

rédaction:
Benoît Rivière
8 rue Victor Hugo
14730 Giberville
FRANCE

ORDIMAGE

ORDIMAGE, le fanzine des passionnés de l'Amstrad CPC et du PC.

le fanzine
CPC
&
PC

numéro 7 - septembre 1994

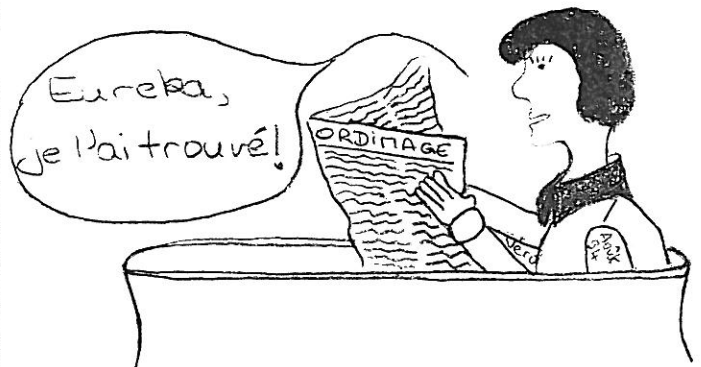
Apériodique gratuit 99 pages

Edito

C'est l'été et pourtant, fidèles au poste, nous nous sommes mis en quatre pour vous satisfaire au mieux. Vous vous délecterez des pages consacrées aux jeux, des astuces et des plans qui vont vous faire venir à bout des stratagèmes les plus audacieux et réduire à néant les adversaires les plus coriaces. Les rubriques sérieuses sont toujours au rendez-vous, aussi bien en ce qui concerne la gestion (Budget, Analyse Financière sur Excel) et programmation (Macro-commande Excel, Basic sur CPC, Assembleur Z80 et 8080, et Turbo Pascal sur PC). Benoit.

Sommaire

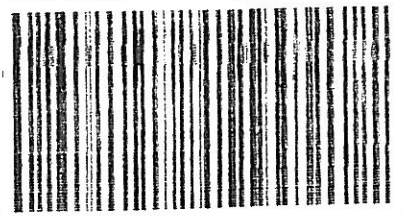
- p2 : Adresses des CPC (suite et fin)
- p7 : Les Tortues Ninja (la solution)
- p10 : Budget sur EXCEL
- p12 : Les nouvelles du Monde Informatique
- p13 : Programmation 8080
- p18 : Batman au banc des accusés
- p19 : Turbo Pascal & TPDSEXT
- p37 : B.A.T.
- p38 : Basic CPC : CPCTAP
- p41 : Turbo Pascal : COPIFIC
- p44 : Basic CPC : ORDICO
- p48 : Macro-commande Excel : DIDACTI
- p54 : RSX sur CPC : SAVECR
- p56 : Dossier Trucs et Astuces sur CPC
- p62 : Les partiels en Turbo Pascal
- p64 : Mathématiques et Informatique
- p65 : Gestion financière sur Excel



Ordimage,
des Idées à chaque page

La représentation
de la page 1 est une
photo numérisée
(photo et
numérisation effec-
tuées par le club in-
formatique WIN'S).

X0000 - 07 - GRATUIT



DOSSIER SPECIAL :

les adresses des CPC

(Acte final)

Voici la suite tant attendue de notre dossier consacré aux adresses de nos bons vieux CPC. Ces adresses sont classées par thème : texte, son, imprimante... Le nom des routines est indiqué (si connu), puis leur utilité, éventuellement leur équivalence Basic, et enfin leurs conditions d'entrée (E) et les sorties (S).

L'utilisation de ces routines à partir de l'assembleur est fort simple. Il suffit de procéder comme suit :

```
ld a,1
call #bc0e
```

Explication : on charge 1 dans le registre A, puis on appelle la routine de changement de résolution d'écran (MODE en Basic).

Voilà, ce dossier est enfin terminé. Il a normalement dû éclairer la lanterne des programmeurs en Assembleur (c'était tout au moins l'objectif de ce dossier). Benoît.

Routines du gestionnaire clavier

BB00 : initialise le clavier, les prédefinitions (KEY, KEYDEF), le tampon clavier (CLEAR INPUT sur 6128), etc..

E : aucune.

S : les registres AF, BC, DE et HL sont modifiés.

BB03 : reset.

BB06 : attente d'une touche du clavier.

E : aucune.

S : A : numéro du caractère (ASCII).

BB09 : teste si une touche est enfoncée. cette routine n'attend pas. (doit être placée dans une boucle d'attente).

E : aucune.

S : si une touche est pressée, carry = 1 et A = numéro du caractère, sinon carry = 0 et A est modifié.

BB0C : réserve un caractère pour le prochain appel de BB09.

BB0F : positionne une chaîne de caractères associée à un code étendu.

BB12 : lit un caractère depuis une chaîne expansée.

BB15 : allocation d'un tampon pour une chaîne expansée.

BB18 : attente de caractère clavier. idem BB06 sauf que BB18 renvoie des valeurs allant de 48 à 57 (codes ASCII) alors que BB06 donne 128 à 137 (valeurs utilisables lors de la redéfinition d'une touche). La différence est vraie avec 139 pour ENTER (au lieu de 13 avec BB18) et 138 pour le "." du pavé numérique (au lieu de 46 avec BB18).

E : aucune.

S : A = numéro du caractère.

BB1B : teste si une touche est enfoncée. Cette routine n'attend pas. (doit être placée dans une boucle d'attente), idem BB09, sauf que A contient le numéro de la touche.

E : aucune.

S : si une touche est pressée, carry = 1 est A = numéro du caractère, sinon carry = 0 et A est modifié.

BB1E : KM TEST KEY
permet par le test de l'appui d'une touche particulière de sortir à tout instant du programme et de restituer l'état initial. (touche ou joystick)

E : A contient le numéro de la touche (ASCII) qui doit être testée.

S : si la touche est pas enfoncée, zero est faux ; sinon il est vrai. A et HL sont modifiés.

C : 128 Ctrl appuyé, 32 Shift appuyé, 160 Shift et Ctrl appuyés.
carry : faux ; autres flags altérés.

BB21 : regarde si Shift et CAPS LOCK sont appuyés.

E : aucune.

S : A altéré ; L : 0 Shift lock enlevé, 255 lock mis.

H : 0 Caps lock enlevé, 255 Caps lock mis.
flags altérés.

BB24 : lecture JOYSTICK.

E : aucune.

S : A : état joystick 0, H : état joystick 0, L : état joystick 1.

Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
réserve	Fire 1	Fire 2	Droite	Gauche	Bas	Haut

Bit 7 = 0 ; Bits ci-dessus mis = état correspondant vrai.

BB27 : positionne un code dans la table sans SHIFT ni CONTROL.

BB2A : fournit le code de la touche pressée sans SHIFT ni CONTROL.

BB2D : positionne un code dans la table SHIFT.

BB30 : fournit le code de la touche pressée avec SHIFT.

BB33 : positionne un code dans la table CONTROL.

BB36 : fournit le code de la touche pressée avec CONTROL.

BB39 : autorise/interdit auto-répétition sur touche spécifique.

E : A : numéro touche (ASCII), B : 0 = pas de répétition, 255 = répétition possible.

S : A, B, C, H, L, et flags altérés.

BB3C : trouve état répétition sur touche spécifique.

E : A : numéro de touche.

S : A, HL : altérés, zero : faux si répétition, sinon vrai
carry : faux

BB3F : positionne le temps avant le début de répétition et la vitesse de répétition. (SPEED KEY en BASIC).

E : H : démarrage retard, L : vitesse de répétition. Ces vitesses sont indiquées en cinquantièmes de seconde (LD H,50 pour une seconde).

S : A et flags : altérés.

BB42 : lecture du temps avant répétition et de la vitesse de répétition.

BB45 : arme le mécanisme de BREAK.

BB48 : désactive la touche ESC (break).

E : aucune.

S : AF et HL sont modifiés.

BB4B : génère un BREAK.

Routines du gestionnaire de lecteur de cassette

BC85 : CAS INITIALISE

initialisation du gestionnaire cassette.

E : aucune.

S : AF, BC, DE et HL sont modifiés.

BC68 : CAS SET SPEED

positionnement de la vitesse d'écriture.

E : HL contient la valeur d'un demi-zéro, A contient la précompensation.

S : AF et HL sont modifiés.

BC6B : autorise ou interdit l'affichage des messages.

BC6E : mise en route du moteur du lecteur.

BC71 : arrêt du moteur du lecteur.

BC74 : repositionne le moteur dans son état précédent.

BC77 : CAS IN OPEN

positionne le tampon pour la lecture et lit le premier bloc.

E : B contient la longueur du nom du fichier :

HL contient l'adresse du nom du fichier, et

DE contient l'adresse du tampon (2k).

S : si Ok, le carry est vrai et le zéro est faux. HL contient l'adresse du tampon qui contient le HEADER, DE contient l'adresse des données, BC contient la longueur du fichier et A, le type de fichier. Si le STREAM (flux) est déjà utilisé, le carry est faux, A, BC, DE, et HL sont modifiés. Dans tous les cas, IX est modifié.

BC7A : CAS IN CLOSE

ferme le fichier d'entrée.

E : aucune.

S : si Ok, le carry est vrai, sinon il est faux.

AF, BC, DE, et HL sont modifiés dans les deux cas.

BC7D : CAS IN ABANDON
abandonne la lecture et ferme le fichier.

BC80 : lecture d'un octet.

BC83 : CAS IN DIRECT
lecture d'un fichier et écriture mémoire.

BC85 : CAS RETURN
met le dernier caractère libre.

BC89 : CAS TEST EOF
teste si on atteint la fin de fichier.

BC8C : CAS OUT OPEN
ouverture d'un fichier en sortie.

BC9F : CAS OUT CLOSE
fermeture d'un fichier en sortie.

BC92 : CAS OUT ABANDON
fermeture immédiate d'un fichier en sortie.

BC95 : CAS OUT DIRECT
écriture d'un caractère sur un fichier de sortie.

BC98 : CAS OUT DIRECT
écriture directe du contenu d'une mémoire vers un fichier de sortie.

BC9B : CAS CATALOG
génère le catalogue cassette.

BC9E : CAS WRITE
écrit un enregistrement sur cassette.
E : HL contient l'adresse des données à écrire,
DE contient le nombre d'octets à écrire.
A contient le caractère de synchronisation.
S : le carry est vrai. En cas de mauvaise lecture, il est faux et A
contient un code d'erreur. De toutes façons, HL et IX sont modifiés.

BCA1 : CAS READ
lit un enregistrement sur cassette.
E : HL contient l'adresse où seront écrites les données,
DE contient le nombre d'octets à lire et
A contient le caractère de synchronisation.
S : le carry est vrai. En cas de mauvaise lecture il est faux et A
contient un code d'erreur. De toutes façons, AF, BC, DE et IX sont
modifiés.

BCA4 : compare un enregistrement sur cassette avec le contenu de la
mémoire.

Routines du gestionnaire de l'imprimante

BD2B : cette routine envoie un caractère a l'imprimante, avec la possibilité de retour si l'imprimante est occupée.

E : A contient le caractère a envoyer.

S : si le caractère a été envoyé, le carry est vrai. Si l'imprimante reste occupée trop longtemps, le carry est faux. De toutes façons, AF est modifiée.

BD2E : MC BUSY PRINTER

teste si l'imprimante est occupée.

E : aucune.

S : si l'imprimante est busy, le carry est vrai, sinon il est faux.

BD31 : MC SEND PRINTER

envoie un caractère a l'imprimante (elle ne doit pas être occupée).

E : A contient le caractère a envoyer.

S : le carry est vrai si le caractère est envoyé et AF est modifiée.

R.S.X.

BCD1 : KL LOG EXT

avertit l'ordinateur de l'existence d'une table de commandes extérieures. Cette table fonctionne comme un pointeur indirect vers les routines de traitements propres a la commande. Ces traitements spécifiques se chargeront du contrôle de syntaxe et de l'exécution de la commande.

E : BC pointe sur la table des sauts d'extension de commande.

HL pointe sur une zone libre de 4 octets en RAM.

S : DE est modifiée.

AU SECOURS ! Help ! ¡ Socorro !

Les Tortues Ninja

Leonardo, Donatello, Raphael et Michellangelo, les plus célèbres mutants des Movies sont de retour. Pour vous aider à arriver jusqu'au bout de ce jeu, voici les solutions et les trucs de TEENAGE MUTANT HERO TURTLE sur Amstrad.

Niveau 1 : APRIL I

Il faut utiliser la plate-forme de la bouche d'égoût pour toucher le monstre (il faut répéter 10 fois cette opération). Le monstre est moins rapide que vous, vous pouvez le bloquer facilement.

Niveau 2 : APRIL II

Là, il faut utiliser les tapis roulants pour bloquer le monstre et le toucher.

Niveau 3 : DAM (Barrage)

Utiliser le plan pour désamorcer les bombes rapidement.

Niveau 4 : SPLINTER

Il est conseillé que deux tortues aient des cordes. Avoir une tortue avec 30 mètres de corde ne sert à rien, car tout est pris lors de

la traversée des toits. En fin de parcours, inutile de se battre avec le monstre, allez libérer directement Splinter. Veiller à bien remettre le van à sa place initiale pour ne pas être embêté quand vous rejouerez.

Niveau 5 : AIRFIELD (Aéroport)

A la fin, sautez vers la tête, se tenir près d'elle et tirer.

Niveau 6 : SHREDDER

Prendre deux tortues et faire le plein de boomerangs (99). Il faut se mettre sur la plate-forme centrale et tirer sans arrêt jusqu'à la mort de Shredder.

A noter

Return : changer de tortues

Shift : prendre le van

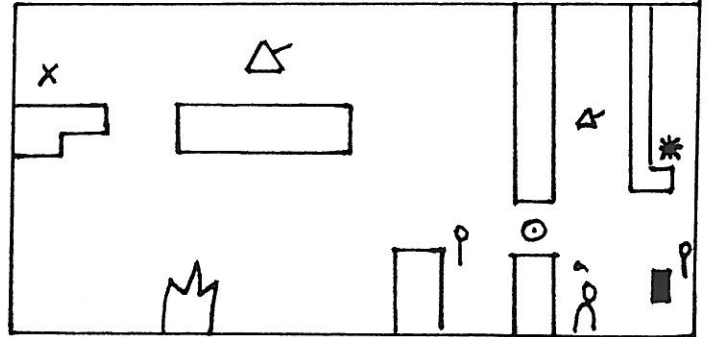
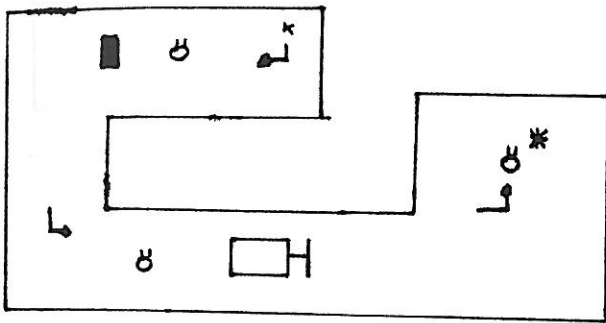
s : changer d'armes

Pour obtenir des VIES INFINIES, appuyez simultanément sur les touches G, 0, 1, et Shift. A.M.

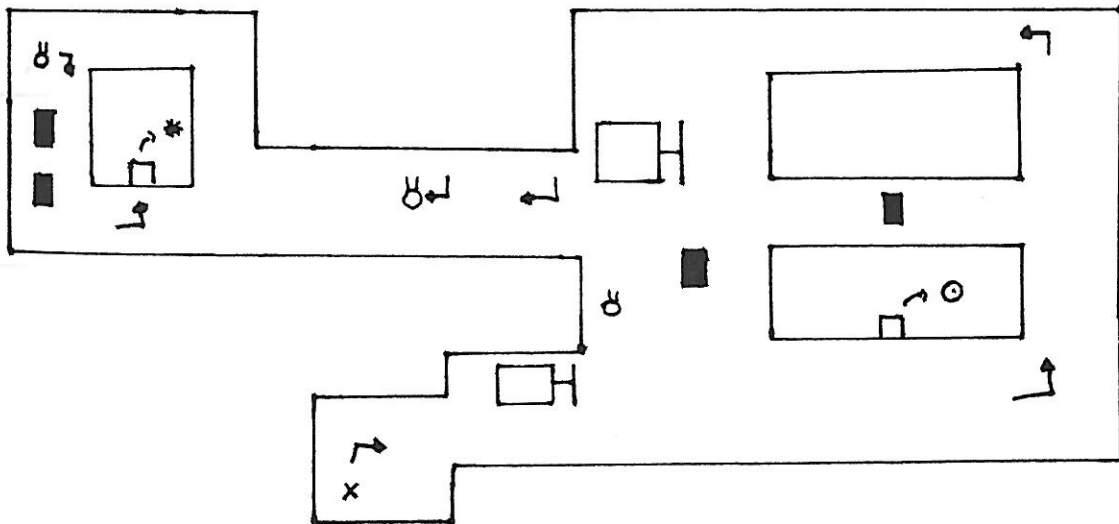
Plan du jeu Tortues Ninja :

(plan réalisé par Alexandre Moreau)

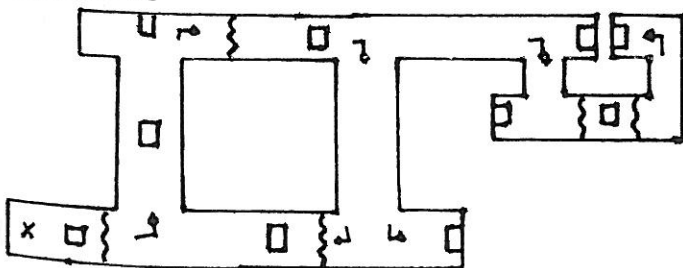
Niveau 1



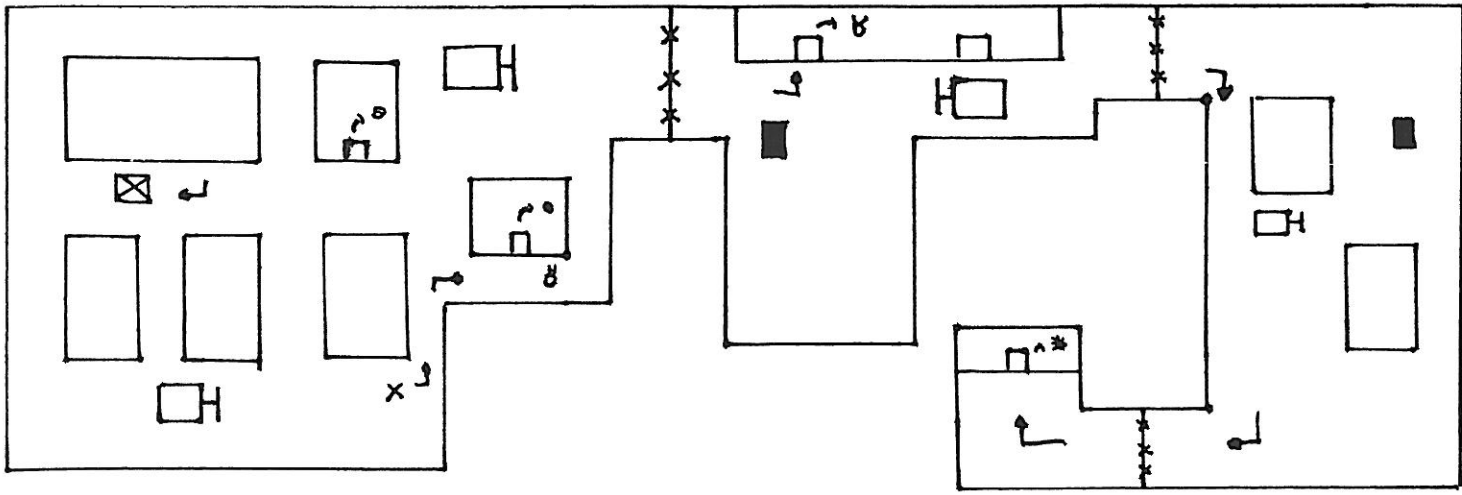
Niveau 2



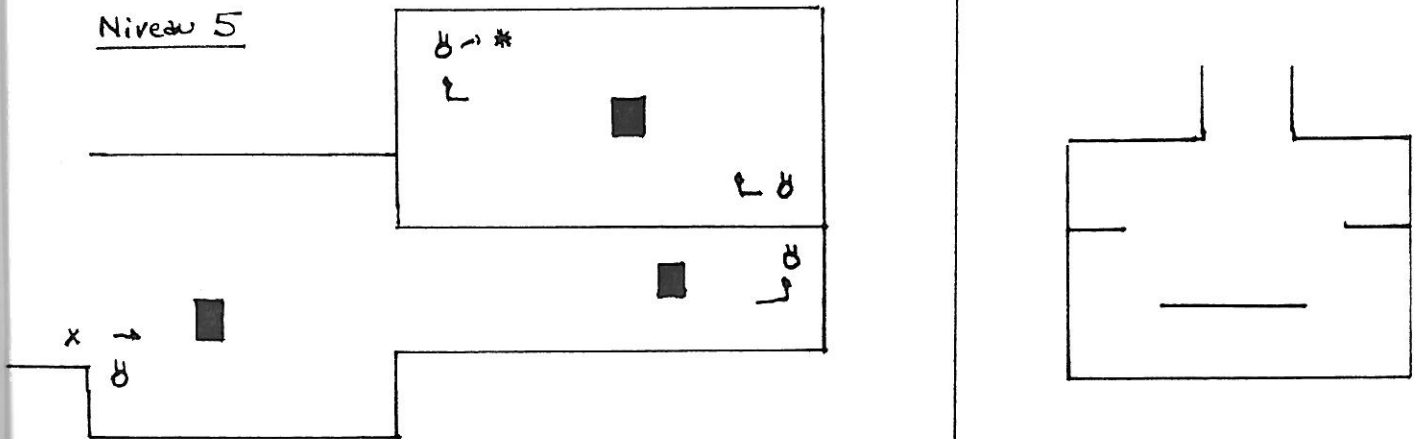
Niveau 3



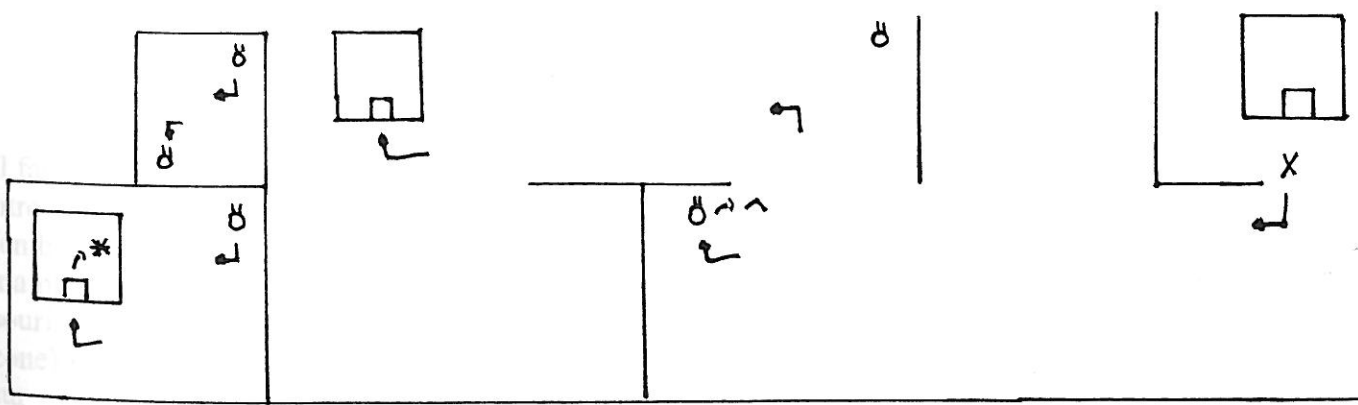
Niveau 4



Niveau 5



Niveau 6



Légende

- | | | |
|--------------------|---------------------|---------------------|
| X : début | ⊙ : pizza | □ : bombe |
| ♂ : bouche d'égout | ♂ : mouche | ⌋ : champ de force |
| ■ : monstre | ♂ : homme en flamme | ⊗ : van |
| ⊠ : bulldozer | ♀ : plante | ● : boule |
| | ♂ : grenouille | ♂ : corde |
| | | * : arrivée |
| | | ↳ : chemin à suivre |

Budget de trésorerie sur Excel

L'application qui sera développée ci-après a pour objet un Budget de trésorerie prévisionnel. Un budget fait état des entrées et des sorties d'argent afin de constater un éventuel déficit ou excédent. S'adressant à tous les utilisateurs (couples, étudiants...), même les plus novices, cette application peut être personnalisée en faisant varier la période étudiée (par semestre, ou sur toute l'année), et le nombre de postes souhaités (salaire, argent de poche...). Véronique.

