et le prénom de la personne.

SORTIE : les initiales et la longueur du nom.

EXERCICE 1.2 (INPUT)

PROBLÈME : à partir de l'année de naissance d'une personne, afficher son âge à l'intérieur d'un texte.

ENTRÉE : le prénom et l'année de naissance d'une personne.

SORTIE : affichage du texte :

**Hé bien bonjour (prénom)
Je ne répéterai à personne
que tu as aujourd'hui (âge) ans.**

ANALYSE

Il faut :

- 1) demander le prénom de la personne.
- 2) demander son année de naissance.
- 3) afficher les lignes de texte (en calculant l'âge).

EXERCICE 1.3 (STRING\$)

PROBLÈME : afficher une ligne et une colonne d'étoiles sur l'écran.

ENTRÉE : un nombre L compris entre 1 et 40, un nombre C compris entre 1 et 25.

SORTIE : affichage d'une ligne de 40 étoiles sur la Lième ligne et d'une colonne de 25 étoiles sur la Cième colonne.

EXERCICE 1.4 (INKEY\$)

PROBLÈME : afficher le code ASCII des touches frappées par l'utilisateur (utiliser la fonction INKEY\$).

ENTRÉE : une touche du clavier.

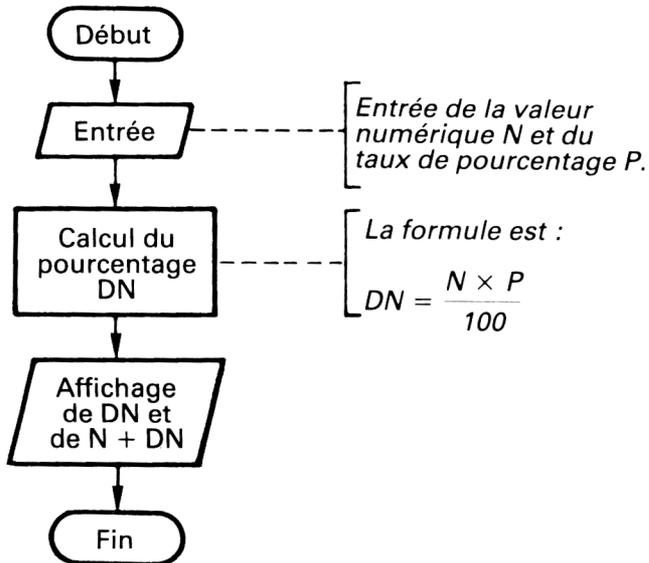
SORTIE : le code ASCII de la touche.

EXERCICE 1.5 (opérations arithmétiques)

PROBLÈME : calcul du pourcentage P d'une valeur numérique N et de la valeur obtenue par application de ce pourcentage.

ENTRÉE : la valeur numérique et le taux de pourcentage.

SORTIE : le pourcentage de la valeur numérique et la valeur finale.

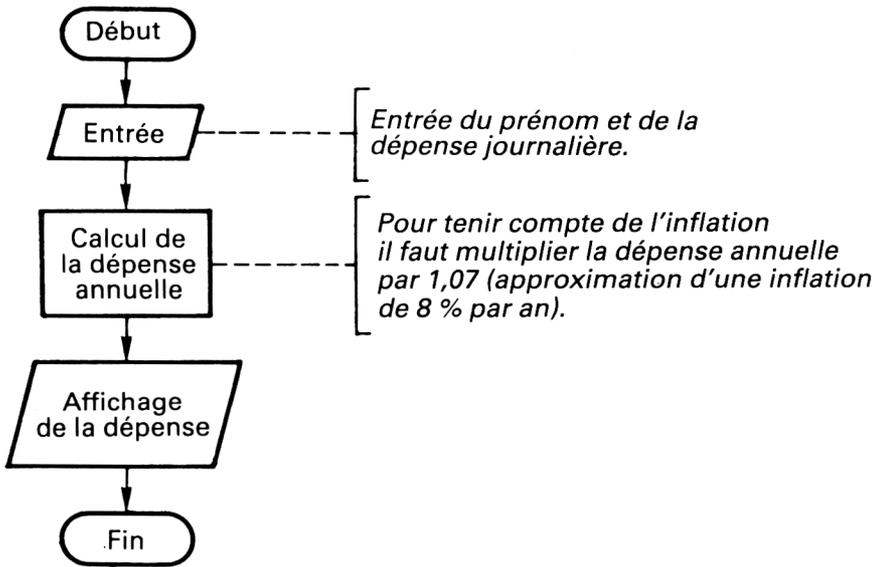
ANALYSE**EXERCICE 1.6** (LOCATE)

PROBLÈME : calcul de la dépense annuelle approximative en nourriture, à partir de la dépense quotidienne moyenne (en tenant compte de l'inflation).

ENTRÉE : dépense quotidienne en nourriture.

SORTIE : la dépense annuelle.

ANALYSE



EXERCICE 1.7 (IF...THEN)

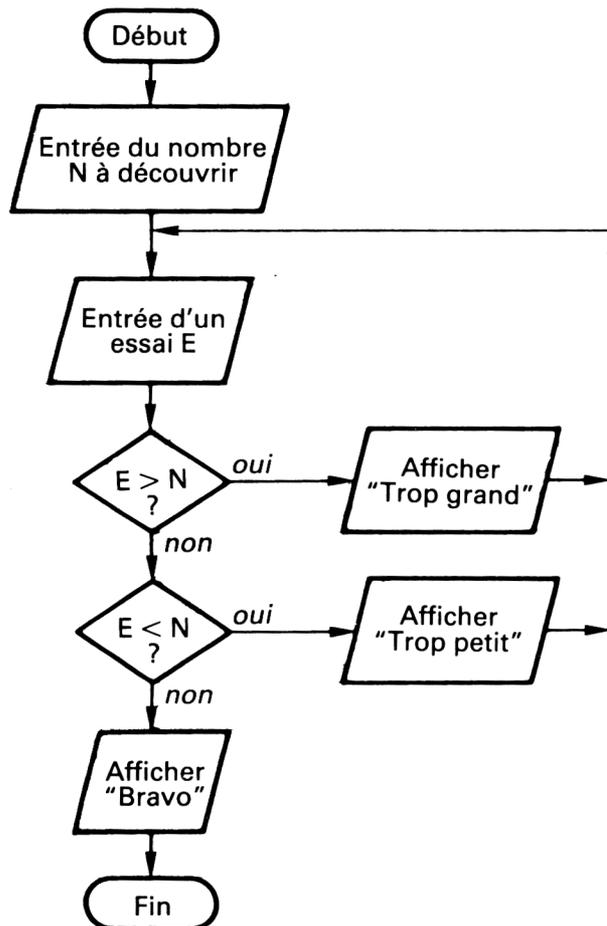
PROBLÈME : l'utilisateur doit découvrir par essais successifs un nombre entier entré auparavant par une autre personne.

ENTRÉE : le nombre à découvrir (compris entre 0 et 1000) puis les essais.

SORTIE : un des messages : "Trop grand", "Trop petit" ou bien "Bravo, vous avez trouvé" selon l'essai.

ANALYSE

Après l'entrée de l'essai, celui-ci est testé par rapport au nombre à deviner. S'il lui est supérieur ou inférieur, le message correspondant est affiché et un nouvel essai est proposé. Sinon le message "Bravo, vous avez trouvé" s'affiche et le programme s'arrête...



EXERCICE 1.8 (RND)

PROBLÈME : simulation du lancer d'un dé.

SORTIE : un nombre entier aléatoire compris entre 1 et 6.

ANALYSE

RND(1) donne un nombre aléatoire entre 0 compris et 1 non compris ; multiplié par 6 il est entre 0 compris et 6 non compris ; il ne reste plus qu'à en prendre la valeur entière (INT) et lui ajouter 1 pour obtenir un nombre entier entre 1 compris et 6 compris.

Ce qui donne en BASIC : $\text{INT}(\text{RND}(1) * 6) + 1$.

EXERCICE 1.9 (FOR/NEXT)

PROBLÈME : simulation du lancer de plusieurs dés.

ENTRÉE : un nombre de dés à jeter (entre 1 et 8).

SORTIE : la somme des nombres sortis.

EXERCICE 1.10 (RND)

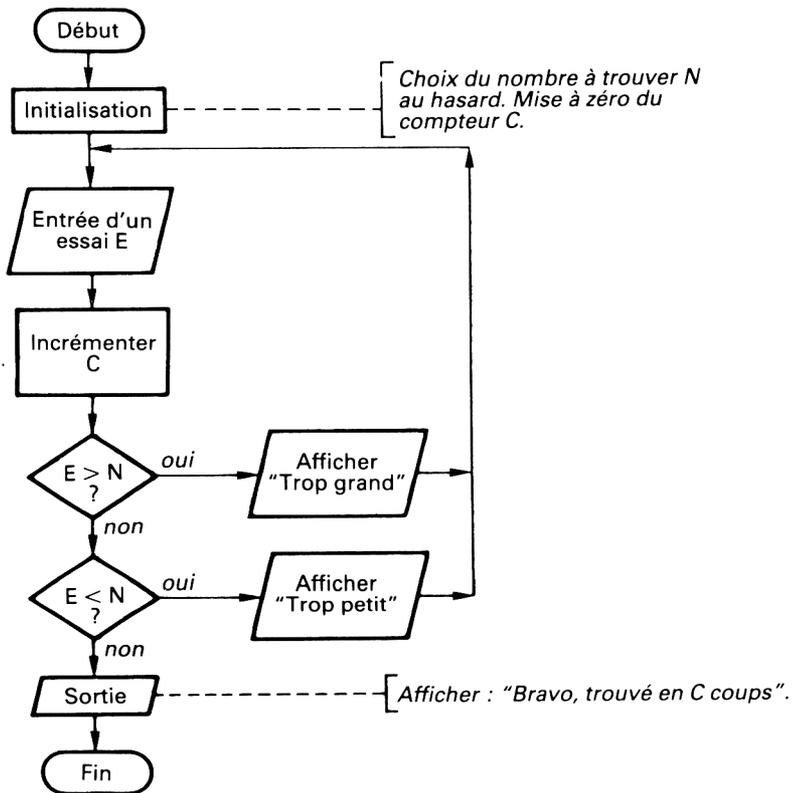
PROBLÈME : version améliorée de l'exercice 1.7. La machine choisit au hasard le nombre à trouver et compte le nombre d'essais nécessaires au joueur pour arriver à la solution.

ENTRÉE : les essais.

SORTIE : un des messages : "Trop grand", "Trop petit" ou "Bravo, vous avez trouvé en C coups", selon l'essai.

ANALYSE

Le nombre d'essais est contenu dans la variable C. Pour cela, elle est mise à zéro au début du programme puis, à chaque essai, elle est augmentée de 1 (elle est "incrémentée"). Une telle variable est appelée "compteur".

**EXERCICE 1.11** (RANDOMIZE, TIME)

PROBLÈME : jeu des lettres. Le joueur choisit une voyelle ou une consonne au hasard et cela 9 fois.

ENTRÉE : choix du joueur - C pour consonne, V pour voyelle.

SORTIE : affichage des 9 lettres.

ANALYSE

Le programme tire un nombre aléatoire, compris entre 1 et 6 si le joueur choisit une voyelle et entre 1 et 20 si le joueur choisit une consonne. Puis ce nombre N est utilisé pour extraire de la chaîne des voyelles ou de la chaîne des consonnes le Nième caractère.

EXERCICE 1.12 (PI)

PROBLÈME : écrire un programme qui calcule le périmètre et la surface d'un cercle de rayon R.

ENTRÉE : le rayon du cercle.

SORTIE : le périmètre et la surface du cercle.

ANALYSE

On rappelle que le périmètre d'un cercle est donné par la formule

$$P = 2 \times 3,14159265 \times R$$

et que sa surface est donnée par

$$S = 3,14159265 \times R \times R$$

Le nombre 3,14159265 qui intervient dans les deux formules peut être obtenu directement en utilisant l'instruction PI.

EXERCICE 1.13 (PRINT USING)

PROBLÈME : écrire un programme qui demande le nom d'un pays, sa superficie et sa population, qui calcule ensuite la densité et qui affiche le résultat.

ENTRÉE : nom du pays, superficie, population.

SORTIE : les résultats seront affichés sous la forme suivante :

Pays	Densité (hab/km²)
FRANCE	96.8921

EXERCICE 1.14 (MIN, MAX)

PROBLÈME : trouver le plus petit élément et le plus grand élément d'une liste de nombres.

ENTRÉE : une liste de nombres.

SORTIE : le plus petit élément et le plus grand élément de la liste.

ANALYSE

Le plus petit élément est trouvé par l'instruction MIN ; le plus grand élément est trouvé par l'instruction MAX. On opère alors de la façon suivante :

- 1) On met dans MI le plus petit élément du tableau
- 2) Pour chacun des éléments du tableau, on met dans MI la plus petite des deux valeurs, calculée par l'instruction MIN et portant sur MI et cet élément du tableau.

Pour trouver l'élément le plus grand du tableau, on opère de la même manière avec l'instruction MAX.

EXERCICE 1.15 (UPPER\$, LOWER\$)

PROBLÈME : affichage du nom et du prénom d'une personne en lettres minuscules, les initiales étant écrites en lettres majuscules.

ENTRÉE : entrer au clavier, le nom et le prénom d'une personne, écrits en lettres majuscules ou en lettres minuscules.

SORTIE : affichage du nom et du prénom.

ANALYSE

L'instruction LEFT\$ permet d'extraire les initiales du nom et du prénom.

EXERCICE 1.16 (ON ERROR, ERR, ERL)

PROBLÈME : écrire un programme qui, en cas d'erreur, exécute un sous-programme d'affichage du numéro de l'erreur et du numéro de la ligne où l'erreur s'est produite. Le programme principal effectue la division de A par B.

ENTRÉE : deux nombres A et B.

SORTIE : le résultat de la division de A par B ou lors d'une erreur (par exemple si $B = 0$) le numéro de l'erreur et le numéro de la ligne où l'erreur s'est produite.

ANALYSE

L'instruction ON ERROR GOTO permet en cas d'erreur de poursuivre à la ligne indiquée par GOTO. Le programme principal lit, par l'instruction INPUT, les deux nombres A et B. Si B est différent de 0, le programme affiche le résultat de la division de A par B. Si B est égal à 0, la division est impossible et l'erreur provoque alors la poursuite du programme à la ligne indiquée.

EXERCICE 1.17 (WINDOW)

PROBLÈME : construire 6 fenêtres d'affichage de couleurs alternées rouge et bleu. Entrer le numéro de l'une des fenêtres et afficher ce numéro au centre de cette fenêtre.

ENTRÉE : un numéro compris entre 1 et 6.

SORTIE : affichage du numéro de la fenêtre au centre de cette fenêtre.

ANALYSE

L'instruction WINDOW définit des fenêtres d'affichage ou canaux, qui sont accessibles de façon directe par les instructions INPUT et PRINT.

EXERCICE 1.18 (*SPEED INK*)

PROBLÈME : affichage clignotant d'un prénom en lettres de taille double.

ENTRÉE : un prénom de personne.

SORTIE : affichage clignotant du prénom en lettres de taille double, en utilisant le mode 0.

EXERCICE 1.19 (*mode transparent*)

PROBLÈME : afficher le nom et le prénom d'une personne en les soulignant.

ENTRÉE : le nom et le prénom d'une personne.

SORTIE : le nom et le prénom soulignés.

ANALYSE

On peut utiliser pour souligner le trait de soulignement ou encore le tiret de soustraction. Dans tous les cas il faut, au préalable, déterminer la longueur du mot à souligner. Pour cela, utiliser la fonction LEN. Si l'on désire, en plus, que le trait se trouve très près du mot à souligner, il faut passer en mode transparent et utiliser l'instruction PRINT CHR\$(22)+CHR\$(1).

EXERCICE 1.20 (*SOUND*)

PROBLÈME : jouer une note de la gamme 0, en tapant DO pour la note do, RE pour la note ré, MI pour la note mi, FA pour la note fa, SO pour la note sol, LA pour la note la, SI pour la note si.

ENTRÉE : l'une des réponses DO, RE, MI, FA, SO, LA, SI.

SORTIE : la note correspondante de la gamme 0.

ANALYSE

Créer une chaîne de caractères correspondant à toutes les réponses possibles. Puis se servir de la position de la réponse dans cette chaîne pour déterminer la note à jouer (instruction INSTR).

EXERCICE 1.21 (ENT, ENV)

PROBLÈME : jouer le morceau de musique suivant (Frère Jacques) :



ANALYSE

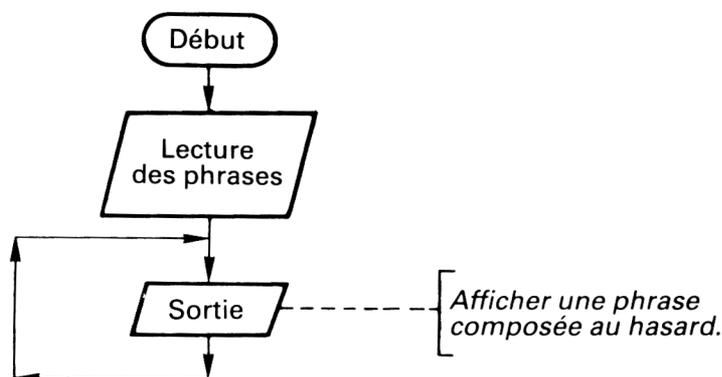
Le morceau peut être enregistré sous forme de DATA.

EXERCICE 1.22 (*manipulation de tableaux*)

PROBLÈME : créer des phrases, composées aléatoirement à partir de noms, verbes et compléments.

ENTRÉE : une liste de noms, verbes et compléments (en DATA).

SORTIE : des phrases aléatoires.

ANALYSE**EXERCICE 1.23** (*ASC, ZONE*)

PROBLÈME : affichage de tous les caractères ASCII de code compris entre 32 et 255.

EXERCICE 1.24 (*ORIGIN, CLG*)

PROBLÈME : dessiner un damier de 100 cases à 2 couleurs alternées.

SORTIE : affichage du damier.

ANALYSE

L'instruction `ORIGIN` permet de définir des zones d'affichage graphique, qui peuvent être coloriées par l'instruction `CLG`.

EXERCICE 1.25 (*SYMBOL*)

PROBLÈME : écrire un programme qui fait se déplacer une balle dans une zone rectangulaire. Les rebonds sur les bords du rectangle seront accompagnés d'un beep musical.

ANALYSE

Pour dessiner la balle, utiliser l'instruction `SYMBOL`. La balle peut se déplacer par pas de un, sur la gauche, sur la droite, vers le haut ou vers le bas. Quand la balle atteint l'un des bords verticaux, on permute les déplacements gauche et droit. Quand la balle atteint l'un des bords horizontaux, on permute les déplacements haut et bas.

CHAPITRE II

CALCULS, MANIPULATIONS DE TABLEAUX NUMÉRIQUES (Recherche, classement, insertion)

EXERCICE 2.1

PROBLÈME : trouver le nombre de valeurs positives ou nulles dans une liste de dix valeurs.

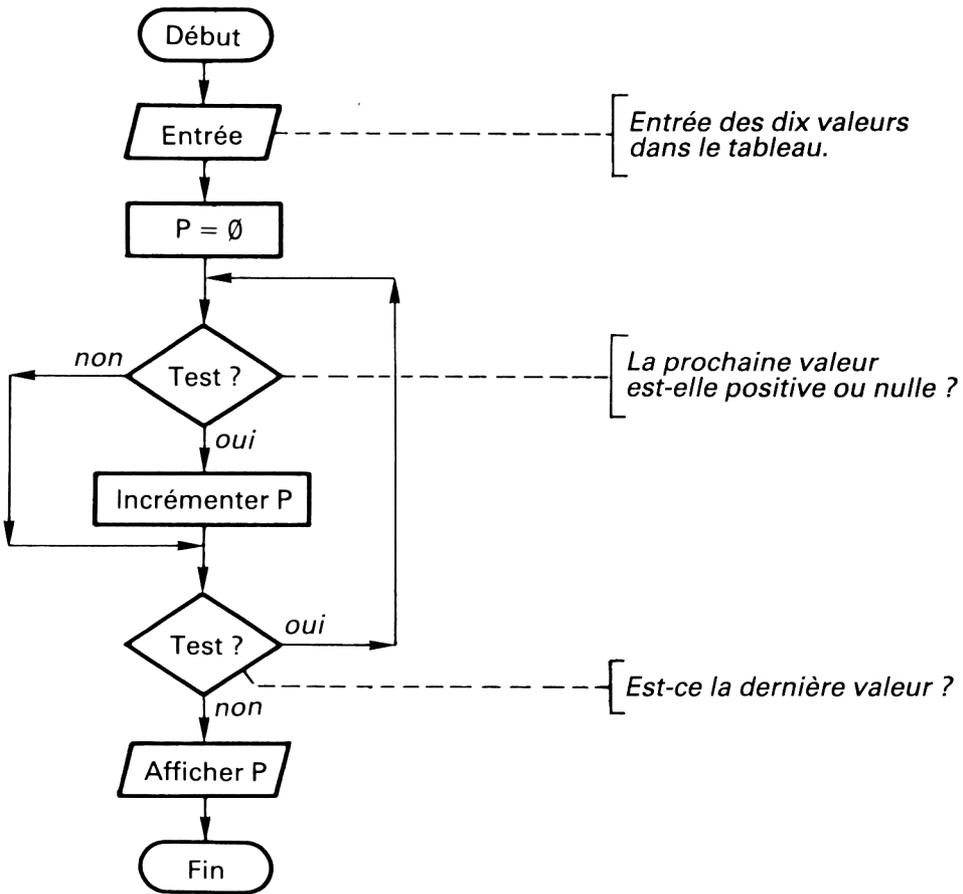
ENTRÉE : les dix valeurs.

SORTIE : le nombre de valeurs positives ou nulles.

ANALYSE

Il est nécessaire d'utiliser un compteur (ici P), qui sera incrémenté (on lui ajoute 1) dans la boucle pour chaque valeur positive trouvée.

L'utilisation d'un tableau, pour enregistrer les valeurs, permet un traitement plus facile.



EXERCICE 2.2

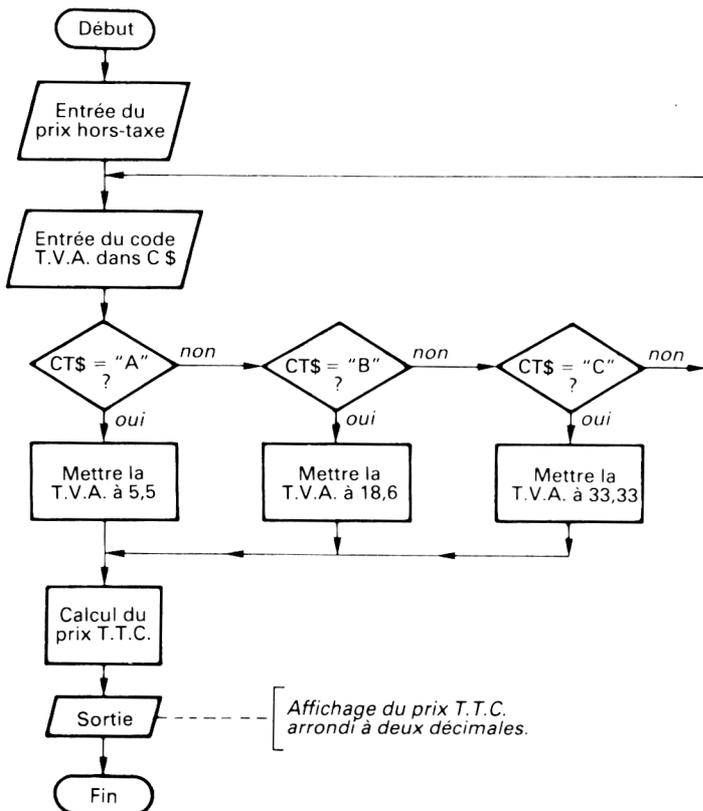
PROBLÈME : calcul d'un prix TTC à partir du prix hors taxe et du code de la TVA à lui appliquer.

ENTRÉE : prix hors taxe et code du taux de TVA (A, B ou C pour 5 %, 18,6 % ou 33,33 % respectivement).

SORTIE : le prix TTC arrondi à deux décimales.

ANALYSE

Le taux de TVA correspondant au code est affecté à une variable, puis le prix TTC est calculé à partir de cette variable. Si le code tapé ne correspond à rien, un autre code TVA est demandé.



EXERCICE 2.3

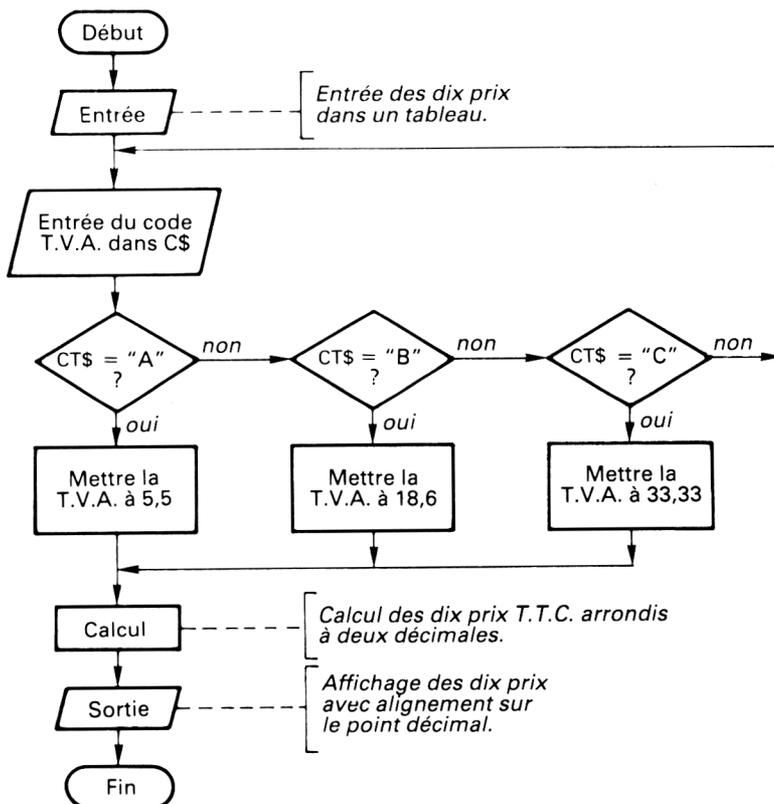
PROBLÈME : même programme que le précédent (EXERCICE 2.2), mais pour un groupe de dix prix.

ENTRÉE : dix prix hors taxes et le code du taux de TVA à leur appliquer.

SORTIE : les dix prix TTC arrondis à deux décimales et alignés sur le point décimal.

ANALYSE

L'utilisation d'un tableau pour enregistrer les dix prix permet de faciliter les entrées, l'affichage et les calculs.



EXERCICE 2.4

PROBLÈME : lecture dans un tableau d'une liste de mots enregistrée dans le programme sous forme de DATA. Donnez différentes solutions pour résoudre le problème de la longueur de la liste.

ENTRÉE : la liste de mots (à partir de lignes DATA).

SORTIE : affichage du tableau.

EXERCICE 2.5

PROBLÈME : à partir d'une liste de noms et de prénoms enregistrée dans la machine, on cherche à savoir si une personne se trouve dans cette liste.

ENTRÉE : une liste de personnes (noms et prénoms). Une personne à chercher (nom et prénom).

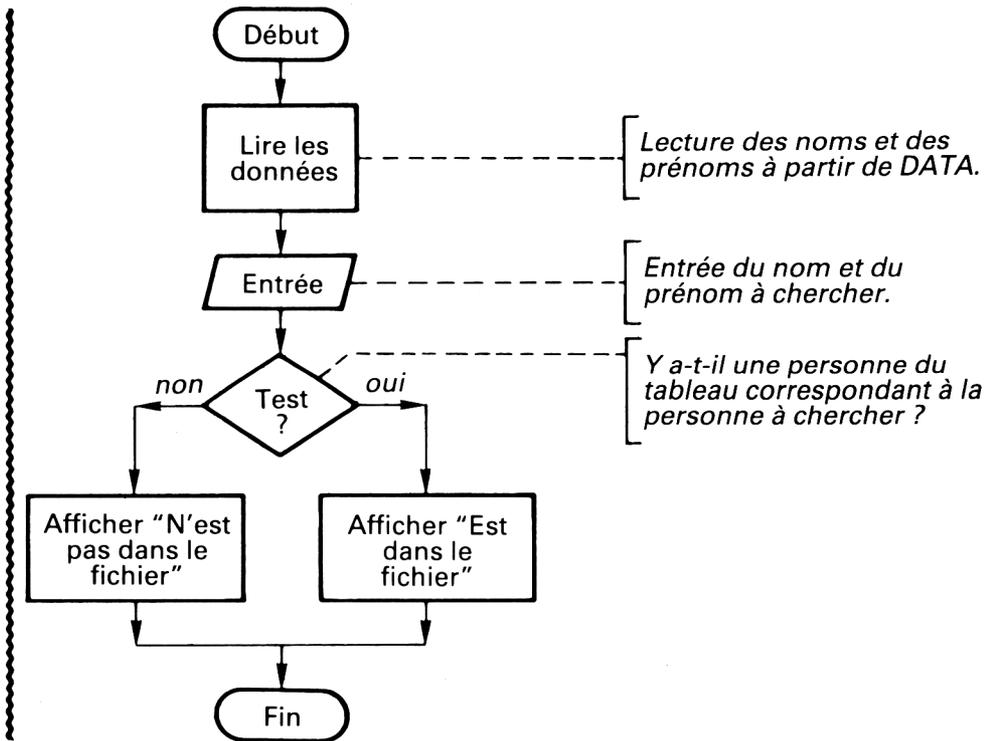
SORTIE : un message indiquant si la personne se trouve, ou ne se trouve pas dans la liste.

ANALYSE

L'utilisation de DATA permet une lecture facile et rapide des données dans un tableau.

Pour savoir si une personne se trouve dans la liste, il faut comparer successivement son nom et son prénom à ceux de toutes les personnes de la liste.





EXERCICE 2.6

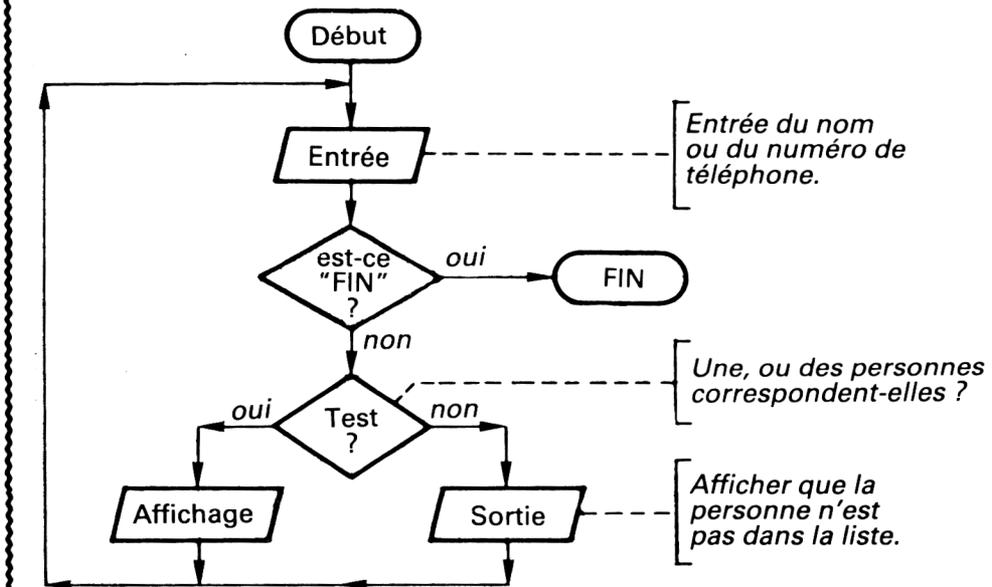
PROBLÈME : recherche, dans un agenda, d'un numéro de téléphone ou d'un nom.

ENTRÉE : le nom de la personne dont on souhaite connaître le numéro de téléphone, ou le numéro de celle dont on cherche le nom.

SORTIE : la personne recherchée avec son numéro de téléphone, si elle se trouve effectivement dans l'agenda (attention, plusieurs personnes peuvent porter le même nom !).

ANALYSE

Les personnes sont enregistrées sous forme de DATA, avec leur numéro de téléphone. La présence ou l'absence dans la liste, de la personne demandée, est signalée par une variable 0 ou 1. Une telle variable s'appelle un drapeau.

**EXERCICE 2.7**

PROBLÈME : trouver dans une liste de personnes la première par ordre alphabétique et indiquer son rang.

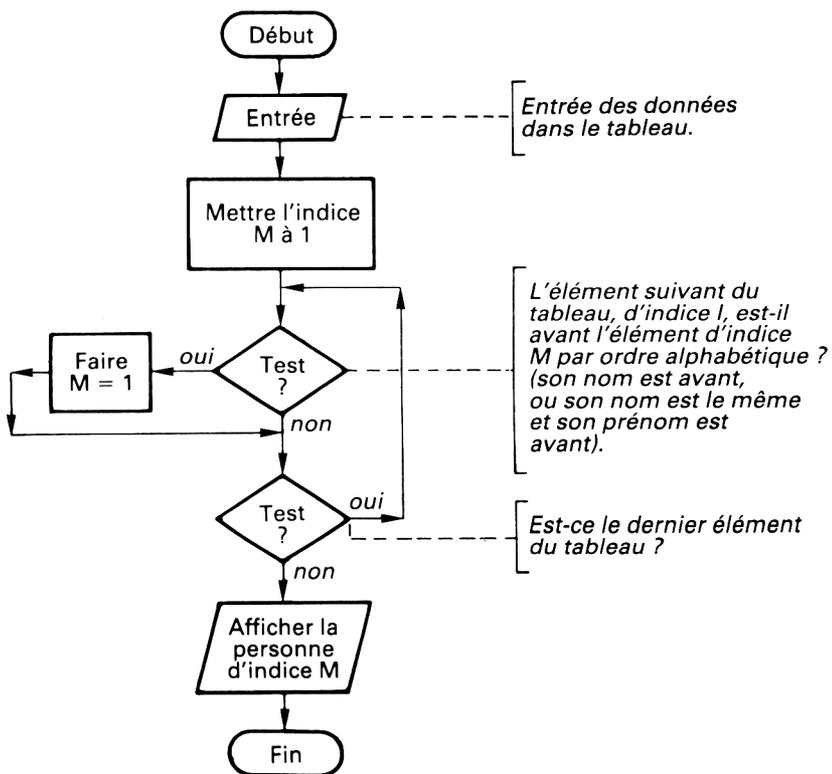
ENTRÉE : une liste de personnes (noms et prénoms).

SORTIE : le rang de la première personne, son nom et son prénom.

ANALYSE

L'ordre alphabétique entre deux chaînes de caractères se détermine en utilisant les opérations de test ">" et "<". Par exemple, si A\$>B\$ est vrai, A\$ suit B\$ dans l'ordre alphabétique..

Dans le problème posé, il y a deux comparaisons à faire, l'une entre les noms, et l'autre, si les noms sont égaux, entre les prénoms.



EXERCICE 2.8

PROBLÈME : classement par ordre décroissant d'une liste de nombres entrés dans la machine dans un ordre quelconque (utiliser deux tableaux, le premier contenant les nombres à classer, le second pour ranger les nombres classés).

ENTRÉE : une liste de nombres (entrée sous forme de DATA).

SORTIE : la liste classée par ordre décroissant.

ANALYSE

Deux manières de résoudre le problème sont proposées :

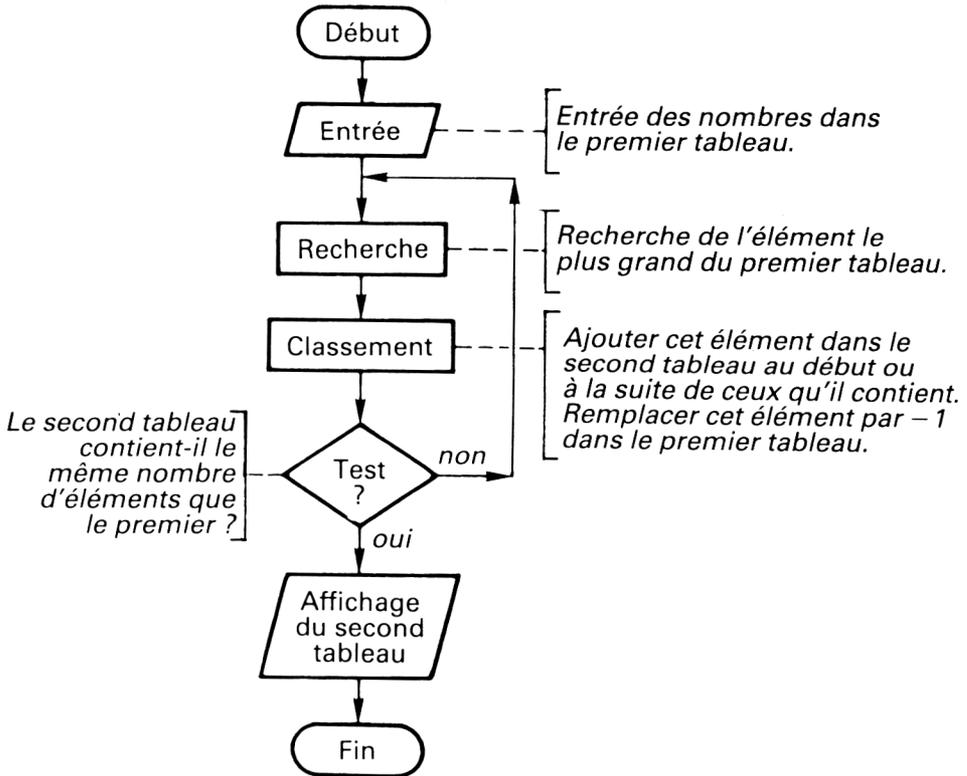
1^{re} méthode :

Elle nécessite l'utilisation d'un nombre particulier ne pouvant appartenir à la liste. Ici nous supposons que tous les nombres à classer sont positifs ou nuls ; un nombre négatif, par exemple -1 , ne peut donc en faire partie.

La machine recherche l'élément le plus grand du premier tableau qu'elle ajoute au second tableau ; cet élément est ensuite remplacé par -1 dans le premier tableau, le processus est recommencé autant de fois qu'il y a de nombres à classer.

A la fin, le premier tableau ne contient plus que des -1 , et le second tableau contient les nombres classés par ordre décroissant.





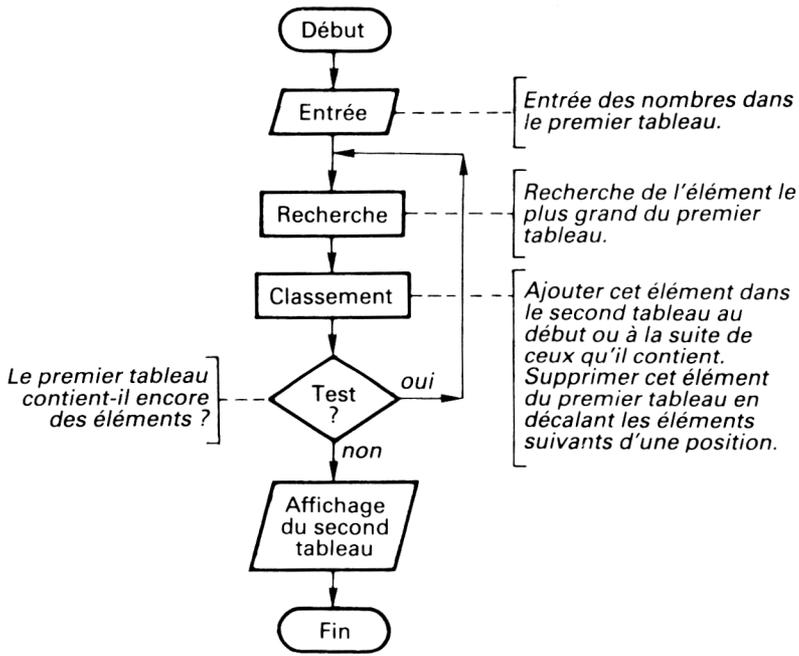
2^e méthode :

Cette méthode est applicable quels que soient les nombres à classer.

Après avoir trouvé l'élément le plus grand du premier tableau et l'avoir transféré dans le second, cet élément est supprimé du premier tableau en faisant "remonter" les éléments suivants d'une position, le tableau des nombres à classer contient donc un élément de moins.

Le processus est recommencé jusqu'à ce qu'il n'y ait plus d'éléments dans le premier tableau.





Exemple :

élément le plus grand →

121	1
1000	2
17	3
-3	4
0	5
2	.
-1	.
0	.
10	.
4	.
1	N1

Tableau 1

1000	N2
x	
x	
x	
x	
x	
x	
x	
x	
x	
x	

Tableau 2

puis décalage dans le premier tableau :

121	1
17	2
-3	3
0	4
-2	5
-1	.
0	.
10	.
4	.
1	N1
x	.

Tableau 1

La première méthode a l'avantage d'être simple, mais elle nécessite l'existence d'une valeur ne pouvant appartenir au tableau initial (ici -1).

La deuxième méthode peut s'appliquer à n'importe quel tableau mais est plus compliquée, et plus longue à exécuter, à cause des décalages que l'on doit effectuer.

EXERCICE 2.9

PROBLÈME : écrire un programme affichant les pays dans l'ordre croissant de leurs populations. Nous donnons ci-dessous une liste de pays et leur population en millions d'habitants :

ALLEMAGNE	: 61,498	BELGIQUE	: 9,889
DANEMARK	: 5,073	FRANCE	: 52,915
IRLANDE	: 3,162	ITALIE	: 56,323
LUXEMBOURG	: 0,358	PAYS-BAS	: 13,825
ROYAUME-UNI	: 55,928		

SORTIE : affichage des noms des pays par population croissante et de leur population en millions d'habitants.

EXERCICE 2.10

PROBLÈME : écrire un programme qui permet de classer une liste de pays suivant les trois critères : population, superficie, densité.

ENTRÉE : choix du critère.

SORTIE : le menu proposé sous la forme :

- 0 - POPULATIONS CROISSANTES
- 1 - SUPERFICIES CROISSANTES
- 2 - DENSITÉS CROISSANTES

l'affichage des résultats demandés.

DONNÉES : on donne :

PAYS

Allemagne, Belgique, Danemark, Espagne, France,
Grèce, Irlande, Italie, Luxembourg, Pays-Bas, Portugal,
Royaume-Uni

POPULATIONS (millions d'hab.)