	A	B	C	D	E	F	G	H	I
1	Budget de trésorerie prévisionnel 2ème semestre								
2									
3		juillet	août	septembre	octobre	novembre	décembre	TOTAL	%
4	Trésorerie début	1 320,00	2 933,50	3 998,50	5 359,50	4 387,00	4 247,50	1 320,00	
5									
6	Argent de poche	1 100,00	100,00	100,00	100,00	100,00	100,00	1 600,00	25,28
7	Salaire	1 320,00	1 320,00	1 320,00	670,00	0,00	0,00	4 630,00	72,91
8	Patrol Caisse d'épargne	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
9	Emprunts	0,00	120,00	0,00	0,00	0,00	0,00	120,00	1,89
10	Divers	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
11									
12	TOTAL	2 420,00	1 540,00	1 420,00	770,00	100,00	100,00	6 350,00	100,00
13									
14	Alimentation	85,00	66,00	0,00	88,00	75,00	80,00	368,00	10,03
15	Transport	31,50	0,00	31,50	94,50	94,50	94,50	346,50	9,01
16	Loyers	630,00	295,00	27,50	0,00	0,00	350,00	1 302,50	35,42
17	Chèques	500,00	500,00	0,00	0,00	0,00	0,00	1 000,00	15,43
18	Trains	400,00	250,00	0,00	0,00	0,00	0,00	650,00	17,00
19	Crus	200,00	50,00	25,00	0,00	0,00	50,00	325,00	8,42
20	Crédit Caisse d'ép	0,00	0,00	0,00	1 500,00	0,00	0,00	1 500,00	38,99
21	Remb emprunts	0,00	0,00	0,00	60,00	70,00	0,00	130,00	3,38
22	Divers	0,00	120,00	0,00	0,00	0,00	0,00	120,00	3,12
23									
24	TOTAL	806,50	475,00	59,00	1 742,50	230,50	524,50	3 847,00	100,00
25									
26	Solde	1 613,50	1 065,00	1 361,00	-372,50	-139,50	-424,50	2 503,00	
27									
28	Trésorerie fin	2 923,50	3 998,50	5 359,50	4 387,00	4 247,50	3 823,00	3 823,00	
29									

Le titre :

Il faut tout d'abord recopier le Titre ainsi que les titres de lignes (colonne A). Cliquez en A1 (endroit où se trouve le titre), puis tout en maintenant appuyé le bouton gauche de la souris, allez jusqu'en I1 (afin de définir une zone) et lâchez. Faites Format/ Position/Centrer sur plusieurs colonnes/OK. Le titre est maintenant centré. Pour l'agrandir : Format/Police/Police. Choisissez par exemple Times New Roman ; style de police : normal ; taille : 18.

Les mois :

Tapez Juillet en B3, déplacez le curseur de la souris jusqu'à obtenir une petite croix noire (en se plaçant en bas à droite de la cellule), puis cliquez et déplacez jusqu'à G3. Les mois sont ainsi incrémentés (Juillet à Décembre).

Les totaux :

En B2 : =somme(B2:B10), à recopier jusqu'en G12 suivant la même procédure que celle

étudiée auparavant. Il existe une formule analogue en B12.

En B24 : =somme(B14:B16):(B20:B22), à copier jusqu'en I24.

La cellule B16 regroupe le montant des cellules B17, B18, B19, celles-ci étant des sous-postes. La ponctuation ";" est un séparateur de zones. Cellule B16 : =somme(B17:B19), à copier jusqu'en I16.

Cellule B26 : =B12-B24. Le chiffre obtenu peut être négatif, puisqu'il s'agit du solde (Entrées-Sorties d'argent), à copier jusqu'en H26.

Cellule B28 : =B24+B26, à copier jusqu'en H28. Le résultat doit toujours être positif.

Cellule C4 : =B28. Dans la cellule C4, nous avons la trésorerie début du mois d'août, identique à la trésorerie fin du mois de Juillet. H4=B4

H6=somme(B6:G6), à copier jusqu'en H10.

H14 : formule analogue à H6(B14:G14), à copier jusqu'en H22.

Les pourcentages :

I6 : si(\$H\$12=0;0;H6/\$H\$12*100).

Si l'entrée d'argent équivaut à 0, le pourcentage est alors nul (sinon, on obtient une division par 0). Dans le cas contraire, lorsque H12 est positif, le calcul peut alors être effectué.

\$: indique une référence cellulaire absolue. Cela signifie que lors de la copie vers le bas, les numéros de cellules ne seront pas incrémentés.

Pour obtenir le pourcentage, effectuez le calcul suivant : (H6/\$H\$12*100)

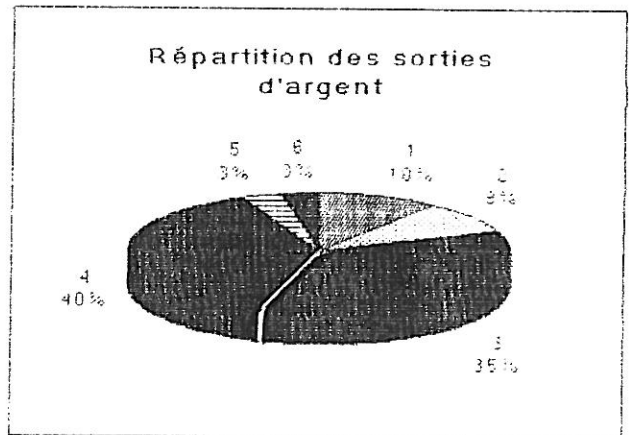
Mises en forme :

Pour centrer les mois : faire une zone de B3 à I3, Format/Position/Centrer horizontalement.

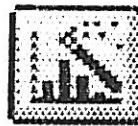
Pour une lecture plus aisée du tableau, faites une zone de B6 à H26, puis Format/Nombre : 2 décimales, séparateurs de milliers, et affichage des nombres en rouge lorsque ceux-ci sont négatifs (# ###, ##(Rouge))

Pour les pourcentages : établissez une zone de I6 en I24, faites Format nombre : 0, 00.

De A17 à I19 : faites Format/Police : on obtient une modification de la taille de l'écriture, qui est maintenant réduite.



Graphique



Afin de clarifier la lecture du tableau, il est conseillé d'accompagner le texte de graphiques. Pour se faire, définissez une zone dans la partie du tableau choisie. Cliquez sur l'icône "graphique" (se reporter à la figure dans ce paragraphe). La forme du curseur souris change (petite croix). Cliquez alors dans la feuille de calcul pour créer une zone où insérer le graphique. Confirmez la plage de données/OK. Choisissez le style de graphique (secteur 3D). Donnez un titre, une légende/OK. Le cadre autour du graphique est alors agrémenté de poignées qui servent à modifier sa taille. Il est préférable de ne pas conserver les couleurs proposées lorsque l'imprimante est noir et blanc. Pour apporter des modifications ; double-cliquez sur la couleur à changer; pour le premier plan : choisissez noir, et pour le motif il existe différents tramages.

Les Nouvelles du Monde Informatique

Sélection d'articles dans la presse spécialisée.

Les WC ne sont plus ce qu'ils étaient.

Cet été, pour les passagers des compagnies aériennes américaines, certains vols intérieurs ont été assez éprouvants. En effet, il est devenu de bon ton, grâce à son PC portable, de mettre à profit les heures de voyage pour continuer à travailler. Mais comme chacun sait, l'alimentation par piles des PC n'est pas tout à fait au point, et il faut recharger ses NiCad en cours de route. La solution ? Les prises pour rasoir électrique dans les toilettes, pardi ! C'est ainsi que les WC ont parfois été occupés pendant toute la durée du vol par des cadres en manque de batteries. De surcroît, plusieurs d'entre eux se sont branchés en même temps, via les rallonges éparpillées à travers la cabine. D'où un problème de taille pour les hôtes et les stewards, emmêlés dans les fils avec leurs chariots de boissons ! Le *San Francisco Chronicle*, journal réputé, rapporte également que le personnel aérien se plaint d'une "ruée vers l'arrière", car les cadres demandent maintenant les sièges les plus proches des WC, "au cas où les piles ne durent pas". Certaines compagnies ont réagi, cependant, en condamnant les prises en question et en expliquant pourquoi. Effet immédiat : les toilettes ont été libérées mais les cadres contraints de se passer de leurs rasoirs électriques. Sauf s'ils marchent aussi à piles, bien sûr !

Compatibles PC MAGAZINE n°61 - Décembre 1992 - p.56

Le champion Microsoft doit retirer un logiciel

Tricherie. Le numéro 1 mondial des logiciels, Microsoft, vient de se voir ordonner par un juge américain de retirer tous les logiciels DOS-6 du marché mondial : ce programme de compression de données sur disque dur a violé deux brevets appartenant à la société Star Electronics qui a obtenu d'un autre tribunal quelque 700 millions de francs de dommages et intérêts, jugement contre lequel Microsoft a fait appel.

Quest France (Normandie, Vendée, Mayenne), Juin 1994.

(En fait Microsoft ne doit retirer du marché qu'une partie de son système DOS 6 : DOUBLE SPACE)

Intel double ses SX

Intel reprend sa technologie de doublement de la fréquence d'horloge inaugurée avec les processeurs 486DX pour l'adapter aux processeurs 486SX. Le 486 SX2 25/50 est désormais disponible. Il assure un gain de performances de 30 % par rapport au processeur SX. Ce processeur signe le renouveau du choix pour la micro-informatique d'entrée de gamme.

Génération PC & WINDOWS n°13 - Mai 1994 - p.14

Un jeu pas assez hasardeux

Un expert en informatique a réussi par trois fois à décrocher le gros lot d'un jeu réputé de hasard au casino de Montreal. Ce jeu électronique, le Keno, consiste, pour le joueur qui parie entre 2 et 5 dollars, à déterminer le plus exactement possible une séquence de 20 chiffres que l'ordinateur retiendra sur un total de 80. Daniel Corriveau a gagné trois fois de suite, totalisant la somme de 620 000 dollars canadiens (environ 2,5 millions de francs). Soumis à une enquête de police, l'heureux parieur a révélé avoir découvert une « faille » dans le système informatique du Keno, après avoir analysé une séquence de 7000 jeux, l'appareil recommençait à émettre les mêmes combinaisons qu'il suffisait de rejouer au bon moment. Les enquêteurs n'ont rien trouvé d'illégal dans la combine du joueur qui pourra donc empocher ses gains. Mais Loto-Québec, l'organisme gouvernemental qui - comme en France - pompe davantage de sous qu'il n'en redistribue, a décidé de retirer purement et simplement le Keno en question. Et se réserve la possibilité de poursuivre la société américaine qui devait fournir un jeu sans faille.

Quest France (Normandie, Vendée, Mayenne), 20/06/94.

Programmation en Assembleur 8080

HELLO.COM et HELP.COM

Je ne proposerai pas d'autres programmes dans ce langage, tout d'abord parce que je n'utilise plus beaucoup mon Amstrad et ensuite parce que ce système est obsolète ce qui ne permet pas, à mon avis, de développer des applications assez puissantes. Les programmes, qui se nomment HELLO, et HELP, présentent de manière concise quelques instructions de ce langage qu'est l'assembleur 8080.

Historique

Ce langage devait assurer une compatibilité maximum entre toutes les machines possédant un microprocesseur Intel 8080 (commercialisé à partir de 1973) ou équivalent (8085 ou Z80). Cette compatibilité ne pouvait être effective que si les programmes développés respectaient certaines règles notamment en ce qui concerne la gestion des entrées/sorties des périphériques. En effet, ces derniers devaient être gérés par des appels à des fonctions BDOS, tout écart à cette méthode risquant d'entraîner des incompatibilités avec des ordinateurs de marque différente. Le système d'exploitation

utilisé était le CP/M de Digital Research. Sur les Amstrad CPC, on utilise le CPM v2.2 sur 464-664 et v.3.0 (ou CP/M Plus) sur 6128.

Démarches

Il faut d'abord saisir le code source qui est constitué de mnémoniques (pour exemple référez-vous aux listings qui figurent ci-après). Ce programme doit être édité à l'aide d'un programme capable de générer un fichier texte ASCII, on peut donc utiliser ED.COM fourni avec le CP/M, ou AMSWORD... Le fichier ainsi sauvegardé doit comporter l'extension '.ASM', par exemple : 'HELLO.ASM'.

Ensuite, on exécute l'assembleur qui permet de transformer le fichier '*.ASM' en un code hexadécimal. Pour se faire, on entre : 'ASM nomprog' (sans l'extension), par exemple 'ASM HELLO', et on obtient deux fichiers, HELLO.HEX (le fichier hexadécimal) et HELLO.PRN (fichier listable comprenant le source et de l'hexadécimal).

Enfin, pour rendre ce programme exécutable, on utilise le programme LOAD.COM sous la syntaxe suivante LOAD HELLO pour aboutir à l'exécutable HELLO.COM.

HELLO.COM

HELLO affiche un message de bienvenue, indique des informations utiles (pour formater une disquette par exemple) puis vous rend la main.

Pour que ce programme s'exécute à chaque mise en marche du CP/M, déprotégez votre disquette en écriture, chargez CP/M, lancez le programme SETUP. A l'affichage de «Command Initialisation», tapez «no» puis entrez «Help.com^M» puis frappez la touche

<entrée>, ensuite répondez «yes» à toutes les questions, y compris à «Do you want to restart CP/M ?» (Voulez-vous reinitialiser CP/M ?)

HELP.COM

Ce programme propose d'aider l'utilisateur, en lui présentant la plupart des commandes CP/M (utilité, syntaxe...). Toutes les commandes ne sont pas présentes, si vous avez le courage nécessaire vous pourrez toujours vous exercer à l'Assembleur pour les y ajouter. Benoît.

```

: HELLO v1.0 (c)1991 I.P.L. Informatique
:
: écrit par Benoit RIVIERE 14.12.91
:
:
: accès par CP/M : HELLO.COM
:
:
: INTERLOGISERV. il n'y'a qu'a demander.
:
:
BDOS: EQU      5
EFECR: EQU     0CH
AFF: EQU       9
CLVIN: EQU     1
CLVOUT: EQU    2
VIDINV: EQU   16H
BELL: EQU      7
ESC: EQU      GFCH
COPY: EQU     GECH
ENT: EQU      GDH
LF: EQU       0AH
:
:
: ORG          0100H
:
PRES: LXI      D,DEFENC
      MVI      C,AFF
      CALL     BDOS
      LXI      D,BVNSG
      MVI      C,AFF
      CALL     BDOS
      JMP      0
:
DEFENC: DB     28,'a','zz' : stylo blanc
        DB     28,'à','aa' : papier noir
        DB     29,'aa' : bord noir
        DB     EFECR,'s' : efface l'ecran
:
BVNSG: DB     31,31,1,'BIENVENU SOUS CP/M'
        DB     31,2,4,'HELLO v1.0 (c)1991 I.P.L. Informatique et Benoit RIVIERE'
        DB     31,21,5,'INTERLOGISERV. il n'.27H,'y a qu'.27H,'à demander'
        DB     31,7,3,'Pour obtenir de l'.27H,'aide. tapez HELP'
        DB     31,7,10,'Pour formater une disquette, tapez FORMAT '
        DB     31,1,13,'CP/M installé'
        DB     's'

```

```

; HELP v1.0 (c)1991 I.P.L. Informatique
;
; écrit par Benoit RIVIERE 14/12/91
;
;
; acces par CP/M : HELP.COM
;
;
; INTERLOGISERV, il n y'a qu'à demander.
;
;

```

```

BDOS: EQU 5
EFPECR: EQU 0CH
AFF: EQU 9
CLVIN: EQU 1
CLVOUT: EQU 2
VIDINV: EQU 1BH
BELL: EQU 7
ESC: EQU 0FCH
COPY: EQU 0EOH
ENT: EQU 0DH
LF: EQU 0AH
;

```

```

ORG 0100H
;

```

```

MENU: CALL PRES
LXI D,MENMS1
MVI C,AFF
CALL BDOS
;

```

```

DEM: LXI D,DEMMSG
MVI C,AFF
CALL BDOS
MVI C,CLVIN
CALL BDOS
CPI 'a'
JZ OPTA
CPI 'b'
JZ OPTB
CPI 'c'
JZ OPTC
CPI 'd'
JZ OPTD
CPI 'e'
JZ OPTE
CPI 'f'
JZ OPTF
CPI 'g'
JZ OPTG
CPI 'h'
JZ OPTH
CPI 'i'
JZ OPTI
CPI 'j'
JZ OPTJ
;

```

```

CPI 'k'
JZ OPTK
CPI 'l'
JZ OPTL
CPI ESC
JZ BYEBYE
;
MVI E,BELL
MVI C,CLVOUT
CALL BDOS
JMP DEM
;

```

```

OPTA: LXI D,OPAMSG
JMP AFFHLP
OPTB: LXI D,OPBMSG
JMP AFFHLP
OPTC: LXI D,OPCMMSG
JMP AFFHLP
OPTD: LXI D,OPDMMSG
JMP AFFHLP
OPTE: LXI D,OPENMSG
JMP AFFHLP
OPTF: LXI D,OPFMMSG
JMP AFFHLP
OPTG: LXI D,OPGMMSG
JMP AFFHLP
OPTH: LXI D,OPHMSG
JMP AFFHLP
OPTI: LXI D,OPIMMSG
JMP AFFHLP
OPTJ: LXI D,OPJMSG
JMP AFFHLP
OPTK: LXI D,OPKMSG
JMP AFFHLP
OPTL: LXI D,OPLMSG
JMP AFFHLP
;

```

```

AFFHLP: MVI C,AFF
CALL BDOS
;

```

```

CHOIX: LXI D,PHMSG
MVI C,AFF
CALL BDOS
MVI C,CLVIN
CALL BDOS
CPI COPY
JZ MENU
CPI ESC
JZ BYEBYE
MVI E,BELL
MVI C,CLVOUT
CALL BDOS
JMP CHOIX
;

```

```

BYEBYE: LXI D,BYEMSG
MVI C,AFF
CALL BDOS
JMP 0
;

```

```

PRES: LXI D,DEFENC
MVI C,AFF
CALL BDOS
LXI D,BVNMSG
MVI C,AFF
;

```

```

CALL      BDDS
RET

;
DEFMSG: DB      28,'a','zz' ; style blanc
DB      28,'a','aa'  ; papier noir
DB      29,'aa'     ; bord noir
DB      EFECR,'$'   ; efface l'écran
;

EVNMSG: DB      31,2,4,'HELP v1.0 (c)1991 I.P.L. Informatique et Benoit RIVIERE'
DB      31,20,5,'INTERLOGISERV. li n',27H,'y a qu',27H,'à demander'
DB      '$'
;

HENMSG1: DB      31,15,07,'AMSDOS...A ASM.....B BOOTGEN...C CLOAD....D'
DB      31,15,08,'CSAVE....D DDT.....E DIR.....F DISCCHK...F',ENT,LF
DB      '          DISCOPY..F DISKIT2..F '
DB      'DRUKEYS...G DUMP.....H'
DB      31,15,10,'ED.....I ERA.....J FILECOPY..F FORMAT2...F'
DB      31,15,11,'FWRESET...G HELP.....H LIB.....I LOAD.....J'
DB      31,15,12,'MAC.....K MOVCPM....L PIP.....M REN.....N'
DB      31,15,13,'RMAC.....O SETUP.....P SID.....Q STAT.....R'
DB      31,15,14,'SUBMIT.... SYSGEN....C TYPE..... XSUB..... '
DB      31,15,16,'Retour au CP/M.....<ESC>'
DB      '$'
;

DEMMSG: DB      31,15,20,'Choisissez une option :$'
;

PHMSG:  DE      31,25,21,VIDINV,' SUITE...<COPY>, FIN...<ESC> ',VIDINV
DB      '$'
;

BYMSG:  DE      ENT,LF,ENT,LF,ENT,LF,'HELP v1.0 terminé','$' ; retour au CP/M
;

OPMSG:  DE      EFECR,VIDINV,' RETOUR AU BASIC                          ',VIDINV
DB      31,08,3,'* AMSDOS : commande externe'
DB      31,10,04,'Syntaxe : AMSDOS'
DB      31,10,05,'Effet : provoque le retour au BASIC$'
OPBMSG: DE      EFECR,VIDINV,' ASSEMBLEUR 8080                          ',VIDINV
DB      31,08,3,'* ASM : commande externe'
DB      31,10,04,'Syntaxe : ASM nfic (fichier .ASM)'
DB      31,10,05,'Effet : assemble un fichier .ASM (assembleur 8080)'
DB      31,08,07,'* DDT : commande externe'
DB      31,10,08,'Syntaxe : DDT nfic',ENT,LF
DB      '          Effet : programme de mise au point assembleur'
DB      ENT,LF,ENT,LF
DB      '          * DUMP : commande externe'
DB      31,10,12,'Syntaxe : DUMP nfic.ext'
DB      31,10,13,'Effet : liste les fichiers sous forme hexadécimale'
DB      31,08,15,'* LOAD : commande externe'
DB      31,10,16,'Syntaxe : LOAD nfic'
DB      31,10,17,'Effet : rend exécutable un fichier assemblé$'
OPCMMSG: DE      EFECR,'$'
OPDMMSG: DE      EFECR,'$'
OPEMMSG: DE      EFECR,VIDINV,' GESTION DU LECTEUR DE DISQUETTE          ',VIDINV
DB      31,08,3,'* DIR : commande interne'
DB      31,10,4,'Syntaxe : DIR / DIR *.ext / DIR nfic.* / ...'
DB      31,10,5,'Effet : liste des fichiers'

```



```

DB      31,08,7,'* ERA : commande interne'
DB      31,10,8,'Syntaxe : ERA *.ext / ERA nfic.* / ERA nfic.ext /...'
DB      ENT,LF,'          Effet : efface des fichiers'
DB      31,08,11,'* REN : commande interne'
DB      31,10,12,'Syntaxe : REN nvnfic.ext=ancntfic.ext'
DB      31,10,13,'Effet : renomme un fichier'
DB      31,08,15,'* STAT : commande externe'
DB      31,10,16,'Syntaxe : STAT / STAT *.ext / STAT nfic.* /...'
DB      31,10,17,'Effet : donne des informations sur fichiers et'
DB      ' disquettes'

OPFMSG: DB      EFECR,VIDINV,' GESTION DES FICHIER          '.VIDINV
DB      31,08,03,'* DISCOPY : commande externe'
DB      31,10,04,'Syntaxe : DISCOPY'
DB      31,10,05,'Effet : copie une disquette sur une autre'
DB      31,08,07,'* $'

OPGMSG: DB      EFECR,VIDINV,' GESTION DE L',27H,'ECRAN          '
DB      VIDINV,31,08,03,'* DRLKEYS : commande externe'
DB      31,10,04,'Syntaxe : DRLKEYS'
DB      31,10,05,'Effet : efface l',27H,'écran en 40 colonnes'
DB      31,08,06,'* FWRESET : commande externe',ENT,LF
DB      '          Syntaxe : FWRESET'
DB      31,10,10,'Effet : efface l',27H,'écran en 80 colonnes$'
OPHMSG: DB      EFECR,VIDINV,' EDITION DE LIGNES          '.VIDINV
DB      31,08,03,'* ED : commande externe'
DB      31,10,04,'Syntaxe : ED nfic.ext'
DB      31,10,05,'Effet : édite des lignes (utilisation'
DB      ' totalement déconseillée)$'
OPIMSG: DB      EFECR,VIDINV,' AIDE A L',27H
DB      'UTILISATION DE CP/M ',VIDINV
DB      31,08,03,'* HELP : commande externe'
DB      31,10,04,'Syntaxe : HELP'
DB      31,10,05,'Effet : aide l',27H,'utilisateur de CP/M en lui'
DB      ' expliquant les principales          fonctions$'
OPJMSG: DB      EFECR,'$'
OPKMSG: DB      EFECR,'$'
OPLMSG: DB      EFECR,'$'

```

AU BANG DES ACCUSEES

BATMAN

Un soir, dans les rues malfamées de Gotham City, le Joker, accompagné de son infernal rictus, écume la ville de crimes... Mais, que vois-je au loin ? Une énorme chauve-souris nous survole, tombe sur le Joker, et en fait de la bouillie pour chat (un souvenir de Catwoman peut-être).

Voilà à peu près comment est le jeu. Entre "Batman" et le jeu "Robocop", vous avez la même qualité : il n'y a pas besoin de voir le film, vous le voyez sur l'écran de votre ordinateur.

Au **premier niveau**, vous êtes dans l'usine chimique AXIS où on vous lance des grenades (les salauds !!). Votre but est de trouver et de descendre Napier. L'originalité est qu'il tombe dans une cuve d'acide une fois que vous êtes parvenu à l'atteindre, comme dans le film.

Au **deuxième niveau**, vous conduisez la «Batmobile» dans les rues de Gotham, utilisez le grappin pour aller plus vite en tournant, mais, méfiez-vous quand même.

Au **niveau 3**, c'est amusant. Vous devez jouer avec des produits chimiques pour contrecarrer l'épidémie qu'a lancé le Joker (comme dans le film).