61,498/9,889/5,073/35,971/52,915/9,165/3,162
56,323/0,358/13,825/9,449/55,928

SUPERFICIES (milliers de km²)

249/31/43/505/547/132/70/301/3/41/92/244

DENSITÉS (hab./km²)

247/319/118/71/97/69/45/187/138/339/103/228

EXERCICE 2.11

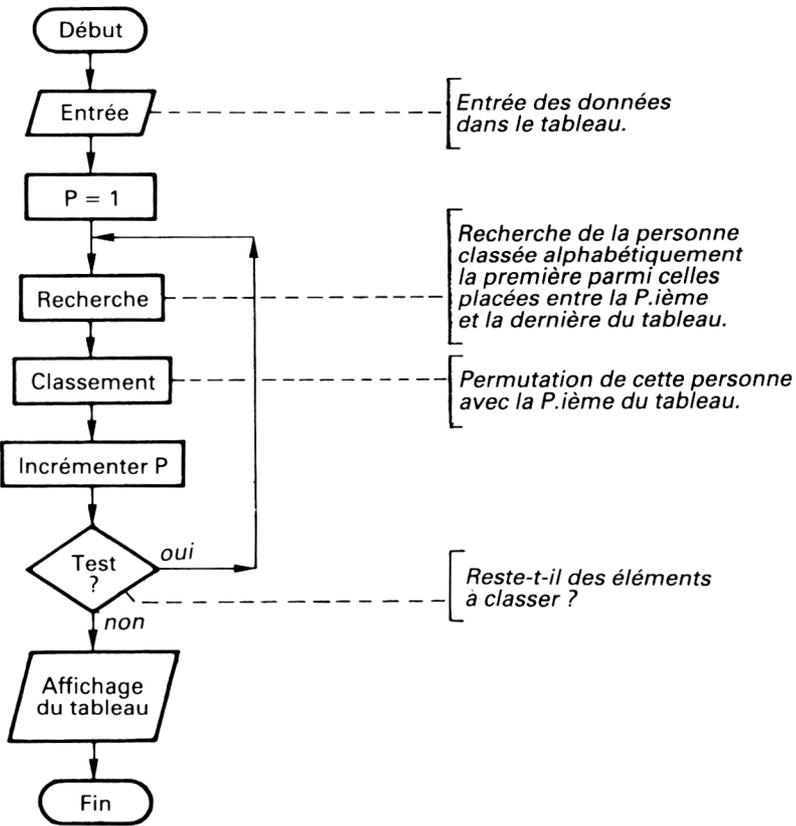
PROBLÈME : classer une liste de personnes par ordre alphabétique en n'utilisant qu'un tableau.

ENTRÉE : une liste de personnes (noms et prénoms).

SORTIE : la liste classée par ordre alphabétique.



ANALYSE



Il est parfois nécessaire de n'utiliser qu'un tableau, par exemple lorsque la capacité mémoire de la machine est insuffisante pour en contenir plusieurs.

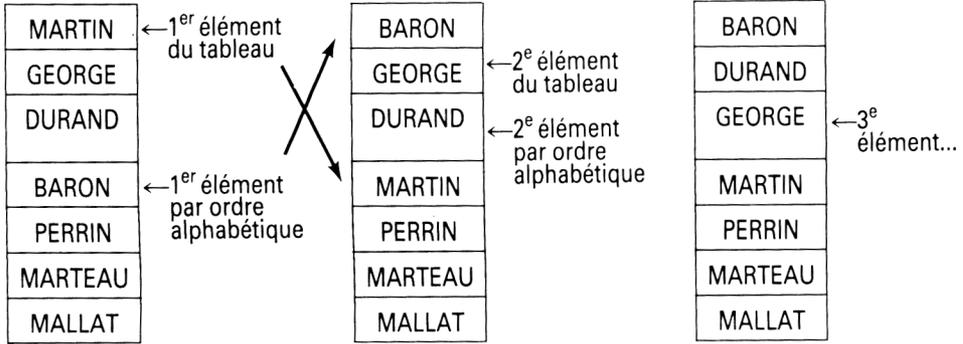
Voici la méthode utilisée :

Rechercher le premier élément du tableau par ordre alphabétique (BARON), puis permuter celui-ci avec le premier élément du tableau (MARTIN).

Le premier élément du tableau est alors aussi le premier par ordre alphabétique.

Recommencer avec les éléments suivants, chercher le premier par ordre alphabétique parmi eux, puis permuter celui-ci avec celui d'indice le plus petit.

Exemple :



EXERCICE 2.12

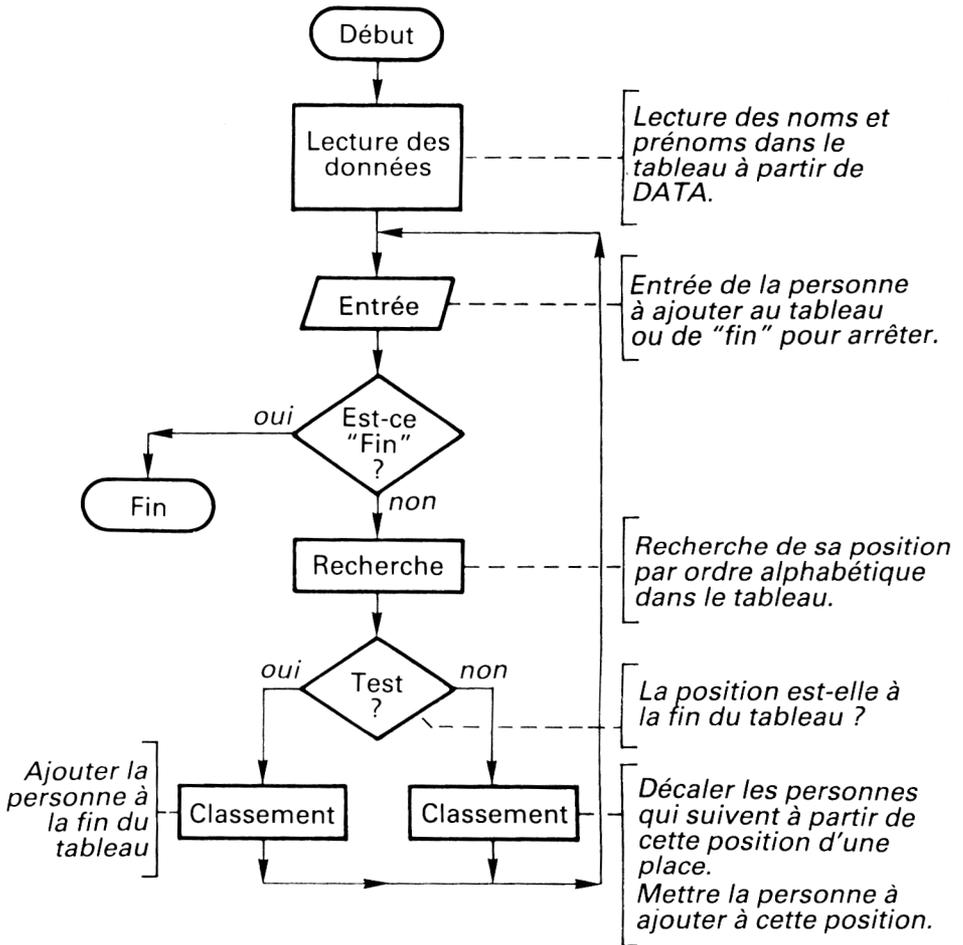
PROBLÈME : ajouter une personne à un tableau, classé alphabétiquement, en l'insérant à sa position dans l'ordre alphabétique.

ENTRÉE : une liste de personnes (noms et prénoms) classées par ordre alphabétique. Une personne à insérer.

SORTIE : affichage du nouveau tableau contenant la personne à insérer à sa position dans l'ordre alphabétique.



ANALYSE



La lecture de la liste de personnes à partir de DATA permet d'être certain, lors de l'exécution du programme, de manipuler une liste classée dans l'ordre alphabétique (en enregistrant les DATA par ordre alphabétique).

Pour insérer la personne dans l'ordre alphabétique, il faut trouver sa position dans le tableau. Si cette position est à la fin du tableau, il suffit d'ajouter la personne à la fin, sinon il faut décaler toutes les personnes suivantes d'un cran pour libérer cette position.

Exemple :

Insérer GEORGE dans :



DUPONT
DURAND
MARTIN
NOEL
ROY
SENDEK
TILLOY

} inchangé

* position
d'insertion

D'où le décalage :



DUPONT
DURAND
MARTIN
MARTIN
NOEL
ROY
SENDEK
TILLOY

] décalage

DUPONT
DURAND
GEORGE
MARTIN
NOEL
ROY
SENDEK
TILLOY

* puis insertion de
GEORGE :**EXERCICE 2.13**

PROBLÈME : écrire un programme qui calcule les racines réelles d'une équation du second degré.

ENTRÉE : les trois coefficients du polynôme du second degré.

SORTIE : les 2 racines réelles distinctes, ou la racine double, ou le message "DEUX RACINES COMPLEXES".

ANALYSE

Entrer les trois nombres A, B et C correspondant à l'équation :

$$A X^2 + B X + C = 0$$

il faut calculer le discriminant DELTA donné par :

$$\text{DELTA} = B^2 - 4AC$$

Si DELTA est négatif, l'équation n'a pas de racines réelles.

Si DELTA est positif, l'équation a deux racines réelles données par :

$$X1 = (-B + \sqrt{\text{DELTA}})/(2A)$$

$$X2 = (-B - \sqrt{\text{DELTA}})/(2A)$$

Si DELTA est nul, l'équation a une racine double donnée par :

$$X = (-B)/(2A)$$

EXERCICE 2.14

PROBLÈME : transformer chaque élément d'un tableau de nombres en son pourcentage par rapport à la somme de tous les nombres.

ENTRÉE : un tableau de nombres.

SORTIE : le tableau composé en transformant chaque élément en son pourcentage par rapport à la somme de tous les nombres. Les pourcentages devront être affichés avec au plus deux nombres après la virgule et alignés sur celle-ci.

ANALYSE

Il faut, tout d'abord, calculer la somme S des valeurs du tableau. Le pourcentage d'une valeur T(I) du tableau est donné par la formule : $100 \times T(I)/S$.

EXERCICE 2.15

PROBLÈME : trouver le nombre supérieur le plus proche d'un nombre donné et ayant un seul chiffre significatif. Par exemple le nombre 0,0057637 est transformé en 0,006 et le nombre 124,978764 en 200. Ce problème se pose, par exemple, lorsque l'on gradue les axes d'un graphique.

ENTRÉE : un nombre.
 SORTIE : le nombre tronqué.

ANALYSE

La partie entière de la fonction LOG 10 d'un nombre donne le nombre de chiffres avant la virgule si le nombre est supérieur à 1 et le nombre de zéros après la virgule, si le nombre est inférieur à 1. Ainsi :

$\text{INT}(\text{LOG}10(124,978764))+1$ est égal à 3
 $\text{INT}(\text{LOG}10(0,0057637))+1$ est égal à -2

EXERCICE 2.16

PROBLÈME : calculer la surface et le volume d'une sphère, d'un cube ou d'un cylindre.

ENTRÉE : taper SP pour la sphère, CY pour le cylindre et CU pour le cube. Entrer ensuite les paramètres nécessaires pour le calcul de la surface et du volume.

SORTIE : la surface et le volume.

ANALYSE

En utilisant l'instruction DEF FN, on peut calculer une expression numérique comme le volume d'une sphère, en passant uniquement la valeur du rayon de la même façon que l'on utilise la fonction SIN ou COS...

CHAPITRE III

TRAITEMENTS DE CHAINES DE CARACTÈRES

*(Recherche, insertion,
suppression de
caractères)*

TRAITEMENTS DE FICHIERS

EXERCICE 3.1

PROBLÈME : suppression de tous les caractères "espace" pouvant exister dans une chaîne de caractères.

ENTRÉE : une ligne de texte.

SORTIE : la ligne sans aucun espace.

EXERCICE 3.2

PROBLÈME : ajouter un espace entre chaque lettre d'une chaîne de caractères.

ENTRÉE : une chaîne de caractères.

SORTIE : la même chaîne de caractères avec un espace ajouté entre chaque caractère.

EXERCICE 3.3

PROBLÈME : supprimer successivement la première lettre d'une chaîne de caractères jusqu'à son épuisement.

ENTRÉE : une phrase ou un mot.

SORTIE : affichages successifs du mot tronqué.

Exemple : BONJOUR donne
BONJOUR
ONJOUR
NJOUR
JOUR
OUR
UR
R

EXERCICE 3.4

PROBLÈME : suppressions successives du premier et du dernier caractère d'un mot.

ENTRÉE : un mot.

SORTIE : affichages successifs du mot tronqué au centre de l'écran.

Exemple : **BONJOUR** donne **BONJOUR**
ONJOU
NJO
J

ANALYSE

Pour obtenir la présentation donnée en exemple, plusieurs méthodes sont possibles : on peut remplacer les caractères supprimés en début de ligne par autant d'espaces, ou bien utiliser à l'affichage la fonction TAB (c'est cette dernière possibilité qui sera utilisée dans la solution proposée).

EXERCICE 3.5

PROBLÈME : supprimer successivement le dernier mot d'une phrase. Effectuer cette opération jusqu'à épuisement.

ENTRÉE : une phrase.

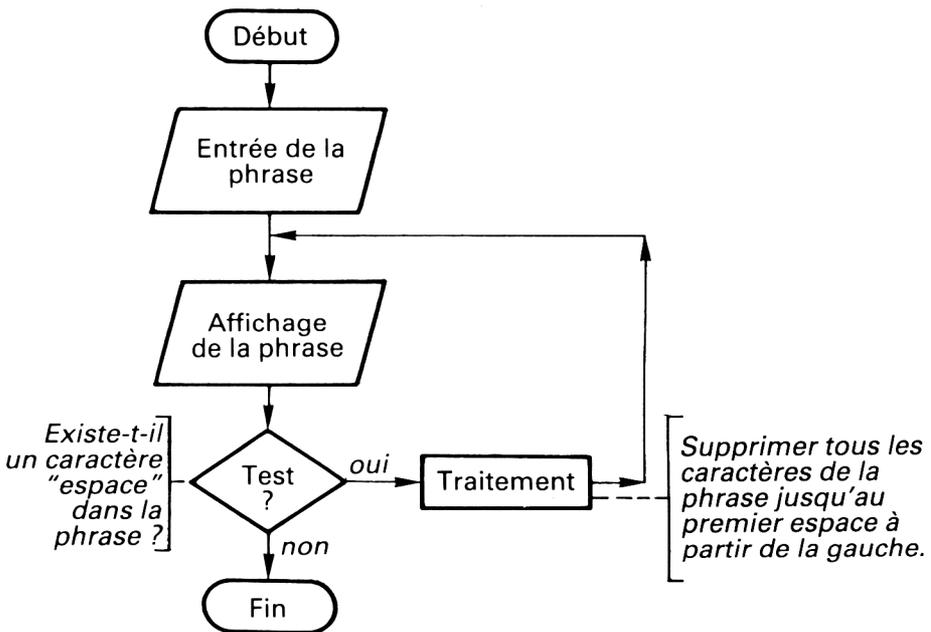
SORTIE : affichages successifs de la phrase tronquée d'un mot à chaque fois.

Exemple : **LE CHAT EST SUR LE TOIT** donne

LE CHAT EST SUR LE TOIT
CHAT EST SUR LE TOIT
EST SUR LE TOIT
SUR LE TOIT
LE TOIT
TOIT

ANALYSE

Dans une phrase, les mots sont séparés par un "blanc" contenant un ou plusieurs "espaces". Le programme va donc rechercher le premier blanc à partir de la gauche de la phrase, supprimer tous les caractères jusqu'à ce blanc, afficher la phrase, et recommencer jusqu'à ce qu'il ne reste plus de mots dans la phrase.

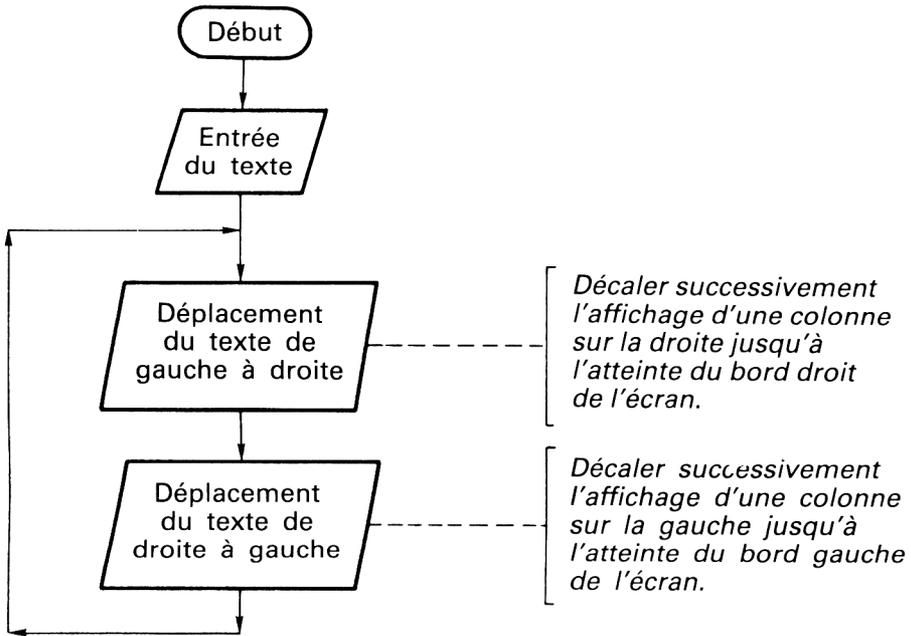


EXERCICE 3.6

PROBLÈME : faire défiler une phrase sur l'écran de gauche à droite puis, lorsqu'elle atteint le bord, de droite à gauche, et ainsi de suite.

ENTRÉE : une phrase de moins de 20 caractères.

SORTIE : faire défiler la phrase sur l'écran.

ANALYSE**EXERCICE 3.7**

PROBLÈME : faire défiler un texte, sans arrêt, de droite à gauche sur une même ligne d'écran.

ENTRÉE : un texte de moins de 39 caractères.

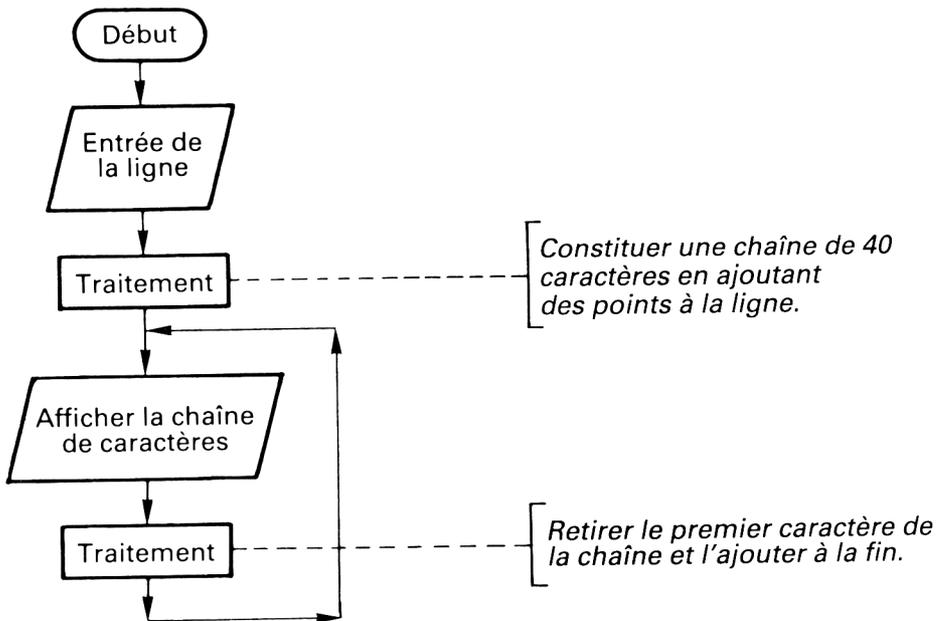
SORTIE : le texte défile sur l'écran.

ANALYSE

Deux méthodes sont proposées pour résoudre le problème. Dans les deux cas, on utilise la possibilité sur AMSTRAD de faire clignoter les couleurs.

1^{re} méthode :

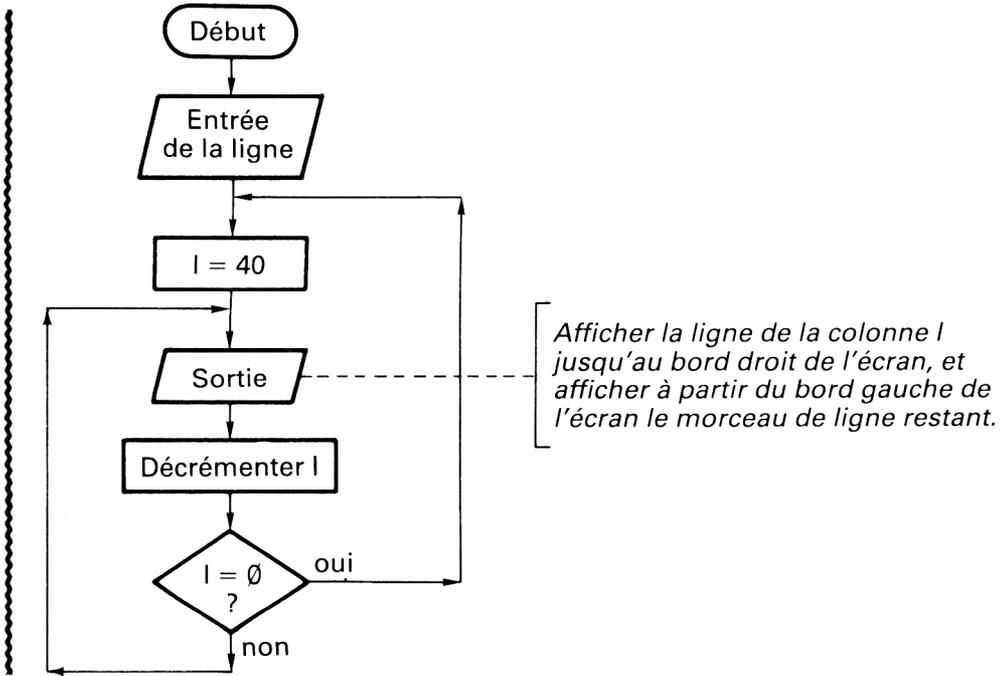
Créer une chaîne de 40 caractères (largeur totale de l'écran) à partir du texte entré en le complétant par suffisamment de points. Pour obtenir le défilement, retirer le premier caractère de la chaîne et le reporter à la fin, puis recommencer l'opération.



2^e méthode :

Afficher le texte depuis la colonne 1 jusqu'au bord droit de l'écran, puis afficher la partie restante de ce texte depuis le bord gauche de l'écran sur la même ligne.





EXERCICE 3.8

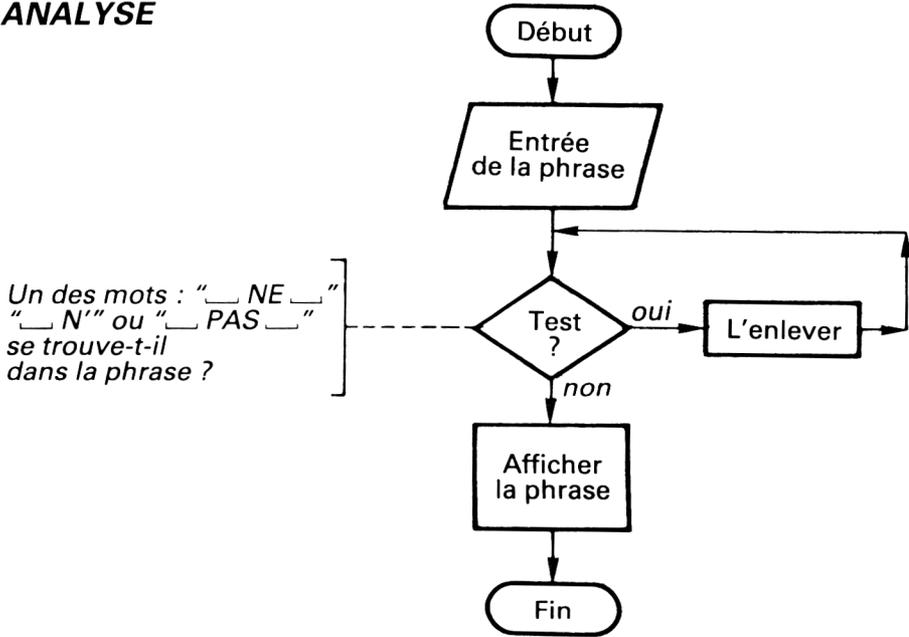
PROBLÈME : transformer une phrase de la forme négative à la forme positive en supprimant de la phrase les mots NE, N' et PAS.

ENTRÉE : une phrase à la forme négative.

SORTIE : la phrase à la forme positive.



ANALYSE



Remarque : les comparaisons s'effectuent avec les mots NE, N' et PAS entourés d'espaces pour les distinguer de parties de mots (exemple : PASserelle).

Attention au cas où la phrase commencerait avec un de ces mots, ou se terminerait avec le mot PAS.

EXERCICE 3.9

PROBLÈME : écrire un programme recherchant, dans un groupe de mots, tous ceux contenant une suite donnée de lettres (clé).

ENTRÉE : la clé à rechercher.

SORTIE : les différents mots trouvés, affichés en alignant sur la même colonne la clé à rechercher. Le nombre de fois où la clé a été trouvée.

Exemple : si la liste de mots est VOITURE, RIVIERE ET AVAIT, et la clé à rechercher IT, on obtient :

VOITURE
AVAIT

La clé a été trouvée 2 fois.

Remarque : la clé pourra être écrite avec une couleur différente ou apparaître en lettres majuscules, le reste du mot étant en lettres minuscules.

ANALYSE

Le texte à analyser est introduit dans le programme sous forme d'une liste d'une ou plusieurs instructions DATA. La clé est ensuite comparée successivement en se plaçant sur le premier caractère d'un mot et en se déplaçant successivement sur la droite.

EXERCICE 3.10

PROBLÈME : insertion de caractères dans une chaîne de caractères à une position donnée.

ENTRÉE : une chaîne de caractères, puis la chaîne de caractères à insérer et la position d'insertion.

SORTIE : la nouvelle chaîne de caractères.

EXERCICE 3.11

PROBLÈME : remplacement d'un mot par un autre dans une ligne de texte ne dépassant pas 128 caractères. Le mot doit être remplacé autant de fois qu'il se trouve dans le texte (le texte modifié ne doit cependant pas dépasser 128 caractères).

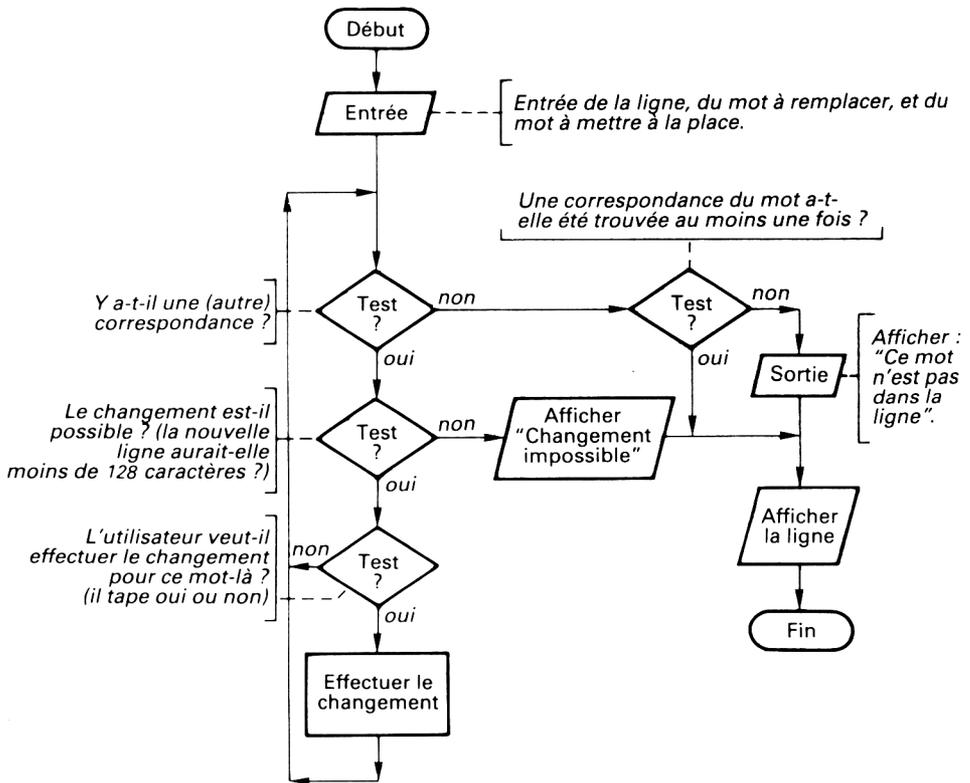
ENTRÉE : un texte d'au plus 128 caractères. Un mot à remplacer et le mot qu'on lui substituera. La réponse pour confirmer la substitution.

SORTIE : le texte obtenu en affichant le mot à remplacer en mode clignotant et si la substitution est confirmée, le nouveau texte obtenu.

ANALYSE

La recherche dans le texte du mot à remplacer est faite en le comparant à des morceaux du texte de même longueur, en commençant au début de la ligne, puis en se déplaçant d'un caractère jusqu'à la fin de la ligne.

A chaque fois que le mot à remplacer est trouvé, la machine demande à l'utilisateur s'il veut ou non remplacer ce mot dans le texte.



EXERCICE 3.12

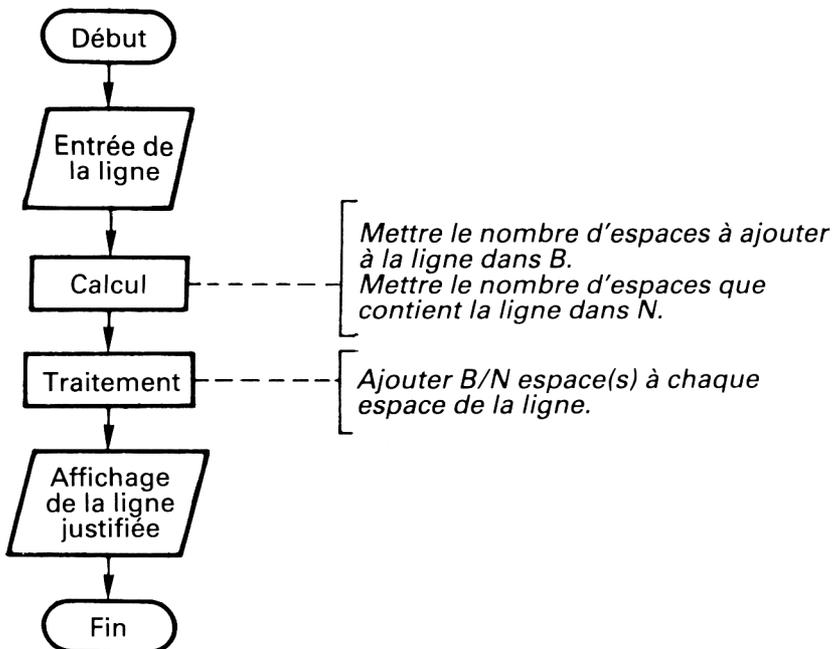
PROBLÈME : justification d'une ligne de texte sur l'écran.

ENTRÉE : une ligne de texte de longueur inférieure ou égale à 40 caractères.

SORTIE : la ligne justifiée sur l'écran (des espaces ont été ajoutés entre chaque mot, de manière à ce que le premier et le dernier mot de la ligne soient sur les bords de l'écran).

ANALYSE

Il faut ajouter des espaces entre chaque mot de façon à ce que le premier caractère se trouve sur la colonne 1 et le dernier caractère sur la colonne 40. Le nombre d'espaces à ajouter est égal à 40 moins le nombre de caractères du texte à justifier. Toutefois, pour obtenir une bonne présentation à l'écran, il est nécessaire de répartir équitablement entre chaque mot, ce nombre d'espaces nécessaires.



EXERCICE 3.13

PROBLÈME : à partir d'un mot ou d'une phrase, créer le mot ou la phrase, construit en lisant les caractères de droite à gauche, puis en les lisant de façon aléatoire.

ENTRÉE : une phrase ou un mot.

SORTIE : la phrase avec les caractères placés dans l'ordre inverse et une phrase constituée exactement des mêmes caractères, mais dans un ordre aléatoire.

Exemple : TABLEAU donne UAELBAT en ordre inverse et, par exemple, BLTUAEA en ordre aléatoire.

ANALYSE

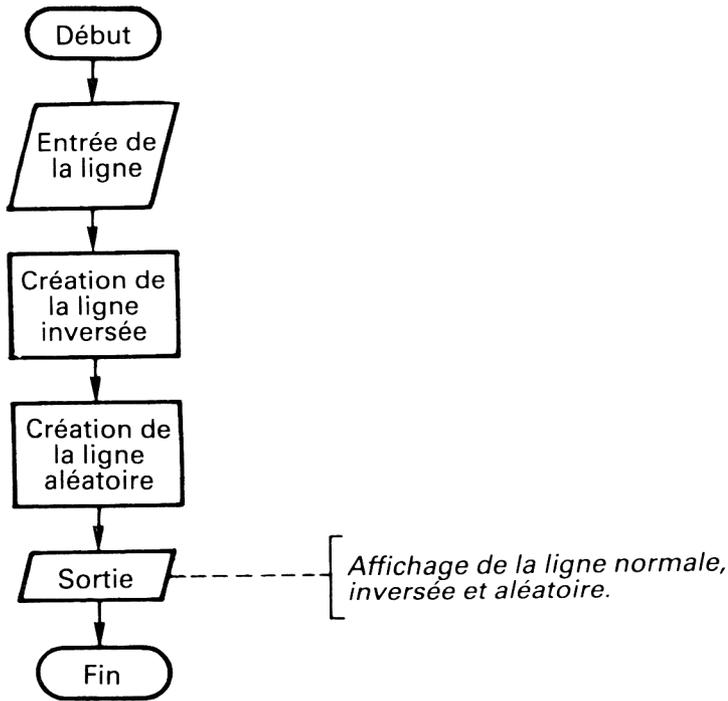
La méthode utilisée pour inverser l'ordre des caractères est la suivante : les caractères de la chaîne à inverser sont ajoutés un à un, en partant de la fin, à une autre chaîne.

Exemple : TABLEAU donne U
 puis UA
 puis UAE
 puis UAEL
 puis UAELB
 puis UAELBA
 et enfin UAELBAT.

Pour créer la chaîne aléatoire, un caractère de la chaîne est choisi au hasard, ajouté à une autre chaîne, puis supprimé de la chaîne initiale. Le processus est ensuite recommencé jusqu'à ce que la chaîne initiale soit vide.

Exemple : avec TABLEAU on aurait par exemple :

TABLEAU	et	B
TAEAU	et	BL
AEAU	et	BLT
AEA	et	BLTU
AE	et	BLTUA
A	et	BLTUAE
	et enfin	BLTUAEA



EXERCICE 3.14

PROBLÈME : donner le pourcentage d'apparition de chaque lettre de l'alphabet dans un texte.

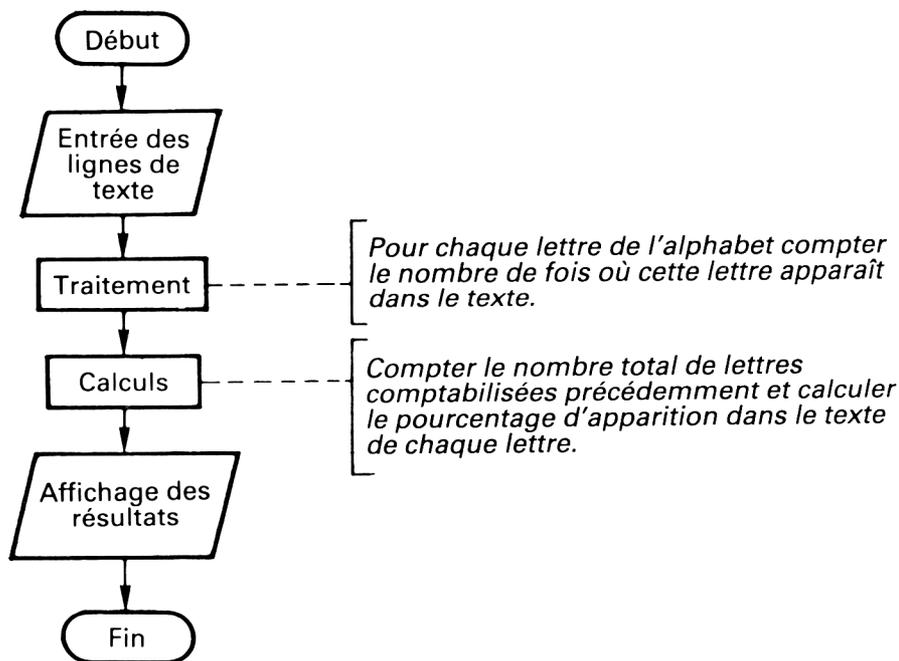
ENTRÉE : une ou plusieurs lignes de texte.

SORTIE : chaque lettre de l'alphabet suivie de son pourcentage d'apparition dans les lignes de texte (arrondi à deux décimales et aligné sur la virgule ou point décimal anglais).

ANALYSE

Créer un tableau contenant 26 éléments, puis à chaque fois que la lème lettre de l'alphabet apparaît dans le texte, incrémenter le lème élément. Pour déterminer le rang l d'une lettre dans l'alphabet, on

peut utiliser la fonction ASC et soustraire 64. On note en effet que les lettres majuscules en code ASCII se suivent par ordre alphabétique, entre le code 64 et 90. On peut encore créer une chaîne contenant toutes les lettres de l'alphabet dans l'ordre et utiliser l'instruction INSTR pour déterminer la position d'une lettre dans cette chaîne. C'est la première méthode qui a été retenue dans la solution proposée. Calculer enfin le pourcentage d'apparition de chacune des lettres dans le texte.



EXERCICE 3.15

PROBLÈME : à partir d'une date écrite en chiffres (ex : 15-7-1790), remplacer le numéro du mois par son nom (ex : 15 JUILLET 1790).

ENTRÉE : trois nombres correspondant à un jour, un mois et une année.

SORTIE : affichage de la chaîne de caractères contenant la date.

ANALYSE

Les noms des 12 mois de l'année sont passés par une liste créée par une ou plusieurs instructions DATA. Ils sont lus successivement et le numéro du mois sert à arrêter cette lecture.

EXERCICE 3.16

PROBLÈME : programme inverse du précédent.

ENTRÉE : une date sous la forme d'une chaîne de caractères.

Exemple : "13 JUILLET 1788"

SORTIE : les trois nombres correspondant au jour, au mois et à l'année de la date.

Exemple : 13/7/1788

ANALYSE

La chaîne de caractères représentant la date est découpée en fonction des espaces qu'elle contient. Le numéro du mois est obtenu à partir de la place qu'il occupe dans la liste de tous les mois enregistrée dans une instruction DATA (le mois de juillet y occupera ainsi la 7ème place).

EXERCICE 3.17

PROBLÈME : afficher un nombre entré au clavier avec des chiffres géants dessinés au moyen d'étoiles sur une matrice de cinq colonnes et de six lignes.

ENTRÉE : un nombre entier positif compris entre 0 et 99 999.


```

Adresse de debut : 512
Adresse de fin : 639
0200 : C4 00 1F 00 3C 00 A3 20 ....<...
0208 : 22 41 64 72 65 73 73 65 "Adresse
0210 : 20 64 65 20 66 69 6E 20 .de.fin.
0218 : 3A 22 2C 0D 10 00 41 C6 :",...A.
0220 : 00 23 00 41 00 BF 20 23 .#.A...#
0228 : 16 2C 22 41 64 72 65 73 .,"Adres
0230 : 73 65 20 64 65 20 66 69 se.de.fi
0238 : 6E 20 3A 20 22 3B 0D 10 n...";..
0240 : 00 41 C6 00 1D 00 46 00 .A....F.
0248 : 9E 20 0D 19 00 C9 EF 0D .....
0250 : 06 00 41 C4 20 EC 20 0D ..A.....
0258 : 10 00 41 C6 20 E6 20 16 ..A.....
0260 : 00 1A 00 50 00 BF 20 23 ...P...#
0268 : 16 2C FF 73 28 0D 19 00 ...s(...
0270 : C9 2C 12 29 22 20 3A 20 ...)"...
0278 : 22 3B 00 11 00 5A 00 9E ";...Z..

```

ANALYSE

Les fonctions PEEK et HEXA\$ permettent d'extraire le contenu sous forme hexadécimale d'une case mémoire. Tester ensuite si la valeur extraite est comprise entre 33 et 126 (caractères affichables).

EXERCICE 3.19

PROBLÈME : codage (cryptage) d'un texte.

ENTRÉE : un texte.

SORTIE : le même texte codé.

ANALYSE

L'alphabet de codage est donné sous forme d'une suite de 26 lettres, dans l'ordre de leur correspondance avec l'ordre alphabétique. Par exemple, si l'alphabet de codage est :

"WQAXSZCDEVFRBGTNHYJUKILOMP"

alors la lettre A est codée W, la lettre B est codée Q, etc.

EXERCICE 3.20

PROBLÈME : établir une fiche de saisie de données. Les données traitées sont le nom, le prénom, l'adresse et le numéro de téléphone.

ENTRÉE : les réponses aux différentes rubriques de la fiche de saisie. Les touches de déplacement vertical, de façon à se déplacer dans la fiche aux différentes rubriques. La touche ENTER, pour valider les données entrées. La touche DEL pour effacer le caractère qui précède le curseur.

SORTIE : la fiche de saisie sous forme de 4 rubriques. Chaque rubrique comprend l'élément à saisir (exemple : le prénom), suivi de points (".") en nombre suffisant pour compléter la ligne d'écran. Au fur et à mesure que l'on complète une rubrique, les caractères frappés remplacent les points sur la ligne.

ANALYSE

Les données sont saisies à la rubrique où le curseur est positionné à l'écran. Pour changer de rubrique, utiliser les 2 touches de déplacement vertical. Pour valider les éléments saisis, utiliser la touche ENTER.

CHAPITRE IV

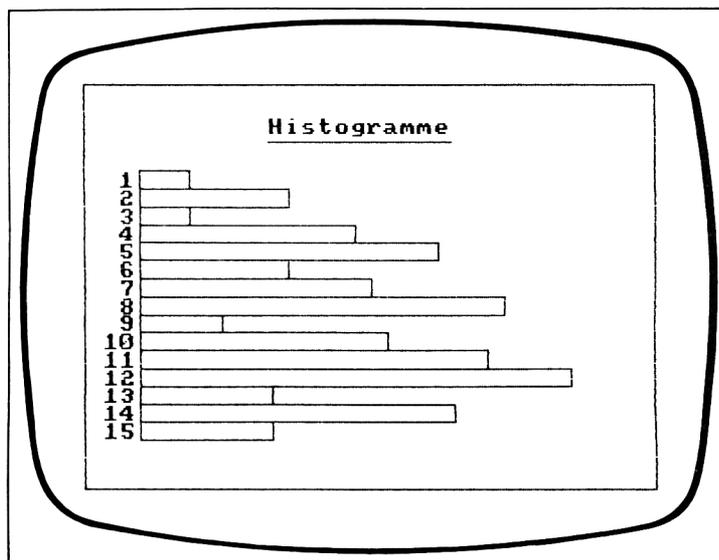
DESSINS, HISTOGRAMMES, GRAPHIQUES, TRACÉS DE FONCTIONS, PROGRAMMES MUSICAUX

EXERCICE 4.1

PROBLÈME : afficher l'histogramme en bâtons horizontaux à partir d'un tableau de quinze valeurs.

ENTRÉE : quinze valeurs comprises entre 0 et 100.

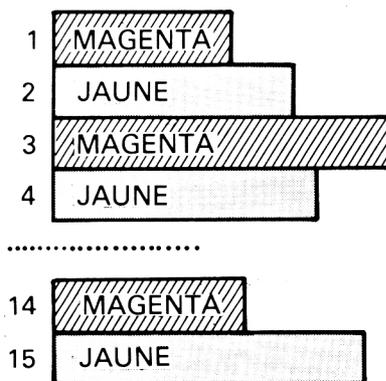
SORTIE : affichage de l'histogramme correspondant.



ANALYSE

Il faut faire correspondre à la valeur 100 une bande horizontale de longueur 30, à la valeur 0 une bande de longueur nulle et aux autres valeurs intermédiaires une bande de longueur proportionnelle. Pour rendre le graphique plus lisible, on utilise deux couleurs en alternance pour dessiner les bandes ; chaque bande est précédée de son rang dans le tableau.

Exemple :



EXERCICE 4.2

PROBLÈME : afficher l'histogramme en bâtons horizontaux à partir d'un tableau de dimension maximale égale à 20.

ENTRÉE : vingt valeurs positives (inférieures à 1000).

SORTIE : affichage de l'histogramme correspondant.

ANALYSE

A la plus grande valeur entrée, il faut faire correspondre une bande horizontale de longueur 30, à la valeur 0 une bande de longueur nulle et aux autres valeurs intermédiaires une bande de longueur proportionnelle. Il faut donc déterminer la plus grande valeur, puis la variation (pas) correspondant à chaque pavé.

EXERCICE 4.3

PROBLÈME : histogramme comparatif : à partir d'un tableau de 20 valeurs représentant les ventes de 2 articles sur 10 années, faire un graphique en bâtons. Les bâtons sont groupés par deux, pour représenter la vente des deux articles sur une même année.

ENTRÉE : deux fois dix valeurs, rentrées deux par deux et séparées par une virgule.

SORTIE : histogramme des valeurs. Utiliser une couleur par article.

Exemple :

1974 MAGENTA
 1974 JAUNE
 1975 MAGENTA
 1975 JAUNE

article A pour la 1^{re} année
 article B pour la 1^{re} année
 article A pour la 2^e année
 article B pour la 2^e année

.....

.....

ANALYSE

La plus grande des 20 valeurs permet de déterminer le pas de quantification, c'est-à-dire la quantité qui correspond à un pavé. Pour enregistrer les 20 valeurs, il est possible d'utiliser un tableau de dimension 2, le premier indice (0 et 1) représentant l'article, le deuxième indice (0 à 9) représentant l'année. Chaque bâton de l'histogramme est précédé de l'année correspondante et la valeur la plus grande est affichée à l'écran.

EXERCICE 4.4

PROBLÈME : graphique en secteurs : à partir d'une série de valeurs, afficher un disque composé de secteurs, chacun ayant une ouverture (angle) proportionnelle au pourcentage de la valeur à représenter. Pour rendre la lecture plus facile, utiliser pour l'affichage deux couleurs de façon alternée.

ENTRÉE : une suite de valeurs.

SORTIE : graphique en secteurs ; valeur associée au secteur de taille maximale.

ANALYSE

Il faut tout d'abord calculer les pourcentages d'apparition des valeurs. Ces pourcentages permettent ensuite de déterminer l'angle d'ouverture de chaque secteur. Cet angle est égal à ce pourcentage multiplié par l'angle total d'un cercle, soit 360 degrés ou $2 \times \text{PI}$ radian. Pour dessiner le secteur correspondant, on peut par exemple, en partant de l'origine, tracer un nombre de segments de longueur donnée (rayon du cercle), régulièrement espacés. Ce nombre de segments est calculé pour être proportionnel à la valeur à représenter. Pour chaque valeur affichée, on changera la couleur du secteur.

EXERCICE 4.5

PROBLÈME : graphique en bandes : à partir d'une série de valeurs, afficher une bande horizontale, constituée de segments, chacun ayant une longueur proportionnelle au pourcentage de la valeur à représenter. Pour rendre la lecture plus facile, utiliser pour l'affichage deux couleurs de façon alternée.

ENTRÉE : une suite de valeurs passées par une liste de DATA.

SORTIE : graphique en bandes ; valeur correspondant au segment de taille maximale.

ANALYSE

Il faut tout d'abord calculer les pourcentages d'apparition des valeurs. Ces pourcentages permettent ensuite de déterminer la longueur de chaque segment. Cette longueur est égale à ce pourcentage multiplié par la longueur de la bande complète (400 points d'affichage graphique). Pour chaque valeur affichée, on changera la couleur du segment.

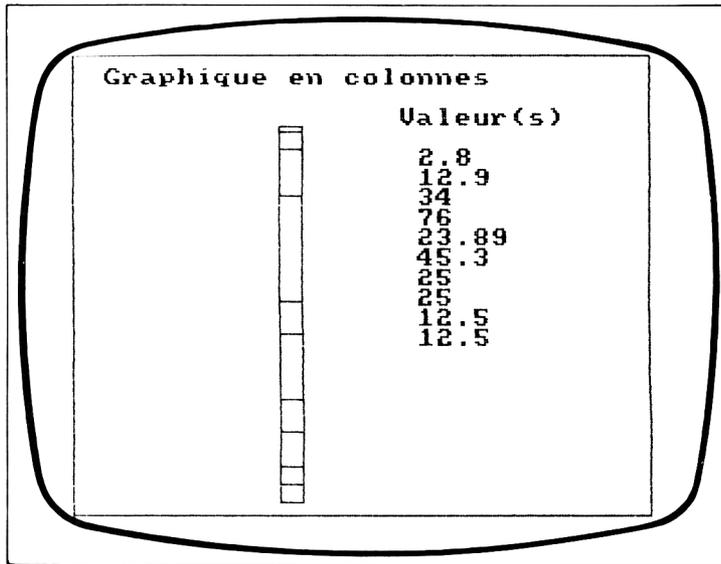
EXERCICE 4.6

PROBLÈME : graphique en colonnes : à partir d'une série de valeurs, afficher une colonne verticale de segments, chacun ayant une longueur proportionnelle au pourcentage de la valeur à représenter. Pour rendre la lecture plus facile, utiliser pour l'affichage deux couleurs de façon alternée.

ENTRÉE : une suite de valeurs passées par une liste de DATA.

SORTIE : graphique en colonnes ; valeur du segment de taille maximale.





ANALYSE

Il faut tout d'abord calculer les pourcentages d'apparition des valeurs. Ces pourcentages permettent ensuite de déterminer la longueur de chaque segment. Cette longueur est égale à ce pourcentage multiplié par la hauteur de la colonne (300 points graphiques). Pour chaque valeur affichée, on changera la couleur du segment.

EXERCICE 4.7

PROBLÈME : étudier l'ensemble des valeurs obtenues lors de 600 lancers de dés. Le nombre de dés est compris entre 1 et 6. Les résultats sont donnés sous forme d'un histogramme, dessiné en mode haute résolution (mode 2).

ENTRÉE : le nombre de dés à jeter à chaque lancer. Une touche quelconque est utilisée pour arrêter au cours des 600 lancers et provoque l'affichage de l'histogramme.

SORTIE : affichage des différents lancers.
Affichage de l'histogramme en graphique haute résolution. Chaque colonne a une hauteur proportionnelle au pourcentage d'apparition de la somme correspondante. On rappelle que, pour 3 dés par exemple, les sommes que l'on peut obtenir sont comprises entre 3 et 18.

ANALYSE

Pour chaque lancer, la somme des faces sorties est affichée. A la fin de tous les lancers, le pourcentage d'apparition de chaque somme possible est calculé en divisant le nombre de fois où cette somme est apparue par le nombre total de lancers.

EXERCICE 4.8

PROBLÈME : dessiner sur l'écran trois stylos de couleurs préalablement choisies ; on utilisera pour dessiner la forme d'un stylo, les possibilités de l'instruction SYMBOL.

ENTRÉE : codes des 3 couleurs choisies.

SORTIE : affichage de la question demandant le choix des 3 couleurs. Affichage de trois stylos sur l'écran, chacun ayant une des couleurs choisies.

ANALYSE

En utilisant l'instruction SYMBOL, on peut à l'aide de 5 caractères graphiques, dessiner la forme du stylo. En mode 1, on ne peut avoir que 4 couleurs simultanément à l'écran et donc on ne peut représenter que 3 stylos de couleurs, l'une des couleurs étant réservée pour le fond (PAPER).

EXERCICE 4.9

PROBLÈME : secteur tournant : dessiner un secteur circulaire d'ouverture donnée ($1/20$ de la surface totale) qui tourne

sur un disque. Le secteur et le disque sont dessinés au moyen de rayons issus du centre et de longueur donnée. Plus l'écartement entre 2 rayons voisins est faible, meilleure est la définition.

ENTRÉE : le niveau de définition.

SORTIE : le secteur tournant.

ANALYSE

Dessiner le secteur en traçant successivement des rayons partant de l'origine, de longueur donnée. Si le nombre de rayons est faible, on obtient des branches séparées. Si le nombre est suffisamment grand, on obtient une surface pleine.

Pour donner l'impression du mouvement et voir le secteur tourner, il suffit de dessiner un nouveau rayon à la suite du dernier et d'effacer le premier. Si le nombre de rayons est petit (faible définition), la vitesse apparente est grande. Si le nombre de rayons est grand (grande définition), la vitesse apparente est faible.

EXERCICE 4.10

PROBLÈME : entrer l'âge (inférieur à 20 ans) d'une personne. Afficher le nombre de bougies allumées nécessaires.

ENTRÉE : l'âge de la personne.

SORTIE : dessins des bougies.

ANALYSE

Le programme teste si le nombre représentant l'âge est inférieur à 20. En utilisant l'instruction INK, le programme dessine des bougies clignotantes, disposées en rangées de 10.

EXERCICE 4.11

PROBLÈME : réaliser un quadrillage de l'écran graphique en mode haute résolution (mode 2).

ENTRÉE : largeur de la maille du quadrillage (nombre de points entre chaque ligne).

SORTIE : quadrillage de l'écran.

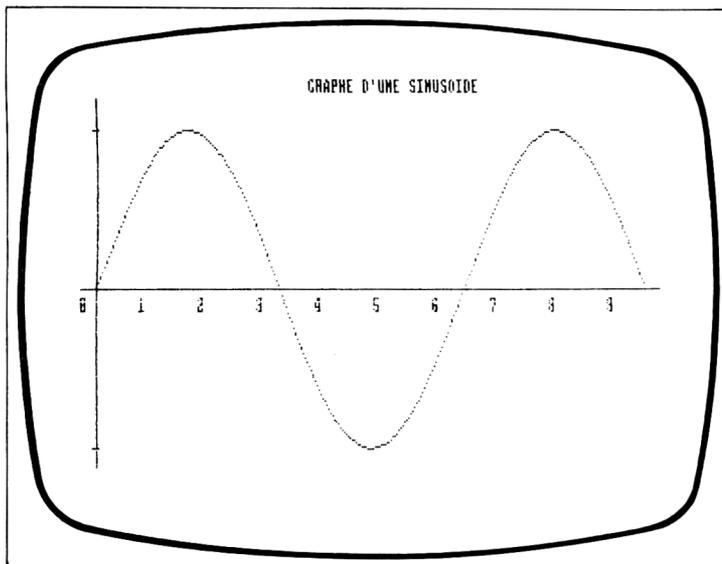
ANALYSE

Pour dessiner un quadrillage régulier, on trace des lignes verticales et horizontales séparées par la largeur de la maille.

EXERCICE 4.12

PROBLÈME : tracer en haute résolution le graphe de la fonction SINUS.

SORTIE : graphe de la fonction SINUS pour des valeurs de la variable comprise entre 0 et 3π (radian). Graduer l'axe des abscisses de 0 à 9 par pas de 1 ; graduer l'axe des ordonnées de -1 à 1 par pas de un.



ANALYSE

La graduation des axes peut être obtenue en déplaçant le curseur graphique par l'instruction MOVE et en lui attachant le curseur de caractères par l'instruction TAG. Utiliser l'instruction ORIGIN pour définir la zone d'écran où se fait l'affichage du graphe.

EXERCICE 4.13

PROBLÈME : crible d'ERATHOSTENE. Afficher suivant un carré de côtés N tous les nombres compris entre 1 et $N*N$. Barrer ensuite par une ou deux croix tous ceux qui ne sont pas premiers.

ENTRÉE : un nombre compris entre 1 et 9.

SORTIE : affichage d'un tableau carré dans lequel est inscrite une suite de nombres. Barrer par une ou deux croix, ceux qui ne sont pas premiers.

ANALYSE

Un nombre premier est un nombre entier positif qui n'admet comme diviseur que 1 et lui-même. Si un nombre N est divisible par D, alors $(N \text{ MOD } D)$ est égal à 0 puisque $(N \text{ MOD } D)$ donne le reste de la division de N par D. Pour savoir si N est premier, il suffit de vérifier qu'il n'a pas de diviseurs compris entre 1 et sa racine carrée.

EXERCICE 4.14

PROBLÈME : composition musicale : faire un programme qui demande une note à jouer sur un canal ainsi que les canaux en rendez-vous avec ce dernier. Le programme donne aussi l'état des files en attente sur les différents canaux.

ENTRÉE : réponses aux différentes questions posées.
SORTIE : affichage du mot SQ(1), SQ(2), SQ(4). Affichage des questions nécessaires pour compléter l'instruction SOUND.

ANALYSE

Pour savoir s'il est possible d'introduire de nouvelles notes dans la file d'attente, lire le mot SQ de ce canal ; calculer son reste modulo 8 (MOD 8). On obtient ainsi une valeur comprise entre 0 et 7 (inclus). En fait les seules valeurs possibles sont comprises entre 0 et 4 (inclus) et correspondent aux nombres de places encore libres dans la file d'attente de ce canal.

EXERCICE 4.15

PROBLÈME : chronomètre. Faire un programme qui réalise les fonctions suivantes :

- 1) chronométrage des secondes
- 2) remise à 0
- 3) arrêt du chronométrage
- 4) redémarrage avec ou sans prise en compte du temps écoulé.

ENTRÉE : les touches suivantes ont pour fonction :

- 1) S pour arrêter le chronomètre
- 2) C pour démarrer ou redémarrer sans prise en compte du temps écoulé
- 3) L pour redémarrer avec prise en compte du temps écoulé
- 4) Z pour la remise à 0 du chronomètre.

SORTIE : affichage dans la partie haute de l'écran des fonctions réalisées par les touches S, C, L et Z. Affichage du chronomètre.

ANALYSE

Le synchronisme des secondes est obtenu par l'instruction EVERY qui est utilisée pour exécuter un sous-programme de façon régulière.

Le temps entre deux exécutions est un multiple de 0.02 seconde. Si ce multiple est égal à 50, le programme principal est ainsi interrompu toutes les secondes et l'exécution du sous-programme a pour fonction l'incréméntation de la valeur affichée. Le programme principal, quant à lui, scrute le clavier :

- 1) Si la touche S est appuyée, un drapeau indique au sous-programme que l'affichage ne doit pas être modifié.
- 2) Si la touche C est appuyée et que le drapeau précédent est levé, le chronométrage reprend avec la valeur qui était celle au moment où la touche S a été appuyée.
- 3) Si la touche L est appuyée et que le drapeau précédent est levé, le chronométrage reprend avec la valeur qui est celle au moment où cette touche a été appuyée.
- 4) Si la touche Z est appuyée, le chronomètre est remis à zéro.

EXERCICE 4.16

PROBLÈME : détermination des paramètres de l'instruction SYMBOL.

ENTRÉE : les touches de déplacement du curseur, la touche COPY et la touche ENTER.

SORTIE : affichage d'une grille 8 sur 8. Les touches suivantes ont pour fonction :

- 1) L'appui sur les touches de déplacement déplace le curseur dans les limites de la grille.
- 2) L'appui sur la touche COPY éclaire le pavé s'il est éteint, et l'éteint s'il est éclairé.
- 3) L'appui sur la touche ENTER affiche le caractère ainsi construit, ainsi que les 8 valeurs décimales nécessaires à l'instruction SYMBOL, pour obtenir ce caractère.

ANALYSE

L'instruction INKEY\$ est utilisée pour saisir une touche au clavier. La valeur ASCII de cette touche est comparée à celles des touches de fonction du programme.

Le calcul des paramètres décimaux de l'instruction SYMBOL est expliqué ci-dessous par un exemple. Supposons que la quatrième ligne de la grille soit :

128	64	32	16	8	4	2	1
■	□	■	■	■	□	□	■

Le quatrième paramètre de l'instruction SYMBOL est alors égal à :

$$128+32+16+8+1=185$$

EXERCICE 4.17

PROBLÈME : transformer les quatre rangées du clavier en clavier musical. Les sept touches de chaque rangée sont utilisées pour jouer les sept notes d'une gamme (–1,0,1 et 2). La durée de la note est fonction de la durée d'appui sur la touche correspondante.

ENTRÉE : l'une des sept touches des quatre rangées.

SORTIE : une des 28 notes choisies.

ANALYSE

L'instruction INKEY\$ est utilisée pour savoir si une touche a été frappée. Si la touche correspond à l'une des 28 touches choisies pour jouer une note, un son de durée brève est émis. Les valeurs de période des gammes sont passés par une liste de DATA. Pour établir la correspondance entre la touche frappée et la note à jouer, on peut :

- 1) ranger les notes dans un tableau
- 2) calculer par l'instruction INSTR le rang de la touche frappée dans la chaîne de caractères constituée par les 28 touches choisies du clavier.
- 3) jouer la note de même rang dans le tableau.

EXERCICE 4.18

PROBLÈME : programme pédagogique d'enseignement de l'addition. Une addition de 2 nombres (chacun compris entre 1 et 1000) est affichée en mode 0. Pour chaque colonne, l'élève doit trouver le chiffre à mettre en dessous du trait d'addition et la retenue s'il y a lieu. Puis par appui sur la barre d'espacement, l'élève peut découvrir le résultat.

ENTRÉE : deux nombres, compris entre 1 et 1000. La barre d'espacement.

SORTIE : à chaque appui sur la barre d'espacement, le résultat de la colonne s'inscrit en dessous du trait d'addition. La retenue, s'il y en a une, s'affiche dans la colonne suivante en clignotant.

ANALYSE

Les chiffres des unités sont isolés puis additionnés. Le résultat partiel obtenu est affiché dans la colonne des unités et, dans le cas où il y a une retenue, elle est affichée dans la colonne des dizaines. Puis on recommence avec les deux chiffres des dizaines, puis avec ceux des centaines. Pour isoler dans un chiffre (unité, dizaine, centaine) on peut, par exemple, utiliser l'instruction MOD ou traiter ce nombre comme une chaîne de caractères.

CHAPITRE V

*JEUX GRAPHIQUES
(Tir à la cible, télécran,
glouton, loto sportif,
loto, machine à sous,
pendu, calcul mental,
mastermind,
leçon de géographie,
tours d'Hanoi,
répertoire
téléphonique)*

EXERCICE 5.1 : *TIR SUR CIBLE*

PROBLÈME : un projectile, tiré par une pression sur la barre d'espace, part du bas de l'écran pour atteindre une cible située en haut de l'écran. Cette cible comporte 6 cases numérotées de 1 à 6. Le programme comptabilise le nombre de points marqués (score), et s'arrête quand le nombre de coups joués atteint la valeur 20 ou par appui sur la touche ENTER.

ENTRÉE : touches frappées au clavier.

SORTIE : affichage de la cible constituée de 6 cases, utilisant deux couleurs en alternance et numérotées de 1 à 6. Affichage du projectile se déplaçant vers la cible en effectuant des écarts aléatoires sur la gauche et sur la droite. Affichage du score et du nombre de coups joués.

ANALYSE

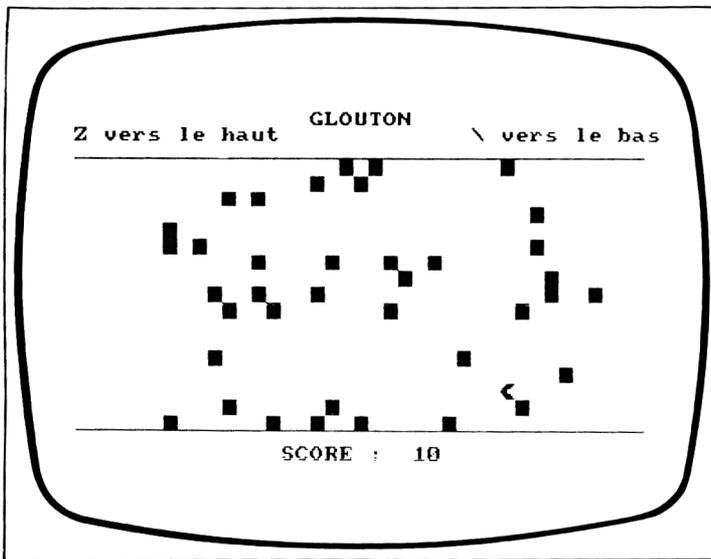
L'affichage de la cible se fait par l'instruction WINDOW qui permet de définir six fenêtres que l'on colorie par l'instruction CLS. L'instruction INKEY\$ permet de saisir un caractère au clavier. Si ce caractère est un espace, le projectile est tiré d'un point situé en bas de l'écran et à une position horizontale, choisie arbitrairement par l'instruction RND. Le projectile se déplace ensuite sur l'écran en effectuant des écarts de une à trois positions, sur la gauche ou sur la droite. Quand le projectile a atteint la cible, sa position horizontale permet de déterminer la case touchée.

EXERCICE 5.2 : *JEU DU GLOUTON*

PROBLÈME : un glouton se dirige du bord gauche vers le bord droit de l'écran en émettant un signal sonore (beep). Le trajet est parsemé d'obstacles, répartis aléatoirement sur l'écran. Le but du jeu est d'atteindre le bord droit en évitant ces obstacles : pour cela, le joueur peut faire monter ou descendre le glouton à l'aide de deux touches du clavier.

ENTRÉE : la barre d'espace, la touche Z pour monter et la touche \ pour descendre.

SORTIE : l'écran parsemé d'obstacles. Le glouton se dirigeant du bord gauche vers le bord droit en émettant un beep.



ANALYSE

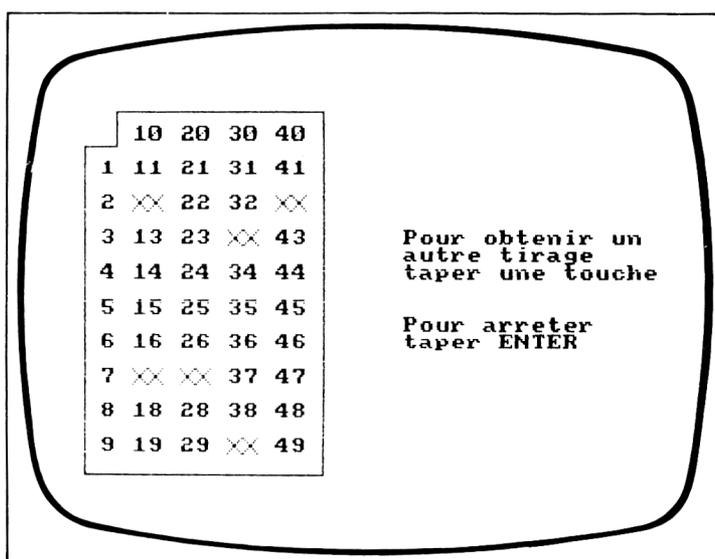
Le nombre d'obstacles sur la zone de jeu croît en fonction du niveau de difficulté. L'utilisation de 2 caractères graphiques, définis par l'instruction SYMBOL, permet de représenter successivement le glouton la bouche ouverte puis la bouche fermée. Pour tester si le glouton a rencontré un obstacle, on utilise l'instruction TEST qui renvoie le code couleur d'un point de l'écran.

EXERCICE 5.3 : LOTO NATIONAL

PROBLÈME : faire un programme qui affiche une grille du loto national et coche les 6 numéros obtenus par tirage aléatoire.

ENTRÉE : une touche est attendue au clavier pour obtenir un nouveau tirage.

SORTIE : la grille du loto national, représentée par un rectangle de couleur jaune sur lequel sont inscrits en rouge les 49 premiers nombres. Les six nombres tirés aléatoirement sont cochés sur cette grille à l'aide d'une croix.



ANALYSE

La grille du loto national comprend les 49 premiers nombres, inscrits en rouge sur fond jaune suivant 5 colonnes de 10 nombres. Si les nombres tirés au hasard sont compris entre 1 et 9, ils sont cochés sur la grille à l'aide d'une croix. S'ils sont compris entre 10 et 49, ils sont cochés à l'aide de deux croix. La croix est obtenue à partir de l'instruction SYMBOL. La grille complète du loto est définie par l'instruction WINDOW. Cela facilite ensuite la détermination de l'emplacement d'un numéro dans cette grille.

EXERCICE 5.4 : *CALCUL MENTAL*

PROBLÈME : le programme choisit de façon aléatoire :

- 1) deux nombres entiers compris entre 1 et 1000
- 2) une opération : addition, soustraction ou multiplication.

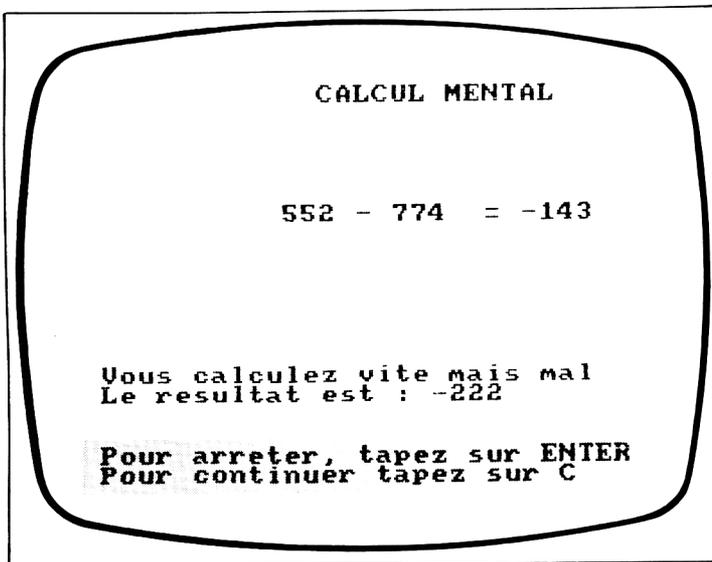
La réponse du joueur est vérifiée et doit être donnée dans un temps limité à 10 secondes.

ENTRÉE : la réponse du joueur.

SORTIE : l'opération à faire.
L'un des 3 messages suivants :

- 1) Vous calculez vite et bien
- 2) Vous calculez vite mais mal
- 3) Vous calculez trop lentement.

Le résultat exact s'il y a lieu.

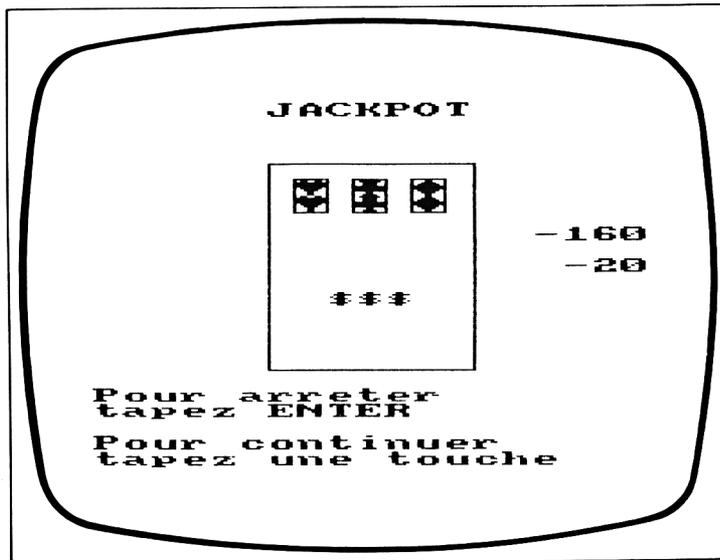


ANALYSE

Suivant l'opération tirée, l'un des 3 sous-programmes (addition, soustraction, multiplication) est exécuté. Le branchement à ce sous-programme est obtenu simplement par l'instruction ON... GOSUB... L'instruction AFTER permet d'interrompre un programme lorsque le temps écoulé est supérieur à une valeur fixée. Toutefois, AMSTRAD n'interrompt jamais une instruction en cours, mais attend son achèvement, avant d'aller exécuter le sous-programme associé à l'instruction AFTER. Par conséquent, il est impossible d'interrompre l'instruction INPUT et donc d'utiliser cette instruction pour entrer une réponse en un temps limité. Le moyen de tourner la difficulté est de saisir la réponse par l'instruction INKEY\$.

EXERCICE 5.5 : MACHINE A SOUS

- PROBLÈME** : réaliser en mode 0 un programme simulant une machine à sous.
- ENTRÉE** : une touche pour lancer la machine.
- SORTIE** : affichage de la machine et du score en mode 0.



ANALYSE

Une machine à sous est constituée de 3 fenêtres. Dans chacune d'elles défile un symbole : pique, cœur ou carreau. Chaque symbole peut prendre une des trois couleurs bleu, blanc ou rouge. Cela fait au total 729 combinaisons différentes. Le joueur marque 1000 points si la combinaison tirée au hasard est : cœur bleu, cœur blanc et cœur rouge. Le joueur marque 500 points s'il tire l'une des 5 configurations différentes de la précédente et constituée d'un cœur bleu, d'un cœur blanc et d'un cœur rouge. Dans tous les autres cas, il perd 20 points. Pour obtenir une impression de défilement des symboles qui apparaissent dans les fenêtres, on effectue un grand nombre d'affichages rapides avant l'arrêt de la machine (affichage de la combinaison tirée). Pour représenter la combinaison tirée au hasard, on numérote de 1 à 9 les différents symboles :

- 0 pour pique de couleur rouge
- 1 pour pique de couleur blanc
- 2 pour pique de couleur bleu
- 3 pour carreau de couleur rouge
- 4 pour carreau de couleur blanc
- 5 pour carreau de couleur bleu
- 6 pour cœur de couleur rouge
- 7 pour cœur de couleur blanc
- 8 pour cœur de couleur bleu

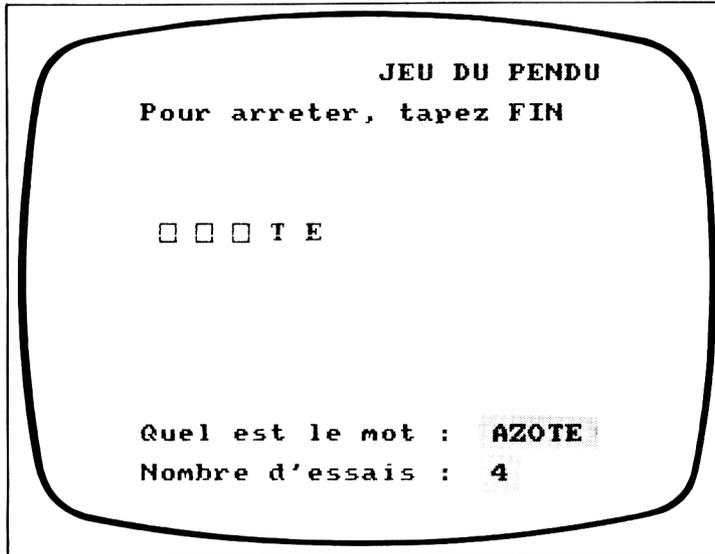
Ainsi le nombre 753 représente la suite des trois symboles : cœur blanc, carreau bleu et carreau rouge.

EXERCICE 5.6 : JEU DU PENDU

PROBLÈME : trouver un mot caché en proposant des mots de même longueur.

ENTRÉE : les mots proposés.

SORTIE : affichage, pendant un temps limité, de la liste des mots cachés, puis de l'écran de jeu. Une partie de l'écran est utilisée pour afficher successivement les lettres découvertes du mot caché. Une autre partie de l'écran est utilisée pour entrer les réponses du joueur.



ANALYSE

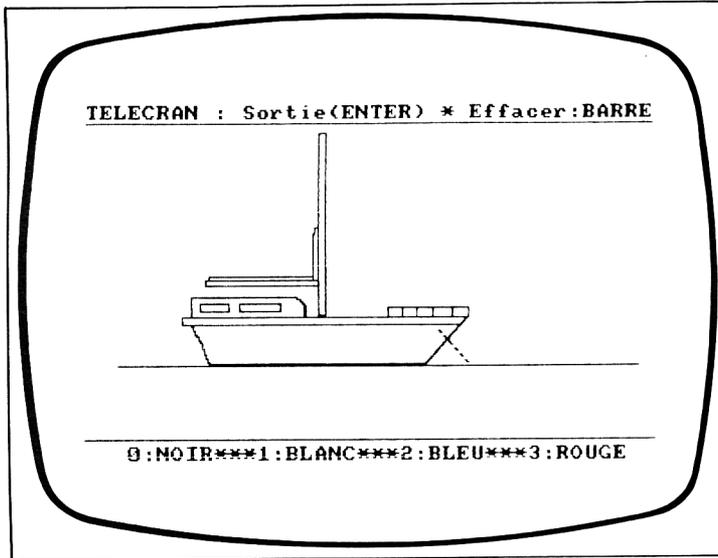
La liste des mots cachés est donnée par une ou plusieurs instructions DATA. Le mot proposé est entré au moyen de l'instruction INPUT. Cette entrée est faite dans une fenêtre d'écran définie par l'instruction WINDOW. Si une des lettres du mot proposé est en bonne position (utiliser l'instruction MID\$), cette lettre s'affiche dans la fenêtre d'écran réservée au mot caché.

EXERCICE 5.7 : TÉLÉCRAN

PROBLÈME : réalisation d'un télécran graphique à 4 couleurs.

ENTRÉE : les touches de déplacement du curseur, les touches 0, 1, 2, et 3, la barre d'espace et la touche ENTER.

SORTIE : les touches de déplacement du curseur sont utilisées pour dessiner sur l'écran à l'aide d'un point lumineux (prévoir la couleur noire pour effacer).



ANALYSE

L'instruction INKEY\$ permet de saisir une touche du clavier. L'instruction SPEED KEY permet d'accélérer la vitesse de répétition d'une touche appuyée en continu. Quand le joueur appuie sur l'une des touches 0, 1, 2 et 3, le point lumineux s'allume. La touche ENTER fait sortir du programme et rétablit dans la couleur correspondante une vitesse normale par l'instruction SPEED KEY.

EXERCICE 5.8 : LEÇON DE GÉOGRAPHIE

PROBLÈME : afficher la carte de France. Puis positionner un point lumineux clignotant. Le joueur doit donner le nom de la ville repérée par ce point lumineux clignotant.

ENTRÉE : le nom de la ville.

SORTIE : affichage de la carte de France, d'un point clignotant, de la réponse d'AMSTRAD.



ANALYSE

Les coordonnées du contour de la carte de France sont enregistrées sous forme d'une liste d'instructions DATA. Pour déterminer l'emplacement des points successifs, on peut dessiner la carte de France sur une feuille de papier quadrillée de 640×400 pavés puis relever ensuite les coordonnées des points sur cette feuille. On fait de même pour les différentes villes à placer. L'instruction RND est utilisée pour tirer une ville au hasard parmi celles qui ont été mises dans la liste.

EXERCICE 5.9 : LOTO SPORTIF

PROBLÈME : sept coureurs à pied effectuent une course le long de sept couloirs différents. Chaque couloir est repéré par son numéro. Pour afficher un joueur, on utilise un caractère graphique construit par l'instruction SYMBOL. Le but du jeu est de pronostiquer le tiercé gagnant.

ENTRÉE : le tiercé joué.

SORTIE : affichage de la piste, déroulement de la course, affichage de l'ordre d'arrivée.

ANALYSE

L'affichage et le déroulement de la course dans les sept couloirs sont facilités par l'utilisation de l'instruction WINDOW. Le programme saisit ensuite le pronostic du joueur et en vérifie la validité (les numéros proposés doivent être entiers, compris entre 1 et 7 et tous différents). L'instruction RND permet de simuler l'avance aléatoire de chaque coureur dans son couloir. Les trois premiers coureurs qui franchissent la ligne d'arrivée sont affichés et le résultat est comparé au tiercé joué.

EXERCICE 5.10 : RÉPERTOIRE TÉLÉPHONIQUE

PROBLÈME : écrire un programme qui effectue sur un fichier de données les opérations suivantes :

- 1) Entrée du fichier à partir du lecteur
- 2) Affichage du fichier à l'écran
- 3) Insertion d'un élément dans le fichier
- 4) Suppression d'un élément dans le fichier
- 5) Enregistrement du fichier sur le lecteur
- 6) Sortie du programme.

ENTRÉE : l'un des codes du menu proposé. Informations contenues dans une fiche du répertoire. Lecteur de cassettes ou de disquettes.

SORTIE : affichage du menu, du fichier total, différents messages indiquant à l'utilisateur la marche à suivre. Lecteur de cassettes ou de disquettes.



```
*****
*   Lecture du fichier :           1   *
*   Affichage du fichier :        3   *
*   Insertion d'une fiche :       4   *
*   Suppression d'une fiche :     4   *
*   Enregistrement du fichier :  5   *
*   Fin :                          6   *
*****
Operation choisie (1-6) : 2
1 : Denise 375 48 44
2 : Pierre 356 75 68
3 : Dominique 356 54 36
4 : Marcel 425 46 76
5 : Emmanuel 425 58 86

Pour revenir au menu, tapez une touche
```

ANALYSE

L'écran sera divisé en trois fenêtres d'affichage :

- 1) La fenêtre supérieure sert à afficher le menu.
- 2) La fenêtre centrale sert à effectuer les différentes opérations sur le fichier.
- 3) La fenêtre inférieure sert à inscrire les différents messages de communication avec l'utilisateur (exemples : "SUPPRESSION IMPOSSIBLE", "Pour revenir au menu, tapez une touche", ...).

Pour lire un fichier, il faut au préalable ouvrir le fichier par l'instruction OPENIN, puis effectuer les lectures au moyen de l'instruction INPUT #9 et enfin fermer le fichier par l'instruction CLOSEIN. L'instruction EOF permet, lors de la lecture, de déterminer de façon indirecte le nombre d'enregistrements du fichier.

De même pour écrire un fichier, il faut au préalable ouvrir le fichier par l'instruction OPENOUT, puis effectuer les écritures au moyen de l'instruction PRINT #9 et enfin fermer le fichier par l'instruction CLOSEOUT. Lors de l'insertion d'un nouvel élément, il faut incrémenter le nombre total d'éléments.

La suppression d'un élément est obtenue en indiquant le numéro de l'élément à supprimer. Dans le cas où la suppression est impossible,

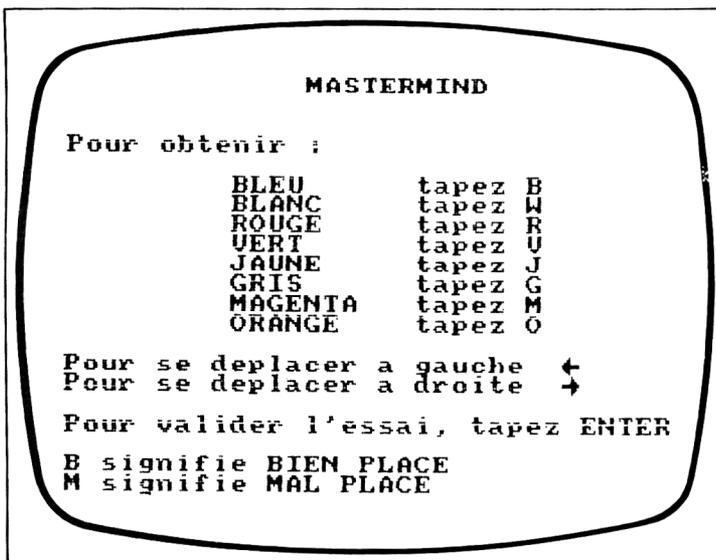
un message d'erreur s'inscrit. L'instruction ON ... GOSUB permet, en fonction du numéro de menu choisi, de se brancher sur les différents sous-programmes.

EXERCICE 5.11 : MASTERMIND

PROBLÈME : l'ordinateur choisit une combinaison de cinq pions de couleurs prises dans un ensemble de huit couleurs que le joueur doit découvrir en proposant des essais successifs.

ENTRÉE : les touches de déplacement horizontal, la touche ENTER, les différents essais entrés par la lettre initiale de la couleur choisie (B : bleu, W : (en anglais) blanc, R : rouge, V : vert, J : jaune, G : gris, M : magenta, O : orange).

SORTIE : un premier écran est utilisé pour afficher les règles du jeu. Un second écran est utilisé pour afficher le nombre de pions bien et mal placés et la configuration cachée après 7 essais infructueux.



ANALYSE

Le nombre de couleurs utilisées (8) nécessite l'affichage en mode 0.

La combinaison à découvrir est créée de façon aléatoire sous la forme d'un tableau de cinq valeurs représentant chacune une couleur. L'essai proposé est constitué d'un tableau de même structure. Par exemple, la suite 5, 2, 3, 3, 1 représente la suite des 5 pions jaune, blanc, rouge, rouge et bleu. Le problème consiste alors à comparer un à un les éléments de ces deux tableaux.

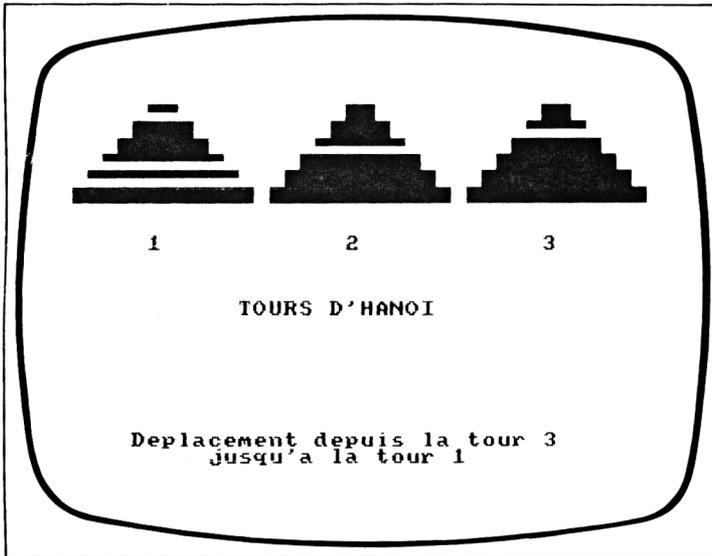
EXERCICE 5.12 : TOURS D'HANOÏ

PROBLÈME : le jeu est constitué de trois tours coniques, dont la première comporte 5 anneaux de tailles différentes. Le but du jeu est de faire passer les 5 anneaux sur la troisième tour en s'aidant des 2 autres. Le joueur peut déplacer uniquement l'anneau supérieur d'une tour et l'amener sur une autre tour, à condition de ne pas superposer un grand anneau sur un petit.

ENTRÉE : le numéro de la tour de départ et le numéro de la tour d'arrivée.

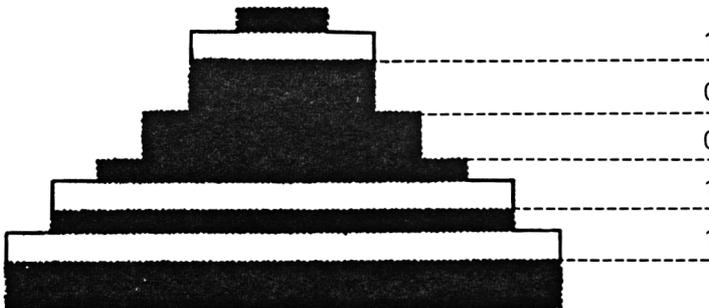
SORTIE : affichage des 3 tours et des 5 anneaux sur la première. Déplacement des anneaux en fonction de la réponse du joueur.





ANALYSE

Chaque tour peut être caractérisée par un nombre binaire à 5 chiffres. Un 0 représente un niveau sans anneau et un 1 représente un niveau occupé par un anneau. Par exemple, la tour ci-dessous est caractérisée par le nombre 10011.



Pour vérifier si le joueur a gagné, il faut tester le nombre binaire associé à la tour 3, en le comparant à 11111.

*DEUXIÈME
PARTIE*

**SOLUTION
DES
EXERCICES**

EXERCICE 1.1

PROGRAMME

Solution 1 :

```
10 CLS
20 INPUT "QUEL EST VOTRE NOM ";N$
30 INPUT "QUEL EST VOTRE PRENOM ";P$
40 PRINT:PRINT
50 PRINT TAB(10)LEFT$(N$,1);". ";LEFT$(P$,1);"
  ."
60 PRINT
70 PRINT "LONGUEUR DU NOM :";LEN(N$)
80 END
```

VARIABLES

N\$ nom
P\$ prénom

COMMENTAIRES

40-60 : passe des lignes pour faciliter la lecture.
50 : affichage, à partir de la colonne 10, des initiales suivies d'un point.