Au **niveau 4**, vous utilisez la «Batwing» contre les ballons du Joker. Si, au début, c'est assez facile, à la fin, utilisez les "droite-gauche" et les ombres au sol pour y arriver.

Au **niveau 5**, le dernier (Ouf !!), vous êtes dans la cathédrale. Faites attention aux tireurs fous (il y en a une quantité astronomique dans cette cathédrale) et, à la fin, lancez votre corde en diagonale pour faire tomber le Joker et vous avez enfin gagné. A.M.

Programmation en Turbo Pascal 6.0

TPDOSEXT.TPU

Turbo Pascal 6.0 est un langage très développé qui offre de nombreuses possibilités. Néanmoins, on constate que de nombreuses lacunes obligent les développeurs à créer leurs propres procédures. Ces lacunes sont souvent comblées à l'aide d'unités. L'unité que j'ai créée ajoute des procédures et fonctions dans de nombreux domaines (gestion graphique, fenêtres, souris, fichiers...).

TPDOSEXT.TPU

L'unité se nomme TPDOSEXT.TPU ce qui signifie : unité d'extension au Turbo Pascal sous MS/DOS. Je ne vais pas

expliquer ici toutes les extensions apportées à ce langage. Cependant, je m'y appliquerai dans un prochain numéro, tout comme je publierai les nombreuses améliorations qui seront apportées à cette unité.

Installation

Pour pouvoir utiliser cette unité dans vos programmes, il faut la compiler sur le disque dur dans un répertoire accessible au compilateur. Il vous suffira ensuite d'ajouter *TPDOSEXT* à la ligne *uses* de votre programme. (par exemple : *uses crt, graph, tpdosex;*). Benoît.

```
unit tpdosex; {v1.1}
{unité d'extension au Turbo Pascal 6.0}
{écrit par Benoît Rivière}
{v1.0 le 20/10/93}
{v1.1 le 28/12/93}
{(c)1994, 1993 I.P.L. Informatique}

interface

uses crt, graph, dos, printer;

{variables et constantes publiques}
var
  {variables graphiques}
  graphdriver : integer;
  graphmode : integer;
  graphpath : string[100];
  errorcode : integer; { erreurs graphiques }
  maxx,maxy,maxcol : integer;
```

```
varsys : array[1..15] of word;

dateInq : integer; {1 : frc, 2 : all, 3 : gb}

{variables souris}
existesouris : boolean;

{variables police}
ScriptFont : byte;
EuroFont : byte;
BoldFont : byte;
SimpFont : byte;
LcomFont : byte;
TSCRFont : byte;

settings : word;
```

```

const
  jourfr : array[0..6] of string[3] =
('LUN','MAR','MER','JEU','VEN','SAM','DIM');
  moisfr : array[0..11] of string[3] =
('JAN','FEV','MAR','AVR','MAI','JUN','JUL','A
OU','SEP','OCT','NOV','DEC');
  jourall : array[0..6] of string[3] =
('MON','DIE','MIT','DON','FRE','SAM','SON');
  moisall : array[0..11] of string[3] =
('JAN','FEB','MÄR','APR','MAI','JUN','JUL','A
UG','SEP','OKT','NOV','DEZ');
  jourgb : array[0..6] of string[3] =
('MON','TUE','WED','THI','FRI','SAT','SUN');
  moisgb : array[0..11] of string[3] =
('JAN','FEB','MAR','APR','MAY','JUN','JUL','
AUG','SEP','OCT','NOV','DEC');

{déclaration des procédures et fonctions
publiques}

{procédure système}
procedure DosShell;

{procédures et fonctions chaînes de
caractères}
function ChaineCapital(c : string) : string;
function Desaccentue(c : string) : string;
function LocaliseChaine(c,t : string) : boolean;
function RepCar(n : word; c : string) : string;
function Espace(n : word) : string;

{procédures graphiques et gestion de l'écran}
procedure Initialize;

procedure Posit(x,y : word);
procedure AffReliefXY(x,y,c1,c2 : integer; txt
: string);
procedure Centre (txt : string);

procedure CercleP(x,y,r,c1,c2 : integer);
procedure Triangle(x1,y1,x2,y2,x3,y3,c :
integer);
procedure TriangleP(x1,y1,x2,y2,x3,y3,c1,c2
: integer);

procedure Drapeau(n,x,y,lx,ly : integer);

{procédures relatives aux polices de
caractères}
procedure InitFont;
procedure TailleTexte (taille : word);

procedure DirectionTexte (dir : byte);
procedure Font (font : byte);

{procédure permettant la saisie de texte}
procedure ModifTexte (x,y : word; c1,c2,lg :
byte; t : string; var rep : string);
procedure EntreTexte (x,y : word; c1,c2,lg :
byte; var rep : string);

{procédures relatives aux fenêtres}
procedure Fen (x1,y1,x2,y2,cf,cb,ct : integer ;
titre : string);
procedure EffFen;
procedure FermFen;
procedure FenMessage(titre, msg, bout1,
bout2 : string ; var rep : byte);

{traitement des listes}
procedure AfficheListe(x,y,lx,ly : word ; c1 :
byte ;
      tab : pointer ; nb,position : word);

{procédures et fonctions fichier}
function ExisteFichier(nomfic : string) :
boolean;

{procédures et fonctions de l'objet souris}
type
  souris = object
    procedure Init;
    procedure Montre;
    procedure Cache;
    procedure LimiteX(x1,x2 : word);
    procedure LimiteY(y1,y2 : word);
    procedure Posit(sx,sy : word);
    function X : integer;
    function Y : integer;
    function Press : boolean;
    function Sorte : byte;
  end;

{procédures boutons}
procedure Bouton (n,x,y,cf,ct,ct2 : word; txt :
string);
procedure EffBouton (n : word);
procedure BoutonApp (n : word);

{procédures et fonctions boutons et souris}
function BoutonPress (n : word) : boolean;

{fonction relative à la gestion du clavier}

```

```

function AttTouche : char;

{procédure de gestion de l'imprimante}
procedure ImprimeGraph(xd,yd,xf,yf :
integer);

{procédures et fonctions mathématiques}
function RealToInt(n : real) : integer;

function Tan(x : real) : real;
function Sec(x : real) : real;
function Cosec(x : real) : real;
function Cotan(x : real) : real;
function ACos(x : real) : real;
function ASin(x : real) : real;
function ASec(x : real) : real;
function ACosec(x : real) : real;
function ACotan(x : real) : real;

function Cosh(x : real) : real;
function Sinh(x : real) : real;
function TanH(x : real) : real;
function SecH(x : real) : real;
function Cosech(x : real) : real;
function CotanH(x : real) : real;
function ACosh(x : real) : real;
function ASinh(x : real) : real;
function ATanh(x : real) : real;

function DegToRad(x : real) : real;
function RadToDeg(x : real) : real;

function Log10(x : real) : real;
function Exp10(x : real) : real;
function LogA(a : integer; x : real) : real;
function ExpA(a : integer; x : real) : real;

{procédures et fonctions relatives au
traitement de la date}
function AnneeBisextile(ann : word) :
boolean;
function NbJourMois(m : byte; ann : word) :
byte;
function NbSec(d1,d2 : string) : longint;
procedure AffDate;
procedure EffDate;

{procédures diverses}
function ExisteWindows : boolean;

implementation

```

```

{variables privées}
var
  {variables souris}
  soux,souy : word;

  {variables boutons}
  boutonc : array[0..15] of integer;
  boutonx : array[0..15] of integer;
  boutony : array[0..15] of integer;
  boutonl : array[0..15] of integer;

  {variable fenêtres}
  imag : pointer;
  ximag,yimag : integer;
  taillimag : integer;
  coulfondfen : byte;

{procédures et fonctions publiques}

{procédure système}
procedure dosshell;
var
  comspec : string;

begin
  textmode(co80);
  textbackground(0);
  textcolor(15);
  write('Tapez EXIT pour revenir. ');
  comspec:=getenv('COMSPEC');
  if comspec="" then
    comspec:='c:\COMMAND.COM';
  swapvectors;
  exec(comspec,"");
  swapvectors;
end; {dosshell}

{procédures et fonctions chaînes de
caractères}

function chaineCapital(c : string) : string;
var
  i : word;
begin
  for i:=1 to length(c) do c[i]:=upcase(c[i]);
  chaineCapital:=c;
end; {chaineCapital}

function desaccentue(c : string) : string;

```

```

var
  p : string;
  g : integer;

begin
  for g:=1 to length(c) do
  begin
    c[g]:=uppercase(c[g]);
    p:=c[g];
    if (p='à') or (p='â') or (p='ä') or (p='Ä') then
      c[g]:='A'
    else
      if (p='ç') then c[g]:='C'
      else
        if (p='é') or (p='è') or (p='ê') or (p='ë') then
          c[g]:='E'
        else
          if (p='ï') or (p='í') then c[g]:='I'
          else
            if (p='ô') or (p='ö') or (p='Ö') then c[g]:='O'
            else
              if (p='ù') or (p='û') or (p='ü') or (p='Û') then
                c[g]:='U';
              end;
            desaccentue:=c;
          end; {desaccentue}

function localisechaine(c,t : string) : boolean;
begin
  localisechaine:=pos(c,t)<>0;
end; {localisechaine}

function repcar(n : word; c : string) : string;
var
  i : integer;
  l : string;
begin
  l:="";
  for i:=1 to n do l:=l+c;
  repcar:=l;
end; {repcar}

function espace(n : word) : string;
begin
  espace:=repcar(n, ' ');
end; {espace}

```

{procédures graphiques}

{mise en place du mode graphique}
 {résolution 640x480, 16 couleurs}

```

procedure initialize;
begin
  graphdriver:=9;
  graphmode:=2;
  graphpath:="";

initgraph(graphdriver,graphmode,graphpath);
graphmode:=getgraphmode;
errorcode := graphresult; { erreur? }
if ErrorCode <> grOk then
begin
  clrscr;
  writeln('Erreur 1 : ',
  GraphErrorMsg(ErrorCode));
  halt(1);
end;
maxx:=getmaxx;
maxy:=getmaxy;
maxcol:=getmaxcolor;
end; {initialize}

procedure posit(x,y : word);
begin
  varsys[4]:=x*8;
  varsys[5]:=y*14;
  moveto(varsys[4],varsys[5]);
end; {posit}

procedure affreliefxy(x,y,c1,c2 : integer; txt :
string);
begin
  setcolor(c2);
  outtextxy(x+2,y+2,txt);
  setcolor(c1);
  outtextxy(x,y,txt);
end; {affreliefxy}

procedure centre (txt : string);
begin
  if varsys[2]=0 then posit(((varsys[9]-
varsys[7])-textwidth(txt)) div 2
div 8 div varsys[1],varsys[5] div 14)
  else posit(varsys[4] div 8,((varsys[10]-
varsys[8])-textwidth(txt)) div 2
div 14 div varsys[1]);
  outtext(txt);
end; {centre}

procedure cerclep(x,y,r,c1,c2 : integer);
begin

```

```

settings:=getcolor;
if c1<>-1 then setcolor(c1);
if c2<>-1 then setfillstyle(solidfill,c2);
fillellipse(x,y,r,r);
setcolor(settings);
end; {cerclep}

procedure triangle(x1,y1,x2,y2,x3,y3,c :
integer);
begin
settings:=getcolor;
if c<>-1 then setcolor(c);
line(x1,y1,x2,y2);
line(x2,y2,x3,y3);
line(x3,y3,x1,y1);
setcolor(settings);
end; {triangle}

procedure trianglep(x1,y1,x2,y2,x3,y3,c1,c2 :
integer);
var
triang : array[1..3] of pointtype;
begin
settings:=getcolor;
triang[1].x:=x1;
triang[1].y:=y1;
triang[2].x:=x2;
triang[2].y:=y2;
triang[3].x:=x3;
triang[3].y:=y3;
setcolor(c1);
setfillstyle(solidfill,c2);
fillpoly(sizeof(triang) div sizeof(pointtype),
triang);
setcolor(settings);
end; {trianglep}

procedure drapeau(n,x,y,lx,ly : integer);
var
c : integer;
g : integer;
polyg : array[1..6] of pointtype;
echx,echy : integer;
ecx,ecy : integer;
begin
settings:=getcolor;
if n=1 then
begin
lx:=lx div 3;
for g:=1 to 3 do

```

```

begin
if g=1 then c:=1 else if g=2 then c:=15
else c:=4;
setfillstyle(solidfill,c);
bar(x+lx*(g-1),y,x+lx*g,y+ly);
end;
end
else
if (n=2) or (n=5) then
begin
ly:=ly div 3;
for g:=1 to 3 do
begin
if n=2 then if g=1 then c:=0 else if g=2
then c:=4 else c:=14
else if g=1 then c:=4 else if g=2 then
c:=15 else c:=1;
setfillstyle(solidfill,c);
bar(x,y+ly*(g-1),x+lx,y+ly*g);
end;
setcolor(4);
if n=2 then rectangle(x,y,x+lx,y+ly);
end
else
if n=3 then
begin
setfillstyle(solidfill,1);
bar(x,y,x+lx,y+ly);
echx:=realtoint(int((3/17)*lx)/2);
echy:=realtoint(int((2/10)*ly)/2);
ecx:=echx;
ecy:=echy;
for g:=1 to 2 do
begin
if g=1 then c:=15 else c:=4;
setcolor(c);
setfillstyle(solidfill,c);
polyg[1].x:=x;
polyg[1].y:=y;
polyg[2].x:=x+ecx;
polyg[2].y:=y;
polyg[3].x:=x+lx;
polyg[3].y:=y+ly-ecy;
polyg[4].x:=x+lx;
polyg[4].y:=y+ly;
polyg[5].x:=x+lx-ecx;
polyg[5].y:=y+ly;
polyg[6].x:=x;
polyg[6].y:=y+ecy;
fillpoly(sizeof(polyg) div
sizeof(pointtype),polyg);

```

```

polyg[1].x:=x+lx;
polyg[1].y:=y;
polyg[2].x:=x+lx;
polyg[2].y:=y+ecy;
polyg[3].x:=x+ecx;
polyg[3].y:=y+ly;
polyg[4].x:=x;
polyg[4].y:=y+ly;
polyg[5].x:=x;
polyg[5].y:=y+ly-ecy;
polyg[6].x:=x+lx-ecx;
polyg[6].y:=y;
fillpoly(sizeof(polyg) div
sizeof(pointtype),polyg);

```

```

    ecx:=echx div 4;
    ecy:=echy div 4;
end;
ecx:=echx;
ecy:=echy;
for g:=1 to 2 do
begin
  if g=1 then c:=15 else c:=4;
  setcolor(c);
  setfillstyle(solidfill,c);
  bar(x+lx div 2-ecx,y,x+lx div 2+ecx,y+ly);
  bar(x,y+ly div 2-ecy,x+lx,y+ly div 2+ecy);
  ecx:=realtoint(echx/2);
  ecy:=realtoint(echy/2);
end;
end
else
if n=4 then
begin
  ly:=ly div 4;
  for g:=1 to 4 do
  begin
    if (g=1) or (g=4) then c:=4
    else c:=14;
    setfillstyle(solidfill,c);
    bar(x,y+ly*(g-1),x+lx,y+ly*g);
    echy:=echy+ly;
  end;
end
else
if n=6 then
begin
  setfillstyle(solidfill,4);
  bar(x,y,x+lx,y+ly);
  echx:=realtoint(int((4/25)*lx)/2);
  echy:=realtoint(int((4/16)*ly)/2);

```

```

    ecx:=echx;
    ecy:=echy;
    for g:=1 to 2 do
    begin
      if g=1 then c:=15 else c:=1;
      setcolor(c);
      setfillstyle(solidfill,c);
      bar(x+lx div 3-ecx,y,x+lx div 3+ecx,y+ly);
      bar(x,y+realtoint(ly/2)-
      ecy,x+lx,y+realtoint(ly/2)+ecy);
      ecx:=realtoint(echx/2);
      ecy:=realtoint(echy/2);
    end;
  end;
  setcolor(settings);
end; {drapeau}

```

```

{procédures relatives aux polices de
caractères}
procedure styletxt; {procédure privée}
begin
{ if not existefichier('goth.chr') then errdsk(4)
  else
  if not existefichier('litt.chr') then errdsk(4)
  else
  if not existefichier('sans.chr') then errdsk(4)
  else
  if not existefichier('trip.chr') then errdsk(4);}
  setttextstyle(varsyst[3],varsyst[2],varsyst[1]);
end; {styletxt}

```

```

procedure initfont;
begin
  varsyst[1]:=1;
  varsyst[2]:=0;
  varsyst[3]:=0;
  styletxt;
end; {initfont}

```

```

procedure tailletexte (taille : word);
begin
  varsyst[1]:=taille;
  styletxt;
end; {tailletexte}

```

```

procedure directiontexte (dir : byte);
begin
  varsyst[2]:=dir;
  styletxt;
end; {directiontexte}

```



```

procedure font (font : byte);
begin
  varsys[3]:=font;
  stylext;
end; {font}

{procédures permettant la saisie de texte}
procedure modifftexte (x,y : word; c1,c2,lg :
byte; t : string;
                    var rep : string);
{saisie de textes}
{x,y : coord., c1 : coul. texte, c2 : coul. points,
lg : longueur texte}
var
  fin : integer;
  pixel,i : integer;
  ch : char;

begin
  settings:=getcolor;
  if lg>255 then lg:=255;
  if length(t)>lg then delete(t,lg+1,length(t));
  fin:=0;
  {lit la couleur du fonds, et la met en couleur
de remplissage}
  pixel:=getpixel(x*8,y*14);

  setcolor(c2);
  posit(x,y);
  for i:=1 to lg do outtext('.');
  setcolor(c1);

  rep:="";
  if t<>"" then
  begin
    posit(x,y);
    outtext(t);
    i:=length(t);
    rep:=t;
  end
  else i:=0;

  repeat
    i:=i+1;
    if i<1 then i:=1;
    setcolor(c1);
    posit(x+i-1,y);
    outtext('_');
    setcolor(c2);
    posit(x+i-1,y);
    if i=lg then outtext(' ') else outtext('.');

```

```

repeat
  affdate;
until keypressed;
ch:=readkey;
if ch=char(8) then
begin
  i:=i-2;
  if length(rep)>0 then
  begin
    delete(rep,length(rep),1);
    setfillstyle(1,pixel);
    bar((x+i)*8*varsys[1],y*14*varsys[1],
((x+i)*8+8)*varsys[1],(y*14+14)*varsys[1]);
    bar((x+i+1)*8*varsys[1],y*14*varsys[1],
((x+i+1)*8+8)*varsys[1],(y*14+14)*varsys[1]);
    posit(x+i+1,y);
    setcolor(c2);
    if length(rep)=lg then outtext(' ') else
outtext('.');
    end;
  end
  else
  if ch=char(#13) then fin:=1
  else
  if ch=char(#27) then rep:=char(#27)
  else
  begin
    setcolor(c1);
    posit((x+i-1),y);
    setfillstyle(1,pixel);
    bar((x+i-1)*8*varsys[1],y*14*varsys[1],
((x+i-
1)*8+8)*varsys[1],(y*14+14)*varsys[1]);
    outtext(ch);
    rep:=rep+ch;
  end;
  if (rep=char(#27)) or (fin=1) then
  begin
    {efface le curseur}
    setfillstyle(1,pixel);
    bar((x+i-
1)*8*varsys[1],(y*14+7)*varsys[1],
((x+i-
1)*8+8)*varsys[1],(y*14+14)*varsys[1]);

    setcolor(settings);
    exit;
  end;
until i=lg;

```

```

setcolor(settings);
end; {modiftexte}

procedure entretex (x,y : word; c1,c2,lg :
byte; var rep : string);
var
  t : string;
begin
  t:= "";
  modiftexte(x,y,c1,c2,lg,t,rep);
end; {entretex}

{procedure relatives aux fenêtres}
procedure fenetre (x1,y1,x2,y2,cf,cb,ct :
integer ; titre : string ;
  typfen : byte); {procedure privée}
{cf : coul.fond, cb : coul.bord, ct : coul.titre,
  typfen : = 0 fenêtre normale, = 1 fenêtre
  message}
begin
  setviewport(0,0,maxx,maxy,clipon);
  settings:=getcolor;
  coulfondfen:=cf;

  taillimag:=imagesize(x1*8,y1*14,x2*8,y2*14);
  getmem(imag,taillimag);
  getimage(x1*8,y1*14,x2*8,y2*14,imag^);
  ximag:=x1*8;
  yimag:=y1*14;

  setfillstyle(1,cf);
  bar(x1*8,y1*14,x2*8,y2*14);

  setcolor(cb);
  rectangle(x1*8+3,y1*14+3,x2*8-3,y2*14-3);

  varsys[7]:=(x1+1)*8;
  varsys[9]:=(x2-1)*8;
  varsys[10]:=y2*14-7;

  if titre="" then
  begin
    varsys[8]:=y1*14+7;

  setviewport(varsys[7],varsys[8],varsys[9],varsys[10],clipon);
  end
  else
  begin
    if typfen=1 then
    begin

```

```

setfillstyle(solidfill,9);
bar(x1*8+9,y1*14+9,x1*8+19,y1*14+19);
setfillstyle(solidfill,2);
bar(x1*8+6,y1*14+6,x1*8+16,y1*14+16);
end;
if length(titre)*varsys[1]>x2-x1-2 then
  delete(titre,x2-x1-
2,length(titre)*varsys[1]-(x2-x1-2));
  setcolor(ct);
  moveto((x1+(x2-x1-length(titre)*varsys[1])
div 2)*8,(y1+1)*14-4);
  outtext(titre);
  setcolor(cb);
  line(x1*8+3,y1*14+7*varsys[1]+14,x2*8-
3,y1*14+7*varsys[1]+14);
  varsys[8]:=y1*14+7*varsys[1]+22;

setviewport(varsys[7],varsys[8],varsys[9],varsys[10],clipon);
end;
setcolor(settings);
end; {fen}

procedure fen (x1,y1,x2,y2,cf,cb,ct : integer ;
titre : string);
{cf : coul.fond, cb : coul.bord, ct : coul.titre}
begin
  fenetre(x1,y1,x2,y2,cf,cb,ct,titre,0);
end; {fen}

procedure efffen;
begin
  setfillstyle(solidfill,coulfondfen);
  bar(0,0,varsys[9]-varsys[7],varsys[10]-
varsys[8]);
end; {efffen}

procedure fermfen;
begin
  varsys[7]:=0;
  varsys[8]:=0;
  varsys[9]:=maxx;
  varsys[10]:=maxy;

setviewport(varsys[7],varsys[8],varsys[9],varsys[10],clipon);
  putimage(ximag,yimag,imag^,normalput);
  freemem(imag,taillimag);
end; {fermfen}

```

```

procedure fenmessage(titre, msg, bout1,
bout2 : string; var rep : byte);
{rep : = 1->bouton 1, = 2->bouton 2, = 3-
>bouton 'annuler', <Echap>...}
var
  i          : word;
  x1,x2,y1,y2,lgx,lgbout : word;
  t          : array[1..2] of string[1];
  ch         : char;

function boutonquittepress : boolean;
var
  mouse : souris;
begin
  if (mouse.x>-5) and (mouse.x<20) and
(mouse.y>-26) and (mouse.y<0) and
  mouse.press then
    boutonquittepress:=TRUE
  else
    boutonquittepress:=FALSE;
end; {boutonquittepress}

procedure boutonquitteapp;
var
  img : pointer;
  cf,co : byte;
begin
  settings:=getcolor;

  setviewport(x1*8,y1*14,x2*8,y2*14,clipon);
  {récupère la couleur du fonds et de l'ombre
du bouton}
  cf:=getpixel(1,1);
  co:=getpixel(18,18);

  getmem(img,imagesize(6,6,16,16));
  getimage(6,6,16,16,img^);

  setfillstyle(1,cf);
  bar(6,6,16,16);
  putimage(9,9,img^,normalput);

  delay(250);

  setfillstyle(1,co);
  bar(9,9,19,19);
  putimage(6,6,img^,normalput);
  freemem(img,imagesize(6,6,16,16));

  delay(100);

```

```

setcolor(settings);
end; {boutonquitteapp}

begin
  settings:=getcolor;
  if length(msg)>60 then
    delete(msg,61,length(msg));
  if length(titre)>40 then
    delete(titre,41,length(titre));
  lgx:=length(msg);
  if length(bout1)>20 then
    delete(titre,21,length(bout1));
  if length(bout2)>20 then
    delete(titre,21,length(bout2));

  lgbout:=length(bout1)+4+2+length(bout2)+4+
2+7+4+2;
  if lgbout>length(msg) then
    lgx:=lgbout;
  lgx:=lgx+4;
  x1:=((maxx-lgx*8) div 2) div 8;
  x2:=x1+lgx;
  y1:=(maxy div 14-10) div 2;
  y2:=y1+10;
  if titre="" then
    begin
      y1:=y1+1;
      y2:=y2-1;
    end;
  fenetre(x1,y1,x2,y2,1,9,4,titre,1);
  affreliefxy((lgx-length(msg))*8 div 2-
8,25,15.8,msg);
  bouton(1,2+(lgx-lgbout) div
2,5.2,11,9,bout1);
  bouton(2,2+(lgx-lgbout) div
2+length(bout1)+4,5.2,11,9,bout2);
  bouton(3,2+(lgx-lgbout) div
2+length(bout1)+4+length(bout2)+4,5.
2,11,9,'~annuler');
  t[1]:=upcase(bout1[pos('~',bout1)+1]);
  t[2]:=upcase(bout2[pos('~',bout2)+1]);
  repeat
    ch:=upcase(afftouche);
  until (ch=t[1]) or boutonpress(1) or
(ch=t[2]) or boutonpress(2) or
(ch='A') or (ch=#27) or
(boutonquittepress and (titre<>"")) or
  boutonpress(3);
  if (ch=t[1]) or boutonpress(1) then
    begin
      boutonapp(1);