Solution 2 :

```
10 CLS
20 INPUT "QUEL EST VOTRE NOM ";N$
30 INPUT "QUEL EST VOTRE PRENOM ";P$
40 PRINT:PRINT
50 PRINT USING "!.";N$;P$
60 PRINT
70 PRINT "LONGUEUR DU NOM :";LEN(N$)
80 END
```

50 : une autre solution consiste à utiliser l'instruction PRINT avec une description (format) de la ligne à afficher. Cette description s'obtient en utilisant l'instruction USING qui indique à l'instruction PRINT de quelle façon doit se faire

l'affichage de N\$ et P\$. Cette description est mise entre guillemets (" ") : le signe ! indique que l'on doit prendre uniquement la première lettre de N\$, puis que l'on doit afficher le point. Comme le format est épuisé, on repart du début du format pour afficher P\$.

EXERCICE 1.2

PROGRAMME

```

10 CLS
20 INPUT "QUEL EST TON PRENOM ";P$
30 PRINT
40 INPUT "QUELLE EST TON ANNEE DE NAISSANCE "
   ;A
50 PRINT
60 PRINT "TON ANNIVERSAIRE"
70 INPUT "EST-IL DEJA PASSE (O/N) ";R$
80 IF R$="N" THEN A=A+1
90 MODE 0
100 PRINT TAB(3)"EH BIEN BONJOUR "
110 PRINT
120 PRINT TAB(10-LEN(P$)/2)P$
130 PRINT
140 PRINT " JE NE REPETERAI A PERSONNE QUE
TU AS"
150 PRINT TAB(5)"AUJOURD'HUI"
160 PRINT:PRINT
170 PRINT TAB(8)1985-A
180 PRINT:PRINT
190 PRINT TAB(9)"ANS"
200 IF INKEY$="" THEN 200
210 MODE 1

```

VARIABLES

A année de naissance
P\$ prénom
R\$ réponse

COMMENTAIRES

- 50 : saute une ligne pour faciliter la lecture.
- 90 : MODE 0 provoque l'affichage de caractères de taille double.
- 170 : (1985-A) donne l'âge (remplacer éventuellement 1985 par l'année actuelle).
- 200 : la frappe d'une touche quelconque arrête le programme et l'affichage repasse en mode 1 (ligne 210).

EXERCICE 1.3

PROGRAMME

```
10 REM
20 REM Affichage d'une ligne et
30 REM d'une colonne d'etoiles
40 REM *****
50 CLS
60 INPUT "Numero de ligne (entre 1 et 25) : "
,LIG
70 IF LIG<1 OR LIG>25 OR LIG<>INT(LIG) THEN 60
80 INPUT "Numero de colonne (entre 1 et 40) : "
,COL
90 IF COL<1 OR COL>40 OR COL<>INT(COL) THEN 80
100 CLS
110 LOCATE 1,1
120 FOR I=1 TO 25
130 LOCATE COL,I:PRINT "*";
140 NEXT I
150 LOCATE 1,LIG:PRINT STRING$(40,"*");
160 IF INKEY$="" THEN 160
170 END
```

VARIABLES

LIG numéro de ligne
COL numéro de colonne

COMMENTAIRES

110-140 : affichage de la colonne d'étoiles.
 150 : affichage de la ligne d'étoiles.
 160 : pour éviter une remontée des lignes et l'effacement partiel de l'écran, provoqué par le message READY, une touche est attendue au clavier pour arrêter le programme. Puisque cette touche n'importe pas, elle n'est pas conservée à travers une variable.

EXERCICE 1.4**PROGRAMME**

```

10 REM
20 REM Affichage des codes ASCII
30 REM *****
40 CLS
50 PRINT TAB(7)"Affichage des codes ASCII"
60 PRINT:PRINT TAB(12)"Tapez une touche":PRIN
T
70 A$=INKEY$:IF A$="" THEN 70
80 PRINT TAB(16)A$;" :";ASC(A$)
90 GOTO 70

```

VARIABLE

A\$ touche frappée au clavier

COMMENTAIRES

70 : la touche frappée au clavier est conservée dans A\$, puis comparée à la chaîne vide. Si le résultat est vrai, le programme boucle en 70. Dans le cas contraire, le code ASCII de A\$ est affiché.

Remarque : pour sortir du programme, tapez deux fois sur la touche ESC.

EXERCICE 1.5

PROGRAMME

```
10 CLS
20 INPUT "QUEL NOMBRE";N
30 INPUT "QUEL TAUX DE POURCENTAGE";P
40 PRINT
50 DN=N*P/100
60 PRINT "VARIATION :";DN
70 PRINT "VALEUR FINALE :";N+DN
80 END
```

VARIABLES

N valeur numérique
P taux de pourcentage
DN pourcentage

COMMENTAIRES

50 : calcul de la variation.
70 : affichage de la valeur numérique finale.

EXERCICE 1.6

PROGRAMME

```
10 CLS
20 INPUT "QUEL EST VOTRE PRENOM";P$
30 INPUT "COMBIEN DEPENSEZ-VOUS QUOTIDIENNE
MENT POUR VOTRE NOURRITURE";D
40 DA=D*365*1.07
50 CLS
60 PRINT "HE BIEN BONJOUR ";P$
70 PRINT:PRINT
80 PRINT "VOUS DEPENSEZ A PEU PRES";DA;"FRANC
S DE NOURRITURE PAR AN"
90 LOCATE 7,10
100 PRINT "MOI JE DEPENSE 0 CENTIME !"
110 LOCATE 1,24
120 END
```

VARIABLES

D dépense quotidienne
DA dépense annuelle
P\$ prénom

COMMENTAIRES

40 : calcul de la dépense annuelle.
110 : positionnement du curseur à la ligne 24, pour éviter l'apparition du message Ready sous la dernière ligne affichée.

EXERCICE 1.7

PROGRAMME

```
10 CLS
20 REM ENTREE DU NOMBRE A DECOUVRIR
30 PRINT "QUEL NOMBRE ENTIER ENTRE 0 ET 1000"
40 INPUT "VOULEZ-VOUS CACHER ";N
50 IF N-INT(N)<>0 OR N<0 OR N>1000 THEN 30
60 REM DEBUT DU JEU
70 CLS
80 PRINT
90 INPUT "ESSAI ";E
100 IF E>N THEN PRINT "TROP GRAND":GOTO 80
110 IF E<N THEN PRINT "TROP PETIT":GOTO 80
120 PRINT "BRAVO VOUS AVEZ TROUVE !"
130 END
```

VARIABLES

N nombre à découvrir
E essai

COMMENTAIRES

50 : le nombre à découvrir est testé sur les 3 conditions suivantes :
- entier
- supérieur ou égal à 0

- inférieur ou égal à 1000.

S'il est inférieur à 0 ou supérieur à 1000, un autre nombre est demandé.

80 : passe une ligne entre chaque essai.

120 : si les résultats des deux tests précédents sont faux (les deux nombres étant égaux), la ligne 120 est exécutée.

EXERCICE 1.8

PROGRAMME

```
10 CLS
20 D=INT(6*RND)+1
30 MODE 0
40 LOCATE 8,7
50 PRINT D
60 LOCATE 1,18
70 PRINT "VOULEZ-VOUS LANCER"
80 PRINT "LE DE` A NOUVEAU"
90 INPUT "(OUI/NON) ";A$
100 PRINT
110 IF A$="OUI" THEN 20
120 MODE 1
130 END
```

VARIABLES

D nombre entier aléatoire entre 1 et 6
A\$ réponse du joueur

COMMENTAIRES

20 : tirage du nombre aléatoire.

70-90 : question posée au joueur (les deux réponses possibles sont incluses dans le texte affiché).

110 : si l'utilisateur a répondu OUI, alors l'exécution du programme continue à la ligne 20.

EXERCICE 1.9**PROGRAMME**

```

10 CLS
20 PRINT "NOMBRE DE DES LANCES (1 A 8)
30 INPUT "O - POUR ARRETER ";N
40 PRINT
50 IF N=0 THEN END
60 IF N<2 OR N>8 THEN 20
70 S=0
80 FOR I=1 TO N
90 S=S+INT(6*RND)+1
100 NEXT I
110 PRINT
120 PRINT TAB(8)"SOMME = ";S
130 PRINT
140 GOTO 20

```

VARIABLES

I compteur
N nombre de dés à lancer
S somme des nombres sortis pour un lancer

COMMENTAIRES

50 : si N = 0, le programme s'arrête.
60 : cette ligne doit se trouver après la ligne 50, sinon le programme ne s'arrêterait pas pour N = 0.

Remarque : pour calculer la somme du lancer de N dés, la variable S est initialisée à 0 en 70. Elle est ensuite augmentée, dans la boucle 80-100, de la valeur obtenue à chacun des N lancers de dés.

EXERCICE 1.10

PROGRAMME

```
10 CLS
20 REM INITIALISATION DES VARIABLES
30 N=INT(1000*RND)+1
40 C=0
45 PRINT:PRINT "Trouvez un nombre entre 1 et
1000":PRINT
50 REM DEBUT DU JEU
60 C=C+1
70 PRINT
80 PRINT "ESSAI N. ";C;:INPUT E
90 IF E>N THEN PRINT "TROP GRAND":GOTO 50
100 IF E<N THEN PRINT "TROP PETIT":GOTO 50
110 PRINT:PRINT
120 PRINT "BRAVO VOUS AVEZ TROUVE EN ";C;:PRINT
UP(S)"
130 PRINT:PRINT
140 END
```

VARIABLES

C compteur des essais
N nombre à découvrir
E numéro de l'essai

COMMENTAIRES

30 : un nombre entier est choisi au hasard entre 1 et 1000.
60 : le compteur C est augmenté de 1 (il est **incrémenté**). Ne pas oublier de le mettre à 0 en début de programme (ligne 40).
110 : deux lignes blanches sont passées. Cette ligne et les suivantes ne seront exécutées que si les résultats des deux tests précédents sont faux.
120 : C contient le nombre d'essais.

EXERCICE 1.11**PROGRAMME**

```

10 REM
20 REM JEU DES LETTRES
30 CLS
40 CO$="BCDFGHJKLMNPQRSTVWXZ"
50 VO$="AEIOUY"
60 RANDOMIZE(TIME)
70 PRINT "POUR TIRER UNE CONSONNE, TAPEZ C"
80 PRINT "POUR TIRER UNE VOYELLE, TAPEZ V"
90 LOCATE 11,12
100 FOR I=1 TO 9
110 REP$=INKEY$
120 IF REP$<>"C" AND REP$<>"V" THEN 110
130 IF REP$="C" THEN AT$=MID$(CO$,20*RND+0.5,
1) ELSE AT$=MID$(VO$,6*RND+0.5,1)
140 PRINT AT$" ";
150 NEXT I
160 PRINT
170 LOCATE 1,22:END

```

VARIABLES

I compteur
CO\$ chaîne de consonnes
VO\$ chaîne de voyelles
REP\$ réponse du joueur
N nombre tiré au hasard
AT\$ lettre tirée

COMMENTAIRES

40 : la chaîne alphanumérique CO\$ contient l'ensemble des 20
 consonnes.
50 : la chaîne alphanumérique VO\$ contient l'ensemble des 6
 voyelles.
60 : la fonction RND est initialisée de façon aléatoire par la
 fonction RANDOMIZE ; le paramètre TIME est une valeur
 stockée dans votre ordinateur AMSTRAD qui s'incrémente

- 130 : (+1) tous les $1/300^{\circ}$ de seconde. Il est impossible de prévoir cette valeur à un instant donné.
: si la réponse est "C", la lettre tirée est prise au hasard parmi les 20 lettres de la chaîne CO\$; en effet, l'instruction MID\$(CO\$,20*RND+0.5,1) extrait de CO\$ la lettre dont la position est la valeur arrondie de $20*RND+0.5$. De même, si la réponse est "V", la lettre tirée au hasard est prise parmi les 6 lettres de la chaîne VO\$.

EXERCICE 1.12

PROGRAMME

```
10 CLS
20 LOCATE 1,10
30 INPUT "RAYON DU CERCLE ";RAYON
40 PERI=2*PI*RAYON
50 AIRE=PI*RAYON*RAYON
60 PRINT
70 PRINT "LE PERIMETRE EST EGAL A ";PERI
80 PRINT
90 PRINT "LA SURFACE EST EGALE A ";AIRE
100 PRINT
110 END
```

VARIABLES

RAYON rayon du cercle
PERI périmètre du cercle
AIRE surface du cercle

COMMENTAIRES

40-50 : PI est un nombre stocké dans votre ordinateur AMS-TRAD ; il a pour valeur 3,14159265.

EXERCICE 1.13**PROGRAMME**

```

10 CLS
20 REM ENTREE DES DONNEES
30 INPUT "NOM DU PAYS : ",P$
35 PRINT
40 INPUT "SUPERFICIE (en km2) : ",S
50 INPUT "POPULATION (en millions d'hab.) : "
   ,P
60 REM CALCUL DE LA DENSITE
70 D=P*1000000/S
80 REM
90 REM AFFICHAGE
100 PRINT
110 PRINT "Pays";TAB(14);"Densite (hab./km2)"
120 PRINT STRING$(4,"-");TAB(14);STRING$(18,"
-")
130 PRINT P$;TAB(17);USING "####.####";D
140 PRINT
150 END

```

VARIABLES

P\$ nom du pays
S superficie du pays
P population du pays
D densité

COMMENTAIRES

70 : D donne la population en habitants par km².
120 : l'instruction STRING\$(l,"-") construit une chaîne de l tirets qui servent ici à souligner les mots Pays et Densité (hab./km²).
130 : P\$ est tout d'abord affiché, puis D s'inscrit, à partir de la colonne 17, sur 4 positions pour la partie entière et 4 positions pour la partie décimale. Si D est supérieur à 10000, le signe % s'inscrit dans l'affichage, pour indiquer l'impossibilité de cadrer sur 4 positions la partie entière (essayer par exemple S = 100 et P = 1).

EXERCICE 1.14

PROGRAMME

```
10 REM
20 CLS
30 INPUT "Nombre de valeurs a comparer : ",N
40 DIM D(N)
50 REM
60 REM ENTREE DES VALEURS
70 FOR I=1 TO N
80 PRINT "Valeur ";I;
90 INPUT " : ",D(I)
100 NEXT I
110 REM
120 REM RECHERCHE DU MINIMUM
125 REM ET DU MAXIMUM
130 MI=D(1):MA=D(1)
140 FOR I=1 TO N
150 MI=MIN(MI,D(I))
160 MA=MAX(MA,D(I))
170 NEXT I
180 REM
190 REM AFFICHAGE
200 PRINT:PRINT
210 PRINT "La plus petite valeur est ";MI
220 PRINT "La plus grande valeur est ";MA
230 PRINT:PRINT
240 END
```

VARIABLES

I compteur
N nombre de valeurs à comparer
D(N) tableau des valeurs
MI plus petite valeur de D
MA plus grande valeur de D

COMMENTAIRES

130 : on initialise les valeurs de MI et MA en leur affectant la première valeur du tableau D ; on évite ainsi l'erreur de

- 140 : démarrer les comparaisons avec la valeur 0 qui peut être plus grande ou plus petite que les valeurs de D.
: on peut démarrer le compteur avec la valeur 2. Toutefois en commençant avec 1, on évite d'avoir une erreur lorsque $N = 1$.

EXERCICE 1.15

PROGRAMME

```

10 CLS
20 INPUT "Quel est votre nom ",N$
30 INPUT "Quel est votre prenom ",P$
40 REM
50 N$=UPPER$(LEFT$(N$,1))+LOWER$(MID$(N$,2))
60 P$=UPPER$(LEFT$(P$,1))+LOWER$(MID$(P$,2))
70 REM
80 LOCATE 19-0.5*LEN(N$+P$),10
90 PRINT N$+" "+P$
100 LOCATE 1,20:END

```

VARIABLES

N\$ nom de la personne
P\$ prénom de la personne

COMMENTAIRES

- 50-60 : l'instruction UPPER\$ transforme une chaîne de caractères en lettres majuscules. Les lettres déjà en majuscules sont conservées. De même, LOWER\$ transforme une chaîne de caractères en lettres minuscules. Les lettres déjà en minuscules sont conservées.

EXERCICE 1.16

PROGRAMME

```

10 REM
20 CLS
30 ON ERROR GOTO 160
40 REM
50 PRINT "Ce programme traite une erreur prod
uite lors de la division de A par B."
60 PRINT "Par exemple division par ZERO"
70 PRINT:PRINT
80 INPUT "A = ",A
90 INPUT "B = ",B
100 C=A/B
110 PRINT:PRINT
120 PRINT "A/B =";C
130 PRINT:PRINT "Tapez ENTER pour arreter":PR
INT "Tapez une touche pour recommencer"
135 A$=INKEY$:IF A$="" THEN 135
140 IF A$=CHR$(32) THEN 20
150 END
160 REM Programme de traitement de
170 REM l'erreur
180 REM *****
190 PRINT:PRINT
200 PRINT "L'erreur de code";ERR;"s'est"
210 PRINT "produite a la ligne";ERL
220 RESUME 130

```

VARIABLES

A dividende
 B diviseur
 C quotient
 A\$ touche frappée au clavier

COMMENTAIRES

30 : si une erreur se produit lors de l'exécution du programme,
 le programme se poursuit à la ligne 160.

- 200 : ERR contient le numéro de code de l'erreur. En standard, ce numéro est compris entre 1 et 31 et sa signification est donnée dans la documentation de votre ordinateur ; par exemple 11 signifie : essai de division par 0.
- 210 : ERL contient le numéro de la ligne de programme où l'erreur s'est produite.
- 220 : RESUME indique la ligne où l'exécution du programme doit reprendre, après traitement de l'erreur.

EXERCICE 1.17

PROGRAMME

```

10 REM
20 REM Deux appuis sur la touche BREAK
30 REM provoque un branchement en 330
40 REM *****
50 ON BREAK GOSUB 330
60 CLS
70 INK 2,2:INK 3,7
80 REM
90 REM Definition de la fenetre 0
100 REM *****
110 PAPER 0
120 WINDOW 1,40,17,24
130 CLS
140 REM
150 REM Definition des 6 fenetres
160 REM *****
170 FOR I=0 TO 5
180 K=I MOD 3
190 WINDOW #I+1,13*K+1,13*K+13,8*INT((I-K)/2)
+1,8*INT((I-K)/2)+8
200 PAPER #I+1,(I MOD 2)+2
210 CLS #I+1
220 NEXT I
230 CLS
240 PRINT:PRINT

```

```
250 INPUT "QUELLE CASE CHOISISSEZ-VOUS ";N
255 N=INT(N)
260 IF N<1 OR N>6 THEN 230
270 LOCATE #N,4,4
280 PRINT #N,"Case ";N
290 PRINT "Tapez une touche pour continuer"
305 PRINT "Tapez 2 fois sur BREAK pour arreter"
320 IF INKEY#="" THEN 320 ELSE 230
330 REM
340 WINDOW 1,40,1,25
350 END
```

VARIABLES

I,J,K compteurs
N numéro de la fenêtre choisie

COMMENTAIRES

50 : cette instruction permet, par deux appuis successifs sur la touche ESC, de redéfinir en 330 la fenêtre 0, de façon à retrouver l'affichage sur tout l'écran.
150-220 : définition des six fenêtres.
270 : LOCATE #N,4,4 permet de se positionner à la ligne 4 et à la colonne 4 de la fenêtre N. On remarque que cette position est donnée en valeurs relatives à la fenêtre.

EXERCICE 1.18

PROGRAMME

```
10 REM
20 REM CLIGNOTANT
30 REM *****
40 INPUT "QUEL EST VOTRE PRENOM : ",P#
50 MODE 0
60 REM
70 REM Parametres de couleur
```

```

80 REM *****
90 INK 3,0
100 INK 2,3,26
110 SPEED INK 8,1
120 PAPER 3
130 BORDER 24
140 PEN 2
150 REM
160 REM Affichage
170 REM *****
180 CLS
190 LOCATE 10-LEN(P$)/2,10
200 PRINT P$
210 IF INKEY$="" THEN 210
220 REM
230 REM Retour au MODE 1
240 REM *****
250 PAPER 0
260 BORDER 1
270 PEN 1
280 MODE 1
290 END

```

VARIABLE

P\$ prénom à afficher

COMMENTAIRES

- 100-110 : l'instruction INK permet de définir simultanément pour l'encre 2 couleurs d'affichage de codes 3 et 26. L'instruction SPEED INK permet de faire varier la vitesse de clignotement de ces deux couleurs. Chaque paramètre indique en 50^e de seconde la durée d'affichage de chaque couleur.
- 220-290 : à la fin du programme, on retourne au mode 1 et aux couleurs initiales d'affichage.

EXERCICE 1.19

PROGRAMME

Solution 1 :

```

10 REM
20 REM Affichage avec soulignement
30 REM *****
40 CLS
50 INPUT "QUEL EST VOTRE NOM : ",N$
60 INPUT "QUEL EST VOTRE PRENOM : ",P$
70 IF P$="" OR N$="" THEN 20
80 PRINT
90 LN=LEN(N$):LP=LEN(P$)
100 TRAIT1$=STRING$(LN,"-")
110 TRAIT2$=STRING$(LP,"-")
120 PO=1+(LN+LP)/2
130 PRINT TAB(20-PO);N$;" ";P$
140 PRINT TAB(20-PO)TRAIT1$;" ";TRAIT2$
150 END

```

VARIABLES

N\$, P\$	nom et prénom
LN, LP	longueur du nom et longueur du prénom
TRAIT1\$, TRAIT2\$	chaînes de "-" servant à souligner
PO	position d'affichage

COMMENTAIRES

100-110 : la longueur du soulignement se fait en fonction des longueurs du nom et du prénom, grâce à l'instruction STRING\$.

120-140 : l'affichage se fait au centre de l'écran.

Solution 2 :

```

10 REM
20 REM Affichage avec soulignement
30 REM *****

```

```

40 CLS
50 INPUT "QUEL EST VOTRE NOM : ",N$
60 INPUT "QUEL EST VOTRE PRENOM : ",P$
70 IF P$="" OR N$="" THEN 20
80 PRINT
90 LN=LEN(N$);LP=LEN(P$)
100 TRAIT1$=STRING$(LN,"_")
110 TRAIT2$=STRING$(LP,"_")
120 PO=1+(LN+LP)/2
130 PRINT TAB(20-PO);N$;" ";P$
140 PRINT CHR$(22)+CHR$(1);
150 LOCATE 1,VPOS(#0)-1
160 PRINT TAB(20-PO)TRAIT1$;" ";TRAIT2$
170 PRINT CHR$(22)+CHR$(0)
180 END

```

VARIABLES

N\$, P\$	nom et prénom
LN, LP	longueur du nom et longueur du prénom
TRAIT1\$, TRAIT2\$	chaînes de "-" servant à souligner
PO	position d'affichage

COMMENTAIRES

140 : passage en mode transparent.
150 : remontée d'une ligne par rapport à la position verticale courante donnée par VPOS(#0).
160 : affichage des 2 traits de soulignement.

Remarque : l'instruction PRINT CHR\$(22)+CHR\$(1) fait passer en mode transparent : deux caractères différents peuvent maintenant s'afficher à la même position de l'écran. L'instruction VPOS(#0) permet ici de venir se repositionner sur le premier caractère à souligner. L'instruction PRINT CHR\$(22)+CHR\$(0) fait repasser en mode normal : à une même position de l'écran, un seul caractère s'affiche.

EXERCICE 1.20

PROGRAMME

```

10 REM
20 REM JOUONS LES NOTES
30 REM *****
40 GAM$="DO,RE,MI,FA,SO,LA,SI"
50 FOR I=1 TO 7:READ TABNOT(I):NEXT I
60 CLS
70 PRINT "Pour arreter taper ENTER"
80 PRINT
90 INPUT "Entrez une note : ",NOTE$
95 NOTE$=UPPER$(NOTE$)
100 IF NOTE$="" THEN PRINT:END
110 P=INSTR(GAM$,NOTE$)
120 IF P=0 THEN PRINT "ERREUR (!)" +CHR$(7):G
    OTO 90
130 INOT=(P+2)/3
140 PERI=TABNOT(INOT)
150 SOUND 1,PERI,100
160 GOTO 90
170 REM
180 REM Perodes de la gamme 0
190 REM *****
200 DATA 478,426,379,358,319,284,253
    
```

VARIABLES

I	compteur
TABNOT	tableau des valeurs de période des 7 notes de la gamme 0
GAM\$	notes de la gamme
NOTE\$	note à jouer
P	position de NOTE\$ dans GAM\$
INOT	indice dans le tableau TABNOT
PERI	période de la note à jouer

COMMENTAIRES

110 : l'instruction INSTR permet de déterminer la position de la note demandée dans GAM\$; supposons, par exemple,

200

que la réponse soit LA, INSTR(GAM\$,NOTE\$) prend alors la valeur 16 ; en ajoutant 2 et en divisant par 3, on obtient 6, qui est l'indice de la note LA dans le tableau TABNOT.
: les valeurs de période des notes de la gamme 0 sont données par l'intermédiaire de l'instruction DATA.

EXERCICE 1.21

PROGRAMME

```

10 REM
20 REM JOUONS UN MORCEAU
30 REM *****
40 GAM$="DO,RE,MI,FA,SO,LA,SI"
50 FOR I=1 TO 7:READ TABNOT(I):NEXT I
60 ENV 1,2,1,1,1,0,20,4,-1,3
70 ENT 1,4,1,1,8,-1,1,4,1,1
80 REM
90 REM Lecture du morceau
100 REM *****
110 READ NB
120 FOR I=1 TO NB
130 READ NOTE$,DUREE
140 P=INSTR(GAM$,NOTE$)
150 INDT=(P+2)/3
160 PERI=TABNOT(INDT)
170 SOUND 1,PERI,30*DUREE,7,1,1
180 NEXT I
190 END
200 REM
210 REM Perodes de la gamme 0
220 REM *****
230 DATA 478,426,379,358,319,284,253
240 REM
250 REM Morceau a jouer
260 REM *****
270 DATA 23
280 DATA DO,1,RE,1,MI,1,DO,1,DO,1,RE,1,MI,1,DO,1,MI,1,FA,1,SO,2,MI,1,FA,1,SO,2,SO,.5,LA,.5,SO,.5,FA,.5,MI,1,DO,1,RE,1,DO,1,DO,2

```

VARIABLES

I	compteur
TABNOT	tableau des valeurs de période des 7 notes de la gamme 0
GAM\$	notes de la gamme
NOTE\$	note à jouer
P	position de NOTE\$ dans GAM\$
INOT	indice dans le tableau TABNOT
PERI	période de la note à jouer
DUREE	durée de la note à jouer

COMMENTAIRES

- 60 : l'instruction ENV permet de définir une variation de l'amplitude du son émis. Dans notre exemple l'enveloppe numéro 1 commence par monter, puis reste constante sur un palier, et décroît enfin. Les vitesses de la montée et de la chute sont indiquées en nombre de pas et en hauteur des pas.
- 70 : l'instruction ENT a un rôle analogue à celui de ENV, mais les variations portent non plus sur l'amplitude mais sur la fréquence (aigu ou grave).
- 280 : morceau donné par une suite de notes et de durée. La valeur 2 indique une blanche, la valeur 1 une noire et la valeur .5 une croche.

EXERCICE 1.22

PROGRAMME

```
10 DIM N$(20),V$(20),C$(20)
20 REM
30 REM LECTURE DES PHRASES
40 READ N
50 FOR I=1 TO N
60 READ N$(I),V$(I),C$(I)
70 NEXT I
80 REM
90 REM AFFICHAGE DES PHRASES
100 CLS
```

```

110 FOR I=1 TO 14
120 PRINT N$(N*RND+0.5); " "; V$(N*RND+0.5); " "
;C$(N*RND+0.5)
130 NEXT I
140 PRINT:PRINT
150 REM
160 REM ATTENTE POUR LA SUITE
170 PRINT "TAPER SUR C POUR CONTINUER"
180 PRINT "TAPER SUR A POUR ARRETER"
190 A$=INKEY$
200 IF A$="C" OR A$="c" THEN 90
210 IF A$="A" OR A$="a" THEN END
220 GOTO 190
500 DATA 6
510 DATA L'ORDINATEUR,EST,UNE MACHINE
520 DATA L'HOMME,EST,UN ETRE VIVANT
530 DATA LE CHIEN,DORT SUR,LE LIT
540 DATA LE MUSEE,DUVRE,LE SOIR
550 DATA LE TRAIN,ROULE SUR,LA VOIE FERREE
560 DATA LA FUSEE,S'ENVOLE VERS,LA LUNE

```

VARIABLES

I compteur
N nombre d'éléments dans N\$(20), V\$(20) et C\$(20)
A\$ touche frappée au clavier
N\$(20) noms
V\$(20) verbes
C\$(20) compléments

COMMENTAIRES

60 et 510-560 : les différents éléments des phrases sont séparés par une virgule pour permettre la lecture de chacun d'entre eux.
120 : affichage d'une phrase. Chacun des éléments la constituant est choisi au hasard.
210-220 : le programme continuera à afficher des phrases jusqu'à ce que l'utilisateur tape "A".

Remarque : pour permettre la lecture des phrases, celles-ci sont affichées par groupe de 14, une pression de la touche C étant attendue au clavier entre chaque affichage.

EXERCICE 1.23

PROGRAMME

```
10 REM
20 REM Affichage des caracteres
30 REM de codes ASCII compris
40 REM entre 32 et 255
50 REM *****
60 CLS
70 ZONE 2
80 PRINT TAB(7)"AFFICHAGE DES CARACTERES ASCII"
90 LOCATE 5,4
100 FOR I=32 TO 255
110 PRINT CHR$(I),
120 NEXT I
130 END
```

VARIABLE

I compteur

COMMENTAIRES

70 : l'instruction ZONE 2 définit à l'écran des zones d'affichage
 de 2 caractères, soit 20 caractères par ligne.
110 : les caractères s'affichant à l'écran sont ceux dont le code
 ASCII est compris entre 32 et 255.

EXERCICE 1.24

PROGRAMME

```
10 REM
20 REM DAMIER A 2 COULEURS
30 REM *****
40 CLS
50 INK 2,8:INK 3,25
```

```

60 FOR I=0 TO 9
70 FOR J=0 TO 9
80 C=1-C:COL=2+C
90 ORIGIN 0,0,40*J,40*J+39,40*I,40*I+39
100 CLG COL
110 NEXT J
120 C=1-C
130 NEXT I
140 GOTO 140

```

VARIABLES

I,J compteurs
C variable prenant la valeur 0 ou 1
COL variable de la couleur

COMMENTAIRES

50 : définit la couleur 2 en magenta (8) et la couleur 3 en jaune (24).
80 et 120 : C prend alternativement les valeur 0 et 1 et donc COL prend alternativement les valeurs 2 et 3.
90 : chaque zone d'affichage graphique est un carré de côté égal à 40 **pixels** (point lumineux élémentaire).
100 : CLG COL efface en COL la zone graphique préalablement définie (90).

EXERCICE 1.25

PROGRAMME

```

10 REM
20 REM REBOND
30 REM *****
40 SYMBOL 250,&0,&3C,&7E,&FF,&FF,&7E,&3C,0
50 CLS
60 INK 1,26:INK 2,18:INK 3,0
70 PAPER 3

```

```
80 CLS
90 WINDOW #1,4,36,4,20
100 PAPER #1,2:PEN #1,1
110 CLS #1
120 X=1:Y=12:DX=1:DY=1
130 LOCATE #1,X,Y:PRINT #1," "
140 LOCATE #1,X+DX,Y+DY:PRINT #1,CHR$(250)
150 X=X+DX:Y=Y+DY
160 IF X>32 OR X<2 THEN DX=-DX:SOUND 1,100,2:
GOTO 130
170 IF Y>16 OR Y<2 THEN DY=-DY:SOUND 1,100,2
180 IF INKEY$="" THEN 130
190 REM
200 REM Retour
210 REM *****
220 INK 0,1:INK 1,24
230 PAPER 0:PEN 1
240 CLS
250 END
```

VARIABLES

X,Y position horizontale et verticale de la balle
DX sens de déplacement horizontal
DY sens de déplacement vertical

COMMENTAIRES

130-140 : affichage d'un espace en X,Y et de la balle en X+DX,Y+DY.
160 : changement de direction de la balle lorsqu'elle atteint l'un des bords verticaux. Ainsi si la balle se déplace sur la droite DX a la valeur +1 ; quand la balle atteint le bord droit, DX prend la valeur -1, et la balle se déplacera alors sur la gauche.
170 : test de changement de direction verticale.

EXERCICE 2.1**PROGRAMME**

```

10 DIM V(10)
20 REM
30 REM CHOIX DES VALEURS
40 CLS
50 FOR I=1 TO 10
60 PRINT "VALEUR ";I;
70 INPUT V(I)
80 NEXT I
90 REM
100 REM POSITIFS OU NULS
110 P=0
120 FOR I=1 TO 10
130 IF V(I)>=0 THEN P=P+1
140 NEXT I
150 REM
160 REM AFFICHAGE
170 FOR I=1 TO 10
180 PRINT V(I)
190 NEXT I
200 PRINT
210 PRINT "NOMBRE DE VALEURS"
220 PRINT "POSITIVES OU NULLES : ";P
230 END

```

VARIABLES

I compteur de boucle
P nombre de valeurs positives
V(10) tableau des 10 valeurs

COMMENTAIRES

10 : dimensionnement du tableau.
50-80 : boucle d'entrée des valeurs : le numéro de chaque valeur est affiché par l'instruction PRINT de la ligne 60 car il est impossible de faire afficher par l'instruction INPUT un message comprenant la valeur de I.

- 110 : initialisation du compteur de valeurs positives.
- 130 : si la lème valeur est positive ou nulle, P est incrémenté.
- 170-190 : les dix valeurs sont affichées.
- 210-220 : affichage du résultat P.

EXERCICE 2.2

PROGRAMME

```
10 CLS
20 REM
30 INPUT "PRIX H.T. ";P
40 INPUT "CODE T.V.A. (A:5.5% B:18.6% C:33%)
";CT$
50 REM
60 REM RECHERCHE DU TAUX DE T.V.A.
70 IF CT$="A" THEN T=5.5 ELSE IF CT$="B" THEN
T=18.6 ELSE IF CT$="C" THEN T=33 ELSE 40
80 REM
90 REM CALCUL ET AFFICHAGE
100 PRINT
110 PT=P+P*T/100
120 PT=ROUND(P,2)
130 PRINT "PRIX T.T.C. : ";PT
140 PRINT:PRINT
150 END
```

VARIABLES

P prix hors taxe
PT prix TTC
T taux de TVA
CT\$ code du taux de TVA

COMMENTAIRES

- 40 : les codes à utiliser sont rappelés dans le texte de l'instruction INPUT.
- 70 : le taux de TVA est affecté à la variable T. Sa valeur est obtenue par test sur la variable d'entrée CT\$.

110 : calcul du prix TTC.
 120 : l'arrondi à deux décimales est obtenu par l'instruction ROUND.

Remarque : lorsque l'arrondi n'est nécessaire qu'à l'affichage, il faut utiliser l'instruction PRINT USING, qui n'affecte pas la valeur affichée. Dans la solution proposée, effacer la ligne 120 et remplacer la ligne 130 par :

```
130 PRINT "PRIX T.T.C.:"; USING "###.##";PT
```

EXERCICE 2.3

PROGRAMME

```
10 CLS
20 DIM P(10)
30 REM
40 REM RENTREE DES DONNEES
50 FOR I=1 TO 10
60 PRINT "PRIX H.T. ";I;
70 INPUT P(I)
80 NEXT I
90 PRINT
100 INPUT "CODE T.V.A. (A:5.5% B:18.6% C:33%)
";CT$
110 REM
120 REM RECHERCHE DU TAUX DE T.V.A.
130 IF CT$="A" THEN T=5.5 ELSE IF CT$="B" THE
N T=18.6 ELSE IF CT$="C" THEN T=33 ELSE 100
140 REM
150 REM CALCUL
160 PRINT
170 FOR I=1 TO 10
180 P(I)=P(I)+P(I)*T/100
190 NEXT I
200 REM
210 REM AFFICHAGE AVEC ALIGNEMENT
```

```
220 PRINT
230 FOR I=1 TO 10
240 PRINT TAB(10)"PRIX "; USING "## #####.##"
;I,P(I)
250 NEXT I
260 END
```

VARIABLES

I compteur de boucle
T taux TVA
CT\$ code du taux de TVA
P(10) tableau des prix

COMMENTAIRES

20 : dimensionnement du tableau.
50-80 : boucle d'entrée des valeurs du tableau P. Avant chaque demande, le rang de la valeur à entrer est affiché.
150-190 : boucle de calcul des prix TTC. Les prix TTC remplacent les prix hors taxe dans le tableau P.
210-250 : boucle d'affichage. L'instruction PRINT USING affiche les 2 nombres séparés par un espace ; le premier est cadré sur 2 positions (entier compris entre 1 et 10) ; le second est cadré sur 5 positions avant la virgule et 2 positions après la virgule (il doit être inférieur à 100.000). Il y a alignement du point décimal pour les différentes lignes successives.

EXERCICE 2.4

PROGRAMME

1^{re} méthode :

```
10 CLS
20 DIM L$(20)
30 READ N
40 FOR I=1 TO N
50 READ L$(I)
60 NEXT I
```

```

70 FOR I=1 TO N
80 PRINT L$(I)
90 NEXT I
100 PRINT:PRINT
110 END
500 DATA 7
510 DATA AMSTRAD, TABLE, VOITURE, ARBRE, MAISON, C
    HAISE, OISEAU

```

2^e méthode :

```

10 CLS
20 DIM L$(20)
30 FOR I=1 TO 20
40 READ L$(I)
50 IF L$(I)="FIN" THEN 70
60 NEXT I
70 N=I-1
80 FOR I=1 TO N
90 PRINT L$(I)
100 NEXT I
110 PRINT:PRINT
120 END
500 DATA AMSTRAD, TABLE, VOITURE, ARBRE, MAISON, C
    HAISE, OISEAU
510 DATA FIN

```

3^e méthode :

```

10 CLS
20 DIM L$(20)
30 ON ERROR GOTO 200
40 FOR I=1 TO 100
50 READ L$(I)
60 NEXT I
70 REM
80 FOR I=1 TO N
90 PRINT L$(I)
100 NEXT I
110 PRINT:PRINT
120 END

```

EXERCICES EN BASIC POUR AMSTRAD

```
200 REM TRAITEMENT EN CAS D'ERREUR
210 N=I-1
220 RESUME 80
500 DATA AMSTRAD, TABLE, VOITURE, ARBRE, MAISON, C
HAISE, OISEAU
```

VARIABLES

I compteur
N nombre de mots enregistrés sous forme de DATA
L\$(20) tableau des mots

COMMENTAIRES

Les trois méthodes donnent des résultats similaires.

Dans la première méthode le nombre d'éléments à lire est passé par l'intermédiaire de l'instruction DATA en ligne 500.

Dans la deuxième méthode, la fin des données à lire est indiquée par le mot réservé FIN. La ligne 70 calcule le nombre de mots de la liste enregistrée sous forme de DATA et l'affecte à la variable N.

La troisième méthode ne nécessite ni de connaître le nombre d'éléments de la liste des données, ni de réserver un mot pour en indiquer la fin. C'est l'erreur provoquée par le manque de données à lire, lors de l'exécution de l'instruction READ, qui est utilisée pour sortir de la boucle de lecture. Cette méthode est à utiliser avec précaution car elle nécessite en principe un test sur la cause de l'erreur.

EXERCICE 2.5

PROGRAMME

```
10 CLS
20 DIM N$(20), P$(20)
30 REM
40 REM LECTURE DES DONNEES
50 READ N
60 FOR I=1 TO N
```

```

70 READ N$(I),P$(I)
80 NEXT I
90 REM
100 INPUT "NOM a chercher ";N$
110 INPUT "PRENOM ";P$
120 PRINT
130 REM
140 REM RECHERCHE
150 FOR I=1 TO N
160 IF N$=N$(I) AND P$=P$(I) THEN PRINT P$;"
";N$;" est dans le fichier":GOTO 190
170 NEXT I
180 PRINT P$;" ";N$;" n'est pas dans le fichi
er "
190 PRINT
200 END
500 DATA 5
510 DATA MARTIN,ALBERT,PERRIN,ADELAIDE
520 DATA BARON,JEAN,DURAND,GEORGES
530 DATA MARTEAU,JACQUES

```

VARIABLES

I	compteur
N	nombre de personnes
N\$	nom de la personne à chercher
P\$	prénom de la personne à chercher
N\$(20)	tableau des noms
P\$(20)	tableau des prénoms

COMMENTAIRES

60-80 : lecture des noms et prénoms enregistrés sous forme d'une liste DATA.

160 : si les nom et prénom sont ceux de la personne cherchée, le programme affiche un message, puis s'arrête. Sinon, la personne suivante dans la liste est testée.

180 : si toutes les personnes ont été testées et qu'aucune n'est celle demandée, le programme l'affiche.

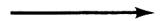
Remarque : N\$ et N\$(I) sont deux variables distinctes, la première est la variable alphanumérique N\$, la seconde est le lième élément du tableau alphanumérique N\$.

EXERCICE 2.6

PROGRAMME

Solution 1 :

```
10 CLS
20 REM
30 REM lecture du nombre de personnes
40 READ N
50 REM
60 REM entree du nom et du numero
70 INPUT "NOM OU NUMERO A RECHERCHER (TAPEZ
FIN POUR ARRETER) ";R$
80 IF R$="FIN" THEN END
90 R=LEN(R$)
100 REM
110 REM recherche
120 F=0
130 FOR I=1 TO N
140 READ NOM$,PRENOM$,NUMERO$
150 IF R$=MID$(NOM$,1,R) OR R$=MID$(NUMERO$,1
,R) THEN F=1:PRINT NOM$;" ";PRENOM$;"....";N
UMERO$
160 NEXT I
170 IF F=0 THEN PRINT R$;" n'est pas dans la
liste"
180 REM
190 RESTORE:PRINT
200 GOTO 30
500 DATA 6
510 DATA MARTIN,ALBERT,275 98 98
520 DATA PERRIN,ADELAIDE,67 76 01
530 DATA DURAND,GEORGES,23 65 72
535 DATA MARTIN,JEAN,575 76 27
540 DATA MARTEAU,JEAN,345 83 54
550 DATA MOORE,RICK,19 1 714 755 50 04
```



Solution 2 :

```

10 CLS
20 REM
30 REM lecture du nombre de personnes
40 READ N
50 REM
60 REM entree du nom et du numero
70 INPUT "NOM OU NUMERO A RECHERCHER (TAPEZ
FIN POUR ARRETER) ";R$
80 IF R$="FIN" THEN END
90 R=LEN(R$)
100 REM
110 REM recherche
120 F=0
130 FOR I=1 TO N
140 READ NOM$,PRENOM$,NUMERO$
150 ZONE 10
160 IF R$<>MID$(NOM$,1,R) AND R$<>MID$(NUMERO
$,1,R) THEN 190
170 F=1
180 PRINT NOM$,PRENOM$,NUMERO$
190 NEXT I
200 IF F=0 THEN PRINT R$;" n'est pas dans la
liste"
210 REM
220 RESTORE:PRINT
230 GOTO 30
500 DATA 6
510 DATA MARTIN,ALBERT,275 98 98
520 DATA FERRIN,ADELAIDE,67 76 01
530 DATA DURAND,GEORGES,23 65 72
540 DATA MARTIN,JEAN,575 76 27
550 DATA MARTEAU,JEAN,345 83 54
560 DATA MOORE,RICK,19 1 714 755 50 04

```

VARIABLES

I	compteur
N	nombre de personnes dans le fichier
R	nombre de caractères dans R\$
F	drapeau (flag) : égal à 0 si la personne cherchée n'est pas dans le fichier et à 1 dans le cas contraire

R\$ nom ou numéro à rechercher
NOM\$ nom de la lème personne de la liste
PRENOM\$ prénom de la lème personne de la liste
NUMERO\$ numéro de téléphone de la lème personne

COMMENTAIRES

Solution 1 :

80 : le programme bouclant en 200 (GOTO 30) est interrompu lorsque la réponse FIN est donnée à la ligne 80.
120 : mise à zéro de F en début de programme.
150 : si le nom ou le numéro de téléphone correspondent, la personne est affichée et le drapeau F est mis à 1. La recherche continue néanmoins car plusieurs personnes peuvent avoir le même nom ou le même numéro de téléphone.
170 : si F contient 0, cela indique qu'aucune personne du fichier n'est celle demandée.
190 : l'instruction RESTORE permet de relire les données en DATA à partir du début (y compris le nombre de personnes).

Remarque : la comparaison s'effectue entre la chaîne entrée (R\$), de longueur R et les R premiers caractères du nom ou du numéro. Ainsi, il est possible de faire afficher la liste des personnes dont le nom commence par "M" en tapant uniquement cette lettre lors de la demande du nom ou du numéro (ligne 70).

Solution 2 :

150 : dans cette solution, une facilité d'affichage est obtenue par l'instruction ZONE qui définit à l'écran des zones d'affichage de 10 caractères.
180 : l'affichage sur une même ligne d'écran des 3 variables NOM\$, PRENOM\$, NUMERO\$ se fait dans chacune des zones. Si le nombre de caractères excède la longueur de la zone, son affichage s'étend à la zone suivante.

EXERCICE 2.7**PROGRAMME**

```

10 DIM N$(1,20)
20 CLS
30 INPUT "COMBIEN DE PERSONNES (DE 1 A 20) ";
N
40 IF N<1 OR N>20 THEN 30
50 PRINT
60 REM
70 REM ENTREE DES DONNEES
80 FOR I=1 TO N
90 PRINT "PERSONNE "I;": "
100 INPUT "NOM ",N$(0,I)
110 INPUT "PRENOM ",N$(1,I)
120 PRINT
130 NEXT I
140 REM
150 REM RECHERCHE
160 M=1
170 FOR I=1 TO N
180 IF N$(0,I)=N$(0,M) AND N$(1,I)<N$(1,M) TH
EN M=I
190 IF N$(0,I)<N$(0,M) THEN M=I
200 NEXT I
210 REM
220 REM AFFICHAGE
230 PRINT
240 PRINT "LA PREMIERE PERSONNE PAR ORDRE
      ALPHABETIQUE EST CELLE D'INDICE";M;": "
245 PRINT:PRINT TAB(15)N$(1,M);" ";N$(0,M)
250 PRINT
260 END

```

VARIABLES

I	compteur
N	nombre de personnes
M	indice de la première personne dans l'ordre alphabétique
N\$(1,20)	tableau à deux dimensions des noms et prénoms
	N\$(0,1),...,N\$(0,20) : noms
	N\$(1,1),...,N\$(1,20) : prénoms

COMMENTAIRES

- 90 : le numéro de la personne à entrer est affiché.
 180 : si les noms sont identiques, l'ordre des prénoms est alors pris en compte.

Remarque : un tableau à deux indices est utilisé. Le premier indice indique s'il s'agit d'un nom (0) ou d'un prénom (1). Le deuxième indice indique le rang dans la liste (1 à 20).

	0	1
1	N\$(0,1) nom 1	N\$(1,1) prénom 1
2	N\$(0,2) nom 2	N\$(1,2) prénom 2
3	•	•
4	•	•
•	•	•
20	•	•

Il est possible de regrouper les tests des lignes 180 et 190 en un seul :
 180 IF N\$(0,I)<N\$(0,M) OR (N\$(0,I) = N\$(0,M) AND N\$(1,I)<N\$(1,M))
 THEN M = I.

EXERCICE 2.8

PROGRAMME

1^{re} méthode :

```

10 DIM TA(50),TB(50)
20 REM ENTREE DES DONNEES
30 READ N1
40 FOR I=1 TO N1
50 READ TA(I)
60 NEXT I
70 REM
80 REM CLASSEMENT
90 N2=0
100 REM RECHERCHE DE L'ELEMENT LE
110 REM PLUS GRAND DU TABLEAU
120 N2=N2+1
    
```

```

130 M=1
140 FOR I=1 TO N1
150 IF TA(I)>TA(M) THEN M=I
160 NEXT I
170 REM
180 REM AJOUT DANS LE SECOND TABLEAU
190 TB(N2)=TA(M)
200 TA(M)=-1
210 IF N2<N1 THEN 120
220 REM
230 REM AFFICHAGE
240 FOR I=1 TO N2:PRINT TB(I),:NEXT I
250 PRINT
260 END
270 DATA 12
280 DATA 3, 10, 15, 1000, 1, 5, 32, 0, 15, 12, 26, 38

```

2^e méthode :

```

10 DIM TA(50),TB(50)
20 REM ENTREE DES DONNEES
30 READ N1
40 FOR I=1 TO N1
50 READ TA(I)
60 NEXT I
70 REM
80 REM CLASSEMENT
90 N2=0
100 REM RECHERCHE DE L'ELEMENT LE
110 REM PLUS GRAND DU TABLEAU
120 N2=N2+1
130 M=1
140 FOR I=1 TO N1
150 IF TA(I)>TA(M) THEN M=I
160 NEXT I
170 REM
180 REM ECHANGE DES ELEMENTS
190 TB(N2)=TA(M)
200 IF M<N1 THEN FOR I=M+1 TO N1:TA(I-1)=TA(I)
):NEXT I
210 N1=N1-1
220 IF N1>0 THEN 120

```

```
230 REM
240 REM AFFICHAGE
250 FOR I=1 TO N2:PRINT TB(I),:NEXT I
260 PRINT
270 END
280 DATA 12
290 DATA 3,-10,15,1000,-1,5,-32,0,15,12,26,38
```

VARIABLES

I compteur
N1 nombre d'éléments du premier tableau
N2 nombre d'éléments du second tableau
M indice de l'élément le plus grand du premier tableau
TA(50) premier tableau : nombres à classer
TB(50) second tableau : nombres classés

COMMENTAIRES

Solution 1 :

30-60 : lecture des données.
120 : incrémentation de l'indice du second tableau.
130-160 : recherche de l'élément le plus grand du premier tableau.
190 : insertion de cet élément dans le second tableau.
200 : mise à (-1) de l'emplacement occupé dans le tableau 1,
 par l'élément rangé dans le tableau 2.
210 : test de fin.

Solution 2 :

140-160 : recherche du plus grand élément du tableau TA.
190 : rangement dans le tableau TB.
200 : remontée d'une position des éléments de TA d'indice supérieur à M+1 sauf si le plus grand élément se trouvait en queue de TA auquel cas le décalage est inutile.

EXERCICE 2.9

PROGRAMME

```

10 REM
20 DIM N$(15),H(15)
30 REM
40 REM LECTURE DES DONNEES
50 FOR I=1 TO 9
60 READ N$(I)
70 NEXT I
80 FOR I=1 TO 9
90 READ H(I)
100 NEXT I
110 REM CLASSEMENT PAR ORDRE CROISSANT
120 FOR I=1 TO 9
130 P=1
140 FOR J=1 TO 9
150 IF H(I)>H(J) THEN P=P+1
160 NEXT J
170 Z(P)=I
180 NEXT I
190 REM AFFICHAGE
200 CLS
210 PRINT STRING$(40,"_")
220 PRINT " PAYS"SPC(14)"POPULATION(millions)
"
230 PRINT STRING$(40,"_")
240 FOR I=1 TO 9
250 PRINT " N$(Z(I))TAB(19),USING "##.###";H
(Z(I))
260 NEXT I
270 PRINT STRING$(40,"_")
280 END
500 DATA ALLEMAGNE,BELGIQUE,DANEMARK,FRANCE
510 DATA IRLANDE,ITALIE,LUXEMBOURG
520 DATA PAYS-BAS,ROYAUME-UNI
530 DATA 61.498,9.889,5.073,52.915
540 DATA 3.162,56.323,.358,13.825,55.928

```



VARIABLES

I,J compteurs
P indice
N\$ tableau des pays
H tableau des populations
Z tableau des indices de N\$, donnant les pays par population croissante

COMMENTAIRES

40-100 : lecture des éléments de N\$ et de H par l'instruction READ d'une liste donnée en DATA.
120-180 : mise en ordre croissant des indices de N\$(I). Par exemple, puisque N\$(6) = "ITALIE" et que l'Italie est le 8ème dans l'ordre croissant des populations de la liste proposée, Z(8) prend la valeur 6. Le lème élément de Z est égal au numéro du pays qui arrive en lème position par population croissante.
210, 230 : STRING\$(40,"-") fabrique une chaîne de tirets pour le soulignement.

EXERCICE 2.10

PROGRAMME

```
10 DIM N$(11),C$(2),D(11,2),Z(11)
20 REM LECTURE DES DONNEES
30 GOSUB 140
40 REM AFFICHAGE DU MENU ET CHOIX
50 GOSUB 280
60 REM CLASSEMENT
70 GOSUB 360
80 REM AFFICHAGE DU TABLEAU
90 GOSUB 460
100 REM CHOIX : ARRETER OU CONTINUER
110 INPUT "Voulez-vous continuer (OUI/NON) ";A$
120 IF A$<>"NON" THEN 40
130 END
140 REM SOUS-PROGRAMME DE LECTURE
```

```

150 READ C$(0),C$(1),C$(2)
160 FOR I=0 TO 11
170 READ N$(I)
180 NEXT I
190 FOR J=0 TO 1
200 FOR I=0 TO 11
210 READ D(I,J)
220 NEXT I
230 NEXT J
240 FOR I=0 TO 11
250 D(I,2)=D(I,0)*1000/D(I,1)
260 NEXT I
270 RETURN
280 REM SOUS-PROGRAMME D'AFFICHAGE DU MENU
290 CLS
300 PRINT "0--POPULATIONS CROISSANTES"
310 PRINT "1--SUPERFICIES CROISSANTES"
320 PRINT "2--DENSITES CROISSANTES"
330 INPUT "QUEL EST VOTRE CHOIX : ",J
340 IF J<0 OR J>2 THEN 280
350 RETURN
360 REM SOUS-PROGRAMME DE CLASSEMENT
370 FOR I=0 TO 11
380 P=0
390 FOR K=0 TO 11
400 IF D(I,J)<=D(K,J) THEN 420
410 P=P+1
420 NEXT K
430 Z(P)=I
440 NEXT I
450 RETURN
460 REM SOUS-PROGRAMME D'AFFICHAGE
470 CLS
480 PRINT STRING$(40,"_")
490 PRINT " PAYS"TAB(20)C$(J)
500 PRINT STRING$(40,"_")
510 FOR I=0 TO 11
520 IF J=1 THEN FORM$="###" ELSE FORM$="###.###"
530 PRINT " N$(Z(I))TAB(19),USING FORM$;D(Z(I),
J)
540 NEXT I
550 PRINT STRING$(40,"_")

```

```
560 RETURN
570 DATA POPULATION(millions), SUPERFICIE(1000 km
2), DENSITE(hab./km2)
580 DATA ALLEMAGNE, BELGIQUE, DANEMARK, ESPAGNE, FRA
NCE
590 DATA GRECE, IRLANDE, ITALIE, LUXEMBOURG
600 DATA PAYS-BAS, PORTUGAL, ROYAUME-UNI
610 DATA 61.498, 9.889, 5.073, 35.971, 52.915, 9.165
620 DATA 3.162, 56.323, .358, 13.825, 9.449, 55.928
630 DATA 249, 31, 43, 505, 547, 132, 70, 301, 3, 41, 92, 244
```

VARIABLES

N\$ tableau des noms de pays
C\$ tableau des critères
D tableau de valeurs numériques à deux dimensions
Z tableau d'indices
A\$ réponse (OUI/NON)
I,J,K compteurs
FORM\$ format de l'instruction USING
P indice de l'élément classé

COMMENTAIRES

150 : lecture du tableau C\$: POPULATION, SUPERFICIE, DEN-
SITE.
160-180 : lecture du tableau N\$ (Pays).
190-230 : lecture du tableau D.
 1) D(I,0) est la valeur de la population du pays de rang I.
 2) D(I,1) est la valeur de la superficie du pays de rang I.
240-260 : calcul de D(I,2), qui est la valeur de la densité du pays de
rang I.
360-440 : sous-programme de classement par ordre croissant pour
le critère I.
460-560 : sous-programme d'affichage.
520 : on note que si J=1 (choix du critère de superficie), l'affi-
chage se fait sur 3 chiffres (FORM\$="###"), sinon l'affi-
chage se fait sur 3 chiffres avant et 3 chiffres après la virgule
(FORM\$="###.###").

EXERCICE 2.11

PROGRAMME

```

10 DIM N$(1,20)
20 REM
30 REM ENTREE DES DONNEES
40 CLS
50 INPUT "NOMBRE DE PERSONNES (DE 1 A 20) : ",N
60 IF N<1 OR N>20 THEN 30
70 FOR I=1 TO N
80 PRINT
90 PRINT "PERSONNE ";I;" : ";
100 PRINT TAB(16);
110 INPUT "NOM.....",N$(0,I)
120 PRINT TAB(16);
130 INPUT "PRENOM..",N$(1,I)
140 PRINT
150 NEXT I
160 REM
170 REM CLASSEMENT
180 P=1
190 M=P
200 FOR I=P TO N
210 IF N$(0,I)<N$(0,M) OR (N$(0,I)=N$(0,M) AND N
$(1,I)<N$(1,M)) THEN M=I
220 NEXT I
230 REM
240 N#=N$(0,P):P#=N$(1,P)
250 N$(0,P)=N$(0,M):N$(1,P)=N$(1,M)
260 N$(0,M)=N#:N$(1,M)=P#
270 P=P+1
280 IF P<N THEN 190
290 REM
300 REM AFFICHAGE
310 FOR I=1 TO N
320 PRINT N$(0,I),N$(1,I)
330 NEXT I
340 END

```



VARIABLES

I	compteur
N	nombre de personnes
P	indice à partir duquel les personnes doivent être classées
M	indice de la première personne par ordre alphabétique
N\$,P\$	variables
N\$(1,20)	tableau des noms et prénoms
	N\$(0,1),...,N\$(0,20) - noms
	N\$(1,1),...,N\$(1,20) - prénoms

COMMENTAIRES

- 180 : initialisation de P à 1.
190-220 : recherche du rang M de la personne (nom et prénom) la première par ordre alphabétique, parmi celle d'indice supérieur ou égal à P.
270 : P est incrémenté.
240-260 : permutation des éléments (nom et prénom) de la personne de rang P avec ceux de celle de rang M.

Remarque : pour permuter deux variables, par exemple N1 et N2, il faut utiliser une troisième variable, par exemple A, et exécuter les opérations suivantes :

- 1) A = N1 ; recopier N1 dans A
- 2) N1 = N2 ; recopier N2 dans N1
- 3) N2 = A ; recopier A dans N2

EXERCICE 2.12

PROGRAMME

```
10 DIM N$(1,20)
20 REM
30 REM ENTREE DE LA LISTE DES PERSONNES
40 READ N
50 FOR I=1 TO N:READ N$(0,I),N$(1,I):NEXT I
60 REM
70 REM ENTREE DE LA PERSONNE A AJOUTER
```

```

80 CLS
90 PRINT "(TAPEZ FIN POUR ARRETER)"
100 INPUT "NOM : ",N$
110 N$=UPPER$(N$)
120 IF N$="FIN" OR N>20 THEN END
130 INPUT "PRENOM : ",P$
140 P$=UPPER$(P$)
150 REM
160 REM RECHERCHE
170 FOR I=1 TO N
180 IF N$<N$(0,I) OR (N$=N$(0,I) AND P$<N$(1,I))
    THEN 250
190 NEXT I
200 REM
210 REM AJOUT A LA FIN DE LA LISTE
220 N$(0,N+1)=N$:N$(1,N+1)=P$
230 GOTO 300
240 REM
250 REM DECALAGE ET AJOUT
260 FOR J=N TO I STEP -1
270 N$(0,J+1)=N$(0,J):N$(1,J+1)=N$(1,J)
280 NEXT J
290 N$(0,I)=N$:N$(1,I)=P$
300 N=N+1
310 REM
320 REM AFFICHAGE
330 CLS
340 FOR I=1 TO N
350 PRINT N$(0,I),N$(1,I)
360 NEXT I
370 PRINT:PRINT
380 GOTO 90
390 DATA 5
400 DATA ALRIC,JEAN,DURAND,GEORGES,MARTEAU
410 DATA JACQUES,PERRIN,ADELAIDE,PERRIN,PAUL

```

VARIABLES

I,J	compteurs
N	nombre de personnes
N\$,P\$	nom et prénom de la personne à insérer

N\$(1,20) tableau des noms et prénoms
N\$(0,1),...,N\$(0,20) - noms
N\$(1,1),...,N\$(1,20) - prénoms

COMMENTAIRES

- 120 : l'utilisateur peut insérer plusieurs personnes (le programme boucle en 380) et doit taper FIN pour arrêter. Si le tableau contient 20 personnes, il est impossible d'en insérer une autre.
- 110, 140 : UPPER\$ transforme une chaîne de caractères en lettres majuscules ; les lettres déjà en majuscules sont conservées.
- 170-190 : si la position d'insertion dans le tableau n'est pas la dernière, alors le programme continue en 250 et I contient cette position. Si c'est la dernière, alors la boucle se termine et le programme continue en 210.
- 300 : ne pas oublier d'incrémenter N puisque le tableau contient une personne de plus.

EXERCICE 2.13

PROGRAMME

```
10 CLS
20 PRINT "RESOLUTION DE L'EQUATION DU SECOND DEG
RE"
30 REM
40 PRINT
50 PRINT TAB(10)"A X2 + B X + C = 0"
60 PRINT
70 INPUT "A = ",A
80 INPUT "B = ",B
90 INPUT "C = ",C
100 DELTA=B^2-4*A*C
110 IF DELTA=0 THEN 250
120 IF DELTA<0 THEN 340
130 REM
140 REM DEUX RACINES REELLES DISTINCTES
150 REM
```

```

160 PRINT
170 X1=(-B-SQR(DELTA))/2/A
180 X2=(-B+SQR(DELTA))/2/A
190 PRINT "Cette equation a deux racines reelles
et distinctes"
200 PRINT
210 PRINT TAB(15)"X1 = ";X1
220 PRINT
230 PRINT TAB(15)"X2 = ";X2
240 END
250 REM
260 REM UNE RACINE REELLE DOUBLE
270 REM
280 PRINT
290 X=-B/2/A
300 PRINT "Cette equation a une racine double"
310 PRINT
320 PRINT TAB(15)"X = ";X
330 END
340 REM
350 REM PAS DE RACINES REELLES
360 REM
370 PRINT
380 PRINT "Cette equation a deux racines complex
es"
390 PRINT
400 END

```

VARIABLES

X racine double
X1,X2 racines distinctes
A,B,C coefficients
DELTA discriminant

COMMENTAIRES

100 : calcul du discriminant.
170-180 : SQR(DELTA) donne la racine carrée de DELTA.
240 : si le discriminant est nul, on poursuit à la ligne 250.
 Si le discriminant est négatif, on poursuit à la ligne 340.
 Si le discriminant est positif, on continue.

EXERCICE 2.14

PROGRAMME

```

10 REM
20 REM CALCUL DE POURCENTAGE
30 REM *****
40 DIM T(50),P(50)
50 ETOILE#=SPACE$(5)+STRING$(19,"*")
60 REM
70 REM Entree des valeurs
80 REM *****
90 CLS
100 PRINT SPC(6)"CALCUL DE POURCENTAGE"
110 PRINT:PRINT
120 INPUT "Nombre de valeurs (1 a 50) : ",N
130 N=INT(N)
140 IF N<1 OR N>50 THEN 90
150 FOR I=1 TO N
160 PRINT "Valeur (entre 0 et 1000) ";I;
170 INPUT " : ",T(I)
180 IF T(I)<0 OR T(I)>=1000 THEN 170
190 NEXT I
200 REM
210 REM Calcul
220 REM *****
230 S=0
240 FOR I=1 TO N
250 S=S+T(I)
260 NEXT I
270 FOR I=1 TO N
280 P(I)=T(I)*100/S
290 NEXT I
300 REM
310 REM Affichage
320 REM *****
330 CLS
340 PRINT SPC(4)"CALCUL DE POURCENTAGE"
350 PRINT:PRINT
360 PRINT ETOILE#
370 PRINT TAB(6)"Valeurs"TAB(16)"Pourcent. "

```

```

380 PRINT ETOILE$
390 FOR I=1 TO N
400 FORM$="###.##"+SPACE$(4)+"###.## %"
410 PRINT TAB(6) USING FORM$;T(I),P(I)
420 NEXT I
430 PRINT ETOILE$
440 PRINT:PRINT
450 END

```

VARIABLES

ETOILE\$ chaîne de 5 espaces et de 19 étoiles servant à souligner
 I compteur
 N nombre de valeurs
 T tableau des valeurs
 P tableau des pourcentages
 S somme
 FORM\$ format d'affichage de l'instruction PRINT USING

COMMENTAIRES

140-190 : entrée des N valeurs du tableau T. Ces valeurs doivent être comprises entre 0 (inclus) et 1000 (exclu).
 230 : S est tout d'abord mis à zéro.
 240-260 : pour calculer la somme des N valeurs du tableau T, il suffit d'incrémenter le compteur I depuis 1 jusqu'à N et à chaque passage d'augmenter S de la valeur T(I).
 280 : le pourcentage P(I) est donné par : $100 \times T(I)/S$.
 410 : les valeurs et leurs pourcentages sont affichés en utilisant le format d'affichage, qui est décrit à la ligne 400.

EXERCICE 2.15

PROGRAMME

```

10 REM
20 REM Troncature d'un nombre :
30 REM conservation d'un chiffre
40 REM significatif par excès

```

EXERCICES EN BASIC POUR AMSTRAD

```
50 REM *****
60 CLS
70 INPUT "Entrez un nombre positif : ",NB
80 IF NB<=0 THEN 70
90 LNB=INT(LOG10(NB))
100 MNB=NB*(10^-LNB)
110 TRO=(INT(MNB)+1)*(10^LNB)
120 PRINT
130 PRINT "Nombre : ",NB
140 PRINT "V.tronq. : ",TRO
150 PRINT
160 END
```

VARIABLES

NB nombre entré
LNB partie entière du logarithme de NB
MNB recadrage de NB entre 1 (inclus) et 10 (exclu)
TRO toncature de NB par excès et un chiffre significatif

COMMENTAIRES

90 : (LNB+1) est égal au nombre de chiffres avant la virgule si NB est supérieur à 1 et au nombre de zéros (précédé du signe -) après la virgule si NB est inférieur à 1.
100 : MNB contient les mêmes chiffres que NB mais cadré entre 1 et 10.
110 : détermination du nombre le plus proche par excès de N avec un seul chiffre significatif.

EXERCICE 2.16

PROGRAMME

```
10 REM
20 REM CALCUL DE VOLUME ET D'AIRES
30 REM *****
40 DEF FN SCU(X)=ROUND(6*X*X,3)
50 DEF FN VCU(X)=ROUND(X*X*X,3)
60 DEF FN SSP(R)=ROUND(4*PI*R*R,3)
```

```

70 DEF FN VSP(X)=ROUND(4*PI*X*X*X/3,3)
80 DEF FN SCY(H,R)=ROUND(2*PI*R*(R+H),3)
90 DEF FN VCY(H,R)=ROUND(PI*R*R*H,3)
100 FORM$="CUSPCY"
110 REM
120 REM Choisissez une forme
130 REM *****
140 CLS
150 PRINT "Choisissez une forme : "
160 PRINT
170 PRINT TAB(10)"CU pour CUBE"
180 PRINT TAB(10)"SP pour SPHERE"
190 PRINT TAB(10)"CY pour CYLINDRE"
200 PRINT
210 INPUT "Quel est votre choix :",REP$
220 IF LEN(REP$)>2 OR INSTR(FORM$,REP$)=0 THEN P
RINT SPC(10)"ERREUR (!!)" +CHR$(7):GOTO 210
230 ON INSTR(FORM$,REP$)/2 GOSUB 250,320,390
240 END
250 REM
260 REM Sous-programme du cube
270 REM *****
280 INPUT "Quel est le cote du cube : ",COT
290 PRINT "Sa surface est : ";FN SCU(COT)
300 PRINT "Son volume est : ";FN VCU(COT)
310 RETURN
320 REM
330 REM Sous-programme de la sphere
340 REM *****
350 INPUT "Quel est le rayon de la sphere : ",RA
Y
360 PRINT "Sa surface est : ";FN SSP(RAY)
370 PRINT "Son volume est : ";FN VSP(RAY)
380 RETURN
390 REM
400 REM Sous-programme du cylindre
410 REM *****
420 INPUT "Quelle est sa hauteur : ",HT
430 INPUT "Quel est le rayon de sa base : ",RAY
440 PRINT "Sa surface est : ";FN SCY(HT,RAY)
450 PRINT "Son volume est : ";FN VCY(HT,RAY)
460 RETURN

```

VARIABLES

FORM\$ variable alphanumérique égale à "CUSPCY"
REP\$ réponse
RAY rayon de la sphère ou du cylindre
HT hauteur du cylindre
COT côté du cube

COMMENTAIRES

40-90 : les 6 fonctions correspondant au volume et à la surface de la sphère, du cylindre et du cube sont définies. Pour obtenir par exemple, le volume d'une sphère de rayon 3,4 il faut exécuter l'instruction FN VSP(3.4) ; pour obtenir la surface d'un cylindre de rayon 3 et de hauteur 2,3, il faut exécuter FN SCY(3,2.3). Les résultats sont arrondis à 3 chiffres après la virgule.

230 : INSTR(FORM\$,REP\$)/2 prend la valeur :
1, si REP\$="CU" et le programme reprend à la ligne 250 ;
2, si REP\$="SP" et le programme reprend à la ligne 320 ;
3, si REP\$="CY" et le programme reprend à la ligne 390 ;
sinon FIN.

EXERCICE 3.1

PROGRAMME

```
10 CLS
20 LINE INPUT "TAPEZ UNE LIGNE " ; A$
30 I=1
40 IF MID$(A$, I, 1) = " " THEN A$ = LEFT$(A$, I-1) +
MID$(A$, I+1) : GOTO 40
50 I=I+1 : IF I <= LEN(A$) THEN 40
60 PRINT "(" ; A$ ; ")"
70 END
```



VARIABLES

I compteur
 A\$ ligne de texte à traiter

COMMENTAIRES

40 : tous les caractères de la chaîne sont comparés un par un à l'espace ; si le lème caractère est un espace, il est supprimé de la chaîne et le programme reprend en 40, sinon I est incrémenté. Le processus se poursuit jusqu'à ce que I soit supérieur à la longueur de la chaîne A\$.

Remarque : il serait possible d'utiliser une boucle FOR-NEXT en ligne 30 sous la forme :

```
30 FOR I=1 TO LEN(A$)
40 .....
50 NEXT I
```

Toutefois cette méthode présente un inconvénient puisque LEN(A\$) est évalué à l'entrée dans la boucle et ne tient donc pas compte de la suppression de caractères "espace" qui réduit ensuite cette longueur.

EXERCICE 3.2

PROGRAMME

```
10 CLS
20 LINE INPUT "TAPEZ UN TEXTE : ",A$
30 IF LEN(A$)<2 THEN 70
40 FOR I=LEN(A$)-1 TO 1 STEP -1
50 A$=LEFT$(A$,I)+" "+MID$(A$,I+1)
60 NEXT I
70 PRINT A$
80 PRINT:PRINT
90 END
```

VARIABLES

I compteur
A\$ chaîne de caractères à traiter

COMMENTAIRES

30 : il est impossible d'insérer des espaces si la longueur de la chaîne est 1 ou 0.
40 : LEN(A\$) est évalué comme la longueur initiale de A\$.

Remarque : les espaces sont ajoutés à partir de la fin de la chaîne. Pour commencer par le début, il faut écrire :

```
40 FOR I=1 TO LEN(A$)*2-2 STEP 2
50 ....
60 NEXT I
```

EXERCICE 3.3

PROGRAMME

```
10 CLS
20 LINE INPUT "TAPEZ UN MOT : ";P$
30 PRINT
40 PRINT
50 PRINT P$
60 P$=MID$(P$,2)
70 IF LEN(P$)>0 THEN 50
80 END
```

VARIABLE

P\$ chaîne à traiter

COMMENTAIRES

60 : MID\$(P\$,2) contient tous les caractères de P\$ sauf le premier.
70 : s'il reste des caractères dans P\$, alors le programme boucle en 50.

EXERCICE 3.4**PROGRAMME**

```

10 CLS
20 LINE INPUT "TAPEZ UN MOT : ";L$
30 PRINT L$
40 IF LEN(L$)<2 THEN 90
50 FOR I=1 TO LEN(L$)/2
60 L$=MID$(L$,2,LEN(L$)-2)
70 PRINT TAB(I+1);L$
80 NEXT I
90 END

```

VARIABLES

I compteur
L\$ chaîne de caractères à traiter

COMMENTAIRES

60 : le premier et le dernier caractère sont supprimés de la chaîne L\$.
70 : l'affichage à la position I + 1 permet de réaliser la tabulation demandée.

EXERCICE 3.5**PROGRAMME**

```

10 CLS
20 LINE INPUT "TAPEZ UNE PHRASE : ";P$
30 REM
40 REM DEBUT DE LA BOUCLE
50 GOSUB 200
60 FOR I=1 TO LEN(P$)
70 IF MID$(P$,I,1)=" " THEN P$=MID$(P$,I+1):G
OTO 50

```

```
80 NEXT I
90 PRINT
100 END
200 REM SOUS-PROGRAMME D'AFFICHAGE
210 IF MID$(P$,1,1)=" " THEN P$=MID$(P$,2):GO
    TO 210
220 PRINT P$
230 RETURN
```

VARIABLES

I compteur
P\$ phrase traitée

COMMENTAIRES

70 : si le lème caractère est un "espace", tous les caractères
 qui le précèdent et lui-même sont supprimés de P\$.
200-230 : sous-programme d'affichage.
210 : si la nouvelle phrase à afficher débute par un blanc, les
 espaces qu'il contient sont préalablement supprimés.

EXERCICE 3.6

PROGRAMME

```
10 CLS
20 LINE INPUT "LIGNE (MAX:20 CAR.) : ";L$
30 L=LEN(L$)
40 IF L<2 OR L>20 THEN 20
50 REM
60 REM De gauche a droite
70 REM *****
80 FOR I=1 TO 40-L+1
90 PRINT TAB(I)L$;
110 NEXT I
120 REM
130 REM De droite a gauche
140 REM *****
```

```

150 FOR I=40-L TO 2 STEP -1
160 PRINT TAB(I)L$;
180 NEXT I
190 GOTO 40

```

VARIABLES

I compteur
L longueur de L\$
L\$ ligne de texte

COMMENTAIRES

80-110 : la ligne de texte est affichée successivement, à partir de la
 colonne 1, puis 2,..., puis (40-L+1).
150-180 : la ligne est affichée successivement, à partir de la colonne
 (40-L) puis (40-L-1),..., 2.

EXERCICE 3.7

PROGRAMME

Solution 1 :

```

10 CLS
20 ON BREAK GOSUB 220
30 SPEED INK 5,2
40 A$=STRING$(40,".")
50 LINE INPUT "TAPEZ UNE LIGNE (MAX:39 CAR.)
: ";L$
60 L=LEN(L$)
70 IF L>39 THEN 50
80 REM
90 REM    Creation de la ligne
100 REM *****
110 L$=L$+RIGHT$(A$,40-L)
120 CLS
130 REM
140 REM    Defilement

```

```
150 REM *****
160 INK 3,0:PAPER 3:CLS
170 INK 1,6,26
180 LOCATE 1,10
190 PRINT L$
200 L$=RIGHT$(L$,39)+LEFT$(L$,1)
210 GOTO 180
220 INK 1,24
230 PAPER 0
240 END
```

VARIABLES

I compteur
L\$ texte à faire défiler
L longueur de L\$
A\$ ligne de points

COMMENTAIRES

- 20 : deux appuis successifs sur la touche BREAK font passer à la ligne 220 (retour à l'affichage initial : en jaune sur fond bleu).
- 30 : lors de l'utilisation d'une encre à 2 couleurs (ligne 170), la première couleur s'allume pendant 250 millisecondes (1/4 seconde) et la deuxième s'allume pendant 100 millisecondes (1/10 seconde).
- 40 : STRING\$ permet de construire une chaîne contenant 40 points.
- 110 : la ligne est complétée par suffisamment de points pour obtenir 40 caractères.
- 170 : l'encre numéro 1 a deux couleurs (6 : rouge, 25 : blanc) qui clignotent suivant un rythme défini par l'instruction SPEED INK (ligne 30).
- 200 : le premier caractère de la ligne est placé en dernière position.

Solution 2 :

```
10 CLS
20 ON BREAK GOSUB 160
30 SPEED INK 10,10
40 LINE INPUT "TAPEZ UNE LIGNE (MAX:39 CAR.)
: ";L$
```

```

50 L$=L$+" "
60 L=LEN(L$)
70 IF L>39 THEN 40
80 REM Défilement
90 REM *****
95 INK 0,0:CLS
100 INK 1,25,6
110 FOR I=40 TO 1 STEP -1
120 LOCATE 1,10
130 PRINT MID$(L$,41-I);TAB(I);LEFT$(L$,40-I)
140 NEXT I
150 GOTO 80
160 INK 1,24
170 END

```

VARIABLES

I compteur
L\$ texte à faire défiler

COMMENTAIRES

- 20 : deux appuis successifs sur la touche BREAK font passer à la ligne 220 (retour à l'affichage initial : en jaune sur fond bleu).
- 30 : définition du rythme de clignotement des couleurs de l'instruction INK.
- 50 : le texte est complété par un espace. Cet espace efface le dernier caractère de la dernière ligne affichée.
- 130 : lorsque le texte est affiché à partir de la lème colonne, une partie de ce texte peut déborder de l'écran. Dans ce cas, la partie restante est affichée à partir de la colonne 1.

EXERCICE 3.8

PROGRAMME

```

10 CLS
20 PRINT "TAPEZ UNE PHRASE A LA FORME NEGATIV
E"
30 LINE INPUT A$

```

```
40 REM
50 REM DEBUT DE LA BOUCLE DE RECHERCHE
60 REM *****
70 FOR I=LEN(A$) TO 1 STEP -1
80 IF MID$(A$,I,4)=" NE " THEN A$=LEFT$(A$,I)+
MID$(A$,I+4)
90 IF MID$(A$,I,3)=" N' " THEN A$=LEFT$(A$,I)+
MID$(A$,I+3)
100 IF MID$(A$,I,5)=" PAS " THEN A$=LEFT$(A$,
I)+MID$(A$,I+5)
110 NEXT I
120 IF RIGHT$(A$,4)=" PAS" THEN A$=LEFT$(A$,L
EN(A$)-4)
130 IF LEFT$(A$,3)="NE " THEN A$=MID$(A$,4)
140 IF LEFT$(A$,2)="N' " THEN A$=MID$(A$,3)
150 PRINT
160 PRINT "LA PHRASE A LA FORME AFFIRMATIVE E
ST : "
170 PRINT A$
180 PRINT
190 END
```

VARIABLES

I compteur
A\$ chaîne de caractères

COMMENTAIRES

- 70-100 : l'un des mots NE, PAS, N' est comparé à la partie de A\$,
 de même longueur. Si une correspondance est trouvée, le
 mot est retiré.
120 : teste le cas où "PAS" serait en fin de phrase.
130 : teste le cas où "NE" serait en début de phrase.
140 : teste le cas où "N" serait en début de phrase.

Remarque : LEFT\$(A\$,I) donne les premiers caractères de A\$, jus-
qu'au Ième inclus. Cela permet ici de laisser un espace entre les 2
mots entourant celui à supprimer.

EXERCICE 3.9

PROGRAMME

Solution 1 :

```

10 REM
20 REM RECHERCHE D'UNE SUITE DE LETTRES
30 REM *****
40 CLS
50 INPUT "QUELLE EST LA CLE";C#
60 C=LEN(C#)
70 PRINT
80 REM RECHERCHE
90 F=0
100 READ A#
110 ON ERROR GOTO 200
120 IF LEN(A#)<C THEN 100
125 PEN 2
130 FOR I=1 TO LEN(A#)-C+1
140 IF MID$(A#,I,C)<>C# THEN 170
150 PRINT TAB(20-I)A#
160 F=F+1
170 NEXT I
175 PEN 1
180 GOTO 100
190 REM
200 REM AFFICHAGE DES RESULTATS
210 REM *****
220 PRINT
230 PRINT "LA CLE ' ";C#;" ' A ETE TROUVEE";F
; "FOIS."
240 END
250 DATA LA, VOITURE, EST, ARRIVEE, QUAND, LE, TRA
IN
260 DATA ETAIT, EN, GARE, LE, BUREAU, DANS, LA, VOIT
URE
270 DATA SOUS, LE, TOIT, AVAIT, DEBORDE
280 DATA LA, RIVIERE, A, DEBORDE, PENDANT, TOUTE, L
A, NUIT

```

Solution 2 :

```

10 REM
20 REM RECHERCHE D'UNE SUITE DE LETTRES
30 REM *****
40 CLS
50 INPUT "QUELLE EST LA CLE";C$
60 C=LEN(C$)
70 PRINT
80 REM RECHERCHE
90 F=0
100 READ A$
110 ON ERROR GOTO 210
120 IF LEN(A$)<C THEN 100
125 PEN 2
130 FOR I=1 TO LEN(A$)-C+1
140 IF MID$(A$,I,C)<>C$ THEN 180
150 GOSUB 300
160 PRINT TAB(20-I)A$
170 F=F+1
180 NEXT I
185 PEN 1
190 GOTO 100
200 REM
210 REM AFFICHAGE DES RESULTATS
220 REM *****
230 PRINT
240 PRINT "LA CLE " ";C$;" " A ETE TROUVEE";F
; "FOIS."
250 END
300 REM
310 REM PASSAGE DE LA CLE EN MAJUSCULE
320 REM *****
330 AF$=LOWER$(A$)
340 AF$=LEFT$(AF$,I-1)+LEFT$(C$,C)+MID$(AF$,C
+I)
350 RETURN
500 DATA LA,VOITURE,EST, ARRIVEE,QUAND,LE,TRA
IN
510 DATA ETAIT,EN,GARE,LE,BUREAU,DANS,LA,VOIT
URE
520 DATA SOUS,LE,TOIT,AVAIT,DEBORDE
530 DATA LA,RIVIERE,A,DEBORDE,PENDANT,TOUTE,L
A,NUIT

```

VARIABLES

C\$ clé à rechercher
 C longueur de C\$
 A\$ mot du texte à étudier
 I compteur
 F nombre de fois où la clé a été trouvée
 AF\$ mot à afficher (solution 2)

COMMENTAIRES

110 : si une erreur intervient lors de la lecture de la liste DATA, la liste est considérée comme épuisée et le programme se poursuit à la ligne 210.
 120 : si la longueur de A\$ est inférieure à C, alors la clé ne peut pas se trouver dans ce mot et l'on passe au mot suivant (ligne 100).
 140 : la clé est comparée à toutes les chaînes de longueur C, construites sur A\$ à partir de la gauche.
 150 (sol. 1) : affichage d'un mot où la clé se trouve. L'affichage de
 160 (sol. 2) la chaîne de caractères, constituant la clé, commence toujours en colonne 20.
 300-350 : dans la solution 2, la clé est affichée en lettres majuscules tandis que le reste du mot est affiché en lettres minuscules. A la ligne 330, un mot qui contient la clé est transformé en lettres minuscules dans AF\$, puis en ligne 340, la clé est introduite dans AF\$.

EXERCICE 3.10

PROGRAMME

```

10 CLS
20 PRINT "ENTREZ UNE CHAINE DE CARACTERES"
30 LINE INPUT C$
40 PRINT
50 INPUT "CARACTERES A INSERER : ",I$
60 INPUT "POSITION D'INSERTION : ",P

```

```
70 IF P<0 OR P>LEN(C$)+1 THEN 60
80 C$=LEFT$(C$,P-1)+I$+MID$(C$,P)
90 PRINT
100 PRINT "RESULTAT : "
110 PRINT C$
120 PRINT
130 END
```

VARIABLES

P position d'insertion
C\$ chaîne initiale et finale
I chaîne à insérer

COMMENTAIRES

70 : test de la validité de la position d'insertion.
80 : insertion.

EXERCICE 3.11

PROGRAMME

```
10 CLS
20 REM
30 REM ENTREE DU TEXTE
40 REM *****
45 INK 2,6,9
50 LOCATE 1,3
60 PRINT "TAPEZ UN TEXTE (MAX:128 CARACTERES)
"
70 PRINT:LINE INPUT L$
80 IF L$="" OR LEN(L$)>128 THEN 60
90 PRINT
100 LINE INPUT "Mot a remplacer : ",R$
110 LINE INPUT "Mot a mettre a sa place : ",M$
$
```

```

120 REM
130 REM DEBUT DE LA RECHERCHE
140 REM *****
150 I=1:F=0:R=LEN(R$):M=LEN(M$)
160 REM
170 IF MID$(L$,I,R)<>R$ THEN 280
180 F=1
190 PRINT:IF LEN(L$)-R+M>128 THEN PRINT "CHAN
GEMENT IMPOSSIBLE":GOTO 340
200 F=1
210 REM DOIT-ON REMPLACER ?
220 PRINT:PRINT LEFT$(L$,I-1);
230 PEN 2:PRINT R$;:PEN 1
240 PRINT MID$(L$,I+R)
250 PRINT:PRINT "VOULEZ-VOUS REMPLACER (O/N)
? ";
260 A$=INKEY$:IF A$="" OR (A$<>"N" AND A$<>"O
") THEN 260 ELSE IF A$="N" THEN 280
270 L$=LEFT$(L$,I-1)+M$+MID$(L$,I+R):I=I+M-1
280 I=I+1:IF I<=LEN(L$)-R+1 THEN 160
290 REM
300 REM AFFICHAGE DU RESULTAT
310 REM *****
320 PRINT
330 IF F=0 THEN PRINT "Ce mot n'est pas dans
le texte "
340 PRINT:PRINT L$
350 END

```

VARIABLES

I compteur
F drapeau (**flag**) : 0 si le mot à remplacer n'est pas dans la ligne
L\$ texte
R\$ mot à remplacer
R longueur de R\$
M\$ mot à substituer
M longueur de M\$
A\$ touche frappée au clavier



COMMENTAIRES

- 80 : si la ligne ne contient aucun mot ou contient plus de 128 caractères, le traitement est impossible.
- 150 : initialisation du compteur, du drapeau et des longueurs de mots.
- 170 : si le mot à remplacer ne se trouve pas à la position I, le compteur est incrémenté (ligne 280).
- 200 : si le mot à remplacer se trouve à la position I, le drapeau est mis à 1, indiquant que le mot existe dans le texte.
- 230 : le mot à remplacer est affiché dans la ligne en mode clignotant.
- 260 : test de la réponse de l'utilisateur.
- 330 : si le drapeau est toujours à zéro, cela indique que le mot à remplacer n'a pas été trouvé.