```

```

rep:=1;
end
else
if (ch=t[2]) or boutonpress(2) then
begin
boutonapp(2);
rep:=2;
end
else
if (ch='A') or (ch=#27) or boutonquittepress
or boutonpress(3) then
begin
if boutonquittepress then boutonquitteapp
else
boutonapp(3);
rep:=3;
end;
for i:=1 to 3 do effbouton(i);
fermfen;
setcolor(settings);
end; {fenmessage}

```

{traitement des listes}

```

procedure afficheliste(x,y,lx,ly : word ; c1 :
byte ;

```

```

tab : pointer ; nb,position : word);

```

```

var

```

```

p : pointer;
i,j : integer;
deb,fin : word;
pasass : word;
mouse : souris;

```

```

procedure fleche(i,j : integer);

```

```

begin

```

```

if i=0 then setcolor(9)

```

```

else

```

```

setcolor(2);

```

```

if j=1 then moveto((x+lx)*8-6-2*i,y*14+3-
2*i)

```

```

else moveto((x+lx)*8-6-2*i,(y+ly)*14-1-
2*i);

```

```

linerel(6,6*j);

```

```

linerel(-4,0);

```

```

linerel(0,7*j);

```

```

linerel(-4,0);

```

```

linerel(0,-7*j);

```

```

linerel(-4,0);

```

```

linerel(6,-6*j);

```

```

if i=0 then

```

```

begin

```

```

setfillstyle(solidfill,9);

```

```

if j=1 then floodfill((x+lx)*8-6-2*i,y*14+5,9)

```

```

else floodfill((x+lx)*8-6-2*i,(y+ly)*14-4,9)

```

```

end

```

```

else

```

```

begin

```

```

setfillstyle(solidfill,2);

```

```

if j=1 then floodfill((x+lx)*8-6-2*i,y*14+5,2)

```

```

else floodfill((x+lx)*8-6-2*i,(y+ly)*14-4,2)

```

```

end;

```

```

end; {fleche}

```

```

function ascenspress : byte;

```

```

{n : = 0 pas ascenseur = 1, flèche haut, = 2
flèche bas,

```

```

= 3 curseur ascenseur, = 4 zone ascenseur
haut, = 5 zone ascenseur bas}

```

```

var

```

```

mouse : souris;

```

```

begin

```

```

if (mouse.x>(x+lx-2)*8) and
(mouse.x<(x+lx)*8) and mouse.press then

```

```

begin

```

```

position:=position-1;

```

```

if (mouse.y>y*14-2) and

```

```

(mouse.y<y*14+18) then ascenspress:=1

```

```

else

```

```

if (mouse.y>(y+ly)*14-18) and

```

```

(mouse.y<(y+ly)*14+2) then ascenspress:=2

```

```

else

```

```

if (mouse.y>y*14+18+(ly div
pasass)*position)

```

```

and (mouse.y<y*14+26+(ly div
pasass)*position) then ascenspress:=3

```

```

else

```

```

if (mouse.y>y*14+26+(ly div
pasass)*position) and (mouse.y<(y+ly)*14-
18)

```

```

then ascenspress:=4

```

```

else

```

```

if (mouse.y>y*14+18) and

```

```

(mouse.y<y*14+17+(ly div pasass)*position)

```

```

then ascenspress:=5;

```

```

position:=position+1;

```

```

end;

```

```

end; {ascenspress}

```

```

procedure ascensapp(n : byte);

```

```

var

```

```

img : pointer;

```

```

cf,co : byte;

```

```

begin
  settings:=getcolor;

{
setviewport(x1*8,y1*14,x2*8,y2*14,clipon);}
  {récupère la couleur du fonds et de l'ombre
du bouton}
  cf:=getpixel(1,1);
  co:=getpixel(18,18);

  getmem(img,imagesize(6,6,16,16));
  getimage(6,6,16,16,img^);

  setfillstyle(1,cf);
  bar(6,6,16,16);
  putimage(9,9,img^,normalput);

  delay(250);

  setfillstyle(1,co);
  bar(9,9,19,19);
  putimage(6,6,img^,normalput);
  freemem(img,imagesize(6,6,16,16));

  delay(100);

  setcolor(settings);
end; {ascensapp}

procedure affass(position,pasass : word);
{affiche l'ascenseur}
begin
  mouse.cache;
  setfillstyle(closedotfill,15);
  bar((x+lx-
2)*8+1,y*14+19,(x+lx)*8+1,(y+ly)*14-19);
  setfillstyle(solidfill,2);
  setcolor(15);
  position:=position-1;
  ly:=ly-1;
{  rectangle((x+lx-2)*8,y*14+18+(ly div
pasass)*position,
  (x+lx)*8+2,y*14+26+(ly div
pasass)*position);
  bar((x+lx-2)*8+1,y*14+19+(ly div
pasass)*position,
  (x+lx)*8+1,y*14+25+(ly div
pasass)*position);}
  rectangle((x+lx-
2)*8,y*14+18+(position*pasass)*6,

```

```

(x+lx)*8+2,y*14+26+(position*pasass)*6);
  bar((x+lx-
2)*8+1,y*14+19+(position*pasass)*6,
(x+lx)*8+1,y*14+25+(position*pasass)*6);
  ly:=ly+1;
  mouse.montre;
  delay(250);
end; {affass}

```

```

begin
{allocation de la mémoire dans le tas}
getmem(p,sizeof(tab));

```

```

settings:=getcolor;
initfont;

```

```

if lx<8 then lx:=8;
if ly<8 then ly:=5;

```

```

rectangle(x*8-2,y*14-
2,(x+lx)*8+2,(y+ly)*14+2);

```

```

if position<1 then position:=1;
if position>nb then position:=nb;

```

```

deb:=position;
fin:=deb+ly-1;
if fin>nb then fin:=nb;

```

```

pasass:=nb div ly;

```

```

if nb>ly then
begin
  line((x+lx-2)*8,y*14-2,(x+lx-
2)*8,(y+ly)*14+2);
  line((x+lx-
2)*8,y*14+18,(x+lx)*8+2,y*14+18);
  line((x+lx-2)*8,(y+ly)*14-
18,(x+lx)*8+2,(y+ly)*14-18);

```

```

{flèche haut et bas}

```

```

for j:=-1 to 1 do

```

```

begin

```

```

if j=0 then j:=1;

```

```

for i:=0 to 1 do fleche(i,j);

```

```

end;

```

```

affass(position,pasass);

```

```

end;

```

```

for i:=deb to fin do
begin
  posit(x,y+i-1);
  { outtext(tab[i][lx]);}
  {si l'ascenseur et présent retirer 2 caractère à
  tab[i] pour l'affichage}
  {si i=position alors changer couleur du texte}
  end;
  mouse.init;
  mouse.montre;

  {uniquement si nb>ly}
  repeat
  if ascenspress=1 then
  begin
    position:=position-1;
    if position<1 then position:=1;
    affass(position,pasass);
  end
  else
  if ascenspress=2 then
  begin
    position:=position+1;
    if position>nb then position:=nb;
    affass(position,pasass);
  end
  else
  if ascenspress=4 then
  begin
    position:=position+pasass;
    if position>nb then position:=nb;
    affass(position,pasass);
  end
  else
  if ascenspress=5 then
  begin
    position:=position-pasass;
    if position<1 then position:=1;
    affass(position,pasass);
  end;

until keypressed;

mouse.cache;

{libération en fin de travail}
freemem(p,sizeof(tab));

setcolor(settings);
end; {afficheliste}

```

```

{procédures fichier}
fonction existefichier(nomfic : string) :
boolean;
var
  f : file;
begin
  {$I-}
  assign(f,nomfic);
  reset(f);
  close(f);
  {$I+}
  existefichier:=(IOResult = 0) and
  (nomfic<>"");
end; {existefichier}

```

```

{procédures souris}
procedure souris.init;
var
  reg : registers;
begin
  reg.ax:=$0;
  intr($33,reg);
  existesouris:=(reg.ax<>0);
  if not existesouris then exit;
  reg.ax:=7;
  reg.cx:=0;
  reg.dx:=getmaxx;
  intr($33,reg);
  reg.ax:=8;
  reg.cx:=0;
  reg.dx:=getmaxy;
  intr($33,reg);
end; {souris.init}

```

```

procedure souris.montre;
var
  reg : registers;
begin
  if not existesouris then exit;
  reg.ax:=$1;
  intr($33,reg);
end; {souris.montre}

```

```

procedure souris.cache;
var
  reg : registers;
begin
  if not existesouris then exit;
  reg.ax:=$2;
  intr($33,reg);

```

```

end; {souris.cache}

procedure souris.limitex(x1,x2 : word);
var
  view : viewporttype;
  reg : registers;
begin
  if not existesouris then exit;
  reg.ax:=$7;
  reg.cx:=x1;
  reg.dx:=x2;
  intr($33,reg);
end; {souris.limitex}

```

```

procedure souris.limitey(y1,y2 : word);
var
  view : viewporttype;
  reg : registers;
begin
  if not existesouris then exit;
  reg.ax=$8;
  reg.cx:=y1;
  reg.dx:=y2;
  intr($33,reg);
end; {souris.limitey}

```

```

procedure souris.posit(sx,sy : word);
var
  view : viewporttype;
  reg : registers;
begin
  if not existesouris then exit;
  getviewsettings(view);
  reg.ax=$4;
  reg.cx:=sx;
  reg.dx:=sy;
  intr($33,reg);
end; {souris.posit}

```

```

function souris.x : integer;
var
  view : viewporttype;
  reg : registers;
begin
  if not existesouris then exit;
  getviewsettings(view);
  reg.ax=$3;
  intr($33,reg);
  x:=reg.cx-view.x1;
end; {souris.x}

```

```

function souris.y : integer;
var
  view : viewporttype;
  reg : registers;
begin
  if not existesouris then exit;
  getviewsettings(view);
  reg.ax=$3;
  intr($33,reg);
  y:=reg.dx-view.y1;
end; {souris.y}

```

```

function souris.press : boolean;
var
  view : viewporttype;
  reg : registers;
begin
  if not existesouris then
  begin
    press:=FALSE;
    exit;
  end;
  getviewsettings(view);
  reg.ax:=5;
  reg.bx:=0;
  intr($33,reg);
  press:=reg.ax<>0;
  soux:=souris.x;
  souy:=souris.y;
end; {souris.press}

```

```

function souris.sorte : byte;
var reg : registers;
begin
  if not existesouris then exit;
  reg.ax:=36;
  intr($33,reg);
  sorte:=reg.cx;
end; {souris.sorte}

```

{procédures relatives aux boutons}

```

procedure bouton (n,x,y,cf.cf,ct2 : word; txt :
string);
var
  lg : integer;
  i : integer;

begin
  settings:=getcolor;

```

```

settextstyle(defaultfont,0,1);
boutonc[n]:=getpixel(x*8,y*14);
boutonx[n]:=x;
boutony[n]:=y;
lg:=length(txt);

if pos('~',txt)<>0 then lg:=lg-1;
boutonl[n]:=lg+2;
setfillstyle(1,ct2);
bar(x*8+4,y*14,(x+2+lg)*8+4,y*14+14);
setfillstyle(1,cf);
bar(x*8,y*14-4,(x+2+lg)*8,y*14+10);
if pos('~',txt)<>0 then lg:=lg+1;

posit(x+1,y);
setcolor(ct2);
if pos('~',txt)<>0 then
begin
  if pos('~',txt)<>1 then
  for i:=1 to pos('~',txt)-1 do
    if i=1 then outtext(uppercase(txt[1]))
    else
      outtext(txt[i]);
  setcolor(ct);
  if pos('~',txt)<>1 then
outtext(txt[pos('~',txt)+1])
  else outtext(uppercase(txt[pos('~',txt)+1]));
  setcolor(ct2);
  for i:=pos('~',txt)+2 to lg do outtext(txt[i]);
end
  else
begin
  outtext(uppercase(txt[1]));
  for i:=2 to lg do outtext(txt[i]);
end;

setcolor(settings);
end; {bouton}

procedure effbouton (n : word);
begin
  setfillstyle(1,boutonc[n]);
  bar(boutonx[n]*8-2,boutony[n]*14-4,

(boutonx[n]+boutonl[n])*8+6,boutony[n]*14+1
4);
end; {effbouton}

procedure boutonapp (n : word);
var

```

```

  img : pointer;
  c : word;
begin
  settings:=getcolor;

  {récupère la couleur de l'ombre du bouton}

  c:=getpixel(boutonx[n]*8+5,boutony[n]*14+13
);

  getmem(img,imagesize(boutonx[n]*8,boutony
[n]*14-4,

(boutonx[n]+boutonl[n])*8,boutony[n]*14+10))
;
  getimage(boutonx[n]*8,boutony[n]*14-4,

(boutonx[n]+boutonl[n])*8,boutony[n]*14+10,i
mg^);

  setfillstyle(1,boutonc[n]);
  bar(boutonx[n]*8,boutony[n]*14-4,

(boutonx[n]+boutonl[n])*8,boutony[n]*14+10);

  putimage(boutonx[n]*8+4,boutony[n]*14,img^
.normalput);

  delay(250);

  setfillstyle(1,c);
  bar(boutonx[n]*8+4,boutony[n]*14,

(boutonx[n]+boutonl[n])*8+4,boutony[n]*14+1
4);
  putimage(boutonx[n]*8,boutony[n]*14-
4,img^,normalput);

  freemem(img,imagesize(boutonx[n]*8,bouton
y[n]*14,

(boutonx[n]+boutonl[n])*8,boutony[n]*14+14))
;

  delay(100);

  setcolor(settings);
end; {boutonapp}

{procédures et fonctions bouton et souris}

```



```

function boutonpress (n : word) : boolean;
var
  mouse : souris;
begin
  if (soux>boutonx[n]*8-4) and
(soux<(boutonx[n]+boutonl[n])*8+4)
  and (souy>boutony[n]*14-8) and
(souy<boutony[n]*14+18)
  and mouse.press then
  boutonpress:=TRUE
  else
  boutonpress:=FALSE;
end; {boutonpress}

```

{procédure relative à la destion du clavier}

```

function atttouche : char;
var
  mouse : souris;
  ch : char;
begin
  mouse.montre;
  ch:=' ';
  repeat
    if varsys[6]=1 then affdate;
    delay(200);
  until keypressed or mouse.press;
  if mouse.press then ch:='U' {ALT+233}
  else
  ch:=readkey;
  mouse.cache;
  atttouche:=ch;
end; {atttouche}

```

{procédure de gestion de l'imprimante}

```

procedure imprimegraph(xd,yd,xf,yf :
integer);
begin
end; {imprimegraph}

```

{procédures mathématiques}

function realtoint(n : real) : integer;

{convertit un réel en entier}

```

var
  buf : string;
  buf2 : integer;
  err : integer;
begin
  str(n:0:0,buf);
  val(buf,buf2,err);

```

```

  realtoint:=buf2;
end; {realtoint}

```

function Tan(x : real) : real;

```

begin
  tan:=sin(x)/cos(x);
end; {Tan}

```

function Sec(x : real) : real;

```

begin
  sec:=1/cos(x);
end; {Sec}

```

function Cosec(x : real) : real;

```

begin
  cosec:=1/sin(x);
end; {Cosec}

```

function Cotan(x : real) : real;

```

begin
  cotan:=1/tan(x);
end; {Cotan}

```

function ACos(x : real) : real;

```

begin
  acos:=arctan(sqrt(1-sqr(x))/x);
end; {ACos}

```

function ASin(x : real) : real;

```

begin
  asin:=arctan(x/sqrt(1-sqr(x)));
end; {ASin}

```

function ASec(x : real) : real;

```

begin
  asec:=arctan(sqrt(sqr(x)-1));
end; {ASec}

```

function ACosec(x : real) : real;

```

begin
  acosec:=arctan(1/sqrt(sqr(x)-1));
end; {ACosec}

```

function ACotan(x : real) : real;

```

begin
  acotan:=arctan(1/x);
end; {ACotan}

```

function Cosh(x : real) : real;

```

begin

```

```
cosh:=(exp(x)+exp(-x))/2;
end; {Cosh}
```

```
function SinH(x : real) : real;
begin
  sinh:=(exp(x)-exp(-x))/2;
end; {SinH}
```

```
function TanH(x : real) : real;
begin
  tanh:=(exp(x)-exp(-x))/(exp(x)+exp(-x));
end; {TanH}
```

```
function SecH(x : real) : real;
begin
  sech:=1/cosh(x);
end; {SecH}
```

```
function Cosech(x : real) : real;
begin
  cosech:=1/sinh(x);
end; {Cosech}
```

```
function CotanH(x : real) : real;
begin
  cotanh:=1/sech(x);
end; {CotanH}
```

```
function ACosh(x : real) : real;
begin
  acosh:=ln(x+sqrt(sqr(x)-1));
end; {ACosh}
```

```
function ASinh(x : real) : real;
begin
  asinh:=ln(x+sqrt(sqr(x)+1));
end; {ASinh}
```

```
function ATanh(x : real) : real;
begin
  atanh:=ln((1+x)/(1-x))/2;
end; {ATanh}
```

```
function DegToRad(x : real) : real;
begin
  degtorad:=x*pi/180;
end; {DegToRad}
```

```
function RadToDeg(x : real) : real;
begin
```

```
radtodeg:=x*180/pi;
end; {RadTodeg}
```

```
function Log10(x : real) : real;
begin
  log10:=ln(x)/ln(10);
end; {Log10}
```

```
function Exp10(x : real) : real;
begin
  exp10:=exp(x*ln(10));
end; {Exp10}
```

```
function LogA(a : integer; x : real) : real;
begin
  loga:=ln(x)/ln(a);
end; {LogA}
```

```
function ExpA(a : integer; x : real) : real;
begin
  expa:=exp(x*ln(a));
end; {ExpA}
```

{procédures et fonctions relatives au traitement de la date}

```
function anneebisextile(ann : word) : boolean;
begin
  if ann div 4=ann/4 then
    anneebisextile:=TRUE else
    anneebisextile:=FALSE;
end; {anneebisextile}
```

```
function nbjourmois(m : byte; ann : word) :
byte;
const
  moi : array[1..12] of byte =
(31,28,31,30,31,30,31,31,30,31,30,31);
begin
  nbjourmois:=moi[m];
  if (m=2) and (anneebisextile(ann)) then
    nbjourmois:=29;
end; {nbjourmois}
```

```
function ecartdate(d : string) : longint;
{fonction privée}
var
  lim1,lim2,jou,moi,ann,heu,min,sec : longint;
  tps : longint;
  err : integer;
```

```

begin
  val(d[1]+d[2],jou,err);
  val(d[3]+d[4],moi,err);
  val(d[5]+d[6]+d[7]+d[8],ann,err);
  val(d[9]+d[10],heu,err);
  val(d[11]+d[12],min,err);
  val(d[13]+d[14],sec,err);
  tps:=sec+min*60+heu*3600+(jou-
1)*24*3600;

  for err:=1 to moi-1 do
    tps:=tps+nbjourmois(err,ann)*24*3600;
    if ann=1973 then
      begin
        lim1:=0;
        lim2:=0;
      end
    else
      if ann>1973 then
        begin
          lim1:=1973;
          lim2:=ann-1;
        end
      else
        begin
          lim1:=ann;
          lim2:=1973-1;
        end;
      end;

    for err:=lim1 to lim2 do
      begin
        tps:=tps+365*24*3600;
        if anneebisextile(err) then
          tps:=tps+1*24*3600;
        end;
        ecartdate:=tps;
      end; {ecartdate}

    function nbsec(d1,d2 : string) : longint;
    begin
      nbsec:=ecartdate(d1)-ecartdate(d2);
    end; {nbsec}

  procedure affdate;
  var
    d,j,m,a,heu,min,sec,sec2 : word;
    j2,a2,heu2,min2,sec22 : string[10];

  begin
    {capture de l'heure et de la date}
    getdate(a,m,j,d);

```

```

    gettime(heu,min,sec,sec2);

    if varsys[11]=heu*3600+min*60+sec then
      exit;
    varsys[11]:=heu*3600+min*60+sec;

    initfont;
    settings:=getcolor;
    varsys[6]:=1;
    if (varsys[9]-varsys[7]<>maxx) or
      (varsys[10]-varsys[8]<>maxy) then
      setviewport(0,0,maxx,maxy,clipon);

    setfillstyle(1,0);
    bar(488,10,504,0);
    if sec=0 then
      begin
        bar(464,10,480,0);
        if min=0 then
          begin
            bar(440,10,456,0);
            if heu=0 then bar(512,10,640,0);
          end;
        end;
      end;

    {formatage et transfert en chaine de la date
    et de l'heure}
    d:=d-1;
    if d>6 then d:=6;
    m:=m-1;

    str(j,j2);
    if (j=1) and (datelng=1) then j2:=j2+'er';
    str(a,a2);
    str(heu,heu2);
    if heu<10 then heu2:=''+heu2;
    str(min,min2);
    if min<10 then min2:='0'+min2;
    str(sec,sec22);
    if sec<10 then sec22:='0'+sec22;

    {affichage de l'heure et de la date}
    setcolor(15);
    posit(55,0);
    outtext(heu2+'.'+min2+'.'+sec22+' ');
    if datelng=1 then outtext(jourfr[d]+''+j2+'
'+moisfr[m]+''+a2)
      else if datelng=2 then outtext(jourall[d]+'
'+j2+'.'+moisall[m]+''+a2)
        else outtext(jourgb[d]+''+j2+'
'+moisgb[m]+''+a2);

```

```

setcolor(settings);

setviewport(vars[7],vars[8],vars[9],vars[10],clipon);
end; {affdate}

procedure effdate;
begin
  vars[6]:=0;

  if (vars[9]-vars[7]<>maxx) or
  (vars[10]-vars[8]<>maxy) then
    setviewport(0,0,maxx,maxy,clipon);
  setfillstyle(1,0);
  bar(440,10,640,0);

setviewport(vars[7],vars[8],vars[9],vars[10],clipon);
end; {effdate}

{procédures diverses}
function existewindows : boolean;
var n : byte;
begin
  existewindows:=FALSE;
  for n:=1 to envcount do
    if copy(envstr(n),1,6)='windir' then
      existewindows:=TRUE;
  end; {existewindows}

```

NOTE : Certaines procédures figurant dans le listing précédent ne fonctionnent pas correctement, puisque cette unité est en cours de développement. Il s'agit notamment des procédures de traitement des listes, Imprimegraph... Néanmoins la plupart d'entre-elles fonctionnent parfaitement bien, parmi celles-ci, la gestion souris, la gestion de fenêtres graphiques avec sauvegarde du fond. Vous remarquerez certaines limitations, par exemple il n'est possible d'ouvrir qu'une seule fenêtre à la fois, la fenêtre ouverte ne doit pas être trop grande sinon il risque d'y avoir un plantage faute de mémoire suffisante. Dans la prochaine version qui sera publiée ces

```

{procédures et fonctions privées}
procedure initvar;
var
  i : word;
begin
  for i:=1 to 15 do vars[i]:=0;
  datelng:=1;
end; {initvar}

procedure initnvpolices;
begin
  ScriptFont:=InstallUserFont('scri');
  EuroFont:=InstallUserFont('euro');
  BoldFont:=InstallUserFont('bold');
  SimpFont:=InstallUserFont('simp');
  LcomFont:=InstallUserFont('lcom');
  TSCRFont:=InstallUserFont('tscr');
end; {initnvpolices}

{initialisation de l'unité}
begin
  initvar;
  initialize;
  initnvpolices;
  closegraph;
end.

```

problèmes devraient être résolus, de même d'autres fonctions et procédures devraient être ajoutées comme des fonctions mathématiques et financières, d'informations sur la mémoire...

Certaines procédures (boutons, fenêtres...) nécessitent l'utilisation de l'unité **graph**, n'oubliez donc pas d'inclure l'appel à cette unité en tête de vos programmes.

Ne recopiez pas les sections du programme faisant références aux nouvelles polices de caractères si vous ne disposez pas de celles-ci.

AU SECOURS ! Help ! i Socorro !

B.A.T.

Pour vous aider au début du jeu, voici les premières solutions. Si vous en avez d'autres, merci de nous les envoyer, nous les publierons.