EXERCICE 3.12

PROGRAMME

```
10 DIM E(40)
20 REM
30 ESP$=SPACE$(40)
40 REM ENTREE DE LA LIGNE DE TEXTE
50 REM *****
60 CLS
70 PRINT "TAPEZ UNE LIGNE (MAX: 40 CARACTERES)
  : "
80 LINE INPUT P$
90 IF RIGHT$(P$,1)=" " THEN P$=LEFT$(P$,LEN(P
$)-1):GOTO 90
100 IF LEFT$(P$,1)=" " THEN P$=MID$(P$,2):GOT
O 100
110 IF LEN(P$)>40 THEN 70
120 REM
130 REM recherche des espaces
140 REM *****
150 N=0
160 FLG=0
```

```

170 I=1
180 IF MID$(P$,I,1)<>" " THEN FLG=0 ELSE IF F
LG=0 THEN N=N+1:E(N)=I:FLG=1 ELSE P$=LEFT$(P$
,I-1)+MID$(P$,I+1):I=I-1
190 I=I+1
200 IF I<=LEN(P$) THEN 180
210 IF N=0 THEN PRINT "PAS ASSEZ DE MOTS":END
220 REM
230 REM justification
240 REM *****
250 B=40-LEN(P$)
260 FOR I=N TO 1 STEP -1
270 P$=LEFT$(P$,E(I))+STRING$(INT(B/I)," ")+M
ID$(P$,E(I)+1)
280 B=B-INT(B/I)
290 NEXT I
300 REM
310 REM affichage
320 REM *****
330 PRINT:PRINT P$
340 END

```

VARIABLES

ESP\$ chaîne de 40 blancs
I compteur
N nombre d'espaces de P\$
B nombre d'espaces à ajouter à P\$ pour obtenir 40 caractères
P\$ ligne à justifier
E(40) tableau des positions des espaces contenus initialement dans P\$
FLG drapeau indiquant qu'un espace contient plusieurs blancs

COMMENTAIRES

90 : suppression des espaces éventuels mis en fin de P\$.
100 : suppression des espaces éventuels mis en début de P\$.
130-200 : les positions des fins de mots dans P\$ sont rangées par ordre croissant, dans le tableau E. Si plusieurs blancs ont été mis entre mots, ils sont comprimés en un seul.
210 : si P\$ ne contient aucun espace, il est impossible de justifier la ligne.

260-290 : les espaces sont ajoutés à partir de la fin de P\$. Ainsi les valeurs précédentes de E contiennent les positions réelles des fins de mots de P\$.

EXERCICE 3.13

PROGRAMME

```
10 CLS
20 PRINT "ENTREZ UNE LIGNE DE TEXTE "
30 PRINT:LINE INPUT L$
40 IF L$="" THEN 20
50 REM
60 REM Creation de la chaine inverse
70 REM *****
80 I$=""
90 FOR I=LEN(L$) TO 1 STEP -1
100 I$=I$+MID$(L$,I,1)
110 NEXT I
120 REM
130 REM Creation de la chaine aleatoire
140 REM *****
150 T$=L$:A$=""
160 FOR I=1 TO LEN(L$)
170 A=INT(RND*LEN(T$))+1
180 A$=A$+MID$(T$,A,1)
190 T$=LEFT$(T$,A-1)+MID$(T$,A+1)
200 NEXT I
210 REM
220 REM Affichage
230 REM *****
240 PRINT:PRINT
250 PRINT L$:PRINT I$:PRINT A$
260 END
```

VARIABLES

I compteur
A position aléatoire d'un caractère de T\$
L\$ ligne de texte

I\$ ligne inverse
 A\$ ligne aléatoire
 T\$ variable alphanumérique

COMMENTAIRES

150 : la ligne de texte est recopiée dans T\$ pour pouvoir être manipulée sans perdre l'original. A\$ est initialisée.
 170 : INT(RND*LEN(T\$))+1 donne un nombre aléatoire entre 1 et la longueur de T\$, qui est affecté à A.
 180-190 : le Aème caractère de T\$ est ajouté à A\$ puis supprimé de T\$.

EXERCICE 3.14

PROGRAMME

```

10 REM
20 REM Pourcentage d'apparition des
30 REM lettres dans un texte
40 REM *****
50 DIM L(26),T$(10)
60 CLS
70 INPUT "Nombre de phrases (1 a 10) ",N
80 IF N<1 OR N>10 THEN 60
90 FOR I=1 TO N
100 PRINT:PRINT TAB(12)"PHRASE ";I;":"
110 LINE INPUT T$(I)
120 T$(I)=UPPER$(T$(I))
130 NEXT I
140 REM
150 REM Calcul des pourcentages
160 REM *****
170 FOR J=1 TO N
180 FOR K=1 TO LEN(T$(J))
190 LSC=ASC(MID$(T$(J),K,1))
200 IF LSC>64 AND LSC<91 THEN L(LSC-64)=L(LSC
-64)+1
210 NEXT K
220 NEXT J

```

```
230 REM
240 S=0
250 FOR I=1 TO 26:S=S+L(I):NEXT I
260 FOR J=1 TO 26
270 L(J)=ROUND(L(J)/S,2)
280 NEXT J
290 REM
300 REM Affichage des resultats
310 REM *****
320 CLS
322 PRINT "Texte etudie : "
325 FOR I=1 TO N:PRINT T$(I):NEXT I
328 PRINT
330 FOR I=1 TO 13
340 PRINT CHR$(64+I); " : ";
350 PRINT USING "###.##";L(I);
360 PRINT TAB(20);CHR$(77+I); " : ";
370 PRINT USING "###.##";L(I+13)
380 NEXT I
390 END
```

VARIABLES

I,J,K compteurs
N nombre de phrases
S nombre total de lettres dans T\$(10)
LSC code ASCII du caractère testé
L(26) tableau des pourcentages d'apparition des lettres de l'alphabet
T\$(10) tableau contenant les phrases du texte

COMMENTAIRES

120 : le texte entier est mis en lettres majuscules.
200 : si le caractère testé est une lettre majuscule, son code ASCII est compris entre 65 et 90. Dans ce cas l'élément du tableau L d'indice égal à (LSC-64) est incrémenté. Par exemple : si la lettre est H de code ASCII 72, l'élément d'indice 8 du tableau L est incrémenté.
250 : calcul du nombre total de lettres contenues dans T\$.
270 : calcul arrondi à deux décimales (instruction ROUND) du pourcentage d'apparition de la Jième lettre.

330-380 : affichage sur deux colonnes des lettres de l'alphabet, suivies de leur pourcentage de présence avec alignement sur le point décimal.

EXERCICE 3.15

PROGRAMME

```

10 CLS
20 INPUT "JOUR (DE 1 A 31) ",J
30 INPUT "NUMERO DU MOIS (DE 1 A 12) ",M
40 INPUT "ANNEE ";A
50 IF J<1 OR J>31 OR M<1 OR M>12 THEN 20
60 FOR I=1 TO M:READ M$:NEXT I
70 D$=STR$(J)+" "+M$+STR$(A)
80 PRINT:PRINT D$:PRINT
90 END
100 DATA JANVIER,FEVRIER,MARS,AVRIL
110 DATA MAI,JUIN,JUILLET,AOUT
120 DATA SEPTEMBRE,OCTOBRE,NOVEMBRE
130 DATA DECEMBRE

```

VARIABLES

I compteur
J numéro du jour
M numéro du mois
A année
D\$ date
M\$ mième mois de l'année

COMMENTAIRES

50 : vérification de la validité de la date d'entrée.
60 : recherche du Mième mois dans la liste de DATA.
70 : création de la date avec le mois en toutes lettres.

EXERCICE 3.16

PROGRAMME

```

10 REM *****
20 CLS
30 ME$="ERREUR (!!)"
40 INPUT "DATE (EX.: 11 JANVIER 1963) : ",DATE$
50 D$=UPPER$(DATE$)
60 REM
70 REM Recherche du jour
80 REM *****
90 IF LEFT$(D$,1)="0" THEN D$=MID$(D$,2)
100 FOR I=1 TO LEN(D$)
110 IF MID$(D$,I,1)=" " THEN 140
120 NEXT I
130 PRINT ME$:GOTO 40
140 J=VAL(LEFT$(D$,I-1))
150 IF J>31 OR J<=0 OR J<>INT(J) OR STR$(J)<>
" "+LEFT$(D$,I-1) THEN PRINT ME$:GOTO 40
160 REM
170 REM Recherche de l'annee
180 REM *****
190 D$=MID$(D$,I+1)
200 FOR I=1 TO LEN(D$)
210 IF MID$(D$,I,1)=" " THEN 230
220 NEXT I
230 B$=MID$(D$,I+1)
240 IF LEFT$(B$,1)="0" THEN B$=MID$(B$,2)
250 A=VAL(B$)
260 IF A<=0 OR A<>INT(A) OR STR$(A)<>" "+B$ T
HEN PRINT ME$:GOTO 40
270 REM
280 REM Recherche du mois
290 REM *****
300 M$=LEFT$(D$,I-1)
310 REM
320 REM Recherche du numero du mois
330 REM *****
340 RESTORE

```

```

350 FOR M=1 TO 12
360 READ A$
370 IF A$=M$ THEN 400
380 NEXT M
390 PRINT ME$;GOTO 40
400 REM
410 REM Affichage du resultat
420 REM *****
430 PRINT
440 PRINT TAB(10)J;" / ";M;" / ";A
450 PRINT
460 END
470 DATA JANVIER,FEVRIER,MARS,AVRIL
480 DATA MAI, JUIN, JUILLET, AOUT
490 DATA SEPTEMBRE, OCTOBRE
500 DATA NOVEMBRE, DECEMBRE

```

VARIABLES

I	compteur
J	numéro du jour
M	numéro du mois
A	année
DATE\$	date
A\$,B\$,D\$	variables alphanumériques
ME\$	message d'erreur
M\$	mois en lettres

COMMENTAIRES

90 : si le numéro du jour commence par un 0, celui-ci n'est pas pris en compte.

110 : recherche du premier espace de D\$.

150 : test de la validité du numéro du jour.

210 : le 2ème espace dans D\$ repère le mois.

240 : si le numéro de l'année commence par un 0, celui-ci n'est pas pris en compte.

260 : test de la validité du numéro de l'année.

340-390 : recherche du numéro du mois.

EXERCICE 3.17

PROGRAMME

```

10 REM
20 REM ECRITURE DE CHIFFRES GEANTS
30 REM *****
40 DIM C$(9,5)
50 REM
60 REM Lecture du graphisme
70 REM   des chiffres
80 REM *****
90 FOR I=0 TO 9
100 FOR J=0 TO 5
110 READ C$(I,J)
120 NEXT J,I
140 REM
150 CLS
160 INPUT "ECRIRE UN NOMBRE ENTRE 0 ET 99999
: ",M$
170 M=LEN(M$)
180 IF M<1 OR M>5 THEN 160
190 REM
200 REM Affichage
210 REM *****
220 PRINT:PRINT
230 FOR I=0 TO 5
240 FOR J=1 TO M
250 D=VAL(MID$(M$,J,1))
260 PRINT C$(D,I);" ";
270 NEXT J
280 PRINT
290 NEXT I
300 END
310 REM
320 REM Graphismes des 10 chiffres
330 REM *****
340 REM
350 REM chiffre 0
360 DATA " *** ", "* *", "* *", "* *", "* *",
* ", " *** "

```

```

370 REM
380 REM chiffre 1
390 DATA " ** ", " * ", " * ", " * ", " * "
    ", "*****"
400 REM
410 REM chiffre 2
420 DATA " *** ", " * ", " * ", " * ", " * "
    ", "*****"
430 REM
440 REM chiffre 3
450 DATA " *** ", " * ", " * ", " * ", " * "
    *", " *** "
460 REM
470 REM chiffre 4
480 DATA "      *", " * ", " * ", " * ", " * "
    **", " * "
490 REM
500 REM chiffre 5
510 DATA "**** ", " * ", " * ", " * ", " * "
    *", "**** "
520 REM
530 REM chiffre 6
540 DATA " *** ", " * ", " * ", " * ", " * "
    *", " *** "
550 REM
560 REM chiffre 7
570 DATA "*****", " * ", " * ", " * ", " * "
    ", " * "
580 REM
590 REM chiffre 8
600 DATA " *** ", " * ", " * ", " * ", " * "
    *", " *** "
610 REM
620 REM chiffre 9
630 DATA " *** ", " * ", " * ", " * ", " * "
    *", " *** "
640 END

```

VARIABLES

I, J compteurs
D chiffre à afficher

M nombre de chiffres à afficher
M\$ nombre à afficher avec les graphismes géants
C\$(9,5) tableau contenant les dessins des chiffres géants

COMMENTAIRES

90-120 : lecture des dessins géants. Le premier indice de C\$
 (compris entre 0 et 9) désigne le chiffre à dessiner, le se-
 cond (compris entre 0 et 5) la ligne du dessin.
230 : l'indice I repère les 6 lignes du graphisme.
260 : affichage de la lième ligne correspondant au Jième gra-
 phisme suivi de 2 espaces pour séparer les différents
 chiffres.
280 : passage à la ligne suivante après M affichages.

EXERCICE 3.18

PROGRAMME

```
10 REM
20 REM Programme de DUMP memoire
30 REM *****
40 REM
50 INPUT "Adresse de debut :",AD
60 INPUT "Adresse de fin :",AF
70 FOR I=AD TO AF STEP 8
80 PRINT HEX$(I,4) " : ";
90 FOR J=0 TO 7
100 CMEM=PEEK(J+I)
110 PRINT HEX$(CMEM,2) " ";
120 NEXT J
130 FOR J=0 TO 7
140 CMEM=PEEK(J+I)
150 IF CMEM<33 OR CMEM>126 THEN A$="." ELSE A
    $=CHR$(CMEM)
160 PRINT A$;
170 NEXT J
180 PRINT
190 NEXT I
200 END
```



VARIABLES

AD adresse de début
 AF adresse de fin
 I,J compteurs
 CMEM contenu d'une case mémoire
 A\$ caractère de code ASCII, CMEM

COMMENTAIRES

110 : conversion du contenu (octet) de la case mémoire en un
 nombre hexadécimal.
 150 : si l'octet CMEM est compris entre 33 et 126, il correspond
 au code ASCII d'un caractère affichable.

EXERCICE 3.19

PROGRAMME

```

10 REM
20 REM CRYPTAGE D'UN TEXTE
30 REM *****
40 CLS
50 REM
60 REM Alphabet et codage
70 REM *****
80 ALPHA$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
90 CODE$="QWERTYUIOPASDFGHJKLZXCVBNM"
100 REM
110 PRINT "ENTREZ LE TEXTE A CODER : "
120 LINE INPUT TEXT$
130 TEXT$=UPPER$(TEXT$)
140 CTXT$=""
150 FOR I=1 TO LEN(TEXT$)
160 A$=MID$(TEXT$, I, 1)
170 P=INSTR(ALPHA$, A$)
180 IF P=0 THEN CTXT$=CTXT$+A$ ELSE CTXT$=CTX
T$+MID$(CODE$, P, 1)
190 NEXT I

```

```
200 CLS
210 PRINT:PRINT
220 PRINT "LE TEXTE CRYPTÉ EST : "
225 PRINT
230 PRINT CTXT$
240 PRINT
250 END
```

VARIABLES

ALPHA\$ alphabet naturel
CODE\$ alphabet de codage
TEXT\$ texte en clair
CTXT\$ texte crypté
P variable numérique
A\$ variable alphanumérique
I compteur

COMMENTAIRES

90 : on affecte à CODE\$ la suite de l'alphabet de cryptage.
130 : transformation du texte en lettres majuscules.
170, 180 : P donne la position dans ALPHA\$ de la lettre à coder. Si P
 est différent de 0, on extrait alors la Pième lettre de CODE\$.

EXERCICE 3.20

PROGRAMME

```
10 REM
20 REM FICHE DE SAISIE DE DONNEES
30 REM *****
40 XINI(1)=6:XINI(2)=9:XINI(3)=10:XINI(4)=12
50 CLS
60 REM
70 REM Initialisation des fenetres
80 REM *****
90 WINDOW #1,1,40,5,6
100 WINDOW #2,1,40,7,8
```

```

110 WINDOW #3,1,40,9,10
120 WINDOW #4,1,40,11,12
130 PRINT #1,"NOM"STRING$(37, ".")
140 PRINT #2,"Prenom"STRING$(34, ".")
150 PRINT #3,"Adresse"STRING$(33, ".")
160 PRINT #4,"Telephone"STRING$(31, ".")
170 REM
180 FOR I=1 TO 4:X(I)=XINI(I):NEXT I
190 I=1
200 LOCATE #I,X(I),1:PRINT #I,CHR$(143);
210 REM
220 REM Acquisition d'une touche
230 REM *****
240 A$=INKEY$:IF A$="" THEN 240
250 REM
260 REM Comparaison a une valeur
270 REM alphanumerique
280 REM *****
290 IF (A$>="A" AND A$<="Z") OR (A$>="a" AND
A$<="z") OR (A$>="0" AND A$<="9") OR A$=CHR$(
32) THEN LOCATE #I,X(I),1:PRINT #I,A$;:GOTO 5
50
300 REM
310 REM La touche n'est pas prise
320 REM en compte
330 REM *****
340 IF A$<>CHR$(13) AND A$<>CHR$(240) AND A$<
>CHR$(241) AND A$<>CHR$(127) THEN 240
350 IF A$=CHR$(240) AND I=1 THEN 240
360 IF A$=CHR$(241) AND I=4 THEN 240
370 LOCATE #I,X(I),1:PRINT #I, ".";
380 REM
390 REM Touche ENTER ?
400 REM *****
410 IF A$=CHR$(13) THEN LOCATE 1,20:END
420 REM
430 REM Touches de deplacement ?
440 REM *****
450 IF A$=CHR$(240) THEN IF I>1 THEN I=I-1:GO
TO 200
460 IF A$=CHR$(241) THEN IF I<4 THEN I=I+1:GO
TO 200

```

```
470 REM
480 REM Touche DEL ?
490 REM *****
500 IF A$=CHR$(127) THEN X(I)=X(I)-2
510 REM
520 REM Emplacement du prochain
530 REM caractere de la zone I
540 REM *****
550 X(I)=X(I)+1
560 IF X(I)<=XINI(I) THEN X(I)=XINI(I)
570 IF X(I)>39 THEN X(I)=40
580 GOTO 200
```

VARIABLES

XINI tableau contenant les positions initiales dans une rubrique
I compteur
X tableau contenant les positions courantes dans une rubrique
A\$ touche frappée au clavier

COMMENTAIRES

40 : initialisation de la position du curseur dans chaque rubrique.
90-120 : définition d'une fenêtre par rubrique.
200 : dans la lième rubrique, on affiche le curseur (pavé jaune), à la position donnée par X(I).
290 : les seuls caractères acceptés sont les lettres (minuscules ou majuscules) et les chiffres.
350, 360 : teste si le curseur reste dans les limites de la fiche.
390-410 : traitement de la touche ENTER.
430-460 : traitement des touches de déplacement vertical.
480-580 : traitement de la touche DEL.

EXERCICE 4.1**PROGRAMME**

```

10 REM
20 REM Histogramme en batons
30 REM *****
40 DIM N(15)
50 CLS
60 FOR I=1 TO 15
70 PRINT "Nombre (0-100)";USING " ##";I;
80 INPUT " : ",N(I)
90 IF N(I)<0 OR N(I)>100 THEN 70
100 NEXT I
110 REM
120 REM Affichage de l'histogramme
130 REM *****
140 INK 2,8:INK 3,0
150 PAPER 3
160 CLS
170 LOCATE 15,4:PRINT "Histogramme"
180 PRINT CHR$(22)+CHR$(1)
190 LOCATE 15,4:PRINT STRING$(11,"_")
200 PRINT CHR$(22)+CHR$(0)
210 LOCATE 1,1
220 FOR I=1 TO 15
230 C=1-C:P=1+C:PEN P
240 NBG=FIX(N(I)*30/100)
250 L$=STRING$(NBG,CHR$(143))
260 LOCATE 1,I+5:PRINT USING " ##"+L$;I
270 NEXT I
280 IF INKEY$="" THEN 280
290 PAPER 0
300 PEN 1
310 END

```

VARIABLES

P,C variables
I compteur

N tableau des valeurs à afficher
NBG nombre de pavés, proportionnel à la valeur à représenter
L\$ bande de longueur proportionnelle à la valeur à représenter

COMMENTAIRES

180, 190 : passage en mode transparent, permettant de souligner les caractères du mot histogramme.
230 : le changement de couleur est obtenu en utilisant la variable auxiliaire C, qui prend la valeur 0 et 1 successivement à chaque exécution de la ligne 230. En effet si C est égal à 0, (1 - C) prend la valeur 1 et si C est égal à 1, (1 - C) prend la valeur 0. P prend donc les valeurs 1 et 2. Cette méthode permet de commuter l'encre du stylo.
240 : NBG représente la valeur entière par défaut, du nombre de positions de la lième bande à afficher. Chaque pavé représente ainsi une variation de 3,33 (100/30).
250 : L\$ est constituée de pavés pleins (caractère CHR\$(143)) en nombre proportionnel à la valeur à afficher.
260 : PRINT USING permet d'afficher la valeur de I devant la bande.

EXERCICE 4.2

PROGRAMME

```
10 REM
20 REM Histogramme en batons
30 REM *****
40 DIM N(20)
50 CLS
60 INPUT "Nombre de valeurs (2-20) : ",N
65 PRINT
70 IF N<2 OR N>20 THEN 60
80 FOR I=1 TO N
90 PRINT "Nombre (0-1000)";USING " ##";I;
100 INPUT " : ",N(I)
110 IF N(I)<0 OR N(I)>=1000 THEN 90
120 NEXT I
```

```

130 REM
140 REM Recherche de la plus
150 REM     grande valeur
160 REM *****
170 G=N(1)
180 FOR I=2 TO N
190 G=MAX(G,N(I))
200 NEXT I
210 REM
220 REM Calcul du pas
230 REM *****
240 PAS=G/30
250 REM
260 REM Affichage de l'histogramme
270 REM *****
280 INK 2,8:INK 3,0
290 PAPER 3
300 CLS
310 LOCATE 15,4:PRINT "Histogramme"
320 PRINT CHR$(22)+CHR$(1)
330 LOCATE 15,4:PRINT STRING$(11,"_")
340 PRINT CHR$(22)+CHR$(0)
350 LOCATE 1,1
360 FOR I=1 TO N
370 C=1-C:P=1+C:PEN P
380 NBG=N(I)/PAS
390 L$=STRING$(NBG,CHR$(143))
400 LOCATE 1,I+5:PRINT USING " ##"+L$+" ###.#
";I;N(I);
410 NEXT I
420 IF INKEY$="" THEN 420
430 PAPER 0
440 PEN 1
450 END

```

VARIABLES

P,C variables
I compteur
N nombre de valeurs à étudier
N (20) tableau des valeurs à afficher

PAS plus petite variation contenue dans un pavé
G plus grande valeur du tableau N
NBG nombre de pavés, proportionnel à la valeur à afficher
L\$ bande formée de pavés pleins (CHR\$(143))

COMMENTAIRES

240 : la valeur correspondant à un pavé sur l'histogramme est égale au rapport de la plus grande valeur par le nombre maximal de pavés.
390 : l'instruction STRING\$ convertit NBG en entier, si nécessaire.
400 : affichage de la ligne de couleurs correspondant à la lième valeur et de la valeur N(I). Le format USING affiche le rang I sur 2 positions, puis L\$ et enfin N(I) sur 5 positions (3 pour la partie entière, 1 pour le point, 1 pour la partie décimale).

EXERCICE 4.3

PROGRAMME

```
10 REM
20 REM HISTOGRAMME COMPARATIF
30 REM *****
40 CLS
50 DIM N(1,9)
60 PRINT "Valeurs (Premiere,seconde) "
70 PRINT
80 FOR I=0 TO 9
90 PRINT SPC(5) "Pour "+STR$(1974+I);: INPUT " :
   ",N(0,I),N(1,I)
100 IF N(0,I)<0 OR N(1,I)<0 THEN 90
110 NEXT I
120 REM
130 REM Recherche de la plus grande
140 REM         valeur
150 REM *****
160 FOR I=0 TO 9
170 G=MAX(G,N(0,I),N(1,I))
```

```

180 NEXT I
190 PAS=G/30
200 REM
210 REM initialisation des
220 REM parametres d'affichage
230 REM *****
240 INK 0,1
250 INK 1,24
260 INK 2,B
270 INK 3,0
280 PAPER 3
290 CLS
300 PRINT TAB(9)"HISTOGRAMME COMPARATIF"
310 PRINT TAB(9)"Produit A : "+CHR$(143)
320 PRINT TAB(9)"Produit B : ";:PEN 2:PRINT C
HR$(143):PEN 1
330 PRINT TAB(9)"VMAX:"ROUND(G,2)
340 PRINT
350 C=1
360 FOR I=0 TO 9
370 FOR J=0 TO 1
380 NBG=N(J,I)/PAS
390 L$=STRING$(NKG,CHR$(143))
400 C=1-C:P=C+1:PEN P
410 PRINT USING "####"+" "+L$;1974+I
420 NEXT J
430 NEXT I
440 IF INKEY$="" THEN 440
450 REM
460 REM Parametres de retour
470 REM *****
480 PAPER 0
490 PEN 1
500 PRINT
510 END

```

VARIABLES

I,J	compteurs
G	plus grande valeur du tableau N
NBG	nombre de pavés correspondant à la lième valeur
L\$	chaîne formée de pavés (CHR\$(143)) en nombre égal à NBG

PAS valeur correspondant à un pavé (pas de quantification)
N(1,9) tableau des valeurs des 2 articles sur dix années
P,C variables

COMMENTAIRES

80-110 : entrée des deux séries de valeurs, année par année.
330, 390 : affichage d'un bâton horizontal de longueur proportionnelle à la valeur à représenter. On utilise la couleur magenta pour un article et la couleur jaune pour l'autre. Chaque bâton de l'histogramme est précédé de l'année plus 2 espaces et est suivi de la valeur correspondante, arrondie à un chiffre après la virgule.

EXERCICE 4.4

PROGRAMME

```
10 REM
20 REM GRAPHIQUE EN SECTEURS
30 REM *****
40 CLS
50 INPUT "VOULEZ-VOUS UN TRACE SOIGNE (O/N) :
  ",REP$
60 IF REP$="O" OR REP$="o" THEN AGPAS=0.001 E
LSE AGPAS=0.1
70 READ NVAL
80 FOR I=1 TO NVAL
90 READ N(I)
100 NEXT I
110 REM
120 REM Initialisation des
130 REM parametres d'affichage
140 REM *****
150 INK 0,1
160 INK 1,24
170 INK 2,8
180 INK 3,0
190 PAPER 3
```

```

200 PAPER #1,3
210 RAD
220 REM
230 REM Valeurs initiales
240 REM *****
250 ORIGIN 200,200:RAYON=150
260 WINDOW #1,30,40,1,25
270 REM
280 REM Calcul du total
290 REM *****
300 G=0
310 FOR I=1 TO NVAL
320 G=G+N(I)
330 NEXT I
340 REM
350 REM Calcul des angles
360 REM *****
370 AG(0)=0
380 FOR I=1 TO NVAL
390 AG(I)=AG(I-1)+2*PI*N(I)/G
400 NEXT I
410 REM
420 REM Trace du graphique en secteurs
430 REM *****
440 CLS
450 PRINT TAB(5)"Graphique en secteurs"
460 PRINT #1,"Valeur(s)"
470 FOR I=1 TO NVAL
480 C=1-C:P=1+C
490 FOR J=AG(I-1) TO AG(I) STEP AGPAS
500 PLOT 0,0,P
510 DRAW RAYON*COS(J),RAYON*SIN(J)
520 NEXT J
530 NEXT I
540 REM
550 REM Affichage des valeurs
560 REM *****
570 PRINT #1
580 FOR I=1 TO NVAL
590 PRINT #1,ROUND(N(I),3)
600 NEXT I
610 IF INKEY$="" THEN 610

```

```
620 PAPER 0
630 LOCATE 1,23
640 END
650 DATA 10
660 DATA 12.5, 12.5, 25, 25, 45.3, 23.89, 56, 34, 12.
9, 12.8
```

VARIABLES

AGPAS pas angulaire
REP\$ réponse
NVAL nombre de valeurs traitées
I,J compteurs
N tableau des valeurs
G valeur maximale de N
AG tableau des angles des secteurs
P,C variables
RAYON rayon du cercle

COMMENTAIRES

60 : suivant la valeur de REP\$, on affecte 0.001 ou 0.1 à AGPAS.
80-100 : lecture des valeurs passées par une liste de DATA.
250 : l'origine graphique est placée au milieu de l'écran.
260 : une fenêtre est définie pour l'affichage du tableau N.
300-330 : calcul de la somme G des valeurs de N.
350-400 : calcul des angles : à la valeur de rang I, on fait correspondre un angle AG(I) égal au pourcentage de N(I) par rapport à G, multiplié par $2 \times \text{PI}$ (cercle complet) et augmenté de AG(I-1).
490-520 : tracé du secteur de rang I : on affiche des segments, de longueur donnée (RAYON), depuis l'origine par pas de AGPAS en partant de AG(I-1) jusqu'à AG(I).
560-600 : affichage des valeurs dans la fenêtre 1.

Remarque : le tracé soigné nécessite quelques minutes. Ne soyez pas trop impatient !

EXERCICE 4.5

PROGRAMME

```

10 REM
20 REM GRAPHIQUE EN BANDES
30 REM *****
40 DIM LG(30),N(30)
50 READ NVAL
60 FOR I=1 TO NVAL
70 READ N(I)
80 NEXT I
90 REM
100 REM Initialisation des
110 REM parametres d'affichage
120 REM *****
130 INK 0,1
140 INK 1,24
150 INK 2,8
160 INK 3,0
170 PAPER 3
180 PAPER #1,3
190 WINDOW #1,30,40,1,25
200 DRIGIN 10,200
210 L100=400
220 REM
230 REM Calcul du total
240 REM *****
250 G=0
260 FOR I=1 TO NVAL
270 G=G+N(I)
280 NEXT I
290 REM
300 REM Calcul des longueurs
310 REM *****
320 LG(0)=0
330 FOR I=1 TO NVAL
340 LG(I)=LG(I-1)+L100*N(I)/G
350 NEXT I
360 REM
370 REM Trace du graphique en bandes

```

```
380 REM *****
390 CLS
400 PRINT TAB(5)"Graphique en bandes"
410 PRINT #1,"Valeur(s)"
420 FOR J=1 TO 20
430 C=0
440 FOR I=1 TO NVAL
450 C=1-C:P=1+C:PEN P
460 PLOT LG(I-1),J,P
470 DRAW LG(I),J
480 NEXT I
490 NEXT J
500 REM
510 REM Affichage des valeurs
520 REM *****
530 PRINT #1
540 FOR I=1 TO NVAL
550 PRINT #1,ROUND(N(I),3)
560 NEXT I
570 IF INKEY$="" THEN 570
580 PAPER 0
590 PEN 1
600 LOCATE 1,23
610 END
620 DATA 7
630 DATA 12.5,12.5,25,25,45.3,23.89,76,34,12.
9,2.8,6.7,32.9,23.5
```

VARIABLES

LG tableau des longueurs des segments de la bande
N tableau des valeurs
NVAL nombre de valeurs
I,J compteurs
L100 nombre de points de la bande complète
G total des valeurs de N
P,C variables

COMMENTAIRES

70-90 : lecture du tableau N à partir de la liste de DATA.
190 : définition de la fenêtre d'affichage 1.

200 : l'origine est mise en 10,200.
 210 : L100 est pris égal à 400.
 230-280 : calcul de G.
 300-350 : calcul de LG(I), distance du point terminal du lième segment à partir du début de la bande.
 420-490 : tracé de la bande LG(I) constituée de 20 lignes horizontales.

EXERCICE 4.6

PROGRAMME

```

10 REM
20 REM GRAPHIQUE EN COLONNES
30 REM *****
40 DIM LG(30),N(30)
50 READ NVAL
60 FOR I=1 TO NVAL
70 READ N(I)
80 NEXT I
90 REM
100 REM Initialisation des
110 REM parametres d'affichage
120 REM *****
130 INK 0,1
140 INK 1,24
150 INK 2,8
160 INK 3,0
170 PAPER 3
180 PAPER #1,3
190 WINDOW #1,30,40,1,25
200 ORIGIN 200,20
210 L100=300
220 REM
230 REM Calcul du total
240 REM *****
250 G=0
260 FOR I=1 TO NVAL
270 G=G+N(I)

```

EXERCICES EN BASIC POUR AMSTRAD

```
280 NEXT I
290 REM
300 REM Calcul des longueurs
310 REM *****
320 LG(0)=0
330 FOR I=1 TO NVAL
340 LG(I)=LG(I-1)+L100*N(I)/G
350 NEXT I
360 REM
370 REM Trace du graphique en colonnes
380 REM *****
390 CLS
400 PRINT TAB(4)"Graphique en colonnes"
410 PRINT #1,"Valeur(s)"
420 FOR J=1 TO 20
430 C=0
440 FOR I=1 TO NVAL
450 C=1-C:P=1+C:PEN P
460 PLOT J, LG(I), P
470 DRAW J, LG(I-1)
480 NEXT I
490 NEXT J
500 REM
510 REM Affichage des valeurs
520 REM *****
530 PRINT #1
540 FOR I=NVAL TO 1 STEP -1
550 PRINT #1, ROUND(N(I), 3)
560 NEXT I
570 IF INKEY$="" THEN 570
580 PAPER 0
590 PEN 1
600 LOCATE 1, 23
610 END
620 DATA 10
630 DATA 12.5, 12.5, 25, 25, 45, 3, 23.89, 76, 34, 12.
9, 2.8, 6.7, 32.9, 23.5
```

VARIABLES

LG tableau des longueurs des segments
N tableau des valeurs

NVAL nombre de valeurs
 I,J compteurs
 L100 longueur de la colonne complète
 G total des valeurs de N
 P,C variables

COMMENTAIRES

470 : affichage d'une colonne verticale de hauteur proportionnelle à la valeur représentée.

EXERCICE 4.7

PROGRAMME

```

10 REM
20 REM STATISTIQUES DE LANCERS DE DES
30 REM *****
40 DIM V(36),N(36)
50 CLS
60 LOCATE 1,10
70 INPUT "Choisir un nombre de des (1 a 6) :
",N
80 IF N<1 OR N>6 OR N<>INT(N) THEN 50
90 RANDOMIZE(TIME)
100 CLS
110 REM
120 REM Lancer des des
130 REM *****
140 PRINT "SOMME DES FACES SORTIES
150 PRINT "Pour arreter, frapper une touche"
160 PRINT
170 FOR I=1 TO 600
180 IF INKEY$<>" " THEN 310
190 S=0
200 FOR J=1 TO N
210 S=S+INT(6*RND)+1
220 NEXT J
230 V(S)=V(S)+1

```

EXERCICES EN BASIC POUR AMSTRAD

```
240 PRINT USING "####";S;
250 G=0
260 FOR J=N TO 6*N
270 N(J)=V(J)/I
280 IF G<N(J) THEN G=N(J)
290 NEXT J
300 NEXT I
310 REM
315 NL=I-1
318 REM
320 REM Affichage de l'histogramme
330 REM *****
340 INK 1,25
350 INK 0,0
360 ORIGIN 25,60
380 PAPER 0
390 PEN 1
395 WINDOW #1,28,40,3,8
396 PAPER #1,0
397 PEN #1,1
400 CLS
405 REM
406 REM Trace
410 REM *****
420 FOR J=N TO 6*N
430 FOR I=0 TO 14
440 PLOT 15*(J-N)+8+I,0
450 IF I=0 THEN DRAWR 0,4:PLDTR 0,-4
460 DRAWR 0,200*N(J)/G
470 NEXT I
480 NEXT J
490 PLOT 75*N+23,0:DRAWR 0,4
500 LOCATE 2,23:PRINT N
520 PRINT #1,"Histogramme"
530 PRINT #1,"de";NL;"lancers"
540 PRINT #1,"VMAX=";ROUND(G,4)
550 IF INKEY$="" THEN 550
560 INK 1,24
565 INK 0,1
570 PEN 1
580 PAPER 0
590 END
```

VARIABLES

R variable d'initialisation
 I,J compteurs
 N nombre de dés
 S somme des dés pour un lancer
 G valeur maximale du tableau N
 V tableau des sommes
 N tableau des pourcentages d'apparition des sommes
 NL nombre de lancers

COMMENTAIRES

90 : initialisation de la séquence de nombres aléatoires par l'instruction RANDOMIZE(TIME).
 170-300 : boucle de calcul et affichage des sommes successives obtenues. La boucle s'arrête au bout de 600 lancers.
 180 : possibilité de sortie avant la fin des 600 lancers par appui sur une touche quelconque.
 230 : calcul de la somme des faces sorties. V(S) contient le nombre de fois où la somme S est sortie.
 270 : N(J) contient le pourcentage d'apparition de la somme J, parmi le nombre total I de lancers.
 410-480 : affichage de l'histogramme.
 450 : affichage des graduations horizontales.
 500 : affichage de la somme minimale possible avec N dés, c'est-à-dire N.

EXERCICE 4.8

PROGRAMME

```
10 REM *****
15 REM * En mode 1 on a 4 stylos *
20 REM * et 4 types de papier. *
30 REM *****
40 REM * Les stylos s'appellent par*
50 REM * PEN 0, PEN 1, PEN 2, PEN 3*
60 REM *****
```

EXERCICES EN BASIC POUR AMSTRAD

```
70 REM      * Les papiers s'obtiennent *
80 REM      * par PAPER 0      PAPER 1 *
90 REM      * PAPER 2      et      PAPER 3 *
100 REM     *****
110 REM     * Pour obtenir la couleur *
120 REM     * rose pour l'encre du stylo*
130 REM     * 3 il faut executer *
140 REM     * l'instruction INK 3,16 *
150 REM     * Mais on change aussi en *
160 REM     * rose le papier numero 3. *
170 REM     *****
180 SYMBOL AFTER 145
190 SYMBOL 146,&FF,&FF,&FF,&FF,&FD,&FD,&FD,&F
D
200 SYMBOL 147,&FD,&FD,&FD,&FD,&FD,&FD,&FD,0
210 SYMBOL 148,&FC,&FC,&FC,&FC,&FC,&FC,&FC,&F
C
220 SYMBOL 149,&FC,&FC,&FC,&FC,&FC,&FC,&FC,&7
8
230 SYMBOL 150,&78,&FC,&FC,&78,&78,&30,&30,0
240 CLS
250 FOR I=1 TO 3
260 PEN I
270 FOR J=1 TO 5
280 LOCATE 17+3*I,J:PRINT CHR$(145+J)
290 NEXT J
300 LOCATE 17+3*I,5:PRINT CHR$(150)
310 NEXT I
320 REM
330 REM Changeons la couleur de l'encre
340 REM *****
350 INK 0,0:PAPER 0
360 INK 1,24
370 PEN 1
380 LOCATE 1,15
390 FOR I=1 TO 3
400 PEN I
410 PRINT "CODE COULEUR DE L'ENCRE DU STYLO";
I;:INPUT " ":N
420 INK I,N
430 NEXT I
440 REM
```

```

450 PRINT:PRINT "Pour arreter taper une touch
e"
460 IF INKEY$="" THEN 460
470 FEN 1:LOCATE 37,15:PRINT 24
480 INK 0,1:PAPER 0
490 INK 1,24:FEN 1
500 LOCATE 1,20:END

```

VARIABLES

I,J compteurs
N code de couleur (0-26)

COMMENTAIRES

190-230 : définition des 5 caractères graphiques, utilisés pour des-
siner un stylo.
390-430 : on demande successivement les trois codes de couleur.

EXERCICE 4.9

PROGRAMME

```

10 REM
20 REM SECTEUR TOURNANT
30 REM *****
40 ON BREAK.GOSUB 390
50 CLS
60 INPUT "Niveau de definition (1-10) ",NV
70 NV=INT(NV)
80 IF NV<1 OR NV>20 THEN 60
90 DIM XVAL(20*NV),YVAL(20*NV)
100 APAS=PI/(NV*10)
110 REM
120 REM Calcul des points du cercle
130 REM *****
140 MODE 0
150 LOCATE 10,15:PRINT "PATIENCE !!"
160 FOR I=0 TO 20*NV-1

```

EXERCICES EN BASIC POUR AMSTRAD

```
170 XVAL(I)=100*COS(APAS*I)
180 YVAL(I)=100*SIN(APAS*I)
190 NEXT I
200 REM
210 REM Valeurs initiales
220 REM *****
230 MODE 1
240 INK 0,1:INK 1,26:INK 2,6:INK 3,13
250 ORIGIN 300,200
260 PAPER 3
270 CLS
280 REM
290 REM Secteur tournant
300 REM *****
310 FOR J=0 TO 20*NV-1
320 XN=XVAL(J):YN=YVAL(J)
330 JD=J-NV:IF JD<0 THEN JD=JD+20*NV
340 XO=XVAL(JD):YO=YVAL(JD)
350 PLOT 0,0,1:DRAW XN,YN
360 PLOT 0,0,2:DRAW XO,YO
370 NEXT J
380 GOTO 310
390 INK 0,1:INK 1,24
400 PAPER 0:PEN 1
410 END
```

VARIABLES

NV	niveau de définition
I,J	compteurs
XVAL,YVAL	tableau des valeurs des points sur le cercle
APAS	angle entre 2 rayons voisins
XN,YN	extrémité du prochain rayon
XO,YO	extrémité du rayon à effacer
JO	rang de XO,YO dans XVAL,YVAL

COMMENTAIRES

100 : APAS représente l'angle entre 2 rayons. Si NV augmente APAS diminue et la définition devient plus grande. Toutefois cela ralentit l'exécution du programme et donc aussi la vitesse de rotation apparente du secteur sur le disque.

- 120-190 : calcul préalable des coordonnées des extrémités des rayons sur le cercle.
 330 : si JO est négatif, il faut effacer un rayon du secteur qui se trouve en fait dans le tableau à l'indice (JO + 20NV), soit 1 tour complet en plus.
 350 : on allume le rayon d'extrémité (XN,YN) en blanc.
 360 : on allume le rayon d'extrémité (XO,YO) en rouge.

EXERCICE 4.10

PROGRAMME

```

10 REM
20 DEFINT N
30 SYMBOL 250,&18,&3C,&3C,&7E,&7E,&7E,&FF,&FF
40 SYMBOL 251,&FF,&FF,&FF,&7E,&7E,&3C,&3C,&18
50 CLS
60 REM Entree de l'age
70 REM *****
80 INPUT "Quel est votre age (inferieur a 20)
";N
90 IF N<1 OR N>20 THEN 80
100 REM Initialisation
110 REM *****
120 PX=5:PY=5
130 REM
140 REM Boucle d'affichage
150 REM *****
160 FOR K=1 TO N
170 GOSUB 250
180 PX=PX+3
190 IF PX>33 THEN PX=5:PY=PY+8
200 NEXT K
210 REM
220 LOCATE 1,23
230 PEN 1
240 END
250 REM
260 REM Dessin d'une bougie dont la
    
```

```
270 REM flamme clignote en position
280 REM          PX,PY
290 REM *****
300 SPEED INK 2,2
310 INK 2,24,15
320 PEN 2
330 LOCATE PX,PY:PRINT CHR$(250)
340 LOCATE PX,PY+1:PRINT CHR$(251)
350 INK 3,26
360 PEN 3
370 FOR I=1 TO 4
380 LOCATE PX,PY+1+I:PRINT CHR$(143)
390 NEXT I
400 RETURN
```

VARIABLES

N âge ou nombre de bougies
I,K compteurs
PX,PY position des bougies

COMMENTAIRES

190 : si PX est supérieur à 33, 10 bougies ont été dessinées sur
 : une même ligne et on passe à la ligne suivante.
300 : l'instruction SPEED INK permet de donner la durée de cli-
 : gnotement des 2 couleurs définies par l'instruction INK
 : (ligne 310). Cette durée est calculée en cinquantièmes de
 : seconde.
310 : l'instruction INK permet ici de définir 2 couleurs qui cli-
 : gnotent (jaune/orange).

EXERCICE 4.11

PROGRAMME

```
10 REM
20 REM TRACE D'UNE GRILLE
30 REM *****
40 ORIGIN 0,0
```

```

50 INK 0,1:INK 1,24:INK 2,0:INK 3,8
60 WINDOW #1,29,40,1,25
70 PAPER #1,2
80 CLS
90 INPUT "LARGEUR DE LA MAILLE : ",ES
100 IF ES<1 OR ES>399 THEN 90
110 CLS:CLS #1
120 N=INT(399/ES)
130 REM
140 REM Droites horizontales
150 REM *****
160 FOR I=0 TO N
170 PLOT 0,I*ES,3:DRAW N*ES,I*ES
180 NEXT I
190 REM
200 REM Droites verticales
210 REM *****
220 FOR I=0 TO N
230 PLOT I*ES,0,3:DRAW I*ES,N*ES
240 NEXT I
250 REM
260 REM Retour
270 REM *****
280 PRINT #1,"Pour arreter";:PRINT #1," tap
ez":PRINT #1," une touche"
290 IF INKEY$="" THEN 290
300 PAPER 0:FEN 1
310 END

```

VARIABLES

I compteur
N nombre de mailles par ligne et par colonne
ES largeur de la maille

COMMENTAIRES

60 : définition d'une fenêtre d'affichage.
140-180 : tracés des lignes horizontales.
200-240 : tracés des lignes verticales.

EXERCICE 4.12

PROGRAMME

```

10 REM
20 REM
30 REM TRACE DE FONCTION
40 REM *****
50 MODE 2
60 INK 0,13
70 INK 1,26
80 CLG 0
90 PRINT TAB(32)"GRAPHE D'UNE SINUSOIDE"
100 ORIGIN 30,200,0,639,0,380
110 REM
120 REM Valeurs initiales
130 REM *****
140 UNX=60:UNY=150
150 XMIN=0:XMAX=3*PI:PAS=0.05
160 REM
170 REM Trace des axes
180 REM *****
190 PLOT -15,0,1:DRAW 580,0
200 PLOT 0,-180,1:DRAW 0,180
210 PLOT -3,UNY:DRAW 3,UNY
220 PLOT -3,-UNY:DRAW 3,-UNY
230 PLOT 0,0
240 REM
250 REM Trace de la fonction
260 REM *****
270 FOR I=XMIN TO XMAX STEP PAS
280 ABX=UNX*I:ABY=UNY*SIN(I)
290 DRAW ABX,ABY
300 NEXT I
310 REM
320 REM Trace des graduations
330 REM de l'axe horizontal
340 REM *****
350 TAG
360 FOR E=0 TO FIX(XMAX)
370 PLOT UNX*E,-5:DRAW UNX*E,5

```

```

380 MOVE E*UNX-24, -8:PRINT E;
390 NEXT E
400 REM
410 REM Retour au mode 1
420 REM *****
430 IF INKEY$="" THEN 430
440 MODE 1
450 INK 0, 1:INK 1, 24
460 PAPER 0:PEN 1
470 END

```

VARIABLES

XMIN,XMAX	plage de variation de X
UNX,UNY	nombre de points graphiques pour l'unité
X,Y	coordonnées du point à afficher
E,I	compteurs
PAS	pas de quantification
ABX,ABY	coordonnées du point de la courbe à afficher

COMMENTAIRES

80 : CLG 0 efface l'écran dans la couleur graphique 0, de code couleur 13 (gris). Cette instruction est importante car si la dernière utilisation de CLG était CLG 1, alors à la ligne 380, on obtiendrait des caractères en blanc sur fond blanc lorsqu'on utilise PEN 1.

140 : la quantité 1 s'affiche sur 60 points graphiques en horizontal et sur 150 points graphiques en vertical.

210-220 : tracé des graduations sur l'axe vertical.

350 : l'instruction TAG permet de positionner le curseur d'affichage des caractères par une instruction MOVE.

360-390 : tracé des graduations sur l'axe horizontal.

370 : dessin de la graduation.

380 : faible décalage du curseur graphique par rapport à la graduation de façon à inscrire la valeur de cette graduation.

EXERCICE 4.13

PROGRAMME

```

10 REM
20 REM CRIBLE D'ERATOSTHENE
30 REM *****
40 SYMBOL 240,&81,&42,&24,&18,&18,&24,&42,&81
50 REM
60 REM Trace de la grille
70 REM *****
75 CLS
80 INPUT "Entrez un nombre (1-9) : ",C
90 IF C<1 OR C>9 OR C<>INT(C) THEN 80
100 CLS:PRINT
110 FOR I=1 TO C
120 FOR J=1 TO C
130 N=N+1
140 PRINT USING "##  ";N;
150 NEXT J
160 PRINT:PRINT
170 NEXT I
180 REM
190 REM Recherche des nombres premiers
200 REM *****
210 FOR N=1 TO C^2
220 FOR D=2 TO C
230 IF (N MOD D)<>0 OR D=N THEN 290
240 COL=N MOD C:LIG=INT(N/C)
250 IF COL=0 THEN COL=C:LIG=LIG-1
260 IF N>9 THEN M$=CHR$(240)+CHR$(240) ELSE M
    $=CHR$(32)+CHR$(240)
270 LOCATE 4*COL-3,2*LIG+2:PRINT M$
280 GOTO 300
290 NEXT D
300 NEXT N
310 LOCATE 1,22:END

```



VARIABLES

I,J compteurs
 C nombre de lignes et de colonnes du carré
 N nombre à tester
 D diviseur
 COL colonne d'affichage
 LIG ligne d'affichage
 M\$ caractère d'effacement

COMMENTAIRES

40 : définition de la croix servant à effacer.
 110-170 : affichage du carré de nombres.
 210-300 : test des nombres.
 260 : le caractère d'effacement contient 1 ou 2 croix selon que le nombre à effacer est inférieur ou supérieur à 10.

EXERCICE 4.14

PROGRAMME

```

10 REM
20 REM ETUDE MUSICALE
30 REM *****
40 CHAINE$="ABBC"
50 NBO(1)=4:NBO(2)=4:NBO(4)=4
60 ENV 1,6,2,2,1,0,80,7,-2,2
70 REM
80 REM Parametre de canal
90 REM *****
100 CLS
110 INPUT "Choix du canal (A, B ou C) : ",CAN
    $
120 IF CAN$<>"A" AND CAN$<>"B" AND CAN$<>"C"
    THEN 110
130 CAN=INSTR(CHAINE$,CAN$)
140 IF NBO(CAN)=0 THEN PRINT:PRINT "La queue
    "CAN$" est saturée":PRINT:GOTO 110
150 ON CAN GOSUB 530,620,620,710
    
```

EXERCICES EN BASIC POUR AMSTRAD

```
160 PAR1=64+RV+CAN
170 REM
180 REM Entree de la periode
190 REM *****
200 INPUT "Choix d'une periode (16 a 3822) :
",NOTE
210 IF NOTE<17 OR NOTE>3821 THEN 200
220 REM Enregistrement
230 REM *****
240 SOUND PAR1,NOTE,100,14,1
250 REM
260 REM Analyse de l'attente
270 REM *****
280 GOSUB 800
290 REM
300 REM Choix : continuer/ecouter
310 REM *****
320 PRINT:PRINT
330 PRINT:PRINT "Pour continuer, tapez ENTER"
340 PRINT:PRINT "Pour ecouter, tapez une autre
touche"
350 A$=INKEY$:IF A$="" THEN 350
360 IF A$=CHR$(13) THEN 70
370 REM
380 REM Audition
390 REM *****
400 PRINT:PRINT
410 PRINT "Queue A","Queue B","Queue C"
420 PRINT
430 WHILE SQ(1)<>4 OR SQ(2)<>4 OR SQ(4)<>4
440 IF SQ(1)=4 AND (MID$(BIN$(SQ(2)),8),5,1)="
1" OR MID$(BIN$(SQ(4)),8),5,1)="1") THEN END
450 IF SQ(2)=4 AND (MID$(BIN$(SQ(1)),8),4,1)="
1" OR MID$(BIN$(SQ(4)),8),4,1)="1") THEN END
460 IF SQ(4)=4 AND (MID$(BIN$(SQ(1)),8),3,1)="
1" OR MID$(BIN$(SQ(2)),8),3,1)="1") THEN END
470 PRINT BIN$(SQ(1),8),BIN$(SQ(2),8),BIN$(SQ
(4),8)
480 RELEASE 7
490 WEND
500 END
510 END
```

```

520 REM *****SOUS-PROGRAMMES*****
530 REM
540 REM Canal A
550 REM *****
560 RV=0
570 INPUT "Rendez-vous avec B (O/N) ",REP$
580 IF REP$="O" THEN RV=16
590 INPUT "Rendez-vous avec C (O/N) ",REP$
600 IF REP$="O" THEN RV=RV+32
610 RETURN
620 REM
630 REM Canal B
640 REM *****
650 RV=0
660 INPUT "Rendez-vous avec A (O/N) ",REP$
670 IF REP$="O" THEN RV=8
680 INPUT "Rendez-vous avec C (O/N) ",REP$
690 IF REP$="O" THEN RV=RV+32
700 RETURN
710 REM
720 REM Canal C
730 REM *****
740 RV=0
750 INPUT "Rendez-vous avec A (O/N) ",REP$
760 IF REP$="O" THEN RV=8
770 INPUT "Rendez-vous avec B (O/N) ",REP$
780 IF REP$="O" THEN RV=RV+32
790 RETURN
800 REM
810 REM Sous-programme d'etude
820 REM des files d'attente
830 REM *****
840 PEN 2
850 NBQ(1)=SQ(1) MOD 8
860 NBQ(2)=SQ(2) MOD 8
870 NBQ(4)=SQ(4) MOD 8
880 PRINT
890 PRINT "File d'attente A : "BIN$(SQ(1),8)
900 PRINT TAB(10)"Places restantes : "NBQ(1)
910 PRINT
920 PRINT "File d'attente B : "BIN$(SQ(2),8)
930 PRINT TAB(10)"Places restantes : "NBQ(2)

```

```
940 PRINT
950 PRINT "File d'attente C : "BIN$(SQ(4),8)
960 PRINT TAB(10)"Places restantes : "NBQ(4)
970 PEN 1
980 RETURN
```

VARIABLES

CHAINES\$ chaîne de caractères contenant les réponses possibles
CAN\$ canal choisi
CAN numéro du canal choisi (1, 2 ou 4)
A\$ touche frappée au clavier
PAR1 premier paramètre de l'instruction SOUND
RV valeur précisant le type de Rendez-vous entre canaux
NOTE période de la note à jouer
REP\$ réponse
NBQ tableau des places libres dans les 3 files A, B et C

COMMENTAIRES

40 : la chaîne "ABBC" contient les noms des 3 canaux. La lettre B est répétée deux fois : ainsi par utilisation de l'instruction INSTR, on peut déterminer les 3 valeurs numériques 1, 2 et 4 associées aux 3 canaux.

50 : au début toutes les files d'attente sont vides et NBQ(I) vaut 4.

140 : teste si la file d'attente du canal choisi est pleine.

150 : la valeur 620 est répétée deux fois, ainsi quand CAN vaut 4, provenant de la lettre C dans CHAINES\$, le programme se poursuit à la ligne 710 (on notera que CAN ne peut jamais prendre la valeur 3).

160 : calcul du premier paramètre de l'instruction SOUND. L'addition du nombre 64 permet de maintenir la note sans la jouer (mode HOLD).

380-490 : audition.

430-490 : boucle WHILE/WEND : tant qu'il y a des notes à jouer, on exécute l'instruction RELEASE (ligne 480) qui libère successivement les différents canaux.

440-460 : testent si les Rendez-vous sont encore possibles.

470 : affichage du mot SQ en binaire sur 8 bits.

520-790 : sous-programmes calculant la valeur de RV.

810-980 : étude des files d'attente.

850-870 : seuls les trois derniers bits de SQ sont conservés : ils représentent le nombre de places libres dans la file d'attente d'un canal. Pour isoler ces trois bits, il suffit de calculer le reste de la division de SQ par 8.

EXERCICE 4.15

PROGRAMME

```

10 REM
20 REM Programme de chronometre
30 REM *****
40 REM
50 CLS
60 PRINT "Pour demarrer taper sur C"
70 PRINT "Pour arreter taper sur S"
80 PRINT "Pour continuer taper sur C"
90 PRINT "Pour reprendre taper sur L"
100 PRINT "Pour remettre a zero taper sur Z"
110 REM
120 LOCATE 1,10:PRINT "CHRONOMETRE";
130 CLEAR
140 LOCATE 18,10:PRINT "00"
150 A$=INKEY$:IF A$="" OR (A$<>"C" AND A$<>"c
") THEN 150
160 EVERY 5,1 GOSUB 240
170 A$=INKEY$:IF A$="" THEN 170
180 A#=UPPER$(A$)
190 IF A#="Z" THEN 130
200 IF A#="S" AND FL=0 THEN FL=1:SECI=SEC:GOT
O 170
210 IF A#="C" AND FL=1 THEN SEC=SECI:FL=0
220 IF A#="L" AND FL=1 THEN FL=0
230 GOTO 170
240 REM
250 REM Sous-programme d'incrementation
260 REM ***** des secondes *****
270 REM *****
280 REM

```

```
290 SEC=SEC+1
300 IF FL=1 THEN RETURN
310 IF SEC>59 THEN SEC=0
320 LOCATE 18,10
330 IF SEC<10 THEN PRINT USING "0#";SEC ELSE
PRINT USING "##";SEC
340 PRINT CHR$(7)
350 RETURN
```

VARIABLES

A\$ touche frappée au clavier
FL drapeau indiquant que la touche S a été frappée
SECI,SEC valeurs des secondes

COMMENTAIRES

180 : transforme la réponse A\$ en lettre majuscule.
190 : si A\$ égale "Z", on retourne à la ligne 130. L'instruction
 CLEAR remet les variables à 0 et arrête les interruptions.
200-220 : si la touche S a été frappée (arrêt du chronomètre), FL est
 mis à 1. Toutefois, le sous-programme d'interruption
 continue de fonctionner. Aussi le programme conserve
 dans SECI la valeur du chronomètre au moment de l'arrêt.
 Si alors la lettre C est frappée, le sous-programme d'inter-
 ruption reprend avec cette valeur. Si la lettre L est frappée,
 le sous-programme d'interruption continue avec la valeur
 courante SEC.
250-350 : sous-programme d'incrémentatation des secondes.
330 : si le nombre des secondes est inférieur à 10, l'affichage de
 la valeur est précédé d'un 0.
340 : chaque seconde passée est ponctuée d'un son.

EXERCICE 4.16

PROGRAMME

```
10 REM VALEURS DECIMALES DE
20 REM L'INSTRUCTION SYMBOL
30 REM *****
40 INK 0,0
```

```

50 REM
60 REM Symboles d'affichage
70 REM *****
80 SYMBOL 248,&FF,&81,&81,&81,&81,&81,&81,&FF
90 SYMBOL 250,&FF,&FF,&FF,&FF,&FF,&FF,&FF,&FF
100 CLS
110 PRINT TAB(11)"Valeurs decimales":PRINT TA
B(8)"de l'instruction SYMBOL"
120 PRINT
130 PEN 2:PRINT "Pour allumer ou eteindre un
pave","utiliser la touche COPY"
140 PRINT "Pour changer de case, utiliser","l
es touches de deplacement":PEN 1
150 REM
160 REM Initialisation des cases
170 REM *****
180 FOR I=1 TO 8
190 FOR J=1 TO 8
200 CASE(I,J)=-1
210 NEXT J,I
220 REM
230 REM Presentation de la grille
240 REM *****
250 LOCATE 1,10
260 FOR J=1 TO 8
270 PRINT TAB(16)STRING$(8,CHR$(248))
280 NEXT J
290 XG=16:YH=10
300 REM
310 REM Acquisition d'une touche
320 REM *****
330 LOCATE XG,YH
340 A$=INKEY$
350 IF A$="" THEN 340
360 A=ASC(A$)
370 IF A=13 THEN 450
380 IF A=240 THEN IF YH>10 THEN YH=YH-1:GOTO
330
390 IF A=241 THEN IF YH<17 THEN YH=YH+1:GOTO
330
400 IF A=242 THEN IF XG>16 THEN XG=XG-1:GOTO
330

```

```
410 IF A=243 THEN IF XG<23 THEN XG=XG+1:GOTO
330
420 IF A=224 THEN CASE(XG-15,YH-9)=-CASE(XG-
15,YH-9)
430 PRINT CHR$(249+CASE(XG-15,YH-9));
440 GOTO 330
450 REM
460 REM Determination des valeurs
470 REM *****
480 FOR I=1 TO 8
490 L(I)=0
500 FOR J=1 TO 8
510 CASE(J,I)=(1+CASE(J,I))/2
520 L(I)=L(I)+CASE(J,I)*(2^(8-J))
530 NEXT J
540 NEXT I
550 REM
560 REM Affichage de la suite
570 REM *****
580 LOCATE 1,20
590 PRINT "SYMBOL XXX";
600 FOR I=1 TO 8
610 PRINT ", "MID$(STR$(L(I)),2);
620 NEXT I
630 REM
640 REM Affichage du symbole
650 REM *****
660 SYMBOL 249,L(1),L(2),L(3),L(4),L(5),L(6),
L(7),L(8)
670 LOCATE 26,14:PRINT CHR$(249);
680 LOCATE 1,22:END
```

VARIABLES

I,J	compteurs
XG,YH	coordonnées d'une case de la grille
A\$	touche frappée au clavier
A	valeur ASCII de A\$
CASE	tableau contenant l'état d'une case (-1 : éteint, 1 : allumé)
L	tableau des valeurs décimales de l'instruction SYMBOL

COMMENTAIRES

- 80, 90 : définitions des caractères graphiques "CASE VIDE" et "CASE PLEINE".
- 160-210 : initialisation du tableau CASE à -1.
- 230-280 : dessin de la grille.
- 290 : positionnement sur la case en haut à gauche.
- 380-410 : déplacement dans les limites de la grille.
- 420 : l'appui sur la touche COPY change alternativement l'état de la case désignée. En effet l'élément CASE(I,J) vaut alternativement -1 et 1.
- 430 : affichage de l'état de la case : si CASE(I,J) vaut -1, la case est allumée (pleine) et si CASE(I,J) vaut 1, la case est éteinte (vide).
- 520 : détermination de L(I).
- 610 : suppression du caractère "ESPACE" qui précède la chaîne de caractères STR\$(L(I)).
- 670 : affichage du caractère graphique en vraie grandeur.

EXERCICE 4.17

PROGRAMME

```

10 REM
20 REM ORGUE ELECTRONIQUE
30 REM *****
40 DIM NOTE(30)
50 ON BREAK GOSUB 400
60 SPEED KEY 1,1
70 TC$="1234567QWERTYUASDFGHJZXCVBNM"
80 CLS
90 LOCATE 10,8:PRINT "ORGUE ELECTRONIQUE"
100 PRINT:PRINT "QUATRE GAMMES : "
110 LOCATE 17,12:PRINT "1234567"
120 LOCATE 17,13:PRINT "QWERTYU"
130 LOCATE 17,14:PRINT "ASDFGHJ"
140 LOCATE 17,15:PRINT "ZXCVBNM"
150 REM
160 REM Lecture des periodes

```

EXERCICES EN BASIC POUR AMSTRAD

```
170 REM *****
180 FOR I=1 TO 28
190 READ NOTE(I)
200 NEXT I
210 REM
220 REM Acquisition d'une touche
230 REM *****
240 A$=INKEY$:IF A$="" THEN 240
250 A$=UPPER$(A$)
260 I=INSTR(TC$,A$)
270 IF I=0 THEN 240
280 SOUND 1,NOTE(I),3
290 GOTO 240
300 REM
310 REM Valeurs des periodes
320 REM *****
330 DATA 956,851,758,716,638,568,506
340 DATA 478,426,379,358,319,284,253
350 DATA 239,213,190,179,159,142,127
360 DATA 119,106,95,89,80,71,63
370 REM
380 REM Retour apres BREAK
390 REM *****
400 SPEED KEY 10,3
410 END
```

VARIABLES

NOTE\$ tableau des 28 périodes des notes choisies
TC\$ chaîne de caractères contenant les lettres des touches choisies
I indice d'une note
A\$ touche frappée

COMMENTAIRES

50 : le programme est interrompu par 2 appuis sur la touche BREAK et se poursuit en ligne 400.
70 : TC\$ est constituée des 28 lettres des touches situées à gauche dans les 4 rangées du clavier.
60 : la vitesse de répétition des touches est mise à sa valeur maximale (1 cinquantième de seconde).

- 260 : l représente le rang de la lettre de la touche frappée dans TC\$.
- 400 : il est important avant de sortir du programme de remettre la vitesse de répétition du clavier à une valeur plus pratique (essayer par curiosité d'utiliser le clavier, après avoir exécuter l'instruction de la ligne 60).

EXERCICE 4.18

PROGRAMME

```

10 REM
20 REM PROGRAMME D'ADDITION
30 REM *****
40 INK 0,13:INK 1,26:INK 2,6:INK 3,18,13
50 ORG=13
60 MODE 1
70 PAPER 0:PEN 1
80 LOCATE 1,10
90 PRINT "Entrer deux nombres, compris entre
0 et"
100 PRINT "1000, separees par une virgule et a
ppuyer"
110 INPUT "sur la touche ENTER : ",N(1),N(2)
120 IF N(1)<0 OR N(1)>1000 OR N(2)<0 OR N(2)>
1000 THEN 60
130 REM
140 REM Passage en MODE 0
150 REM *****
160 MODE 0
170 PRINT:PRINT "PROGRAMME D'ADDITION"
180 WINDOW #1,1,20,20,25
190 PAPER #1,2:PEN #1,1
200 CLS #1
210 PRINT #1:PRINT #1,"Pour afficher le
resultat"," d'une colonne"," taper sur l
a barre"
220 LN1=LEN(STR$(N(1)))-1
230 LN2=LEN(STR$(N(2)))-1

```

EXERCICES EN BASIC POUR AMSTRAD

```
240 LNMAX=MAX(LN1, LN2)
250 POSI1=ORG-LN1
260 POSI2=ORG-LN2
270 POSI3=MIN(POSI1, POSI2)-2
280 TRAIT$=STRING$(MAX(LN1, LN2)+3, "_")
290 LOCATE POSI1, 10:PRINT N(1)
300 LOCATE POSI3, 12:PRINT "+"
310 LOCATE POSI2, 12:PRINT N(2)
320 LOCATE POSI3, 13:PRINT TRAIT$
330 LOCATE POSI3, 15:PRINT "="
340 REM
350 REM Determination des chiffres
360 REM qui constituent N1 et N2
370 REM *****
380 FOR J=1 TO 2
390 FOR I=1 TO LNMAX
400 C(J, I)=N(J) MOD 10
410 N(J)=(N(J)-C(J, I))/10
420 NEXT I
430 NEXT J
440 REM Addition en colonnes
450 REM *****
460 RETENUE=0
470 FOR I=1 TO LNMAX+1
480 A$=INKEY$
490 IF A$<>CHR$(32) THEN 480
500 AUX=C(1, I)+C(2, I)+RETENUE
510 RES=AUX MOD 10
520 RETENUE=(AUX-RES)/10
530 LOCATE ORG-I, 15:PRINT STR$(RES);
540 IF RETENUE<>0 THEN LOCATE ORG-1-I, 8:PEN 3
:PRINT STR$(RETENUE):PEN 1
550 IF I=LNMAX AND RETENUE=0 THEN 570
560 NEXT I
570 LOCATE 1, 20
580 CLS #1
590 PRINT #1
600 PRINT #1, "ENTER pour arreter"
610 PRINT #1
620 PRINT #1, "C pour continuer"
630 A$=INKEY$
640 IF A$="" THEN 630
```

```

650 IF A$="C" THEN 60
660 IF A$<>CHR$(13) THEN 630
670 INK 0,1:INK 1,24
680 MODE 1
690 PAPER 0:FEN 1
700 END

```

VARIABLES

ORG variable
 N(1),N(2) nombres à additionner
 LN1,LN2 longueurs de N(1) et N(2)
 LNMAX maximum entre LN1 et LN2
 POSI1 position de N(1)
 POSI2 position de N(2)
 POSI3 position des signes opératoires (+, =, trait)
 TRAIT\$ chaîne de caractères utilisée pour tirer un trait
 I,J compteurs
 C(J,I) tableau contenant le Jième chiffre du lième nombre
 RETENUE retenue sur une colonne
 RES,AUX variables
 A\$ touche frappée au clavier

COMMENTAIRES

40 : l'encre 3 clignote (vert/gris).
 220, 230 : calcul du nombre de chiffres de N(1) et N(2). On remarque
 que STR\$ fait précéder la chaîne d'un "espace".
 380-430 : calcul des éléments de C(J,I).
 400, 410 : l'opération $N(J) \text{ MOD } 10$ calcule le reste de la division de
 N(J) par 10. On réitère l'opération par décalage sur le chiffre
 des dizaines puis celui des centaines.
 460 : initialisation de la retenue pour les deux premiers chiffres
 à additionner.
 470 : le compteur va de 1 à LNMAX+1 car le résultat de l'addi-
 tion peut comporter un chiffre de plus que le plus grand
 des deux nombres à additionner.
 550 : si l'addition dans la dernière colonne est effectuée et que
 la retenue est nulle, on sort de la boucle.

EXERCICE 5.1

PROGRAMME

```

10 REM
20 REM TIR ALEATOIRE SUR
30 REM UNE CIBLE GRADUEE
40 REM *****
50 CLS
60 REM Valeurs initiales
70 REM *****
80 COUP=-1
90 INK 0,13:INK 1,25:INK 2,8:INK 3,0
100 WINDOW #0,1,40,4,24
110 PAPER 3
120 CLS
130 REM
140 REM Dessins des cases
150 REM **de la cible**
160 REM *****
170 FOR I=1 TO 6
180 WINDOW #I,6*I-3,6*I+2,1,3
190 B=1-B:C=1+B:PAPER #I,C:CLS #I
200 PEN #I,3:LOCATE #I,3,2:PRINT #I,I;
210 NEXT I
220 WINDOW #7,1,40,25,25
230 PAPER #7,0
240 PEN #7,3
250 REM
260 REM Entree d'une touche
270 REM *****
280 CLS
290 COUP=COUP+1
300 IF COUP=20 THEN 470
310 PRINT #7,"Jeux :";COUP;"***** Score :";SC
ORE
320 A$=INKEY$:IF A$="" THEN 320
330 IF A$=CHR$(13) THEN 510
340 IF A$<>CHR$(32) THEN 320
350 X=INT(640*RND):PLOT X,26,1
360 FOR J=0 TO 31:DX=INT(90*RND)-44:X=X+DX:PL
OTR 0,0,3:DRAWR 0,-10:PLOTR DX,10,1:DRAWR 0,1
0:NEXT J

```

```

370 PRINT CHR$(7)
380 REM
390 REM Calcul du nombre de points
400 REM *****
410 IF X<32 OR X>608 THEN POIN=0 ELSE POIN=INT((X-32)/96)+1
420 SCORE=SCORE+POIN
430 GOTO 250
440 REM
450 REM Affichage du resultat
460 REM *****
470 WINDOW #7,1,40,24,25
480 CLS #7
490 PRINT #7,"**Nombre de coups :";COUP,"**
Nombre de points : ";SCORE
500 IF INKEY$="" THEN 500
510 REM
520 REM Valeurs de retour
530 REM *****
540 INK 0,1:INK 1,24
550 WINDOW 1,40,1,25
560 PAPER 0:PEN 1
570 CLS
580 END

```

VARIABLES

I	compteur
B,C	variables
POIN	valeur de la case touchée
SCORE	score
COUP	nombre de coups joués
X	position horizontale du projectile
DX	écart horizontal
A\$	touche frappée au clavier

COMMENTAIRES

140-210 : affichage de la cible. CLS#I efface la fenêtre de rang I dans la couleur C de PAPER #I. C prend alternativement les valeurs 1 et 2.

- 310 : affichage du score et du nombre de coups joués dans la fenêtre 7.
- 350 : affichage de la position de départ du projectile par l'instruction PLOT.
- 360 : l'effet de mouvement est obtenu par effacement et affichage successifs du projectile. Ainsi PLOTR 0,0,3 et DRAWR 0,-10 effacent la dernière position du projectile (couleur noire) et PLOTR DX, 10,1 et DRAWR 0,10 dessinent le projectile dans sa nouvelle position.
- 410 : calcul du nombre de points marqués. POIN est déterminé en fonction de la position horizontale du projectile au moment de son impact avec la cible. Si X est inférieur à 32 ou supérieur à 608, le projectile sort de la cible, sinon $INT((X-32)/96)+1$ donne la valeur de POIN.

EXERCICE 5.2

PROGRAMME

```
10 REM
20 REM JEU DU GLOUTON
30 REM *****
40 REM
50 REM Valeurs initiales
60 REM *****
70 ON BREAK GOSUB 580
80 NIV=1
90 SPEED KEY 1,1
100 SYMBOL 250,&18,&3E,&7F,&FF,&FF,&7F,&3E,&18
110 SYMBOL 251,&1F,&3E,&7C,&FO,&FO,&7C,&3E,&1F
120 INK 0,13:INK 1,25:INK 2,0: INK 3,8
130 WINDOW #1,1,40,1,4
140 WINDOW 1,40,5,21
150 WINDOW #2,1,40,22,25
160 PAPER 2:PAPER #1,0:PAPER #2,0
170 PEN 3:PEN #1,1:PEN #2,1
180 CLS #1:CLS #2
190 REM
200 REM Affichage de la piste
210 REM *****
```

```

220 LOCATE #1,17,2:PRINT #1,"GLOUTON"
230 PRINT #1,"Z vers le haut"SFC(13)"\ vers le b
as";
240 LOCATE #2,15,2:PRINT #2,"SCORE : ";SCORE
250 PEN 3:CLS
260 NIV=NIV-1
270 IF NIV<1 THEN NIV=1
280 IF NIV>5 THEN NIV=5
290 FOR I=1 TO 15+5*NIV
300 X=INT(30*RND)+7:Y=INT(17*RND)+1
310 LOCATE X,Y:PRINT CHR$(143);
320 NEXT I
330 IF INKEY$="" THEN 330
340 REM
350 REM Debut du jeu
360 REM *****
370 PEN 1
380 X=1:Y=INT(6*RND)+10
390 LOCATE X,Y:PRINT CHR$(250);
400 DY=0
410 A$=UPPER$(INKEY$):IF A$="" THEN 440
420 IF A$="\ " AND Y<17 THEN Y=Y+1:DY=1:GOTO 440
430 IF A$="Z" AND Y>1 THEN Y=Y-1:DY=-1
440 X=X+1
450 REM
460 REM Test de collision
470 REM *****
480 YG=391-(Y+3)*16:XG=16*X-8
490 IF TEST(XG,YG)=3 THEN SOUND 1,120,50,13:GOTO
240
500 LOCATE X-1,Y-DY:PRINT " ";
510 LOCATE X,Y:C=1-C:PRINT CHR$(250+C);
520 IF C=0 THEN PRINT CHR$(7)
530 IF X<40 GOTO 400
540 SCORE=SCORE+NIV
550 NIV=NIV+2
560 GOTO 240
570 END
580 REM
590 REM Retour
600 REM *****
610 SPEED KEY 10,3

```

```
620 INK 0,1:INK 1,24
630 WINDOW 1,40,1,25
640 PAPER 0:PEN 1
650 END
```

VARIABLES

I	compteur
A\$	touche frappée au clavier
NIV	niveau de difficulté
X,Y	coordonnées d'affichage
DY	déplacement vertical relatif du glouton
NC	nombre de coups joués
SCORE	score
C	variable
XG,YG	coordonnées du point graphique

COMMENTAIRES

70 : l'instruction ON BREAK permet de continuer le programme à la ligne 580, après deux appuis successifs sur la touche BREAK.

100-110 : le caractère 250 représente le glouton la bouche fermée, et le caractère 251 représente le glouton la bouche ouverte.

270 : le niveau de difficulté est défini par la variable NIV qui prend ses valeurs entre 1 et 5.

290-320 : positionnement des obstacles dans la zone de jeu en nombre proportionnel à NIV.

400-430 : analyse de la touche frappée au clavier. Si la chaîne est vide DY=0, si la chaîne est Z DY=-1 et si la chaîne est \ DY=1.

480 : calcul des coordonnées en mode graphique du point situé en avant du glouton.

490 : si le point de coordonnées XG et YG est de couleur magenta, cela signifie que le glouton touche un obstacle.

500-510 : déplacement du glouton. L'effet de mouvement est obtenu en affichant en position (X-1) un espace (glouton éteint) et en affichant en position X alternativement les caractères graphiques 250 ou 251.

550 : le niveau de difficulté est augmenté à chaque traversée réussie et diminué en cas d'échec.

590-650 : Il est important de rétablir les valeurs 10 et 3 dans l'instruction SPEED KEY, avant de sortir du programme (Essayez d'utiliser le clavier avec les valeurs 1 et 1 !!).

EXERCICE 5.3

PROGRAMME

```

10 REM
20 REM LOTO
30 REM ****
40 REM Valeurs initiales
50 REM *****
60 SYMBOL 250,&B1,&42,&24,&18,&18,&24,&42,&B1
65 CLS
70 INK 0,24:INK 1,6:INK 2,13:INK 3,19
80 WINDOW #1,25,40,1,25
90 PAPER 2:PAPER #1,2:PAPER #2,0
100 WINDOW #2,5,19,3,23
110 PEN #1,3:PEN #2,1
120 CLS:CLS #1
130 LOCATE #1,1,10
140 PRINT #1,"Pour obtenir un"
150 PRINT #1,"autre tirage"
160 PRINT #1,"taper une touche"
170 PRINT #1:PRINT #1,"Pour arreter"
180 PRINT #1,"taper ENTER"
190 REM
200 REM Trace de la grille
210 REM *****
220 CLS #2
230 PEN #2,2:PRINT #2,CHR$(143)+CHR$(143):PEN #2
,1
240 FOR J=0 TO 9
250 FOR I=0 TO 4
260 IF I=0 AND J=0 THEN PEN #2,2:PRINT #2,CHR$(1
43)+CHR$(143);:PEN #2,1:GOTO 280
270 PRINT #2,STR$(10*I+J);

```

```
280 NEXT I
290 PRINT #2:PRINT #2
300 NEXT J
310 REM
320 REM Choix de 6 nombres aleatoires
330 REM *****
340 NU(1)=INT(49*RND)+1
350 FOR I=2 TO 6
360 NU(I)=INT(49*RND)+1
370 FOR J=1 TO I-1
380 IF NU(I)=NU(J) THEN 350
390 NEXT J
400 NEXT I
410 REM
420 REM Affichage du tirage
430 REM *****
440 FOR I=1 TO 6
450 UNI=NU(I) MOD 10:DIZ=(NU(I)-UNI)/10
460 IF DIZ=0 THEN SUP$=" "+CHR$(250) ELSE SUP$=C
HR$(250)+CHR$(250)
470 LOCATE #2,3*DIZ+1,2*(UNI+1):PRINT #2,SUP$;
480 NEXT I
490 A$=INKEY$:IF A$="" THEN 490
500 IF A$<>CHR$(13) THEN 200
510 REM
520 REM Retour
530 REM *****
540 INK 0,1:INK 1,24
550 PAPER 0:PEN 1
560 END
```

VARIABLES

I,J	compteurs
PT	pointeur de position d'affichage
NU	tableau des tirages dimensionné implicitement
DIZ	chiffre des dizaines
UNI	chiffre des unités
SUP\$	chaîne de caractères, utilisée pour cocher
A\$	touche frappée

COMMENTAIRES

- 200-300 : affichage de la grille du loto.
 260 : le coin en haut et à gauche est omis.
 320-400 : tirage des 6 numéros.
 370-390 : comparaison du numéro tiré avec les précédents déjà tirés.
 420-480 : les numéros tirés sont cochés sur la grille avec une ou deux croix suivant qu'ils sont compris entre 1 et 9 ou entre 10 et 49.
 450 : le chiffre des unités de NU(I) est obtenu par le reste de la division de NU(I) par 10 ; c'est le calcul qui est effectué par l'instruction MOD. Le chiffre des unités est ensuite utilisé pour calculer celui des dizaines.
 470 : la position horizontale sur la grille du numéro tiré est calculée à partir de DIZ. Sa position verticale est calculée à partir de UNI.

EXERCICE 5.4**PROGRAMME**

```

10 REM
20 REM CALCUL MENTAL
30 REM *****
40 CLS
50 EI:SPEED KEY 100,100
60 SIGNOP$="+-*"
70 MESS1$="Vous calculez vite et bien"
80 MESS2$="Vous calculez vite mais mal"
90 MESS3$="Vous calculez trop lentement"
100 MESS4$="Le resultat est : "
110 MESS5$=CHR$(10)+SPACE$(12)+"CALCUL MENTAL
"
120 MESS6$="Pour arreter, tapez sur ENTER"
130 MESS7$="Pour continuer tapez sur C"
140 REM
150 REM Valeurs initiales
160 REM *****
170 INK 0,0:INK 1,25:INK 2,13:INK 3,26
180 WINDOW #1,1,40,1,15

```

EXERCICES EN BASIC POUR AMSTRAD

```
190 WINDOW #2,1,40,16,25
200 PAPER #1,0:PEN #1,1
210 PAPER #2,2:PEN #2,3
220 CLS #1:CLS #2
230 PRINT #1,MESS5$
240 BORDER 4
250 REM
260 REM Tirage aleatoire de l'operation
270 REM *****
280 A1=INT(1000*RND)+1
290 A2=INT(1000*RND)+1
300 OPE=INT(3*RND)+1
310 OPE$=MID$(SIGNOF$,OPE,1)
320 ON OPE GOSUB 560,610,660
330 LOCATE #1,10,8:PRINT #1,A1;OPE$;A2;" = ";
340 FL=0
350 AFTER 500 GOSUB 710
360 GOSUB 760 'Saisie de la reponse
370 IF REP$="-" OR REP$="+" THEN REPONSE=0 ELSE REPONSE=VAL(REP$)
380 IF FL=1 THEN MA$=MESS3$:GOTO 410
390 IF REPONSE=RES THEN PRINT #2,MESS1$:GOTO 420
400 MA$=MESS2$
410 PRINT #2,MA$:PRINT #2,MESS4$;RES
420 PRINT #2:PRINT #2
430 PRINT #2,MESS6$:PRINT #2,MESS7$
440 A$=INKEY$:IF A$="" THEN 440
450 IF A$=CHR$(13) THEN 480
460 IF A$="C" OR A$="c" THEN 220
470 GOTO 440
480 REM
490 REM Retour
500 REM *****
510 INK 0,1:INK 1,24
520 PAPER 0:PEN 1:BORDER 1
530 CLS
540 SPEED KEY 10,3
550 END
560 REM
570 REM Addition
580 REM *****
```

```

590 RES=A1+A2
600 RETURN
610 REM
620 REM Soustraction
630 REM *****
640 RES=A1-A2
650 RETURN
660 REM
670 REM Multiplication
680 REM *****
690 RES=A1*A2
700 RETURN
710 REM
720 REM Temps ecoule
730 REM *****
740 FL=1
750 RETURN
760 REM
770 REM Saisie de la reponse
780 REM *****par INKEY$*****
790 REM *****
800 REP$=""
810 IF FL=1 THEN RETURN ELSE A$=INKEY$: IF A$=
"" THEN 810
820 IF A$=CHR$(13) THEN RETURN
830 IF (A$<"0" OR A$>"9") AND A$<>"+" AND A$<
>"-" THEN 810
840 REP$=REP$+A$
850 PRINT #1,A$;
860 IF FL=1 THEN RETURN ELSE A$=INKEY$: IF A$=
"" THEN 860
870 IF A$=CHR$(13) THEN RETURN
880 IF (A$<"0" OR A$>"9") AND A$<>CHR$(127) T
HEN 860
890 IF A$=CHR$(127) AND LEN(REP$)=0 THEN 810
900 IF A$=CHR$(127) THEN REP$=LEFT$(REP$,LEN(
REP$)-1):LOCATE #1,POS(#1)-1,VPOS(#1):PRINT #
1," ";:LOCATE #1,POS(#1)-1,VPOS(#1):IF LEN(RE
P$)=0 THEN 810 ELSE 860
910 REP$=REP$+A$
920 PRINT #1,A$;
930 GOTO 860

```

VARIABLES

SIGNOP\$	variable alphanumérique contenant les 3 signes opératoires
ME1\$, ME2\$, ME3\$, ME4\$, ME5\$, ME6\$	différents messages possibles
A1, A2	opérandes
OPE	code de l'opération (1:addition, 2:soustraction, 3:multiplication)
OPE\$	contient le symbole de l'opération
RES	résultat de l'opération
REP\$, REponse	réponses du joueur
MA\$	variable
FL	drapeau indiquant que les 10 secondes sont écoulées
A\$	touche frappée

COMMENTAIRES

- 50 : activation des interruptions. SPEED KEY permet ici d'éviter les répétitions lors de l'appui sur les touches.
- 320 : suivant la valeur de OPE (1, 2 ou 3), le programme poursuit à la ligne 560 (addition), 610 (soustraction) ou 660 (multiplication).
- 340 : initialisation du drapeau. Si FL=0, le temps écoulé pour donner le résultat est inférieur à 10 secondes.
- 350 : au bout de 10 secondes (500 cinquantièmes de seconde), le sous-programme (lignes 710-750) est exécuté.
- 360 : l'apostrophe (forme abrégée de REM) est utilisée ici pour introduire un commentaire dans la ligne.
- 370 : si REP\$ se réduit à l'un des deux caractères "+ ou -", l'exécution de VAL(REP\$) provoque une erreur.
- 380-430 : la réponse d'AMSTRAD dépend du résultat de la comparaison de FL à 0 et 1, et de RES et REponse.
- 740 : le drapeau FL est mis à 1 lorsque le temps écoulé dépasse 10 secondes.
- 760-930 : saisie de REP\$.
- 810-830 : entrée du premier caractère de REP\$. Ce caractère peut être un chiffre (0,1,2,...,9) ou un caractère de signe (+, -).
- 860-900 : entrée des autres caractères de REP\$. Ce caractère peut être un chiffre ou celui de la touche DEL. Si la touche frappée est DEL, le dernier caractère entré au clavier est effacé et la chaîne REP\$ est réduite.