Au premier tableau, allez au fond des toilettes pour prendre vos affaires. Allez vers le commissariat pour prendre un «permis». Allez vers la Rue Chaude, puis empruntez l'A-312, et entrez la salle de jeux. Une pétasse vous demandera si vous voulez jouer, envoyez-la bouler. Allez vers une machine, insérez votre carte de crédit pour aussitôt la retirer. Faites «Recherche» et prenez le tout:

Armes : Voktrasoft, L15 Beckman, Hacker 30, Haas 10, Canon Toz, Lance Nova.

Boissons : Double Splat, Nitrokola, Seven Down, Centauriax, Mettalurc.

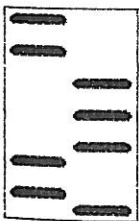
Munitions : LP 12, LP 57, AG 180, N 29, TO 2, E 10, E 300.

Médicaments : Hydratants, Vitamine, Dopant.

Champ de force : P 6, P 7, P 8.

Nourriture : Saucisse Brady, Femac Frites, Frites, Hot Dog, Sushi, Eindijest Frit, Melux Nitrokol, Kradokinoid, Jonc du Sable, Orange bleue, Plat du Jour.

Divers : Carte Xifo, Clé Moteur, Carte Drag, Carte de crédit, Krells, Hologramme, Clé d'Hôtel, «Permis».



Prenez des médicaments, de la nourriture, des boissons pour la forme, des champs de force P 8, des munitions, une arme et toutes les cartes dont quatre fois le «permis» pour les revendre et avoir 5500 krells pour louer un «drag». Allez vers la porte Nord et vers les ordinateurs. Demandez à l'ordinateur le code pour la base Epsilon (peut-être RRDDRRD, cf. figure).

Vous vous trouvez face à Wrangor, allez vers le carré du champ de force puis celui des munitions et FEU. Vous avez tué Wrangor, il vous reste à trouver Merigo et à désamorcer la bombe. A.M.

Programmation Basic CPC

CPCTAP.BAS

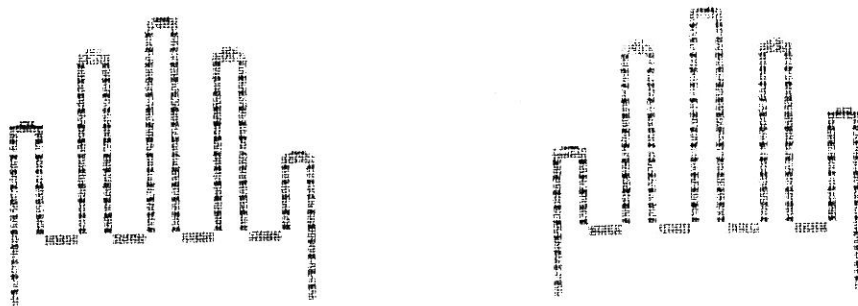
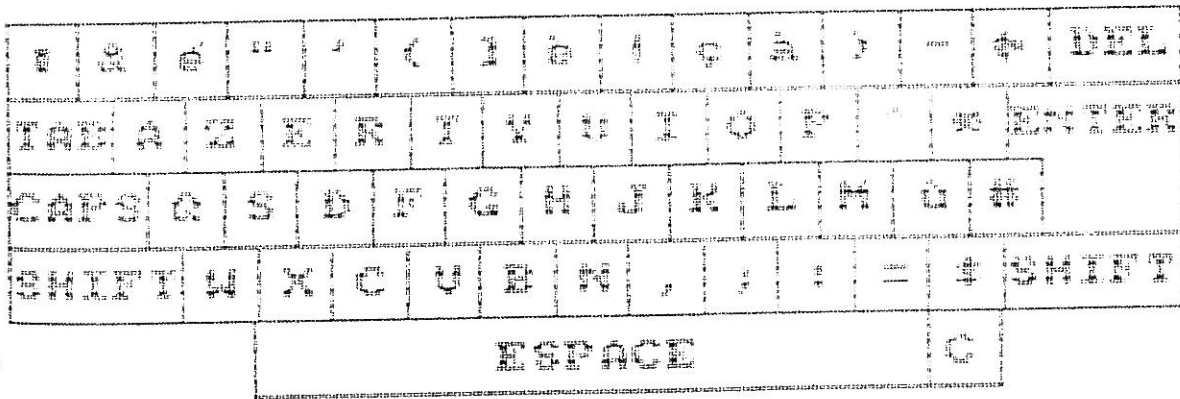
Vous êtes débutant, vous désirez améliorer votre vitesse de frappe ? Alors CPCTAP est fait pour vous. Après l'affichage du clavier et de deux figures symbolisant vos mains droite et gauche, l'ordinateur vous demande de frapper, le plus rapidement possible, des touches choisies aléatoirement, à l'aide du doigt qui clignote (puisque vous n'êtes pas sans savoir que chaque touche est affectée à un doigt particulier, dans un souci d'efficacité). Si au

bout de quelques heures d'utilisation, vous n'avez pas accéléré votre vitesse de frappe, allumez votre ordinateur (vous avez peut-être oublié de l'allumer...) ou faites du tricot.

Démarches

Saisissez le programme, sauvez-le sous le nom de "CPCTAP", puis faites «RUN». Benoît.

Tapez : I



```

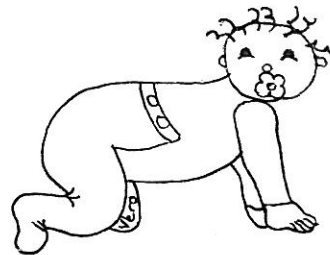
10 'CPC TAP v1.0
20 '
30 '(crit par Benoit RIVIERE (27/12/91)
40 '(c)1991 I.P.L. Informatique
50 '
60 ON BREAK GOSUB 820
70 MODE 1
80 SYMBOL AFTER 222
90 SYMBOL 223,195,195,195,195,195,195,195
100 SYMBOL 224,0,0,0,0,0,60,255,255
110 INK 0,0:INK 1,26:INK 2,0,26:BOARDER 0:PEN 1:PAPER 0:CLS
120 SPEED INK 15,25
130 GOSUB 830:INK 1,0:CLS
140 ON BREAK GOSUB 810
150 PLOT 30,350,1:DRAW 580,350
160 PLOT 30,315:DRAW 580,315
170 PLOT 30,280:DRAW 514,280
180 PLOT 30,245:DRAW 580,245
190 PLOT 30,210:DRAW 580,210
200 PLOT 146,175:DRAW 495,175
210 PLOT 30,350:DRAW 30,210:PLOT 582,350:DRAW 582,210
220 PLOT 146,175:DRAW 146,210
230 t2=350
240 RESTORE 920
250 FOR y=1 TO 5:READ n,t1
260 FOR i=1 TO n:PLOT t1+(i-1)*35,t2:DRAW t1+(i-1)*35,t2-35:
NEXT
270 t2=t2-35
280 NEXT
290 TAG
300 t2=340
310 FOR y=1 TO 4:READ t1,n
320 FOR i=1 TO n:MOVE t1+(i-1)*35,t2:READ c$:PRINT c$::NEXT
330 t2=t2-35:NEXT
340 MOVE 145,340:PRINT CHR$(34):
350 MOVE 32,305:PRINT"TAB":MOVE 32,270:PRINT"CAPS":
360 MOVE 33,235:PRINT"SHIFT":MOVE 500,235:PRINT"SHIFT":
370 MOVE 502,305:PRINT"ENTER":MOVE 260,198:PRINT"ESPACE":M
OVE 470,198:PRINT"C":
380 TAGOFF
390 h$=CHR$(224):c$=CHR$(223):b$=CHR$(208):x$=CHR$(8)+CHR$(1
0)
400 d$=h$:FOR i=1 TO 3:d$=d$+x$+c$:NEXT
410 d1$=d$
420 d2$=d$+x$+c$+x$+c$
430 d3$=d2$+x$+c$
440 d4$=d3$
450 d5$=h$+x$+c$+x$+c$
460 c1$=CHR$(211)+x$+CHR$(211)
470 c2$=CHR$(209)+x$+CHR$(209)
480 dd=0:FOR d=1 TO 5:ON d GOSUB 970,980,990,1000,1010:NEXT:
dd=1:FOR d=5 TO 1 STEP -1:ON d GOSUB 970,980,990,1000,1010:N
EXT
490 PRINT CHR$(31)+CHR$(8)+CHR$(23)+c1$
500 PRINT CHR$(31)+CHR$(32)+CHR$(23)+c2$
510 PRINT CHR$(31)+CHR$(16)+CHR$(23)+c2$
520 PRINT CHR$(31)+CHR$(24)+CHR$(23)+c1$
530 FOR y=0 TO 1:FOR i=1 TO 4:LOCATE 7+(16*y)+i*2,23:PRINT b
$:NEXT:NEXT
540 INK 1,26
550 RANDOMIZE TIME:c=INT(RND*80)
560 d=1:RESTORE 1020
570 READ o$:IF o$="f" THEN 550 ELSE IF o$="*" THEN d=d+1:GOT
O 570
580 READ q$
590 IF VAL(o$)<>c THEN 570
600 IF d=9 THEN PEN 2:dd=0:GOSUB 1010:dd=1:GOSUB 1010:PEN 1
610 IF d<5 THEN PEN 2:dd=0:ON d GOSUB 970,980,990,1000:PEN 1
620 IF d>4 AND d<9 THEN PEN 2:dd=1:ON d-4 GOSUB 1000,990,980
,970:PEN 1
630 IF o$="57" THEN q$=CHR$(34)
640 LOCATE 10,2:PRINT"Tapez : "q$
650 t1=TIME
660 a=-1
670 WHILE a=-1
680 a=INKEY(c)
690 WEND

```

```

700 t2=TIME
710 t=t2-t1
720 tt=tt+t:tk=tk+1
730 LOCATE 1,25:PRINT USING"nb frappes ###, temps moy ##.#
## sec.";tk,tt/tk/300;
740 ON d GOSUB 970,980,990,1000,1000,990,980,970,1010,970
750 IF d=9 AND dd=1 THEN dd=0:GOTO 740
760 CALL &BB03
770 LOCATE 10,2:PRINT SPACE$(28)
780 FOR i=1 TO 20:IF INKEY$<>" " THEN RUN 140
790 NEXT
800 GOTO 550
810 INK 1,0:CLS:RUN 140
820 RUN
830 CLS
840 LOCATE 12,3:PRINT"C P C T A P v1.0"
850 LOCATE 8,6:PRINT"écrit par Benoit RIVIERE
860 LOCATE 6,9:PRINT"le vendredi 27 décembre 1991
870 LOCATE 2,24:PRINT"(c)1991 I.P.L. Informatique
880 FOR i=1 TO 300:CALL &BD19:NEXT
890 CALL &BB18
900 RETURN
910 END
920 DATA 14,65,13,80,13,97,12,113,2,462
930 DATA 40,15,"!","&","(",",",";",":","=","$"
" "-" ,"r",DEL
940 DATA 90,12,A,Z,E,R,T,Y,U,I,O,P," ".*
950 DATA 105,12,Q,S,D,F,G,H,J,K,L,M,"!","#
960 DATA 125,11,W,X,C,V,E,N,"",",":";":"=","$"
970 PRINT CHR$(31)+CHR$(8+24*dd)+CHR$(19)+d1$:RETURN
980 PRINT CHR$(31)+CHR$(10+20*dd)+CHR$(17)+d2$:RETURN
990 PRINT CHR$(31)+CHR$(12+16*dd)+CHR$(16)+d3$:RETURN
1000 PRINT CHR$(31)+CHR$(14+12*dd)+CHR$(17)+d4$:RETURN
1010 PRINT CHR$(31)+CHR$(16+8*dd)+CHR$(20)+d5$:RETURN
1020 DATA 66,ESC,64,"&",68,TAB,67,A,70,CAPS LOCK,69,Q,71,W,*
1030 DATA 65,"!",59,Z,60,S,63,X,*
1040 DATA 57,,58,E,61,D,62,C,*
1050 DATA 56,"",49,"(",50,R,51,T,53,F,52,G,55,V,54,B.*
1060 DATA 48,"!",41,")",43,Y,42,U,44,H,45,J,46,N,38,"",",",*
1070 DATA 40,"!",35,I,37,K,39,"":",.*
1080 DATA 33,"\",34,O,36,L,31,"":",.*
1090 DATA 32,"@",25,")",24,"-",16,CLR,79,DEL,27,P,26,"",17.
"<",18,ENTER,29,M,28,"!",19,#,30,"=",".",22,"$",*
1100 DATA 47,ESPACE,I

```



Ordimage
se conjugue à tous les âges

Programmation en Turbo Pascal 6.0

COIFIC.PAS

Le but du programme proposé est de recopier les fichiers du *répertoire racine* d'une disquette vers le répertoire racine d'une autre disquette. Ce programme doit être simple à utiliser puisqu'il est surtout destiné à des non-initiés, par exemple des personnes qui tapent leur rapport de stage, et qui par mesure de sécurité désirent obtenir une copie de sauvegarde. Ceci exclut donc la saisie de commandes en ligne.

Déroulement

Le déroulement du programme est simple. Le programme demande le nom du lecteur de disquette (A ou B), le nom du disque dur (C ou D). Puis, on doit entrer la *disquette source*, c'est à dire la disquette contenant les fichiers à recopier. Ces fichiers sont recopiés temporairement dans un répertoire sur le disque dur indiqué par l'utilisateur. Ensuite, on insère la *disquette destination* sur laquelle tous les fichiers seront transférés. Il est ensuite proposé de recopier ces mêmes fichiers sur une autre disquette, puis de copier le répertoire racine d'une autre disquette. Le répertoire temporaire créé sur le disque dur est effacé automatiquement à la fin de l'utilisation du programme.

Améliorations

Le programme tel qu'il figure ci-dessous peut être amélioré d'au moins trois façons :

* Tout d'abord, par ajout d'une gestion des erreurs disque (signalisation du manque de place sur le disque dur, sur la disquette destination, détection de l'absence de la disquette...).

* Ensuite, il pourrait être envisageable de gérer la copie d'autres répertoires que la racine.

* Et enfin, on pourrait gérer la copie de fichiers en double en comparant par exemple les heure et date d'enregistrement, ou la taille...

Mise en route

Copiez le listing, puis exécutez-le. Je vous conseille de le compiler sur le disque dur et de le conserver dans la racine ou bien dans un répertoire accessible par le 'Path'. Ainsi les débutants pourront copier leurs disquettes en toute tranquillité. Benoit.

ATTENTION !! Pour fonctionner ce programme nécessite l'unité TPD05EXT.TPU dont le listing figure dans les pages précédentes.

```

program copifc; {v1.1}
{recopie des fichiers du répertoire courant d'une
disquette sur une autre}
{écrit par Benoît Rivière}
{(c)1994 I.P.L. Informatique}
{v1.0 du 15/03/94}
{v1.1 du 02/04/94}

```

```
uses crt,dos,tpdosext;
```

```

var
autrecopie, autredisque : boolean;
disqtte,dd : string;
t : string;

```

```

const
nomprg : string[10] = 'COPIFIC';
ver : string[4] = 'v1.1';
copyrgt : string[60] = 'écrit par Benoît Rivière'+
' (c)1994 I.P.L. Informatique';
nomdir : string[12] = 'copidir.$$$';

```

```
procedure msgprg;
```

```

var
i : integer;

```

```

begin
writeln;
writeln(nomprg+' '+ver+' '+copyrgt);
textcolor(9);
for i:=1 to 75 do write('-');
textcolor(2);
writeln;
writeln;
writeln('Ce programme recopie les fichiers
contenus dans la racine "\",
' d'une");
writeln('disquette SOURCE vers une disquette
DESTINATION. ');
textcolor(15);
writeln;
end; {msgprg}

```

```
procedure efftoutfic;
```

```

var
dirinfo : SearchRec;
f : file;

```

```

begin
findfirst(dd+'\'+nomdir+'\*.*',archive,dirinfo);
while DosError=0 do
begin
assign(f,dirinfo.name);
erase(f);
findnext(dirinfo);
end;

```

```
end; {efftoutfic}
```

```
procedure creedir;
```

```

begin
if existefichier('\'+nomdir) then
begin
chdir(dd+'\'+nomdir);
efftoutfic;
end
else
begin
mkdir(dd+'\'+nomdir);
chdir(dd+'\'+nomdir);
end;
end; {creedir}

```

```
procedure effdir;
```

```

begin
efftoutfic;
chdir(dd+'\');
rmdir(dd+'\'+nomdir);
end; {effdir}

```

```
procedure copiefic(s,d : string);
```

```

var
dirinfo : searchrec;

DepuisF, DansF : file;
NumLus, NumEcrits : word;
Tampon : array[1..2048] of char;

```

```

begin
writeln;
findfirst(s+'*.*',archive,dirinfo);
while DosError=0 do
begin
assign(DepuisF,s+dirinfo.name);
reset(DepuisF,1);
writeln(' Copie de ',s,dirinfo.name,'
,filesize(depuf), ' octets...');

```

```

assign(DansF,d+dirinfo.name);
rewrite(DansF,1);
repeat
{$I-}

```

```

blockread(DepuisF,Tampon,SizeOf(Tampon),N
umLus);

```

```

blockwrite(DansF,Tampon,NumLus,NumEcrits);
{$I+}
if ioresult<>0 then
begin
effdir;
writeln('Erreur disquette...');
textcolor(7);

```

```

    halt(0);
    end;
    until (NumLus = 0) or (NumEcrits <>
NumLus);

    close(DepuisF);
    close(DansF);

    findnext(dirinfo);
    end;
end; {copiefic}

procedure quitte(o : byte);
begin
    if o=1 then effdir;
    writeln;
    writeln('Application avortée par l'utilisateur.');
```

```

    writeln;
    halt(1);
    textcolor(7);
end; {quitte}

procedure attdsk(n : byte);
var
    t : string;
const
    typdsk : array[1..2] of string[11] =
('SOURCE', 'DESTINAT.');
```

```

begin
    textcolor(14);
    writeln('Insérez disquette '+typdsk[n]+' puis
tapez <Entrée>'+
' ou pressez <Echap> pour sortir.');
```

```

    textcolor(15);
    repeat
        t:=readkey;
    until (t=#13) or (t=#27);
    if t=#27 then quitte(1);
end; {attdsk}

{programme principal}
begin
    textcolor(15);
    textbackground(0);
    msgprg;

    writeln('Disquette en lecteur A: ou B: ou
pressez <Echap> pour quitter.');
```

```

    repeat
        t:=upcase(readkey);
    until (t='A') or (t='B') or (t=#27);
    if t=#27 then quitte(0);
    disqtte:=t+'.';

    writeln;
```

```

    writeln('Disque dur C: ou D: ou pressez
<Echap> pour quitter.');
```

```

    repeat
        t:=upcase(readkey);
    until (t='C') or (t='D') or (t=#27);
    if t=#27 then quitte(0);
    dd:=t+'.';

    creedir;

    repeat
        repeat
            writeln;
            attdsk(1);
            textcolor(11);
            copiefic(disqtte+'\', dd+'\'+nomdir+'\');
```

```

            textcolor(15);
            until ioreult=0;

        repeat
            writeln;
            attdsk(2);
            textcolor(12);
            copiefic(dd+'\'+nomdir+'\', disqtte+'\');
```

```

            textcolor(15);
            until ioreult=0;
            writeln;
            writeln('Voulez-vous copier ces fichiers sur
une autre disquette ?',
' (O/N)');
```

```

            repeat
                t:=upcase(readkey);
            until (t='O') or (t='N') or (t=#27);
            if t=#27 then quitte(1);
            if t='O' then autrecopte:=true else
autrecopte:=false;
            until autrecopte=false;

            writeln;
            writeln('Voulez-vous copier une autre
disquette ? (O/N)');
```

```

            repeat
                t:=upcase(readkey);
            until (t='O') or (t='N') or (t=#27);
            if t=#27 then quitte(1);
            if t='O' then autredisque:=true else
autredisque:=false;
            efftoutfic;
            until autredisque=false;
            effdir;
            writeln;
            writeln('Application terminée.');
```

```

            writeln;
            textcolor(7);
        end.
```

Programmation Basic CPC

ORDICO.BAS

ORDICO est un lexique trilingue. Il permet de rechercher des mots français, allemand, et anglais dans deux lexiques français/allemand et français/anglais, ce qui exclut par conséquent la recherche allemand/anglais et anglais/allemand. Ces deux lexiques sont chacun organisés en 26 fichiers (1 par lettre). Avant de lancer ORDICO pour la première fois, il faut exécuter INITDICA et INITDICG qui vont se charger de créer les deux lexiques vides. Il ne faut donc pas exécuter ces deux petits programmes une fois que vous aurez commencé à saisir du vocabulaire sous peine de voir votre travail anéanti ! Je vous conseille vivement de mettre ORDICO sur

une face de disquette affectée uniquement à ce programme, car les lexiques sont gourmands en octets.

Depuis j'ai créé un petit frère sur PC, qui est beaucoup plus rapide et pratique, de plus il utilise pleinement l'unité TPDOSEXT (cf. page 19 et suivantes).

Démarches

Recopiez INITDICA, exécutez-le. Faites de même avec INITDICG. Saisissez le programme ORDICO, sauvez-le sous le nom de "ORDICO", puis faites «RUN». Benoît.

```

1 'INITDICA v1.0 pour ALLDICO v1.x
2 '
3 '(crit par Benoit RIVIERE (aout 91)
4 '(c)1991 I.P.L. Informatique
5 '
6 PRINT"ATTENTION : Initialisation des fichiers ALLDICO !"
7 PRINT"faire RUN 9":END
8 nfile$="ALLDICO"
9 FOR i=65 TO 90
10 OPENOUT nfile$+CHR$(i)
11 PRINT#9,0
12 CLOSEOUT
13 NEXT

```

```

1 'INITDICG v2.1 pour GBDICO v2.x
2 '
3 '(crit par Benoit RIVIERE (mai 91)
4 '(c)1991 I.P.L. Informatique
5 '
6 PRINT"ATTENTION : Initialisation des fichiers GBDICO !"
7 PRINT"faire RUN 9":END
8 nfile$="GBDICO"
9 FOR i=65 TO 90
10 OPENOUT nfile$+CHR$(i)
11 PRINT#9,0
12 CLOSEOUT
13 NEXT

```

```

10 'ORDICO v2.0 (15/02/92)
20 '
30 'écrit par Benoit RIVIERE (v1.0 30/11/91)
40 '(c)1992 I.P.L. Informatique
50 '
60 ON ERROR GOTO 60000
70 SPEED !NK 10,10
80 MODE 1:INK 0,0:INK 1,26:INK 2,0,26:PEN 1:PAPER 0:BORDER 0:CLS
90 ver$="v2.0":date$="15/02/92"
100 GOSUB 1000
110 OPENOUT"m":MEMORY HIMEM-1:CLOSEOUT
120 limite=300
130 ajout=0
140 DIM trad$(limite),det$(limite),pl$(limite),fr$(limite)
150 lang$="anglais":fic$="GB"
160 nbmt=0
170 nfc=65:GOSUB 340
180 MODE 2:CLS
190 '
200 PRINT:PRINT fic$>"CHR$(nfc)"/":PRINT USING"###/":nbmt;:LINE INPUT mt$
210 IF mt$="" OR LEN(mt$)>40 THEN 120
220 IF LEFT$(mt$,1)="/" THEN GOSUB 490:GOTO 120
230 trv=0
240 m=0
250 IF RIGHT$(mt$,1)="/" THEN lg=LEN(mt$)-1:m=1
260 IF LEFT$(mt$,1)="/" THEN lg=LEN(mt$)-1:IF loc=1 THEN m=2 ELSE GOSUB 250:nfc=65:GOSUB 340:m=2
270 IF (m=0 OR m=1) AND (LOWER$(LEFT$(mt$,1))<"a" OR LOWER$(LEFT$(mt$,1))>"z") THEN PRINT"Erreur mot.":CHR$(7):GOTO 120
280 IF m<2 THEN IF nfc<>ASC(UPPER$(LEFT$(mt$,1))) THEN GOSUB 250:nfc=ASC(UPPER$(LEFT$(mt$,1))):GOSUB 340
290 IF imp=1 THEN GOSUB 800:IF errimp=1 THEN imp=0 ELSE PRINT#8,lang$," Commande : ";mt$;:IF loc=1 THEN PRINT#8," (local)" E
300 SEE PRINT#8
310 FOR i=1 TO nbmt
320 IF m=2 THEN IF LOWER$(RIGHT$(mt$,lg))=LOWER$(RIGHT$(trad$(i),lg)) THEN 210
330 IF m=1 THEN IF LOWER$(LEFT$(mt$,lg))=LOWER$(LEFT$(trad$(i),lg)) THEN 210
340 IF LOWER$(mt$)>LOWER$(trad$(i)) THEN 220
350 IF det$(i)="" THEN sp1=0 ELSE sp1=1
360 IF pl$(i)="" THEN sp2=0 ELSE sp2=1
370 IF imp=1 THEN PRINT#8,det$(i)SPACE$(sp1)trad$(i)SPACE$(sp2)STRING$(sp2,"(")pl$(i)STRING$(sp2,")")" = "fr$(i)
380 PRINT det$(i)SPACE$(sp1)trad$(i)SPACE$(sp2)STRING$(sp2,"(")pl$(i)STRING$(sp2,")")" = "fr$(i):trv=1
390 NEXT
400 IF m=2 AND loc=0 AND nfc<90 THEN nfc=nfc+1:GOSUB 340:GOTO 170
410 IF trv<1 AND m=0 THEN IF lang$="allemand" AND ajout=1 THEN LINE INPUT"d(terminant/<ENTER> :",det$:LINE INPUT"pluriel/pr
420 rit/auxiliaire... :",pl$:GOSUB 420 ELSE GOSUB 420
430 GOTO 130
440 IF flagchang=0 THEN RETURN ELSE IF sauveclasse=1 THEN GOSUB 560
450 OPENOUT fic$+"dico"+CHR$(nfc)
460 PRINT#9,nbmt
470 FOR i=1 TO nbmt
480 IF fic$="ALL" THEN WRITE#9,det$(i)
490 WRITE#9,trad$(i)
500 IF fic$="ALL" THEN WRITE#9,pl$(i)
510 WRITE#9,fr$(i)
520 NEXT
530 CLOSEOUT
540 xx$="*.bak":!ERA,@xx$
550 RETURN
560 ERASE trad$,det$,pl$,fr$:DIM trad$(limite),det$(limite),pl$(limite),fr$(limite)
570 OPENIN fic$+"dico"+CHR$(nfc)
580 INPUT#9,nbmt
590 FOR i=1 TO nbmt
600 IF fic$="ALL" THEN INPUT#9,det$(i)
610 INPUT#9,trad$(i)
620 IF fic$="ALL" THEN INPUT#9,pl$(i)
630 INPUT#9,fr$(i)
640 NEXT
650 CLOSEIN
660 flagchang=0
670 RETURN
680 PRINT:PRINT"Mot inconnu."
690 IF ajout=0 THEN RETURN ELSE INPUT"traduction (ENTER=annuler) :",mt2$
700 IF mt2$="" THEN PRINT CHR$(7);:RETURN
710 nbmt=nbmt+1
720 IF nbmt>limite THEN PRINT"trop de mots"CHR$(7):RETURN
730 trad$(nbmt)=mt$:IF fic$="ALL" THEN det$(nbmt)=det$:pl$(nbmt)=pl$
740 fr$(nbmt)=mt2$
750 flagchang=1
760 RETURN
770 mt$=LOWER$(RIGHT$(mt$,LEN(mt$)-1))
780 IF mt$="sauve" THEN GOSUB 250:RETURN