EXERCICE 5.5

PROGRAMME

```

10 REM
20 REM MACHINE A SOUS
30 REM *****
40 SYMBOL 250,&18,&3C,&7E,&FF,&FF,&7E,&3C,&18
50 SYMBOL 251,&7E,&3C,&3C,&FF,&FF,&FF,&18,&3C
60 SYMBOL 252,&66,&FF,&FF,&FF,&7E,&3C,&18,&18
70 REM Valeurs initiales
80 REM *****
90 MODE 0
100 INK 0,24:INK 1,2:INK 2,26:INK 3,6:INK 4,0
:INK 5,13:INK 6,18
110 BORDER 13
120 PAPER 4:CLS:PEN 0
130 WINDOW #1,7,13,5,18
140 PAPER #1,5:PEN #1,0
150 CLS #1
160 WINDOW #2,8,8,7,8
170 PAPER #2,4:CLS #2
180 WINDOW #3,10,10,7,8
190 PAPER #3,4:CLS #3
200 WINDOW #4,12,12,7,8
210 PAPER #4,4:CLS #4
220 WINDOW #5,1,40,20,25
230 PAPER #5,4:PEN #5,2
240 WINDOW #6,15,20,10,12
250 PAPER #6,4:CLS #6:PEN #6,6
260 SCORE=0
270 REM
280 REM Dessin de la machine
290 REM *****
300 PRINT:PRINT TAB(7)"JACKPOT"
310 PRINT #5,"Pour arreter":PRINT #5,"tapez E
NTER"
320 PRINT #5
330 PRINT #5,"Pour continuer":PRINT #5,"tapez
une touche"
340 LOCATE #1,3,10:PRINT #1,"$$$";

```

EXERCICES EN BASIC POUR AMSTRAD

```
350 FOR I=1 TO 3
360 FOR J=1 TO 2
370 PEN #I+1,I
380 PRINT #I+1,CHR$(249+I);
390 NEXT J
400 NEXT I
410 REM
420 REM La machine tourne
430 REM *****
440 A$=INKEY$:IF A$="" THEN 440
450 IF A$=CHR$(13) THEN 790
460 FOR I=1 TO 10
470 RANDOMIZE(TIME)
480 FOR J=1 TO 3
490 FIG(J)=INT(RND*9)
500 COL=FIG(J) MOD 3:CAR=INT(FIG(J)/3)
510 PEN #J+1,COL+1:PRINT #J+1,CHR$(250+CAR);
520 SOUND 1,223,2,15
530 NEXT J
540 NEXT I
550 FOR J=1 TO 3
560 COL=FIG(J) MOD 3:CAR=INT(FIG(J)/3)
570 PEN #J+1,COL+1:PRINT #J+1,CHR$(250+CAR);
580 NEXT J
590 REM
600 REM Determination du gain
610 REM ou des pertes
620 REM *****
630 RES=FIG(3)+10*FIG(2)+100*FIG(1)
640 IF RES=876 THEN POINT=1000:GOTO 740
650 IF RES=867 THEN POINT=500:GOTO 740
660 IF RES=687 THEN POINT=500:GOTO 740
670 IF RES=678 THEN POINT=500:GOTO 740
680 IF RES=786 THEN POINT=500:GOTO 740
690 IF RES=768 THEN POINT=500:GOTO 740
700 POINT=-20
710 REM
720 REM Affichage du score
730 REM *****
740 SCORE=SCORE+POINT
750 PRINT #6,USING "####";SCORE
760 PRINT #6
```

```

770 PRINT #6, USING "#####";POINT
780 GOTO 410
790 REM
800 REM Retour
810 REM *****
820 INK 0,1:INK 1,24
830 PAPER 0:PEN 1
840 BORDER 1
850 MODE 1
860 END

```

VARIABLES

I,J compteurs
FIG tableau des trois chiffres représentant la combinaison tirée
POINT gain d'un coup
SCORE gain d'une partie
COL couleur de la figure à dessiner
CAR numéro de la figure à dessiner
A\$ touche frappée au clavier
RES nombre représentant la combinaison tirée

COMMENTAIRES

280-400 : dessin de la machine à sous.
350-400 : dessin des trois fenêtres.
460-540 : avant l'affichage final, la machine effectue dix rotations.
550-580 : le dernier affichage est répété car les fenêtres contiennent des symboles doublés.
630 : RES se calcule par la formule :
 $FIG(3) + 10 \times FIG(2) + 100 \times FIG(1)$
640-700 : étude du gain d'un coup. Par exemple, la configuration qui donne 1000 points, est représentée par le nombre 876.
720-780 : affichage du résultat.

Remarque : pour les mauvais joueurs, il est possible de gagner plus souvent en réduisant le nombre de couleurs.

EXERCICE 5.6

PROGRAMME

```

10 REM
20 REM *****
30 REM *   JEU DU PENDU   *
40 REM *****
50 REM
60 SYMBOL 250,&FF,&81,&81,&81,&81,&81,&81,&81,&FF
70 I=-1
80 I=I+1:READ A$:IF A$<>"FIN" THEN 80
90 NBMOT=I
100 RESTORE
110 INK 0,1:INK 1,24:INK 2,26:INK 3,13
120 PAPER 3
130 WINDOW #1,1,19,20,22
140 WINDOW #2,20,40,20,20
150 WINDOW #3,19,25,22,22
160 PAPER #1,3:PEN #1,2
170 PAPER #2,3:PEN #2,2
180 PAPER #3,3:PEN #3,2
190 CLS
200 PRINT "Vous avez 10 secondes pour lire"
210 PRINT "ces ";NBMOT;"mots :":PRINT
220 FOR I=1 TO NBMOT
230 READ A$:PRINT A$,
240 NEXT I
250 RESTORE
260 TT=TIME
270 IF TIME-TT<3000 THEN 270
280 CLS
290 PRINT:PRINT TAB(14)"JEU DU PENDU"
300 PRINT:PRINT "Pour arreter, tapez FIN"
310 REM
320 REM Choix du mot a decouvrir
330 REM *****
340 MOTIR=INT(NBMOT*RND)+1
350 FOR I=1 TO MOTIR
360 READ MOTIR$
370 NEXT I

```

```

380 RESTORE
390 LM=LEN(MOTIR$)
400 CLS #1
410 REM
420 REM Dessin de la grille
430 REM * du mot cache *
440 REM *****
450 FOR I=1 TO LM
460 LOCATE 2*I,10:PRINT CHR$(250);
470 PCAR(I)=0
480 NEXT I
490 REM
500 REM Saisie du mot joue
510 REM *****
520 PRINT #1,"Quel est le mot : "
530 PRINT #1
540 PRINT #1,"Nombre d'essais : ";
550 NBC=0
560 NBC=NBC+1
570 PRINT #3,NBC;
580 CLS #2:INPUT #2,"",MOTP$
590 MOTP$=UPPER$(MOTP$)
600 IF MOTP$="FIN" THEN 810
610 IF LEN(MOTP$)<>LM THEN 560
620 REM
630 REM Recherche des lettres
640 REM * bien placees *
650 REM *****
660 FOR I=1 TO LM
670 IF PCAR(I)=1 THEN 690
680 IF MID$(MOTP$,I,1)=MID$(MOTIR$,I,1) THEN
PCAR(I)=1:LOCATE 2*I,10:PRINT MID$(MOTP$,I,1)
690 NEXT I
700 IF MOTP$<>MOTIR$ THEN 560
710 REM
720 REM Fin d'une partie
730 REM *****
740 CLS #1:CLS #2:CLS #3:PRINT #1,"Vous avez
trouve en";NBC;"coups"
750 A$=INKEY$:IF A$="" THEN 750
760 IF A$<>CHR$(13) THEN 190
770 INK 0,1:INK 1,24

```

```
780 PAPER 0:PEN 1
790 CLS
800 END
810 REM
820 CLS #1:CLS #2:CLS #3
830 WINDOW #1,1,40,22,23
840 PRINT #1,"Le mot cache est : ";MOTIR$;
850 GOTO 750
860 REM
870 REM Liste des mots
880 REM terminee par FIN
890 REM *****
900 DATA CLAVIER, MONTRE, CHEVAL, ATOME
910 DATA FUSEE, LIVRE, ROSIER, MUSEE
920 DATA MAMAN, ICONE, AZOTE, VAISSEAU
930 DATA PIERRE, MOUCHE, ROUTE, CLOWN
940 DATA FIN
```

VARIABLES

NBC nombre de coups joués
I,TT compteurs
A\$ variable alphanumérique
NBMOT nombre de mots dans la liste DATA
MOTIR rang du mot caché MOTIR\$ dans la liste DATA
MOTIR\$ mot choisi au hasard dans la liste DATA
LM longueur de MOTIR\$
PCAR tableau contenant les drapeaux indiquant les lettres trouvées
MOTP\$ mot proposé par le joueur

COMMENTAIRES

260-270 : boucle servant à obtenir une durée d'affichage à l'écran de dix secondes. TIME est une horloge interne de l'AMSTRAD qui s'incrémente tous les 3 centièmes de seconde. Pour compter 10 secondes, il faut donc tester si 3 000 incréments ont été faites (ligne 270).
320-370 : tirage aléatoire de MOTIR\$ dans la liste.
420-480 : zone d'affichage du mot à découvrir. Dès qu'une lettre est découverte, elle est affichée à la position qu'elle occupe dans le mot à découvrir.

- 580 : acquisition du mot MOTP\$.
 630-690 : comparaison du mot MOTP\$ et du mot MOTIR\$.
 680 : si la lettre en position I est bien placée et qu'elle n'a pas été déjà découverte (PCAR(I)=0), elle est affichée et PCAR(I) est mis à 1.
 700 : si toutes les lettres sont découvertes, le joueur a gagné.
 870-940 : liste des mots sous forme de DATA, terminée par le mot réservé FIN.

EXERCICE 5.7

PROGRAMME

```

10 REM
20 REM TELECRAN
30 REM *****
40 INK 0,0:INK 1,26:INK 2,11:INK 3,6
50 WINDOW #0,1,40,2,24
60 PAPER 0:ORIGIN 0,0
70 WINDOW #1,1,40,1,1
80 PAPER #1,1:PEN #1,0
90 CLS #1
100 PRINT #1,"TELECRAN : Sortie(ENTER) * Effacer:BARRE";
110 WINDOW #2,1,40,25,25
120 PAPER #2,1:PEN #2,0
130 CLS #2
140 PRINT #2,SPC(3)"0:NOIR***1:BLANC***2:BLEU***3:ROUGE";
150 CLS
160 SPEED KEY 1,1
170 F=1:X=300:Y=200
180 PLOT X,Y,F
190 REM
200 REM Entree d'une touche
210 REM *****
220 A$=INKEY$:IF A$="" THEN 220
230 IF A$>="0" AND A$<="3" THEN F=VAL(A$):GOTO 180

```

```
240 AS=ASC(A$)
250 IF AS=242 THEN IF X<2 THEN X=1:GOTO 180 E
LSE X=X-1:GOTO 180
260 IF AS=243 THEN IF X>638 THEN X=639:GOTO 1
80 ELSE X=X+1:GOTO 180
270 IF AS=241 THEN IF Y<17 THEN Y=16:GOTO 180
ELSE Y=Y-1:GOTO 180
280 IF AS=240 THEN IF Y>381 THEN Y=382:GOTO 1
80 ELSE Y=Y+1:GOTO 180
290 IF AS=32 THEN 150
300 IF AS<>13 THEN 190
310 WINDOW #0,1,40,1,25
320 SPEED KEY 10,3
330 END
```

VARIABLES

F code de l'encre utilisée (0, 1, 2 ou 3)
X,Y coordonnées du point à afficher
A\$ touche tapée par le joueur
AS code ASCII de la touche tapée

COMMENTAIRES

170 :initialisation au centre de la zone de dessin, avec point d'affichage allumé.
250-280 : les mouvements du point lumineux ne sont possibles que dans la limite de la zone d'affichage.
290, 300 : une pression de la barre d'espacement efface l'écran et repositionne le point au milieu de l'écran. Une pression sur la touche ENTER arrête le jeu.

EXERCICE 5.8

PROGRAMME

```
10 REM
20 REM GEOGRAPHIE DE LA FRANCE
30 REM *****
40 REM *****
```

```

50 INK 0,6
60 INK 1,26
70 INK 2,13
80 INK 3,24,0
90 SPEED INK 5,5
100 PAPER 2
110 WINDOW #1,1,40,24,25
120 PAPER #1,1:PEN #1,0
130 CLS:CLS #1
140 SCA=1.4
150 ORIGIN -90*SCA,420*SCA
160 REM
170 REM Trace de la carte de FRANCE
180 REM *****
190 READ X,Y:PLOT X*SCA,-Y*SCA,1
200 FOR I=2 TO 43
210 READ X,Y:DRAW X*SCA,-Y*SCA
220 NEXT I
230 READ NVIL
240 REM
250 REM Choix d'une ville au hasard
260 REM *****
270 NVC=RND*NVIL+1
280 RESTORE 740
290 FOR I=1 TO NVC
300 READ VP$,XVC,YVC
310 NEXT I
320 PLOT XVC*SCA,-YVC*SCA,3
330 DRAWR 2,0:DRAWR 0,2:DRAWR -2,0:DRAWR 0,-2
340 INPUT #1,"Quelle est cette ville : ",VIL$
350 VIL$=UPPER$(VIL$)
360 REM
370 REM Affichage du resultat
380 REM *****
390 IF VIL$=VP$ THEN ME$="BRAVO ! Vous etes b
on en GEOGRAPHIE"+CHR$(7) ELSE ME$="FAUX. Cet
te ville est : "+VP$+CHR$(7)
400 PRINT #1,ME$
410 A$=INKEY$:IF A$="" THEN 410
420 IF A$=CHR$(13) THEN 470
430 PLOT XVC*SCA,-YVC*SCA,2
440 DRAWR 2,0:DRAWR 0,2:DRAWR -2,0:DRAWR 0,-2

```

EXERCICES EN BASIC POUR AMSTRAD

```
450 CLS #1
460 GOTO 270
470 INK 0,1:INK 1,24:PAPER 0:PEN 1
480 CLS
490 END
500 REM
510 REM Contour de la carte
520 REM *****
530 DATA 308,165,346,180
540 DATA 394,190,392,220,365,265
550 DATA 390,260,391,280,395,310
560 DATA 395,322,400,332,390,340
570 DATA 362,345,360,342,330,335
580 DATA 314,358,314,368,290,362
590 DATA 260,360
600 DATA 250,365,218,350,226,314
610 DATA 226,296,226,280
620 DATA 210,254,180,236,175,234
630 DATA 170,228
640 DATA 160,225,160,218,178,210
650 DATA 186,210,198,212,218,218
660 DATA 226,212,218,198,218,185
670 DATA 227,188
680 DATA 234,198,250,198,260,180
690 DATA 264,162,280,150,308,165
700 REM
710 REM Nombre et position des villes
720 REM *****
730 DATA 8
740 DATA PARIS,290,210
750 DATA BORDEAUX,240,312
760 DATA STRASBOURG,389,210
770 DATA MARSEILLE,360,340
780 DATA GRENOBLE,370,294
790 DATA LYON,345,280
800 DATA BREST,165,222
810 DATA PAU,220,348
```

VARIABLES

SCA facteur d'échelle
X,Y coordonnées graphiques d'un point à l'écran

I	compteur
NVIL	nombre de villes dans la liste de DATA
VP\$	nom de la ville à trouver
NVC	rang de VP\$ dans la liste de DATA
XVC,YVC	coordonnées graphiques de VP\$
VIL\$	réponse
ME\$	message
A\$	touche frappée au clavier

COMMENTAIRES

- 140 : le facteur d'échelle SCA permet avec les mêmes données numériques de dilater ou de réduire le dessin.
- 170-220 : dessin de la carte. Le choix de l'origine (ligne 150) est tel que l'axe horizontal est situé en haut de l'écran et donc tous les points de la carte ont des ordonnées négatives.
- 250-310 : tirage aléatoire du nom de la ville à découvrir.
- 320-330 : positionnement d'un petit carré clignotant.
- 370-400 : affichage du résultat à la réponse donnée.
- 430-440 : on efface le point lumineux clignotant en coloriant en gris (couleur du fond) le rectangle précédemment dessiné (ligne 330).
- 500-690 : coordonnées des 46 points définissant le contour de la carte.
- 700-810 : liste des villes suivies des coordonnées de leur position.

EXERCICE 5.9

PROGRAMME

```

10 REM *****
20 REM *   LOTO SPORTIF   *
30 REM *****
40 ON BREAK GOSUB 590
50 SYMBOL 250, &13, &D, &E, &18, &1E, &24, &48, &80
60 INK 0, 13
70 INK 1, 26
80 INK 2, 9
90 PAPER 0: PEN 1

```

EXERCICES EN BASIC POUR AMSTRAD

```
100 CLS
110 PRINT TAB(14)"LOTO SPORTIF"
120 REM
130 REM Affichage de la piste
140 REM *****
150 WINDOW #0,1,40,3,6
160 FOR I=1 TO 7
170 WINDOW #I,1,40,2*I+8,2*I+8
180 PAPER #I,2:PEN #I,1
190 CLS #I
200 PRINT #I,USING "#&";I;CHR$(250);
210 NEXT I
220 CLS #0
230 INPUT "Donnez un pronostic (ex:1,3,6) : "
,N(1),N(2),N(3)
240 IF N(1)=N(2) OR N(2)=N(3) OR N(1)=N(3) TH
EN 220
250 FOR I=1 TO 3
260 IF N(I)<1 OR N(I)>7 OR N(I)<>INT(N(I)) TH
EN 220
270 NEXT I
280 PRINT "Arrivee gagnante : ";
290 REM
300 REM Debut de la course
310 REM *****
320 NC=0
330 FOR I=1 TO 7:P(I)=1:NEXT I
340 K=INT(RND*7)+1
350 IF P(K)=39 THEN 340
360 P(K)=P(K)+1
370 LOCATE #K,POS(#K)-1,1:PRINT #K," "+CHR$(2
50);
380 IF P(K)<>39 THEN 340
390 NC=NC+1
400 IF NC=7 THEN 460
410 IF NC<4 THEN PRINT K;
420 GOTO 340
430 REM
440 REM Fin de la course
450 REM *****
460 PRINT "Pour continuer tapez une touche"
470 PRINT "Pour arreter tapez ENTER"
```

```

480 A$=INKEY$:IF A$="" THEN 480
490 IF A$=CHR$(13) THEN 590
500 CLS #0
510 FOR I=1 TO 7
520 LOCATE #I,40,1:PRINT #I," ";
530 LOCATE #I,2,1:PRINT #I,CHR$(250);
540 NEXT I
550 GOTO 220
560 REM
570 REM Retour
580 REM *****
590 INK 0,1:PAPER 0
600 INK 1,24:PEN 1:WINDOW 1,40,1,25
610 END

```

VARIABLES

NC nombre de coureurs ayant franchi la ligne d'arrivée
 N tableau de dimension implicite contenant le tiercé joué
 I compteur
 P tableau contenant la position du coureur de rang I
 K rang du coureur qui avance d'une case
 A\$ touche frappée au clavier

COMMENTAIRES

160-210 : affichage des 7 coureurs dans les cases départ et inscription des sept numéros en début de couloir.
 190 : le format utilise "#&" permet l'affichage d'un chiffre (#) et d'un caractère (&).
 230-260 : test de validité du pronostic d'arrivée.
 330-360 : un des sept coureurs, tiré au hasard, avance d'une position sauf s'il a atteint l'arrivée.
 380-410 : si un coureur franchit la ligne d'arrivée dans les trois premiers, son numéro est affiché.

EXERCICE 5.10

PROGRAMME

```

10 REM
20 REM *****
30 REM * FICHER/REPERTOIRE *
40 REM *****
50 REM
60 DIM ELEM$(500)
70 ME$(1)="Lecture du fichier : "
80 ME$(2)="Affichage du fichier : "
90 ME$(3)="Insertion d'une fiche : "
100 ME$(4)="Suppression d'une fiche : "
110 ME$(5)="Enregistrement du fichier : "
120 ME$(6)="Fin : "
130 INK 0,0:INK 1,24:INK 2,6:INK 3,13
140 CLS
150 WINDOW #1,1,40,1,8
160 PAPER #1,0:PEN #1,1:CLS #1
170 WINDOW #2,1,40,9,23
180 PAPER #2,3:PEN #2,1:CLS #2
190 WINDOW #3,1,40,24,25
200 PAPER #3,0:PEN #3,2:CLS #3
210 REM
220 REM Affichage du menu
230 REM *****
240 PRINT #1,STRING$(40,"*");
250 FOR I=1 TO 6
260 PRINT #1,"*  "ME$(I)TAB(32);I;SPACE$(5)"*
";
270 NEXT I
280 PRINT #1,STRING$(40,"*");
290 CLS #2:INPUT #2,"Operation choisie (1-6)
":",R
300 IF R<>INT(R) OR R<1 OR R>6 THEN 290
310 IF R=6 THEN END
320 ON R GOSUB 350,480,610,730,940
330 GOTD 290
340 END
350 REM

```

```

360 REM Sous programme de lecture
370 REM      sur cassette
380 REM *****
390 INPUT #2,"Nom du fichier : ",NFIC$
400 NFIC$="!" +NFIC$
410 OPENIN NFIC$
420 WHILE EOF=0
430 NBEL=NBEL+1
440 LINE INPUT #9,ELEM$(NBEL)
450 WEND
460 CLOSEIN
470 RETURN
480 REM
490 REM Sous programme d'affichage
500 REM      du fichier
510 REM *****
520 CLS #3
530 PRINT #3:PRINT #3,"Pour revenir au menu,
tapez une touche"CHR$(7)
540 IF NBEL=0 THEN 580
550 FOR I=1 TO NBEL
560 PRINT #2,USING "### : &";I,ELEM$(I)
570 NEXT I
580 IF INKEY$="" THEN 580
590 CLS #3
600 RETURN
610 REM
620 REM Sous programme d'insertion
630 REM *****
640 CLS #3
650 LOCATE #3,1,15:PRINT #3,"Pour sortir du m
ode insertion":PRINT #3,"reponder : ***"CHR$(
7)
660 PRINT #2
670 NBEL=NBEL+1
680 PRINT #2,"Fiche a inserer : ";:LINE INPUT
#2,ELEM$(NBEL)
690 IF ELEM$(NBEL)<>"***" AND ELEM$(NBEL)<>"
THEN 670
700 NBEL=NBEL-1
710 CLS #3
720 RETURN

```

EXERCICES EN BASIC POUR AMSTRAD

```
730 REM
740 REM Sous programme de suppression
750 REM *****
760 CLS #3
770 PRINT #2,"Numero de l'enregistrement"
780 INPUT #2,"a supprimer : ",K
790 IF K<>INT(K) OR K<=0 OR K>NBEL THEN 890
800 PRINT #2:PRINT #2,"Enr. ";K;" : "ELEM$(K):
PRINT #2
810 INPUT #2,"Pour confirmer, tapez OUI : ",R
EP$
820 IF REP$<>"OUI" AND REP$<>"oui" THEN 920
830 NBEL=NBEL-1
840 FOR I=K TO NBEL
850 ELEM$(I)=ELEM$(I+1)
860 NEXT I
870 CLS #3
880 RETURN
890 PRINT #3,"SUPPRESSION IMPOSSIBLE"+CHR$(7)
900 PRINT #3,"Pour revenir au menu, tapez une
  touche"
910 IF INKEY$="" THEN 910
920 CLS #3
930 RETURN
940 REM
950 REM Sous programme d'enregistrement
sur cassette
960 REM
970 REM *****
980 INPUT #2,"Nom du fichier : ",NFIC$
990 NFIC$="!" +NFIC$
1000 OPENOUT NFIC$
1010 FOR I=1 TO NBEL
1020 PRINT #9,ELEM$(I)
1030 NEXT I
1040 CLOSEOUT
1050 RETURN
```

VARIABLES

ME\$	tableau des messages
R	réponse au menu
K	numéro de l'élément à supprimer

REP\$	réponse de l'utilisateur
NFIC\$	nom du fichier à lire ou à enregistrer
I	compteur
ELEM\$	tableau contenant la liste des éléments
NBEL	nombre total d'éléments du fichier traité

COMMENTAIRES

- 10-330 : programme principal, comprenant l'affichage du menu et le branchement (ligne 320) sur les différents sous-programmes.
- 360-470 : sous-programme de lecture du fichier sur cassette.
- 410 : l'instruction OPENIN ouvre le fichier NFIC\$ sur le canal 9 en lecture (IN) et précise ainsi que l'échange des données se fait du lecteur de cassettes vers l'ordinateur. D'autre part, le point d'exclamation en début de la chaîne NFIC\$ empêche l'apparition des messages usuels lors de l'utilisation du lecteur de cassettes.
- 420-450 : boucle WHILE/WEND. L'instruction EOF (**End Of File** : fin de fichier) indique la fin du fichier sur le canal 9 et permet de déterminer le nombre NBEL d'éléments contenus dans le fichier. En effet à chaque lecture d'un enregistrement et tant que la valeur de EOF est différente de -1 (instruction WHILE), la variable NBEL est incrémentée (ligne 430).
- 460 : l'instruction CLOSEIN ferme le fichier sur le canal 9 en lecture.
- 490-600 : sous-programme d'affichage du fichier à l'écran.
- 620-720 : sous-programme d'insertion d'éléments.
- 740-930 : sous-programme de suppression d'un élément.
- 830-860 : après suppression d'un élément, tous les éléments suivants sont décalés d'un rang, sauf si l'élément à supprimer est le dernier du fichier.
- 950-1050 : sous-programme d'enregistrement du fichier sur le lecteur de cassettes.
- 1000 : l'instruction OPENOUT ouvre le fichier NFIC\$ sur le canal 9 en écriture (OUT) et précise ainsi que l'échange des données se fait de l'ordinateur vers le lecteur de cassettes.
- 1040 : l'instruction CLOSEOUT ferme le canal 9.

Remarque : pour que ce programme tourne sur un ordinateur de la famille AMSTRAD, équipé d'un lecteur de disquettes, supprimer les lignes 400 et 990. En effet, il n'est plus nécessaire de faire précéder le nom du fichier d'un "!", aucun message n'apparaissant lors de l'utilisation du lecteur de disquettes.

EXERCICE 5.11

PROGRAMME

```

10 REM *****
20 REM *   MATERMIND   *
30 REM *****
40 ON BREAK GOSUB 700
50 REM
60 REM Donnees initiales
70 REM *****
80 SYMBOL 250,&C3,&99,&F9,&F3,&E7,&FF,&E7,&FF
90 B$="BWRVJGMD"
100 DATA NOIR,0,BLEU,2,BLANC,26,ROUGE,6,VERT,
108,JAUNE,24,GRIS,13,MAGENTA,8,ORANGE,15
110 FOR I=0 TO 8
120 READ COL$(I),COL:INK I,COL
130 NEXT I
140 REM
150 REM Affichage des regles du jeu
160 REM *****
170 PAPER 0:BORDER 0:CLS
180 LOCATE 14,2
190 FOR I=1 TO 10
200 PEN (I MOD 3)+1:PRINT MID$("MASTERMIND",I
,1);
210 NEXT I
220 LOCATE 1,5
230 PRINT "Pour obtenir ":PRINT
240 FOR I=1 TO 8
250 PRINT TAB(10)COL$(I)TAB(20)"tapez ";MID$(
B$,I,1)
260 NEXT I
270 PRINT
280 PRINT "Pour se deplacer a gauche ";CHR$(
242)
290 PRINT "Pour se deplacer a droite ";CHR$(
243)
300 PRINT
310 PRINT "Pour valider l'essai, tapez ENTER"
320 PRINT

```

```

330 PRINT "B signifie BIEN PLACE"
340 PRINT "M signifie MAL PLACE"
350 PRINT
360 PRINT "Pour commencer tapez une touche"
370 IF INKEY#="" THEN 370
380 REM
390 REM   Choix aleatoire de la
400 REM   configuration a decouvrir
410 REM *****
420 RANDOMIZE(TIME)
430 FOR I=1 TO 5
440 P(I)=INT(RND*8)+1
450 NEXT I
460 REM
470 REM Affichage de l'ecran de jeu
480 REM *****
490 MODE 0
500 PRINT "Essais"SPC(5)"Reponses"
510 PRINT
520 FOR I=1 TO 7
530 REM
540 GOSUB 740 "Saisie de l'essai
550 GOSUB 960 "Reponse d'AMSTRAD
560 REM
570 IF NBIEN=5 THEN PRINT "GAGNE":GOTO 660
580 NEXT I
590 PEN 2:PRINT
600 PRINT "Vous avez perdu "
610 PRINT "il fallait donner : "
620 PRINT:PRINT SPC(8);
630 FOR I=1 TO 5
640 PEN P(I):PRINT CHR$(143),
650 NEXT I
660 REM
670 REM Retour en mode 1
680 REM *****
690 IF INKEY#<>CHR$(13) THEN 690
700 INK 0,0:INK 1,24
710 PAPER 0:PEN 1
720 MODE 1
730 END
740 REM *****

```

EXERCICES EN BASIC POUR AMSTRAD

```
750 REM *   Sous programme           *
760 REM *d'acquisition de l'essai*
770 REM *****
780 ZONE 2
790 FOR J=1 TO 5
800 Q(J)= 2
810 PEN 2:PRINT CHR$(143),
820 NEXT J
830 J=1
840 IF J>5 THEN J=5
850 IF J<1 THEN J=1
860 PEN Q(J):LOCATE 2*J-1,VPOS(#):PRINT CHR$(250);
870 REM
880 A$=UPPER$(INKEY$)
890 IF A$="" THEN 880
900 IF A$=CHR$(13) THEN LOCATE 2*J-1,VPOS(#):PRINT CHR$(143);:RETURN
910 IF A$=CHR$(242) THEN LOCATE 2*J-1,VPOS(#):PRINT CHR$(143);:J=J-1:GOTO 840
920 IF A$=CHR$(243) THEN LOCATE 2*J-1,VPOS(#):PRINT CHR$(143);:J=J+1:GOTO 840
930 CC=INSTR(B$,A$):IF CC=0 THEN 880
940 Q(J)=CC
950 GOTO 840
960 REM *****
970 REM *Sous programme de*
980 REM *   comparaison   *
990 REM *****
1000 ZONE 1
1010 NBIEN=0:NMAL=0
1020 FOR H=1 TO 5
1030 IF P(H)=Q(H) THEN NBIEN=NBIEN+1:Q(H)=-1
1040 NEXT H
1050 FOR H=1 TO 5
1060 IF Q(H)=-1 THEN 1100
1070 FOR K=1 TO 5
1080 IF Q(K)=P(H) THEN NMAL=NMAL+1:Q(K)=-2:GOTO 1100
1090 NEXT K
1100 NEXT H
1110 REM
```

```

1120 REM Affichage
1130 REM *****
1140 PEN 2
1150 LOCATE 12,VPDS(#0)
1160 IF NBIEN=0 THEN 1200
1170 FOR H=1 TO NBIEN
1180 PRINT "B",
1190 NEXT H
1200 IF NMAL=0 THEN 1240
1210 FOR H=1 TO NMAL
1220 PRINT "M",
1230 NEXT H
1240 PRINT:PRINT
1250 RETURN

```

VARIABLES

B\$ chaînes de caractères représentant les initiales des 8 couleurs utilisées. Comme B a été choisi pour désigner le bleu, **W (White)** a été utilisé pour désigner le blanc

NBIEN nombre de pions découverts en bonne position

NMAL nombre de pions découverts en mauvaise position

I,J,K,H compteurs

Q tableau contenant le rang des initiales des couleurs des pions de l'essai joué

A\$ caractère frappé au clavier

CC position de A\$ dans B\$

P tableau contenant le rang des initiales des couleurs des pions de la configuration à découvrir

COL\$ tableau des couleurs utilisées

COL code de la couleur COL\$

COMMENTAIRES

100 : les couleurs utilisées et leurs codes associés sont passés par une liste DATA.

110-130 : lecture des DATA et affectation des couleurs d'encre.

150-360 : affichage des règles du jeu.

380-450 : choix de la configuration à découvrir.

570 : si en moins de 7 coups, les cinq pions sont bien placés, le joueur a trouvé la configuration à découvrir et le message GAGNE s'affiche.

- 630-650 : affichage de la configuration à découvrir après 7 essais infructueux.
740-950 : sous-programme de saisie d'un essai.
790-820 : l'essai est initialisé (5 pions blancs).
860 : la position du pion dont la couleur peut être choisie est repérée par un point d'interrogation.
900 : si le joueur tape sur la touche ENTER, l'essai est validé.
910-920 : traitement des touches de déplacement gauche et droit.
930 : détermination de la couleur à partir de l'initiale entrée au clavier par INSTR(B\$,A\$) qui donne la position de A\$ dans B\$ et donc le rang de la couleur choisie dans le tableau COL\$.
940 : le code de la couleur choisie pour le pion de la Jième position est affecté à l'élément de rang J du tableau Q.
960-1250 : sous-programme de comparaison des tableaux P et Q.
1010-1040 : calcul du nombre de pions découverts et bien placés. On compare pour cela position par position les tableaux P et Q. Si le résultat est vrai, on met à (-1) l'élément de Q.
1050-1100 : calcul du nombre de pions découverts et mal placés.
1060 : les pions découverts bien placés ne sont pas pris en compte.
1170-1190 : affichage d'autant de lettres B que de pions bien placés.
1210-1230 : affichage d'autant de lettres M que de pions mal placés.

EXERCICE 5.12

PROGRAMME

```
10 REM
20 REM TOURS D'HANDI
30 REM *****
40 REM
50 REM Valeurs initiales
60 REM *****
70 SYMBOL 243,0,0,0,0,&FF,&FF,&FF,&FF
80 NBANN=5
```

```

90 TR$(0)="11111":TR$(1)="00000":TR$(2)="0000
0"
100 INK 0,1:INK 1,24:INK 2,0:INK 3,8
110 PAPER 2
120 WINDOW #1,1,40,22,25
130 PAPER #1,0
140 ZONE 13
150 PRINT CHR$(22)+CHR$(1)
160 REM
170 REM Dessin des tours
180 REM *****
190 GOSUB 510
200 REM
210 REM Reponse du joueur
220 REM *****
230 FOR T=1 TO 1000:NEXT T:CLS #1
240 LOCATE #1,5,2:PRINT #1,"Deplacement depuis la tour ";
250 A$=INKEY$:IF A$="" THEN 250
260 IF A$=" " THEN 490
270 IF A$<>"1" AND A$<>"2" AND A$<>"3" THEN 250
280 PRINT #1,A$:LOCATE #1,10,3:PRINT #1,"jusqu'a la tour ";
290 B$=INKEY$:IF B$="" THEN 290
300 IF B$<>"1" AND B$<>"2" AND B$<>"3" OR B$=A$ THEN 290
310 PRINT #1,B$
320 REM
330 REM Traitement de la reponse
340 REM *****
350 REM
360 ND=VAL(A$)-1:NA=VAL(B$)-1
370 PND=INSTR(TR$(ND),"1"):PNA=INSTR(TR$(NA),"1")
380 IF PND=0 OR (PND>PNA AND PNA<>0) THEN PRINT #1,"IMPOSSIBLE !!":CHR$(7):GOTO 230
390 PN=PND:L$="0":R$=TR$(ND)
400 GOSUB 730 'calcul de TR$(ND)
410 TR$(ND)=R$
420 PN=PND:L$="1":R$=TR$(NA)
430 GOSUB 730 'calcul de TR$(NA)

```

EXERCICES EN BASIC POUR AMSTRAD

```
440 TR$(NA)=R$
450 GOSUB 790 'deplacement de l'anneau
460 IF TR$(2)<>"11111" THEN 230
470 LOCATE 17,18:PRINT "GAGNE"+CHR$(7)
480 PRINT:PRINT
490 PRINT CHR$(22)+CHR$(0)
500 END
510 REM
520 REM Sous-programme de dessin
530 REM des 3 tours et des 5 anneaux
540 REM *****
550 REM
560 CLS
570 PEN 3
580 PRINT:PRINT
590 FOR I=1 TO 6
600 FOR J=1 TO 3:PRINT SPACE$(6-I)+STRING$(2*I,CHR$(143)),:NEXT J
610 NEXT I
620 LOCATE 1,3
630 FOR I=1 TO NBANN
640 PEN 1
650 PRINT SPACE$(5-I)+STRING$(2*I+2,CHR$(243)
)
660 NEXT I
670 LOCATE 1,11
680 FOR J=1 TO 3
690 PRINT SPC(4)J,
700 NEXT J
710 LOCATE 12,15:PRINT "TOURS D'HANDI";
720 RETURN
730 REM
740 REM Calcul de R$
750 REM *****
760 IF PN>1 AND PN<5 THEN R$=LEFT$(R$,PN-1)+L
$+MID$(R$,PN+1):RETURN
770 IF PN=1 OR PN=0 THEN R$=L$+MID$(R$,2):RET
URN
780 IF PN=5 THEN R$=LEFT$(R$,4)+L$:RETURN
790 REM
800 REM Deplacement de l'anneau joue
810 REM *****
```

```

820 PRINT CHR$(22)+CHR$(0)
830 LOCATE 1+13*ND,2+PND:PEN 3:PRINT SPACE$(5
-PND)+" "STRING$(2*PND,CHR$(143))" "
840 PRINT CHR$(22)+CHR$(1)
850 LOCATE 1+13*NA,PND+2:PEN 1:PRINT SPACE$(5
-PND)+STRING$(2*PND+2,CHR$(243))
860 RETURN

```

VARIABLES

NBANN nombre d'anneaux
TR\$ tableau donnant l'état d'occupation d'une tour
A\$,B\$ touches frappées
I,J,T compteurs
ND numéro désignant la tour de départ
NA numéro désignant la tour d'arrivée
PND position de l'anneau supérieur de la tour ND
PNA position de l'anneau supérieur de la tour NA
PN variable de position
R\$ variable désignant un élément de TR\$
L\$ variable alphanumérique

COMMENTAIRES

90 : initialisation de TR\$. TR\$(0)=11111, TR\$(1)=00000 et TR\$(2)=00000.
 150 : passage en mode transparent de façon à superposer le dessin des anneaux sur les tours.
 210-320 : saisie de la réponse du joueur dans la fenêtre 1.
 380 : si PND est supérieur à PNA, le joueur essaie de superposer un anneau sur un anneau plus petit. Si PND est nul, le joueur essaie de déplacer un anneau depuis une tour vide.
 390-440 : le sous-programme (ligne 730-790) effectue le changement dans R\$ de l'élément de rang PN et le remplace par L\$: L\$="0" si l'anneau est à enlever et L\$="1" si l'anneau est à mettre. Ce sous-programme est exécuté la première fois pour la tour ND et la seconde fois pour la tour NA.
 460 : test pour savoir si le joueur a gagné.
 510-720 : sous-programme de dessin de l'écran initial de jeu.

CONSEILS DE LECTURE

Pour maîtriser le Basic Amstrad, connaître toutes les finesses de la programmation et découvrir le système des CPC 464, 664 et 6128, P.S.I. vous propose une palette d'ouvrages utiles.

POUR MAITRISER LE BASIC AMSTRAD

- **Découverte de l'Amstrad** – Daniel-Jean David (Editions du P.S.I.)
Une découverte exhaustive et claire des CPC 464, 664 et 6128, un apprentissage progressif du Basic Amstrad pour débutants complets.
- **Basic Amstrad, CPC 464, 664 et 6128 – 1. Méthodes pratiques** – Jacques Boisgontier et Bruno Césard (Editions du P.S.I.)
Pour ceux qui ont déjà pratiqué un Basic, un ouvrage de perfectionnement du Basic Amstrad illustré par de nombreux programmes-exemples.
- **BASIC Amstrad, CPC 464, 664 et 6128 – 2. Programmes**, par Jacques Boisgontier (Editions du P.S.I.)
Un excellent complément du « Méthodes pratiques » permettant de mettre en pratique vos connaissances en BASIC Amstrad.

POUR MIEUX PROGRAMMER VOTRE AMSTRAD

- **102 programmes pour Amstrad** – Jacques Deconchat (Editions du P.S.I.)
Les bases de la programmation en cinq niveaux de difficulté pour apprendre le Basic Amstrad en jouant.
- **Super jeux Amstrad** – Jean-François Sehan (Editions du P.S.I.)
50 programmes commentés pour vous permettre de créer vos propres applications ludiques.
- **Amstrad en famille** – Jean-François Sehan (Editions du P.S.I.)
40 utilitaires de gestion, de jeu, etc. pour vous aider à organiser efficacement vos activités à la maison.
- **Basic +, 80 routines sur Amstrad** – Michel Martin (Editions du P.S.I.)
Pour pousser votre Amstrad au maximum de ses capacités : 80 routines de simulation d'instructions qui n'existent pas en Basic Amstrad.

POUR PROGRAMMER EN LANGAGE MACHINE

- **Assembleur de l'Amstrad** – Marcel Henrot (Editions du P.S.I.)
Une initiation à l'assembleur du Z80 avec de nombreux programmes en langage machine.
- **Clefs pour Amstrad – 1. Système de base** – Daniel Martin (Editions du P.S.I.)
Mémento présentant synthétiquement le jeu d'instructions du Z80, les points d'entrée des routines système, les connecteurs et brochage, etc. Le livre de chevet du programmeur sur Amstrad.
- **Clefs pour Amstrad – 2. Système disque** – par Daniel Martin (Editions du P.S.I.)
Les principales adresses des 16K de ROM disque, agrémentées de commentaires indispensables au programmeur sur Amstrad.

POUR ETRE INFORME REGULIEREMENT DE L'ACTUALITE DES MICROS AMSTRAD

- **MICROSTRAD** revue bimestrielle du Groupe Tests.
Pour exploiter au mieux les capacités de votre micro, vous trouverez au sommaire de chaque numéro un rendez-vous avec les rubriques clés :
 - **DECOUVREZ... LA FACE CACHEE DE VOTRE CPC** : astuces, idées, conseils, tout pour comprendre votre micro, son anatomie, son fonctionnement, sa programmation et exploiter ses capacités graphiques et sonores.
 - **DOMPTEZ... VOTRE CPC 464, 664 OU 6128** : passionnés, petits ou grands, spécialistes ou débutants, une information pratique et la compétence d'experts au service de votre micro.
 - **PROGRAMMEZ... VOTRE MICRO AMSTRAD** : dans chaque numéro de MICROSTRAD, un cocktail de programmes (dessins, jeux, utilitaires, gestion, etc.) et des trucs de programmation.

EN CAS D'ERREUR

Tous les listings de cet ouvrage sont des originaux sortis directement de l'imprimante. Toutes les solutions des exercices proposés ici ont été testées attentivement sur Amstrad. Toute erreur serait donc exceptionnelle.

Si, malgré tout, l'un des programmes ne fonctionnait pas, assurez-vous d'avoir correctement recopié le listing en Basic.

Voici quelques conseils qui vous aideront à déceler l'erreur :

- ne confondez pas 0 (zéro) et O (lettre) ;*
- assurez-vous d'avoir placé correctement les "." et ";" nécessaires ;*
- comptez les lignes du programme à recopier et faites attention à ne pas en oublier ;*
- donnez toujours le même nom à vos variables d'un bout à l'autre du programme.*

Et surtout... Armez-vous de courage !

Votre avis nous intéresse

- Pour nous permettre de faire de meilleurs livres, adressez-nous vos critiques sur le présent livre.
- Si vous souhaitez des éclaircissements techniques, écrivez-nous, nous adresserons votre demande à l'auteur qui ne manquera pas de vous répondre directement.

- Ce livre vous donne-t-il toute satisfaction?

- Y a-t-il un aspect du problème que vous auriez aimé voir abordé?

Comment avez-vous eu connaissance de ce livre?

- | | |
|---|-------------------------------------|
| <input type="checkbox"/> publicité | <input type="checkbox"/> cadeau |
| <input type="checkbox"/> catalogue | <input type="checkbox"/> librairie |
| <input type="checkbox"/> boutique micro | <input type="checkbox"/> exposition |
| <input type="checkbox"/> autres | |

Avez-vous déjà acquis des livres PSI?

lesquels? _____

qu'en pensez-vous? _____

Nom _____ Prénom _____ Age _____

Adresse _____

Profession _____

Centre d'intérêt _____

CATALOGUE GRATUIT

Vous pouvez obtenir un catalogue complet des ouvrages PSI, sur simple demande, ou en retournant cette page remplie à votre libraire, à votre boutique micro ou aux

Editions du PSI
BP 86
77402 Lagny-sur-Marne Cedex



EXERCICES EN BASIC POUR AMSTRAD

Quoi de mieux qu'un exercice pour se graver en mémoire les multiples instructions et commandes BASIC Amstrad ?

Exercices en BASIC pour Amstrad", loin d'être un recueil fastidieux de questions et de réponses, adopte une démarche progressive et pédagogique. L'ouvrage présente chaque mot BASIC par niveau, selon un plan clair et complet :

En première partie : énoncé du problème, données en entrée et en sortie, analyse.

En deuxième partie : solution du problème, variables utilisées, commentaires.

Tous les programmes de cet ouvrage fonctionnent parfaitement sur Amstrad CPC 464, 664 et 6128.

Exercez-vous à maîtriser votre Amstrad !



ÉDITIONS DU P.S.I.
BP 86 - 77402 LAGNY S/MARNE CEDEX - FRANCE

ISBN 2-86595-271-1

130 FF

EXERCICES EN BASIC POUR AMSTRAD





Document **numérisé**
avec amour par :

AMSTRAD

CPC 

MÉMOIRE ÉCRITE



<https://acpc.me/>