```

```

501 IF mt$="imp" THEN IF INP(&F500) AND &X1000000 THEN PRINT"Erreur imprimante.":imp=0:RETURN ELSE imp=1:RETURN ELSE IF mt$="
animp" THEN imp=0:RETURN
505 IF mt$="sauveclasse" THEN sauveclasse=1:RETURN ELSE IF mt$="ansauveclasse" THEN sauveclasse=0:RETURN
507 IF mt$="classe" THEN GOSUB 560:RETURN
508 IF mt$="liste" AND imp=1 THEN GOSUB 800:IF errimp=1 THEN RETURN
509 IF mt$="liste" AND imp=1 THEN RETURN PRINT#8,"Liste lettre "CHR$(nfc)";":FOR i=1 TO nbmt:PRINT#8, det$(i)SPACE$(ABS(LEN(
det$(i)))>0))trad$(i)SPACE$(ABS(LEN(pl$(i)))>0))STRING$(ABS(LEN(pl$(i)))>0),"pl$(i)STRING$(ABS(LEN(pl$(i)))>0),"") = "fr$(i)
:NEXT
510 IF mt$="liste" THEN FOR i=1 TO nbmt:PRINT det$(i)SPACE$(ABS(LEN(det$(i)))>0))trad$(i)SPACE$(ABS(LEN(pl$(i)))>0))STRING$(ABS
(LEN(pl$(i)))>0),"pl$(i)STRING$(ABS(LEN(pl$(i)))>0),"") = "fr$(i):NEXT:RETURN
515 IF mt$="aide" THEN GOSUB 5000:RETURN
519 IF mt$="rajoute" AND ajout=0 THEN RETURN
520 IF mt$="rajoute" THEN PRINT"mot "lang$":":LINE INPUT mt1$:IF mt1$="" THEN RETURN
521 IF mt$="rajoute" THEN IF nfc(>ASC(UPPER$(LEFT$(mt1$,1)))) THEN GOSUB 250:nfc=ASC(UPPER$(LEFT$(mt1$,1))):GOSUB 340:IF lan
g$="allemand" THEN INPUT"determinant/<ENTER> :",det$:LINE INPUT"pluriel/prit(rit/auxiliaire... :",pl$:mt$=mt1$:GOSUB 425:RETU
RN
522 IF mt$="rajoute" THEN mt$=mt1$:GOSUB 425:RETURN
525 IF mt$="ajout" THEN ajout=1:RETURN ELSE IF mt$="interajout" THEN ajout=0:RETURN
530 IF mt$="eff" THEN GOSUB 670:RETURN
532 IF mt$="bil" THEN totmt=0:FOR i=65 TO 90:OPENIN fic$+"DICO"+CHR$(i):INPUT#9,mt:CLOSEIN:PRINT"Lettre ";CHR$(i);" :":PRINT
USING"###" mot(s).":mt:totmt=totmt+mt:NEXT:PRINT"Nombre total de mots :":totmt:RETURN
533 IF mt$="fin" THEN STOP
534 IF mt$="local" THEN loc=1:RETURN ELSE IF mt$="anlocal" THEN loc=0:RETURN
535 IF mt$="version" THEN PRINT ver$:RETURN
537 IF mt$="langue" THEN PRINT lang$:RETURN
538 IF mt$="anglais" THEN GOSUB 250:fic$="GB":lang$="anglais":GOSUB 340:RETURN ELSE IF mt$="allemand" THEN GOSUB 250:fic$="AL
L":lang$="allemand":GOSUB 340:RETURN
540 PRINT "Commande inconnue.":CHR$(7)
550 RETURN
560 flagchang=1:n=nbmt:e=INT(n/2)
570 IF e<1 THEN RETURN
580 nd=1
590 IF nd<(n-e) OR nd=(n-e) THEN 610
600 e=INT(e/2):GOTO 570
610 na=nd
620 IF LOWER$(trad$(na))<=LOWER$(trad$(na+e)) THEN 650
630 per$=trad$(na):per2$=fr$(na):IF lang$="allemand" THEN per3$=det$(na):per4$=pl$(na)
632 trad$(na)=trad$(na+e):fr$(na)=fr$(na+e):IF lang$="allemand" THEN det$(na)=det$(na+e):pl$(na)=pl$(na+e)
635 trad$(na+e)=per$:fr$(na+e)=per2$:IF lang$="allemand" THEN det$(na+e)=per3$:pl$(na+e)=per4$
637 na=na+e
640 IF na>1 OR na=1 THEN 620
650 nd=nd+1
660 GOTO 590
670 INPUT"mot @ effacer :",mt$
680 IF mt$="" THEN RETURN
685 IF nfc(>ASC(UPPER$(LEFT$(mt$,1)))) THEN GOSUB 250:nfc=ASC(UPPER$(LEFT$(mt$,1))):GOSUB 340
687 trv=0
690 FOR i=1 TO nbmt
700 IF trad$(i)=mt$ THEN trv=1:PRINT det$(i)SPACE$(ABS(LEN(det$(i)))>0))trad$(i)SPACE$(ABS(LEN(pl$(i)))>0))pl$(i) = "fr$(i):IN
PUT"Efface ? (o/n)",r$:IF UPPER$(r$)="O" THEN GOSUB 730
710 NEXT
715 IF trv=0 THEN PRINT"Mot(s) non trouv((s).":CHR$(7)
720 RETURN
730 trad$(i)="" :fr$(i)="" :IF lang$="allemand" THEN det$(i)="" :pl$(i)=""
740 FOR y=i TO nbmt
750 trad$(y)=trad$(y+1):fr$(y)=fr$(y+1):IF lang$="allemand" THEN det$(y)=det$(y+1):pl$(y)=pl$(y+1)
760 NEXT
770 nbmt=nbmt-1
775 flagchang=1
780 RETURN
800 errimp=0:IF INP(&F500) AND &X1000000 THEN PRINT"Erreur imprimante.":errimp=1:RETURN
810 WIDTH 80
820 PRINT#8,CHR$(27);"@";
830 PRINT#8,CHR$(27);"P";CHR$(27);"W";CHR$(18);
840 PRINT#8," ORDICO ";ver$;" ";date$;" (v1.0 30/11/91) (c)I.P.L. Informatique
850 PRINT#8
860 PRINT#8," Computer Aided Word-bank (C.A.W.-b.) (crit par Benoit RIVIERE)"
870 PRINT#8
880 RETURN
9000 LOCATE 13,4:PRINT"O R D I C O "ver$
9010 LOCATE 9,8:PRINT"(crit par Benoit RIVIERE"
9015 LOCATE 13,12:PRINT"15 février 1992"
9016 LOCATE 9,17:PRINT"GBDICO v1.0 du 25 mai 91":LOCATE 5,19:PRINT"ORDICO v1.0 du 30 novembre 1991"
9017 LOCATE 5,22:PRINT"Tapez /AIDE pour obtenir la notice"
9020 LOCATE 3,25:PRINT"(c)1992 I.P.L. Informatique
9100 GOSUB 3000
9500 FOR i=1 TO 200:CALL &BD19:NEXT
9600 RETURN

```

```

3000 CALL &BB4E:SYMBOL AFTER 172
3010 SYMBOL 173,108,108,120,12,124,204,118,0
3020 SYMBOL 174,108,108,60,102,126,96,60,0
3030 SYMBOL 175,204,204,102,102,102,102,62,0
3040 SYMBOL 176,108,108,60,102,102,102,60,0
3050 SYMBOL 178,108,108,0,0,126,0,0,0
3051 SYMBOL 193,24,36,120,12,124,204,118,0
3052 SYMBOL 194,24,36,60,102,126,96,60,0
3053 SYMBOL 195,24,36,58,24,24,24,60,0
3054 SYMBOL 196,24,36,60,102,102,102,60
3055 SYMBOL 197,24,36,102,102,102,102,62
3060 KEY DEF 64,1,36,173
3070 KEY DEF 65,1,123,174
3080 KEY DEF 57,1,34,176
3090 KEY DEF 58,1,33,175
3100 KEY DEF 49,1,40,177
3110 KEY DEF 48,1,93,178
3120 KEY DEF 41,1,126,190
3130 KEY DEF 40,1,33,194
3140 KEY DEF 33,1,21,196
3150 KEY DEF 3,1,24,198
3160 KEY DEF 25,1,41,199
3500 RETURN
5000 RESTORE 6000
5010 READ ph$:IF ph$="tin" THEN RETURN
5015 IF ph$="at" THEN at$=INKEY$:CALL &BD19:IF at$="" THEN 5015 ELSE 5010
5020 PRINT ph$
5030 GOTO 5010
6000 DATA " " " Les commandes : "
6005 DATA " /a : aide : vous etes en train de la lire.
6010 DATA " /ajout : autorise l'ajout de mots (interajout : annule cette commande)
6015 DATA " /bit : donne le nombre de mots par lettre puis pour tout le dictionnaire.
6020 DATA " /classe : classe les mots dans l'ordre alphanbetique.
6025 DATA " /eff : efface un ou plusieurs mots.
6030 DATA " /fin : retour au BASIC.
6035 DATA " /imp : impression des listages (/animp)
6040 DATA " /liste : liste les mots de la lettre en cours.
6050 DATA " /local : *xxx ne listera que les mots de la lettre en cours (/anlocal : annule cette instruction).
6060 DATA " /sauve : sauve la lettre en cours.
6065 DATA " /sauveclasse : lors de la sauvegarde les mots seront classés alphanbetiquement. (/ansauveclasse : annule /sauveclasse)
6070 DATA " /ajoute : ajoute un mot qui existe deja.
6075 DATA " /version : indique le numero de la version du logiciel utilise.
6080 DATA " /langue : donne la langue en cours
6085 DATA " /anglais : ouvre le dictionnaire d'anglais
6090 DATA " /allemand : ouvre le dictionnaire d'allemand
6095 DATA "at",""," " Les jockers : "
6100 DATA " *ish : liste tous les mots se terminant pas -ish (voir /local).
6105 DATA " ce* : liste tous les mots commençant par ce-
6999 DATA tin
60000 PRINT"Erreur de programmation";CHR$(7):RESUME 120

```

Macro-commande sur Excel

DIDACTI.XLM

Une macro-commande est une suite d'instructions permettant d'automatiser des tâches. EXCEL offre un interpréteur de macro-commandes tellement puissant que l'on peut l'apparenter à un langage classique, mais à ceci près que ce simili-langage est doté de fonctions tableurs fort intéressantes. C'est ce que je vais tâcher de vous démontrer avec DIDACTI.

DIDACTI comme son nom l'indique est un didacticiel c'est à dire une macro-commande d'enseignement. Il s'agit de répondre à des questions sous forme de Q.C.M. choisies aléatoirement par l'innocente main de l'ordinateur. Les réponses sont elles aussi affichées dans un ordre aléatoire afin que l'élève ne mémorise pas une suite de réponses défilant dans un ordre établi, mais associe bien chaque réponse à la question adéquate.

Mise en oeuvre

Cette macro-commande nécessite deux feuilles ; tout d'abord une feuille macro (Fichier/Nouveau document.../Macro) qui

contiendra la macro proprement dite et que l'on nommera DIDACTI.XLM. puis une feuille de calcul toute simple qui supportera le fichier de questions-réponses et qui s'appellera DIDACTI.XLS.

Une fois les deux feuilles ouvertes, faites Options/Environnement... et cochez la case L1C1, afin que les colonnes soient numérotées au lieu d'être référencées par des lettres.

Allez dans la feuille de calcul DIDACTI.XLS, recopiez la base de questions-réponses (veillez à ce que la 1ère question se trouve bien en L21C1). Puis positionnez-vous en L3C2, faites Sélection/Définir un nom et entrez : nb_quest.

Ensuite cliquez sur la petite flèche qui pointe vers le bas (tout en haut à droite de la feuille de calcul DIDACTI.XLS), puis Fichier/Enregistrer.

Pour ajouter des questions, il suffit de changer le nombre en L3C2, puis d'ajouter vos questions à la suite des autres, tout en sachant que vous devez toujours mettre la bonne réponse en premier.

Ensuite, positionnez-vous sur la feuille macro (DIDACTI.XLM), et cliquez sur les cellules indiquées dans le tableau ci-après. Pour nommer ces cellules passez par Sélection/Définir un nom.

Liste des noms de la feuille macro

DIDACTI.XLM

Références cellules	Noms
L2C1	Auto_ouvrir
L21C1	debut
L15C1	ent
L122C1:L126C7	fen_accueil
L129C1:L133C7	fen_etresur
L137C1:L149C7	fen_question
C1	fonction_enregistrement
L142C7	nr_rep
L149C6	question
L137C6	questionnr
L115C1	quitter
L143C6	reponse1
L144C6	reponse2
L145C6	reponse3
L146C6	reponse4
L147C6	reponse5
L129C6	titrefen_etresur
L80C1	zone_quest

Copiez ensuite toutes les lignes de la macro-commande, en respectant bien les numéros de ligne et de colonne.

La base de questions/réponses (feuille DIDACTI.XLS)

cf. page suivante

NOTE IMPORTANTE : La barre verticale (par exemple dans «nom=ENTRER("Veuillez entrer votre nom :||<Echap> pour quitter";2;"Votre nom...")») s'obtient en faisant Alt+Entrée ; cela ordonne à EXCEL d'exercer un retour à la ligne.

Réduisez la feuille macro DIDACTI.XLM à l'état d'icône (de la même manière qu'avec DIDACTI.XLS, c'est à dire en cliquant sur la flèche pointant vers le bas, située tout en haut à droite de la feuille DIDACTI.XLM). Sauvegardez la feuille macro.

Pour exécuter, cette application faites Fichier/Ouvrir et indiquez DIDACTI.XLM. La macro s'exécute automatiquement, ceci est dû au fait que nous avons nommé L1C1 «Auto_ouvrir».

Améliorations possibles

- on pourrait construire plusieurs fichiers de questions-réponses sur divers thèmes ; l'utilisateur choisirait le fichier à réviser.
- une feuille de calcul pourrait enregistrer les performances de chaque utilisateur afin de suivre la progression individuelle de ceux-ci pour chaque sujet.
- possibilité de choisir le nombre de questions à traiter.
- handicaps, points négatifs... Benoît.

1	2	3	4	5	6	7
Base des questions/réponses						
nombre de questions :	15					
QUESTION	(La bonne réponse)					
Quelle est la capitale de la Belgique ?	REPONSE1 Bruxelles	REPONSE2 Berne	REPONSE3 Libourne	REPONSE4 Paris	REPONSE5 Luxembourg	DEJAFORTE
Qui est le Président de la République ?	M. Mitterrand	M. Kohl	M. Clinton	M. Chirac	M. Juppé	
Une obligation est :	une part d'emprunt	un synonyme de 'must'	un synonyme de part social	une déclaration fiscale	une étape de l'O.P.A.	
Quel est le libellé du compte '2111' ?	Terrains nus	Logiciel	Terrains de gisement	Terrain pétrolière	Pot de vin	
Qu'évoque pour vous 2058A ?	Un imprimé fiscal	Un nom de code obscur	Un problème insoluble	Un gros calibre	Un code minitel	
Que signifie le sigle P.C. ?	Personal Computer	Professional Computer	Petit crélin	Payé Comptant	Plan comptable	
Que signifie le sigle I.B.M. ?	International Business Machines	Institut de Botanique Mexicain	International Burnt Microprocessors	International Bioethical Meeting des ordinateurs	International Binary Machines des fours à micro-ondes	
Que commercialise Intel ?	microprocesseurs	du pop corn	des logiciels de jeu			
MS/DOS signifie :	Microsoft Disk Operating System	Mega Super Death Or Sleep	Microsoft System Directory Operating Signal	Microsoft System Directory Of Softwares	Memory Software Disk Operating System	
Quel est le successeur du microprocesseur 486 ?	le Pentium	le 586	le P5	le Tridium	le Z80	
Quelle est la norme du clavier français ?	AZERTY	QWERTY	AWERTY	QSDFMLKJ	WXXCVB	
Une disquette de format 3,5" Haute Densité peut contenir :	1,44 Mo	740 Ko	1,2 Mo	1,4 Mo	2 Mo	
L'extension 'XLS' des feuilles de calculs EXCEL signifie :	EXCEL Spreadsheet	EXCEL System	EXCEL Soap	EXCEL Sign	EXCEL Signal	
Bagpipe se traduit par :	cornemuse	sac à pipe	boîte à tabac	sac à main	flûte	
Die Wiedervereinigung veut dire :	la réunification	la dislocation	la réunion	meeting de Wiederd	la liberté	

1ère partie de la macro-commande (à recopier à partir de la cellule L2C1 jusqu'en L118C1)

1

DIDACTI v1.0

système didacticiel

écrit par Benoît Rivière

(c)1994 I.P.L. Informatique

v1.0 du 23/04/94

```

=FENETRE.REDUCTION("didacti.xlm")
=OUVRIR("c:\excel\macrolib\fichfnct.xla")
chemin="C:\EXCEL\DIDACTI\"
=SI(EXISTE.FICHER(chemin&"didacti.xls")=FAUX)
=ALERTE("Le fichier "&chemin&"DIDACTI.XLS' est manquant.")
=RETOUR()
=FIN.SI()
=OUVRIR(chemin&"didacti.xls")
=ACTIVER("didacti.xls")
=FENETRE.REDUCTION("didacti.xls")

=ZONE.DE.DIALOGUE(fen_accueil)

debut
nom=ENTRER("Veuillez entrer votre nom :||<Echap> pour quitter":2:"Votre nom...")
=SI(nom=FAUX)
=FORMULE("Quitter DIDACTI...";titrefen_etresur)
=ZONE.DE.DIALOGUE(fen_etresur)
=SI(L(-1)C=4;ATTEINDRE(quitter))
=RETOUR()
=FIN.SI()
=SI(nom="";ATTEINDRE(debut))
nom=NOMPROPRE(GAUCHE(nom;15))
=ALERTE("Bonjour "& nom &". vous allez maintenant subir un test|de connaissances. Bonne
chance.".2)

initialise
i=0
=TANT.QUE(i+1<=DIDACTI.XLS!nb_quest)
=FORMULE("":ADRESSE(21+i;7;1;FAUX;"didacti.xls"))
i=i+1
=SUIVANT()

question
score=0
i=1
=TANT.QUE(i<=DIDACTI.XLS!nb_quest)
q=0
=TANT.QUE(OU(q<1;EVALUER(ADRESSE(21+q-1;7;1;FAUX;"didacti.xls"))=1))
q=ARRONDI(ALEA.ENTRE.BORNES(1;DIDACTI.XLS!nb_quest);0)
=SUIVANT()
=FORMULE(1;ADRESSE(21+q-1;7;1;FAUX;"didacti.xls"))

=FORMULE("Question "&i&" sur "&DIDACTI.XLS!nb_quest;QuestionNr)

```

```

= FORMULE(REFTEXTE(ADRESSE(21+q-
1;1;1;FAUX;"C:\EXCEL\IDACT\IDACTI.XLS")); "didacti.xml!question")

    r=1
    bonne_rep=0
    lig_rep="00000"
= TANT.QUE(r<=5)
    n=ARRONDI(ALEA.ENTRE.BORNES(1;5);0)
= SI(ESTERREUR(TROUVE(CAR(CNUM(n)+48);lig_rep))=VRAI)
= SI(CNUM(n)=1)
    bonne_rep=r
= FIN.SI()
= SI(r=1)
= FORMULE(REFTEXTE(ADRESSE(21+q-
1;1+n;1;FAUX;"C:\EXCEL\IDACT\IDACTI.XLS")); "didacti.xml!reponse1")
= SINON.SI(r=2)
= FORMULE(REFTEXTE(ADRESSE(21+q-
1;1+n;1;FAUX;"C:\EXCEL\IDACT\IDACTI.XLS")); "didacti.xml!reponse2")
= SINON.SI(r=3)
= FORMULE(REFTEXTE(ADRESSE(21+q-
1;1+n;1;FAUX;"C:\EXCEL\IDACT\IDACTI.XLS")); "didacti.xml!reponse3")
= SINON.SI(r=4)
= FORMULE(REFTEXTE(ADRESSE(21+q-
1;1+n;1;FAUX;"C:\EXCEL\IDACT\IDACTI.XLS")); "didacti.xml!reponse4")
= SINON.SI(r=5)
= FORMULE(REFTEXTE(ADRESSE(21+q-
1;1+n;1;FAUX;"C:\EXCEL\IDACT\IDACTI.XLS")); "didacti.xml!reponse5")
= FIN.SI()
    lig_rep=GAUCHE(lig_rep;r-1)&CAR(CNUM(n)+48)&DROITE(lig_rep;NBCAR(lig_rep)-r)
    r=r+1
= FIN.SI()
= SUIVANT()

zone_quest
= FORMULE(1;"didacti.xml!nr_rep")
= ZONE.DE.DIALOGUE(fen_question)
= SI(OU(L(-1)C=FAUX;L(-1)C=3))
= FORMULE("Arrêter le test...";titrefen_etresur)
= ZONE.DE.DIALOGUE(fen_etresur)
= SI(L(-1)C=4;ATTEINDRE(debut))
= ATTEINDRE(zone_quest)
= SINON.SI(nr_rep=bonne_rep)
    score=score+1
= ALERTE("Très bien "&nom&".|Votre note est de "&score&"/"&i&".";3)
= SINON.SI(nr_rep<>bonne_rep)
= ALERTE("Faux. La réponse était : "&REFTEXTE(ADRESSE(21+q-
1;2;1;FAUX;"C:\EXCEL\IDACT\IDACTI.XLS"))&".")
= FIN.SI()
    i=i+1
=SUIVANT()

pourcent=(score/(i-1))*100
note="Votre note est de : "&score&"/"&i-1
=SI(pourcent=100)
= ALERTE(note&"||EXCELLENT "&nom&" !!! Vous avez répondu bon à toutes les
questions.";3)

```

```
=SINON.SI(pourcent>=75)
= ALERTE(note&"||PAS MAL "&nom&" !!! Vous vous défendez bien.";3)
=SINON.SI(pourcent>=50)
= ALERTE(note&"||C'est MOYEN "&nom&" !!! Vous avez la moyenne, mais il faudra réviser.";3)
=SINON.SI(pourcent>=30)
= ALERTE(note&"||PAS TERRIBLE "&nom&" !!! Il est temps de réviser.";3)
=SINON.SI(pourcent>0)
= ALERTE(note&"||TRES PASSABLE "&nom&" !!! Que se passe-t-il ?";3)
=SINON.SI(pourcent=0)
= ALERTE(note&"||NUL, ARCHI NUL "&nom&" !!! Vous vous êtes trompé à toutes les
questions !!!";3)
=FIN.SI()
```

```
=ATTEINDRE(debut)
```

```
quitter
```

```
=ACTIVER("didacti.xls")
```

```
=FICHIER.FERMER(FAUX)
```

```
=FICHIER.FERMER(FAUX)
```

2ème partie de la macro-commande (à recopier à partir de la cellule L120C1 jusqu'en L149C7)

	1	2	3	4	5	6	7
fen_accueil							
fenêtre d'accueil							
				378	144		
5		134	18			DIDACTI v1.0	
5		8	53			Système didacticiel sous Microsoft EXCEL 4.0	
5		92	88			écrit par Benoît Rivière	
5		7	125			(c)1994 I.P.L. Informatique	
fen_etresur							
				362	106	Quitter DIDACTI...	
5		127	27			Êtes-vous sûr ?	
17		15	12			1	
4		260	75	88		Annuler	
3		157	75	88		Oui	
fen_question							
fenêtre affichant les questions/réponses							
				650	300	Question 1 sur 11	
1		452	265	179		Question suivante	
2		344	265	88		Résumé	
3		233	266	88		Arrêter	
14		17	92	612	158	Propositions de réponses	
11							1
12		40	120		20	du pop corn	
12						des ordinateurs	
12						des fours à micro-ondes	
12						des logiciels de jeu	
12						des microprocesseurs	
17		13	13			1	
5		65	12	560	70	Que commercialise Intel ?	

R.S.X. sur CPC

SAVECR

Une RSX (Resident System eXtension) est un programme en langage machine qui ajoute des commandes au Basic. Par exemple : ùBASIC, ùERA, ùDISC... Ces commandes, créées par l'utilisateur (Amstrad pour celles précitées) sont stockées en mémoire vive (RAM), c'est pourquoi il faut recharger les RSX à chaque allumage. Pour les instructions ajoutées par Amstrad, elles sont stockées en ROM, et automatiquement transférées en RAM à l'allumage.

Pour illustrer ce qui va suivre, nous allons prendre pour exemple le source du programme SAVECR qui a été publié en page 6 dans ORDIMAGE n°3. Il avait pour objet de sauver des pages écran en mémoire ou sur disque, puis de les faire réapparaître ; pour se faire, il créait quatre nouvelles instructions : ùSAUVECR, ùCHARGECR, ùSAUVE,@a\$, ùCHARGE,@a\$. Pour plus d'informations sur ces commandes, consultez l'article concerné.

NOTE : Les «ù» pour les claviers AZERTY et «|» pour les claviers QWERTY, précédant chaque RSX servent à indiquer à l'interpréteur BASIC qu'il s'agit de RSX.

Démarches

Pour créer des RSX, il faut tout d'abord réserver de la place en RAM ; pour cela on charge le nombre de RSX à ajouter dans la variable TAMPON (DEFS 4), on nomme la table de RSX (COMEXT par exemple), et on appelle &BCD1, afin d'initialiser les nouvelles RSX.

La table COMEXT est une sorte d'aiguilleur. Suivant la commande à exécuter (elles sont décrites dans la partie RSX, des lignes 230 à 310), l'ordinateur renvoie l'interpréteur vers un sous programme déterminé.

La partie RSX (lignes 230 à 310) doit être décrite pour toutes les commandes de la façon suivante :

```
DEFM "COMMAND"
DEFB "E"+&80
...
DEFB 0
```

Bibliographie

Pour plus de détails, vous pouvez consulter les publications suivantes :

- Programmation du Z80 par Rodney Zaks, édité par Sybex (1984),
- Auto formation à l'Assembleur, édité par Micro-Application,

- Le livre de l'Amstrad CPC464-CPC664, Tome 1, par Daniel Martin et Philippe Jadoul, édité par BCM (1985),

- La boîte à outils du programmeur, Comment créer de nouvelles instructions, par Jean-Claude Despoine, édité par Sybex (1987).

Benoît.

HONEYASSEM V2.1 1985

```

10 ;SAVECR : SAUVEGARDE D'ECRAN
20 ;
30 ;SAUVECR : SAUVE L'ECRAN EN MEMOIRE
40 ;CHARGEBCR : CHARGE L'ECRAN SAUVE
50 ;SAUVE,BA# : SAUVE L'ECRAN SUR DSK/K7
60 ;CHARGE,BA# : CHARGE UN ECRAN DU DSK/K7
70 ;
80 ;ECRIT PAR BENEDIT RIVIERE LE 25/4/92
90 ;(C)1992 I.F.L. INFORMATIQUE
100 ;
110 ;          ORG  &6000
120 INIRSA: EQU  &BCD1
130 TAMPON: DEFS 4
140         LD  BC,COMEXT:
150         LD  HL,TAMPON:
160         CALL INIRSA:
170         RET
180 COMEXT: DEFW RSX:
190         JP  SAUVE:
200         JP  CHARGE:
210         JP  SAUV2:
220         JP  CHARG2:
230 RSX:    DEFM "SAUVEC"
240         DEFB "R"+&80
250         DEFM "CHARGEBC"
260         DEFB "E"+&80
270         DEFM "SAUV"
280         DEFB "E"+&80
290         DEFM "CHARG"
300         DEFB "E"+&80
310         DEFB 0
320 ;SAUVE : SAUVE ECRAN EN MEMOIRE
330 SAUVE: CP 0
340 ;S'IL Y A DES PARAMETRES -> RET
350         RET NZ
360         LD  HL,&C000
370         LD  DE,&6500
380         LD  BC,&4000
390         LDIR
400         RET
410 ;CHARGE : MET A L'ECRAN L'IMAGE SAUVEE
420 CHARGE: CP 0
430 ;S'IL Y A DES PARAMETRES -> RET
440         RET NZ
450         LD  HL,&6500
460         LD  DE,&C000
470         LD  BC,&4000
480         LDIR
490         RET
500 ;
510 ;SAUV2 : SAUVEGARDE SUR DSK
520 ;
530 SAUV2: CP 1
540 ;SI NB PARAMETRES <> 1 -> RET
550         RET NZ
560         CALL BLC:
570         LD  B,12
580         LD  HL,NOM2:
590         LD  DE,X2:
600         CALL &BC8C
610         JR  NC,ERR2:
620         LD  HL,&C000
630         LD  DE,&4000
640         LD  BC,0
650         LD  A,2
660         CALL &BC98
670         JR  NC,ERR2:
680         JP  &BC8F
690 ;
700 ;CHARG2 : CHARGE ECRAN DU DSK
710 ;
720 CHARG2: CP 1
730 ;SI NB PARAMETRES <> 1 -> RET
740         RET NZ
750         CALL BLC:
760         LD  HL,NOM2:
770         LD  DE,X2:
780         LD  B,12
790         CALL &BC77
800         JR  NC,ERR2:
810         AND  &10
820         JP  NZ,ERR2:
830         LD  HL,X2:
840         CALL &BC83
850         JP  &BC7A
860 ERR2:   LD  A,7
870         JP  &BB5A
880 ;MISE A BLANC DE LA ZONE NOM2:
890 BLC:    LD  HL,NOM2:
900         LD  B,8
910 BLC2:   LD  (HL),32
920         INC HL
930         DJNZ BLC2:
940 ;RECUPERE LE NOM DU FICHIER
950 PRDNOM: LD  L,(IX+0)
960         LD  H,(IX+1)
970         LD  B,(HL)
980         INC HL
990         LD  E,(HL)
1000        INC HL
1010        LD  D,(HL)
1020        LD  HL,NOM2:
1030        LD  A,B
1040        CP  B
1050        JP  NC,COPNOM:
1060        LD  B,B
1070 ;RECOPIE LE NOM DU FICHIER DANS NOM
1080 COPNOM: LD  A,(DE)
1090        LD  (HL),A
1100        INC DE
1110        INC HL
1120        DJNZ COPNOM:

```

Dossier Trucs et Astuces sur Amstrad CPC

Il s'agit d'un recueil de programmes que je n'ai personnellement pas écrits (hormis SAVECR), mais recopiés de revues ou livres. Etant, me semble-t-il, intéressants par leur particularité, par les techniques de programmation mises en oeuvre, et par les résultats obtenus, je vous les soumettrai afin que vous puissiez les apprécier à leur valeur la plus juste. Ces programmes peuvent bien entendu être intégrés aux vôtres afin d'ajouter à ces derniers une touche plus «pro». Ce dossier est donc à conserver précieusement au même titre que le dossier des "Adresses des CPC". Benoît.

KELCPC

Ce court listing offre la possibilité de connaître le modèle d'Amstrad sur lequel, un programme est en train de tourner, ce qui peut être particulièrement utile, lorsqu'il s'agit d'adapter un programme en langage machine en fonction de l'ordinateur utilisé.

CLAVSON

Sonorise l'appui des touches par un bip.

OK

Permet de changer le message d'invite (Ready) à votre convenance (ici on l'a remplacé par Ok comme sur les MO5).

CURSCLIG

Le curseur inerte que vous aviez l'habitude de voir disparaîtra pour faire place à un curseur clignotant.

BORD

Anime le bord de l'écran.

FOND RAYE

Sans commentaire. Sur 664 et 6128, remplacez &B290 par &B730.

ECRITURE BICOLORE

Sans commentaire. L'écriture ne sera effectivement bicolore que dans les MODEs 0 et 1, puisqu'en MODE 2, il n'est possible d'afficher qu'une seule couleur texte.

DECODINK

Affiche les codes des couleurs de la palette.

ATTRB

Une des grandes carences des CPC est partiellement comblée, puisqu'il est maintenant possible d'afficher des caractères en grande taille.

SPRITE

Permet le déplacement de caractères dans toutes les directions sans effacer le fond de l'écran.

ECHOIMP

Attention aux espions : ce programme retranscrit toutes les sorties texte écran vers l'imprimante.

GOTO

Autorise enfin le paramétrage des mots-clefs GOTO, GOSUB... par l'intermédiaire de variables.

MERGE

Une erreur dans la conception de l'AMSDOS rend impossible l'utilisation des instructions MERGE, CHAIN MERGE sur les CPC464 équipés d'un lecteur de disquette. Ce défaut est maintenant réparé, grâce à ce programme fourni en son temps par Amstrad.

CATALOG

Affiche le catalogue de la disquette ; il est donc possible d'imprimer ce catalogue en détournant l'affichage écran vers l'imprimante par le biais du canal #8 (PRINT#8,... au lieu de PRINT), l'archivage des disquettes en est donc facilité.

ANALPRG

Fournit des informations relatives à la longueur, à l'adressage... du dernier fichier chargé en mémoire.

LOCKDSK

Permet de forcer Retry, Ignore ou Cancel lors d'erreurs disque signalées par l'AMSDOS.

FICHIERS EFFACES et USER

Si vous désirez récupérer des fichiers effacés (par inadvertance), sachez que dans certains cas il est possible de les retrouver. En effet, l'AMSDOS les affecte à l'USER 229, inaccessible avec la RSX ùUSER. Pour accéder à ce USER, il faut entrer la ligne suivante, en mode direct ou dans un programme : POKE &A701,229. Ensuite affichez le catalogue de la disquette (CAT), notez les fichiers désirés. Puis faire ùREN,@a\$,@b\$ où a\$="FICHIER" et b\$="0:FICHIER", puis ùUSER,0. Normalement les fichiers sont à nouveau présents. Dans certains cas, il est impossible de recouvrer des fichiers, notamment si depuis le malencontreux effacement des sauvegardes ont été effectuées sur cette même disquette.

Par ailleurs, on peut ainsi utiliser 256 USERS au lieu des 16 offerts par l'AMSDOS, et ce en entrant POKE A701,USER.

DEPROTEC

Déprotège des programmes BASIC sauvés avec l'option P (protégé), et uniquement des programmes BASIC. Une autre solution consiste à opérer un «détournement de vecteur» : POKE AC03,&AE: POKE &AC02,&45

:POKE &AC01,&32 (attention, contrairement au programme, cette succession de POKEs ne fonctionne que sur CPC 464).

RALENTI

Aussi surprenant que cela puisse paraître, il peut être utile de ralentir l'ordinateur, pour cela exécutez RALENTI.

RSX A NEANT

Pour que le système ne reconnaisse plus aucune RSX, y compris les RSX AMSDOS (ùDISC, ùERA...), tapez et validez : POKE &BCD4,&AF:POKE &BCD5,&C9.

IMPRIMANTE ES -TU LA ?

Pour savoir si l'imprimante est prête, insérez cette ligne dans vos programmes : IF INP(&F500) AND &X1000000 THEN PRINT"Imprimante non reliée !". Veillez à bien taper INP et non IMP. En jargon plus technique, cela consiste à tester si le BUSY est en mode READY ou non.

SAVECR

RSX permettant de sauver/charger des pages écran sur disquettes, cassettes ou en mémoire. Le source figure en page 54.

NOTEZ QUE :

Certaines routines, telle que ECHOIMP, gagneraient à être transformées en RSX, à ce sujet je vous invite à consulter l'article consacré aux RSX page 54.

Tous les programmes et routines fonctionnent sur 464, et doivent par conséquent tourner sans problème particulier sur les autres Amstrad de la gamme CPC (sauf mention contraire), hormis quelques problèmes d'adressage inhérent à l'architecture interne de ces machines.

Certains de ces programmes peuvent fonctionner ensemble, d'autres «planteront» le système. Si ce dernier événement venait à se produire, il est fort à parier que cela proviendrait du fait que des routines en langage machine seraient logées dans un même emplacement mémoire.

BIBLIOGRAPHIE

- Auto formation à l'Assembleur, édité par Micro-Application,
- Le livre de l'Amstrad CPC464-CPC664, Tome 1, par Daniel Martin et Philippe Jadoul, édité par BCM (1985).
- La boîte à outils du programmeur. Comment créer de nouvelles instructions, par Jean-Claude Despoine, édité par Sybex (1987),
- et bien sûr, les vieilles revues, riches en merveilles de programmation.

```

10 '464, 664 ou 6128 ???
20
30 DATA 14,0,205,21,185,124,50,32,128,201
40 RESTORE 30
50 FOR i=32768 TO 32777
60 READ b
70 POKE i,b
80 NEXT
90 CALL 32768
100 i=PEEK(32800)
110 IF i=0 THEN cpc=464 ELSE IF i=1 THEN cpc=664 ELSE IF i=2
    THEN cpc=6128
120 PRINT "vous avez un CPC "cpc".

```

```

10 'CLAVESON : Clavier sonore
20
30 DATA f5,e5,21,ff,b4,7e,fe,00,c2,18,a0,23,7d,fe,08,c2,05,a
0,e1,f1,c3,4a,13,c9,3e,07,cd,21,a0,b7,c3,12,a0,cf,00,94
40 FOR x=&A000 TO &A023:READ a$:POKE x,VAL("%"+a$):NEXT
50 POKE &B003,&C3:POKE &B004,&0:POKE &B005,&A0

```

```

10 'Changeur d'invite : No READY out UK...
20
30 MEMORY &A560
40 CALL &BFE2,&AD01,&3203,&3EAC,&222,&23EB,&74A5,&9221,&B8ED
,&1ED0,&0CB21,&65,&1B9,&0D,&12AF,&A5CC
50 POKE &A5CC,62:POKE &A5CC,0
60 a#="0"+CHR$(10)+CHR$(13)+CHR$(10)
70 CALL &E9F2,&F9,&1613,&3EE7,&127E,&23B0,&8D2A,&A5CC

```

```

10 'CURSCLIG : Curseur clignotant
20
30 MEMORY &7FFF
40 FOR n=&8000 TO &8029:READ d$:POKE n,VAL("%"+d$):NEXT
50 CALL &8000
60 DATA 21,2a,80,06,B1,0e,00,11,0e,80
70 DATA cd,d7,bc,c9,c5,d5,e5,f5,3a,ff
80 DATA 7f,e5,10,2B,05,cd,7b,bb,18,03
90 DATA cd,7e,bb,21,ff,7f,34,f1,e1,d1
100 DATA c1,c9

```

```

10 'Bord
20
30 CALL &BFEA,&EC1B,&1551,&ED49,&ED7F,&1001,&5740,&F61F,&E&
A,&DBBB,&9CD

```

```

10 'Fond ray(
20
30 FOR i=0 TO 255
40 POKE &B290,i
50 PRINT "Salut !!"
60 NEXT
70 POKE &B290,0

```

```

10 'Ecriture bicolore
20
30 MODE 1
40 FOR i=0 TO 255
50 POKE &B1CF,i
60 PRINT "Salut !!"
70 NEXT
80 POKE &B1CF,128

```

```

10 MERGE : le chainage de programme enfin possible sur 4k4
@ Disquette
20
30 MEMORY HIMEM-41
40 DEF FNmsb(a)=%FF AND INT(a/256)
50 DEF FNlsb(a)=%FF AND INT(a)
60 FOR i=HIMEM+1 TO HIMEM+38
70 READ b,te:POKE i,byte
80 NEXT
90 POKE HIMEM+3, FNlsb(HIMEM+39)
100 POKE HIMEM+4, FNmsb(HIMEM+39)
110 POKE HIMEM+9, FNlsb(HIMEM+41)
120 POKE HIMEM+10, FNmsb(HIMEM+41)
130 POKE HIMEM+18, FNlsb(HIMEM+1)
140 POKE HIMEM+19, FNmsb(HIMEM+1)
150
160 POKE HIMEM+39, PEEK(&BC80+0)
170 POKE HIMEM+40, PEEK(&BC80+1)
180 POKE HIMEM+41, PEEK(&BC80+2)
190 POKE &BC80+3, &C3
200 POKE &BC80+1, FNlsb(HIMEM+1)
210 POKE &BC80+2, FNmsb(HIMEM+1)
220 DATA &e5, &2a, &00, &00, &22, &80, &bc
230 DATA &0a, &00, &00, &32, &82, &bc
240 DATA &cd, &80, &bc, &21, &00, &00
250 DATA &22, &21, &bc, &21, &80, &bc
260 DATA &7a, &c3, &e1, &d8, &c8, &fe, &1a
270 DATA &37, &3f, &c0, &b7, &37, &c9

```

10 MERGE : le chainage de programme enfin possible sur 4k4

```

20
30 DATA &e5, &2a, &00, &00, &22, &80, &bc, &0a, &00, &00, &32, &82, &bc, &cd, &80, &bc, &21, &00, &00, &22, &21, &bc, &21, &80, &bc, &7a, &c3, &e1, &d8, &c8, &fe, &1a, &37, &3f, &c0, &b7, &37, &c9
40 DATA &e5, &2a, &00, &00, &22, &80, &bc, &0a, &00, &00, &32, &82, &bc, &cd, &80, &bc, &21, &00, &00, &22, &21, &bc, &21, &80, &bc, &7a, &c3, &e1, &d8, &c8, &fe, &1a, &37, &3f, &c0, &b7, &37, &c9
50 DATA &e5, &2a, &00, &00, &22, &80, &bc, &0a, &00, &00, &32, &82, &bc, &cd, &80, &bc, &21, &00, &00, &22, &21, &bc, &21, &80, &bc, &7a, &c3, &e1, &d8, &c8, &fe, &1a, &37, &3f, &c0, &b7, &37, &c9
60 DATA &e5, &2a, &00, &00, &22, &80, &bc, &0a, &00, &00, &32, &82, &bc, &cd, &80, &bc, &21, &00, &00, &22, &21, &bc, &21, &80, &bc, &7a, &c3, &e1, &d8, &c8, &fe, &1a, &37, &3f, &c0, &b7, &37, &c9
70 'call 43700, n1, n2, n3, n4, n5$
80 'n1 -> coordonnée x (d(part)
90 'n2 -> coordonnée y (d(part)
100 'n3 -> coordonnée h (arrivé)
110 'n4 -> coordonnée y (arrivé)
120 'n5 -> caractère à d(placer)

```

10 DATALOG : Catalogueur de disquette

```

20
30 MODE 2
40 t:=1
50 POKE &BB5A, &C9: CALL &BC9B, &4000: POKE &BB5A, &CF
60 PRINT "nom fich.ext taille nom fich.ext taille nom fich.ext
t taille nom fich.ext taille"
70 PRINT STRING$(80, "-");
80 FOR i=4000 TO &4800 STEP 14
90 IF PEEK(i)<255 THEN 150
100 PRINT TAB(t);
110 FOR n=1 TO 8:PRINT CHR$(PEEK(i+n));NEXT:PRINT " ";
120 FOR e=0 TO 2:PRINT CHR$(PEEK(i+n+e) AND 127);NEXT
130 PRINT USING " ## ko";PEEK(i+12);
140 ta:=ta+20:IF ta=80 THEN ta=0:PRINT
150 NEXT
160 PRINT

```

```

10 ANALPRG : Affichage des caractéristiques d'un fichier
20
30 CLEAR:ON ERROR GOTO 40:MODE 2:INK 0,0:INK 1,9:BORDER 0:WI
NDOW 1,39,1,25:PLOT 310,0:DRAWR 0,400,1:PEN 1:DEF FNp(a)=PEE
K(a)+256*PEEK(a+1):DIR:a$(0)="bas":a$(1)="bin":a$(2)="ecr":
a$(3)="asc":a$(5)="prot":WINDOW 41,80,1,25:INPUT "Programme "
;n$:OPENIN n$
40 p=PEEK(&A7&7):PRINT "Nom:";FOR i=&A70A TO &A70A+12:PRINT
CHR$(PEEK(i));NEXT:PRINT:PRINT "Type:";a$(p AND 14)/2);a$(p
AND 1)+4):PRINT "Debut:";HEX$(FNp(&A7&A));PRINT " Longueur
:";HEX$(FNp(&A7&D));PRINT "(Execution:";HEX$(FNp(&A7&F))":":E
ND

```

10 LOCKDSK : Annulation des erreurs disque

```

20
30 MODE 1
40 LOCATE 1,1:PRINT "1) Retry automatique"
50 LOCATE 1,3:PRINT "2) Ignore automatique"
60 LOCATE 1,5:PRINT "3) Cancel automatique"
70 LOCATE 3,10:INPUT "CHOIX ? ",choix
80 ON CHOIX GOTO 100,120,140
90 GOTO 70
100 POKE &BB0b, &3E:POKE &BB07, ASC("R"):POKE &BB08, &C9
110 GOTO 150
120 POKE &BB0b, &3E:POKE &BB07, ASC("I"):POKE &BB08, &C9
130 GOTO 150
140 POKE &BB0b, &3E:POKE &BB07, ASC("C"):POKE &BB08, &C9
150 END

```

10 VALENTI : Valentit l'ordinateur

```

20
30 MEMORY &7FFF
40 FOR f=327&8 TO 32902
50 READ a:POKE f,a
60 NEXT
70 CALL &B000
80 NEW
90
100 'utilisation :
110 'call &B00b:Run"nomprg
120 'call &B017 -> annule valenti
130
140 DATA 33,41,128,8,129,17,29,128,195,239,188,33,35,128,17,
50,0,1,50,0,195,233,188,33,35,128,195,234,188,205,24,187,195
,3,187

```

10 GOTO : permet de faire suivre Goto d'une variable

```

20
30 MEMORY 41999:FOR x=42000 TO 42043:READ a$:POKE x,VAL("&"+
a$):NEXT:POKE &AC04, &C3:POKE &AC05, &10:POKE &AC0b, &A4
40 DATA c1, c1, e5, 2b, 2b, 2b, 2b, 2b, 7e, fe, 9f, 28, 10, fe, a0, c2, 97, c
a, 23, 23, cd, 91, ce, cd, 9a, e7, c1, 2b, c9, 23, 23, cd, 91, ce, cd, 9a, e7, e
b, e1, cd, f0, c8, 2b, c9

```

SECTEURS

Pour connaître le numéro du premier secteur de la disquette, il faut entrer :
 OPENOUT"B": CLOSEOUT:PRINT
 PEEK(&A89F).

LECTEUR COURANT

Pour savoir quel est lecteur de disque sur lequel l'ordinateur va opérer (lecteur courant), exécutez : PRINT PEEK(&A700).

MODE sans CLS

Changez de MODE sans effacer l'écran (quel rêve !) : POKE &BDEB,&C9.

LIST SPECIAL

Si vous entrez en mode direct : POKE 368,0:POKE 369,255, la commande LIST, n'affichera que la première ligne de votre programme.

De même : POKE 370,0:POKE 371,0 : la première ligne devient invisible (tout simplement parce qu'elle devient ligne 0) ce qui permet d'entrer une ligne de protection. Si vous sauvegardez le programme après l'exécution de cette ligne, votre programme demeurera toujours avec une ligne 0. Pour annuler ces POKES, tapez RENUM.

```

10 'DECODINK : D(codage de la palette
20
30 DIM enc(16),cod(32)
40 FOR i=0 TO 31:READ a#:cod(VAL("&"+a#))=i:NEXT
50 DATA 14,04,15,1c,18,1d,0c,05
60 DATA 0d,16,06,17,1e,00,1f,0e
70 DATA 07,0f,12,02,13,1a,19,1b
80 DATA 0a,03,0b,01,08,09,10,11
90 a=&B7D5:IF PEEK(16)=128 THEN a=&B1EB
100 MODE 1:LOCATE 1,1:FOR i=0 TO 15:enc(i)=cod(PEEK(a+i)):PR
INT "INK"i", "enc(i),:NEXT

10 'DEPROTEC : Diprotecteur de programme Basic/P
20
30 FOR i=0 TO 9:READ a#:POKE &B739+i,VAL("&"+a#):NEXT:POKE &
3A,&B7
40 DATA f5,3e,00,32,45,ae,f1,c3,39,b9

10 'ECHOIMP : Imprime tout ce qui est tapé
20
30 call &a600 -> active echoimp
40 call &a617 -> d(active echoimp
45
50 DATA 21,5a,bb,11,26,a6,01,03,00,ed,b0,21,23,a6,11,5a,bb,0
1,03,00,ed,b0,c9
60 DATA 21,26,a6,11,5a,bb,01,03,00,ed,b0,c9
70 DATA c3,29,a6,00,00,00,cd,2e,bd,38,fb,cd,2b,bd,18,f3
80 FOR a=&A600 TO &A632
90 READ v#:POKE a,VAL("&"+v#):NEXT
100 'save"echoimp",b,&a600,&33

```

```

10 'Ecriture en différentes tailles
20
30 MEMORY &7FFF
40 a=&B000:FOR i=100 TO 400 STEP 10:s=0
50 FOR j=1 TO 10
60 READ x#:xx=VAL("&"+x#):POKE a,xx:s=s+xx:a=a+1
70 NEXT
80 READ xx:IF s<>xx THEN PRINT"erreur ligne":END
90 NEXT
100 CALL &B000
110
120 'syntaxe : DH, DL, DT, ST
130 'ne pas oublier le ' devant les RSX !
140
150 DATA 2a,d4,bd,22,2b,80,01,13,80,21,829
160 DATA 0f,80,c3,d1,bc,00,00,00,00,21,768
170 DATA 80,c3,2d,80,c3,34,80,c3,3b,80,1253
180 DATA c3,42,80,44,cc,44,c8,44,d4,53,1292
190 DATA d4,00,c3,ff,ff,21,49,80,22,d4,1397
200 DATA bd,c9,21,97,80,22,d4,bd,c9,21,1371
210 DATA ce,20,22,d4,bd,c9,2a,2b,80,22,1217
220 DATA d4,ed,c9,f5,e5,1e,02,cd,a5,bb,1665
230 DATA 3e,19,cd,5a,bb,3e,ff,cd,5a,bb,1368
240 DATA 0e,06,06,04,56,af,cb,4b,20,0e,617
250 DATA cb,22,cb,22,cb,22,cb,22,18,04,976
260 DATA cb,27,cb,27,cb,22,30,02,fb,03,1020
270 DATA 10,f4,cd,5a,bb,23,0d,20,db,3e,1103
280 DATA ff,e1,d5,e5,cd,2a,80,e1,d1,1d,1760
290 DATA c8,3e,09,cd,5a,bb,f1,24,e5,18,1283
300 DATA b6,f5,2d,e5,1e,02,cd,a5,bb,3e,1352
310 DATA 19,cd,5a,bb,3e,ff,cd,5a,bb,06,1312
320 DATA 04,cb,4b,20,04,23,23,23,23,7e,594
330 DATA cd,5a,bb,cd,5a,bb,23,10,fb,3e,1323
340 DATA ff,e1,d5,e5,cd,2a,80,e1,2c,d1,1775
350 DATA 1d,c8,f1,e5,18,ce,f5,2d,e5,1e,1478
360 DATA 04,cd,a5,bb,3e,19,cd,5a,bb,3e,1192
370 DATA ff,cd,5a,bb,cb,43,28,04,23,23,1121
380 DATA 23,23,0e,04,06,04,56,7b,fe,03,564
390 DATA 30,08,cb,22,cb,22,cb,22,cb,22,1004
400 DATA af,cb,27,cb,27,cb,22,30,02,fb,1192
410 DATA 03,10,f4,cd,5a,bb,cd,5a,bb,23,1262
420 DATA 0d,20,d9,3e,ff,e1,d5,e5,cd,2a,1493
430 DATA 80,e1,d1,cb,43,c2,21,81,2c,7b,1355
440 DATA fe,03,c2,2e,81,3e,09,cd,5a,bb,1179
450 DATA 2d,24,f1,1d,c8,f5,e5,18,9e,00,1207

```

Les Partiels en Turbo Pascal

Un partiel ou fichier de recouvrement (overlay en anglais) est une partie de programme ne résidant en mémoire que de façon temporaire. Il porte l'extension «.OVR». Le programme exécutable (.EXE) en charge les parties dont il a besoin. Cette technique de programmation a un avantage de taille : économiser de la mémoire.

L'exemple ci-après est organisé en trois fichiers :

- le programme principal (EXEMPLE.PAS),
- l'unité à transformer en partiel (UNIT1.PAS),
- et l'unité qui doit s'occuper de gérer le partiel (UNITOVR.PAS).

Le **programme principal** (ici EXEMPLE) doit faire appel à l'unité OVERLAY dans sa déclaration d'unités (USES). Ensuite toujours sur la ligne de déclaration d'unités, il faut indiquer l'unité qui installe les partiels (UNITOVR), puis les unités à installer en partiel (UNIT1).

L'**unité à transformer en partiel** doit contenir la directive {\$O+} afin d'indiquer

au compilateur que cette unité doit être préparée à une hypothétique transformation en partiel. A noter que vous pouvez spécifier plusieurs unités à transformer en fichier de recouvrement, la procédure à suivre est rigoureusement identique. Même si plusieurs unités sont transformés en partiel, elles seront fondues en un seul fichier «OVR».

L'**unité de gestion du partiel** ne peut elle même être un partiel. Elle a pour objet de vérifier si la compilation s'est bien déroulée, si le fichier de recouvrement est disponible, et éventuellement, car cela n'est pas indispensable, de charger le partiel en mémoire EMS, si celle-ci est disponible en capacité suffisante. L'intérêt de loger un partiel en EMS est la vitesse d'exécution qui en résulte, en évitant au maximum les accès disque. En outre le chargement de partiels en EMS libère la mémoire conventionnelle (les fameux 640 premiers Ko), déjà bien entamée par le DOS et par le programme principal.

Pour utiliser l'exemple, mais également pour compiler vos propres programmes, sachez que vous devez toujours utiliser l'option «Compiler sur Disque» et non en mémoire à cause du fichier de recouvrement. Vous devez compiler les unités (qui seront ou non transformées en partiel), puis le programme principal. Benoît.

PROGRAMME EXEMPLE

```

program exemple:
{programme utilisant la technique du fichier de
recouvrement}
uses
  crt, dos,
  overlay,
  unitovr,
  unit1:
  {$O unit1}

begin
  clrscr;
  direbonjour(15,20,'Jean Paul');
  readln;
end.

```

UNITE UNITOVR

```

unit unitovr;
{inclut des unité en tant qu'OVR dans le
programme exemple}

interface

implementation

uses overlay;

const
  ficovr : string[12] = 'EXEMPLE.OVR';

procedure Initialise_Ovr;
begin
  OvrInit(ficovr);
  if OvrResult<>ovrOk then
  begin
    case OvrResult of
      ovrError:
        writeln('Programme sans partiel. ');
      ovrNotFound:
        WriteLn('Fichier '+ficovr+' non trouvé. ');
    end;
    Halt(1);
  end;
  OvrInitEMS;
  case OvrResult of

```

```

    ovrIOError :
      begin
        writeln('Erreur d"E/S sur le fichier de
recouvrement. '); readln;
      end;
    ovrNoEMSDriver :
      begin
        writeln('Mémoire EMS non trouvée. ');
        readln;
      end;
    ovrNoEMSMemory:
      begin
        writeln('Pas assez de mémoire EMS. ');
        readln;
      end;
  end;
end; {Initialise_Ovr}
begin
  Initialise_Ovr;
end.

```

UNITE UNIT1

```

unit unit1;
{$O+} {Cette unité est préparée à une éventuelle
transformation en OVR}

interface

uses crt;

procedure DireBonjour(x,y : integer; nom :
string);

implementation

procedure DireBonjour(x,y : integer; nom :
string);
begin
  gotoxy(x,y);
  writeln(nom);
end; {DireBonjour}

begin
end.

```

Dossier : les mathématiques et l'informatique

Je vais apporter quelques éclaircissements concernant les routines mathématiques et trigonométriques parues dans le dernier numéro, ainsi que quelques compléments. Benoît.

Routines arithmétiques

Dans le numéro précédent, certaines routines ont été indiquées sans description précise quant à leur utilisation.

464	664	6128	EFFET
BD40	BD61	BD64	Transforme une représentation entière en une représentation à virgule flottante. L'entier doit être chargé dans HL et l'adresse où doit être stockée la représentation en virgule flottante dans DE
BD5B	BD7F	BD82	Effectue l'opération HL-DE. Le résultat est rangé dans HL
BD5E			Effectue l'opération DE-HL. Le résultat est rangé dans HL
BD64	BD64	BD88	Effectue l'opération HL/DE. Le résultat est rangé dans HL
BD88	BDA9	BDAC	Calcule le sinus d'un angle. HL doit être pointé sur l'adresse de la représentation en virgule flottante de l'angle. Le résultat est rangé à cette adresse
BD8B	BDAC	BDAF	Calcule le cosinus d'un angle. HL doit être pointé sur l'adresse de la représentation en virgule flottante de l'angle. Le résultat est rangé à cette adresse

Compléments

Pour connaître le **mode trigonométrique** en cours : PRINT PEEK(&B8F7). Si le résultat est 0, il s'agit du mode RADIAN, sinon si la réponse est 255, il s'agit du mode DEGRE. Sur CPC664, il faut remplacer &B8F7 par &B113.

Les concepteurs du logiciel interne du CPC464 ont commis une erreur sur la fonction DECS. Pour l'utiliser, il faut entrer

a\$=DECS((a;"###.##) au lieu de a\$=DECS(a;"###.##). Cette erreur a été heureusement réparée sur les autres modèles de la gamme.

Pour calculer la **dérivée de f(x)** au point x, il faut utiliser la formule suivante : $f'(x) = (f(x+a) - f(x-a)) / 2a$. Plus a sera petit plus le résultat sera précis (a=0,00001 semble être un bon compromis).



Analyse fonctionnelle sur tableur

Page 2 du numéro précédent, nous avons abordé les principaux éléments à connaître afin de manipuler EXCEL. L'application proposée ci-après est beaucoup plus ambitieuse, puisqu'elle permet, à partir de bilans comptables et d'autres informations, de dresser des bilans fonctionnels et de calculer des ratios... Le travail de l'analyste financier est ensuite, à partir des documents ainsi établis, d'apporter des solutions à certains problèmes, de rendre des conclusions. Bien que ces calculs s'adressent plus particulièrement à des comptables de formation, toute personne intéressée par le maniement d'un tableur tel EXCEL pourra tirer profit de cette application, puisqu'il y sera notamment évoqué :

- la notion de classeur,
- les liaisons interfeuilles,
- les macrocommandes.

Avant de se plonger plus avant dans la lecture de cet article, il est préférable tout d'abord d'avoir déjà utilisé EXCEL (pour cela vous pouvez vous référer à l'initiation pratique dont il est fait allusion précédemment), et ensuite d'avoir quelques notions de programmation afin de maîtriser plus facilement la construction des macrocommandes. Il est à noter que la macrocommande n'est pas indispensable au bon fonctionnement de l'application, mais constitue tout de même un perfectionnement non négligeable. Benoît.

1. Tout d'abord

A l'aide de logiciels, tel EXCEL, il est maintenant possible d'établir des comparaisons, des prévisions en toute simplicité, les données étant réactualisées en

temps réel à chaque modification. L'application qui va être développée est d'un concept très simple : à partir des documents remis annuellement à l'Administration fiscale (bilans, comptes de résultat, annexes...) par l'entreprise, il s'agira de demander à

l'ordinateur, à l'aide formules de calcul, de calculer automatiquement des documents d'analyse (bilans fonctionnels, ratios, tableaux de financement...). Les premiers seront nommés *documents de saisie* (on se contente de recopier une à une les données des documents de fin d'exercice), les seconds seront appelés *documents calculés* (grâce aux documents de saisie, on obtiendra des résultats calculés automatiquement).

Afin de rendre concret ce qui suit, nous allons utiliser un exemple fictif. Cette application, grâce à sa relative souplesse, peut être réutilisée pour d'autres exemples, et éventuellement dans une petite entreprise. Il

suffit pour cela de changer la raison sociale et les autres renseignements requis dans 'INFOD.XLS' ainsi que rentrer les bilans fiscaux, de retraiter le crédit-bail...

Si vous ne désirez pas approfondir votre connaissance en matière de tableur, mais uniquement disposer d'une application établissant automatiquement des documents d'analyse, sachez que, cette application étant livrée «clef en main», vous pouvez ne consulter que les premiers paragraphes du chapitre qui suit (à propos de la mise en place des feuilles de calcul) puis copier directement les formules de calcul dans les feuilles adéquates.

2 Concepts utilisés et mise en oeuvre de l'application

Cette application nécessite l'utilisation de plusieurs feuilles de calcul dépendantes les unes des autres. Afin de simplifier l'utilisation de cette application les feuilles de calcul sont regroupées dans un classeur nommé 'XLFI.XLW'.

Les feuilles de calcul de l'application

Le tableau intitulé *Liste des feuilles de calcul du classeur 'XLFI.XLW'* indique le nom des feuilles de calcul qui devront être créées. Pour rappel, pour créer une nouvelle feuille de calcul, il faut faire 'Fichier/Nouveau document.../Feuille' et pour créer une feuille macrocommande : 'Fichier/Nouveau/Macro'. Enregistrez les feuilles au fur et à mesure de leur création.

Le classeur de l'application

Une fois que vous aurez créé toutes les feuilles, vous ouvrirez un classeur. Un classeur regroupe des feuilles. Il ajoute à chaque feuille intégrée, de petites icônes en bas à droite, permettant de «naviguer» d'une feuille à l'autre en toute liberté.

Pour créer le classeur faites 'Fichier/Nouveau document.../Classeur'. Pour insérer toutes les feuilles dans le classeur, cliquez sur le bouton 'ajouter', il s'affichera la liste des feuilles ouvertes, vous devrez cliquer sur le nom des feuilles, puis pointer sur 'ajouter' et réitérer cette opération jusqu'à ce que toutes les feuilles soient insérées dans le classeur.

Ensuite, pour chaque feuille vous ferez 'option...' et cochez 'Fichier séparé (non inclus)'. N'oubliez pas de sauvegarder le classeur : 'Fichier/Enregistrer un classeur...' sous le nom de XLFI.XLW.

Points de suite

Afin que les tableaux soient plus lisibles, on peut ajouter des points de suite. Faites 'Format/Nombre', cliquez dans la zone de saisie 'Code' et entrez '@*.'. Ensuite faites 'Format/Nombre'+ '@*.' et des points de suite apparaîtront dans les cellules indiquées.

Impression

Avant d'imprimer les documents, définissez une zone autour des tableaux, puis faites 'Options/Zone d'impression'. Allez également dans 'Fichier/Mise en page', demandez le centrage horizontal et vertical, enlevez le bas et haut de page, décochez les cases quadrillage de cellules et en-tête de colonnes et lignes, au besoin changez l'échelle d'impression (94 % par exemple).

Utilisations ultérieures de l'application

Lorsque vous voudrez réutiliser cette application, dans 'Fichier/Ouvrir' vous n'indiquerez que 'XLFI.XLW', puisque le classeur se charge d'ouvrir toutes les feuilles qu'il contient.

Liste des feuilles de calcul du classeur 'XLFI.XLW'

Feuille de calcul	Contenu
INFOD.XLS	Informations diverses
RESULTAT.XLS	Comptes de résultat
BILACTI1.XLS	Actif du bilan fiscal de l'exercice 1991
BILACTI2.XLS	Actif du bilan fiscal de l'exercice 1992
BILACTI3.XLS	Actif du bilan fiscal de l'exercice 1993
BILPASS.XLS	Passif du bilan fiscal des exercices 1991 à 1993
CRDBAIL.XLS	Retraitement du crédit-bail
BILFO.XLS	Bilans fonctionnels sur 3 ans (exercices 1991 à 1993)
RATIOFO.XLS	Ratios issus des bilans fonctionnels
ANABILFO.XLS	Calcul des F.R.N.G., B.F.R....
VERIF.XLM	Macrocommande.

Les formules de calcul de l'application

Les formules qui sont exposées ci-dessous, ne concernent que la première année étudiée (ici 1991). Les formules des exercices 1992 et 1993 sont identiques, mis à part le fait qu'il faut leur affecter des références cellulaires différentes (C12 au lieu de B12 par exemple), ou modifier les noms de cellule (AmortFict2 au lieu de AmortFict1).

Après avoir ouvert toutes les feuilles de calcul, entrez une à une toutes les formules indiquées ci-après dans les feuilles adéquates, puis faites les encadrements, les mises en forme... N'oubliez pas de sauvegarder les feuilles sous les noms indiqués précédemment.

Notion de liaison

Chaque document a été créé sur une feuille de calcul différente. Pour communiquer des données d'une feuille à l'autre, la formule appelante doit indiquer le nom de la feuille de calcul appelée suivi des références cellulaires ou du nom de la ou des cellules appelées. Par exemple pour calculer le ratio de *financement des investissements*, on entre la formule suivante :

```
=BILFO.XLS!ressetabl/BILFO.XLS!empetabl
*100
```

Cette formule indique qu'il faut prendre le total des ressources stables de l'exercice 1 (ici 1991) dans le Bilan fonctionnel et le

diviser par les emplois stables de la même année.

Macro-commande

Une des carences de cette application est le fait que l'ordinateur calcule des résultats à partir de données saisies par l'utilisateur. Le risque d'erreur peut être important (faute de frappe...). Il est donc souhaitable de vérifier la véracité des données saisies, par exemple en testant l'égalité entre Passif et Actif. Ce test étant automatisable, je propose donc de créer une macro-commande.

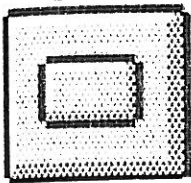
Une macro-commande est une suite de commandes permettant l'exécution automatisée de tâches répétitives plus ou moins complexes. Pour un approfondissement sur les macro-commandes, je vous invite à vous reporter page 48.

La macro-commande développée compare le passif et l'actif des bilans sur les 3 exercices, et les résultats comptables. Elle tolère une différence de 10 FF maximum (les sommes apparaissant dans les documents fiscaux étant toujours arrondies). Cette comparaison s'effectue par soustraction. En cas de différence notable, une fenêtre affiche un message d'erreur et indique les résultats incriminés.

Avant de recopier la macro, cochez L1C1 dans Options/Environnement.../groupe Afficher. Copiez la macro-commande sur la colonne 1 de la feuille macro VERIF.XLM. Veillez à bien entrer la macro à partir de la

cellule L1C1. Une fois la macro saisie, faites 'Sélection/Définir un nom' puis entrez 'Fonction_enregistrement', référence : 'L1C1:L21C1'.

Afin de faciliter l'utilisation de cette macro-commande, je propose de la lier à un bouton. Pour se faire, 'Options/Barres d'outils', cliquez 'Personnaliser', 'catégories : Dessin'. Dans le groupe outils, cliquez sur la cinquième icône en haut en partant de la gauche (cf. figure) et faites la glisser sur la barre d'outils d'Excel (sous la barre de menus).



Ensuite, placez-vous sur la feuille de calcul 'INFOD.XLS', cliquez sur l'icône que vous venez d'ajouter à la barre d'outils, le

curseur change de forme. Dessinez le bouton, par exemple sur la cellule A44. Une fenêtre intitulée 'Affecter à l'objet' apparaît, entrez alors 'Verif.xml!L1C1' dans la zone de saisie 'Référence', puis validez (Ok). Cliquez sur le bouton que vous venez de créer et appelez le 'TEST' (vous pouvez changer de police, de taille, de mise en forme...), puis cliquez en dehors du bouton. La macro Verif.XLM est maintenant liée au bouton 'TEST'.

Il faut maintenant tester si l'application fonctionne correctement. Cliquez sur le bouton 'TEST', si une fenêtre affiche le message 'Aucune erreur de saisie apparente', vous n'avez pas dû commettre de faute en saisissant vos données, sinon vérifiez les données fautives...

```
Macro-commande destinée à vérifier l'intégrité des données saisies
Application 'XLF1.XLM'
erreursaisie=0
=POUR("lig"36:41:0)
=POUR("col"2:4:1)
=SI(ABS(EVALUER(ADRESSE(lig;col)-FAUX("infod.xls"))-EVALUER(ADRESSE(lig+1;col)-FAUX("infod.xls")))>10)
  erreursaisie=1
=SI(lig=36)
  ALERTE("Différence Actif/Passif Bilan fiscal de l'exercice "&"D:\SEV\WLF\INFOD.XLS"&exercice1+col-2&" trop importante.")
=SI(lig=38)
  ALERTE("Différence Résultat comptable de l'exercice "&"D:\SEV\WLF\INFOD.XLS"&exercice1+col-2&" trop importante.")
=SI(lig=40)
  ALERTE("Différence Actif/Passif du Bilan fonctionnel de l'exercice "&"D:\SEV\WLF\INFOD.XLS"&exercice1+col-2&" trop importante.")
=FIN.SI()
=FIN.SI()
=SUivant()
=SUivant()
=SI(erreursaisie=0)
  ALERTE("Aucune erreur de saisie apparente.")
=FIN.SI()
=RETOUR()
```

3. Formules et Tableaux de l'application

Formules de la page d'informations diverses

La feuille d'informations diverses est un tableau dans lequel il faut indiquer :

- des informations d'ordre général (Raison Social, exercice comptable...),
- des informations complémentaires permettant d'établir les bilans fonctionnels (effets escomptés non échus, impôt sur les sociétés...)

Cette feuille fournit un tableau récapitulatif des résultats issus des bilans fiscaux, comptes de résultats, des bilans fonctionnels afin que l'utilisateur puisse vérifier qu'il n'a commis aucune erreur de saisie.

	A	B	C	D
1	Informations de base			
2				
3	RAISON SOCIALE			
4	Année premier exercice traité	ORDIMAGE S.A.		
5	l'exercice débute le 1er avril et s'arrête au 30 mars de l'année suivante	1991		
6	Nombre de jours d'activité (*)	1er exercice traité : 1/1/91-31/12/91		
7	Période annuelle durant l'entreprise cesse ses activités	330		
8	du 1er au 30 août			
9	(*) par simplification, on considère des mois de 30 jours.			
10	Informations complémentaires nécessaires à l'élaboration du bilan fonctionnel			
11	ORDIMAGE S.A.			
12		1991	1992	1993
13	Immobilisations - "exploitation" (en % des immobilisations)	100%	100%	100%
14	Immobilisations - "hors exploitation" (en %)	0%	0%	0%
15	E.E.N.E. (effets escomptés non échus)/affacturage	25 532,00	41 936,00	52 935,00
16	Autres créances d'exploitation	0	0	0
17	Autres créances hors exploitation			
18	Charges constatées d'avance hors exploitation	0	0	0
19	Produits constatés d'avance hors exploitation	0	0	0
20	Intérêts courus non échus	5 250	22 500	31 500
21	Impôts sur les bénéfices	0	0	0
22	Autres dettes d'exploitation	0	0	0
23	Autres dettes hors exploitation	0	0	0
24	Ecart de conversion actif - Créance d'exploitation	0	0	0
25	Ecart de conversion actif - Créance hors exploitation	0	0	0
26	Ecart de conversion actif - Dette d'exploitation	0	0	0
27	Ecart de conversion actif - Dette hors exploitation	0	0	0
28	Ecart de conversion passif - Créance d'exploitation	0	0	0
29	Ecart de conversion passif - Créance hors exploitation	0	0	0
30	Ecart de conversion passif - Dette d'exploitation	0	0	0
31	Ecart de conversion passif - Dette hors exploitation	0	0	0
32				
33	Synthèse permettant une vérification rapide			
34	ORDIMAGE S.A.			
35		1991	1992	1993
36	Bilan - total ACTIF	870 945,00	737 460,00	741 655,00
37	Bilan - total PASSIF	870 945,00	737 460,00	741 655,00
38	Résultat figurant au bilan	-36 955,00	75 415,00	-25 016,00
39	Résultat figurant au compte de résultat	-36 955,00	75 415,00	-25 016,00
40	Bilan fonctionnel - total ACTIF	751 027,00	907 896,00	942 480,00
41	Bilan fonctionnel - total PASSIF	751 027,00	907 896,00	942 480,00

Formules de calcul de la feuille d'informations diverses

Références cellules	Nom cellules	Formule
B3	RaisonSoc	
B4	exercice1	
B6	NbJActivite	
B13	immexpl1	
B14	immhexpl1	=1-immexpl1
B15	EENE1	
B16	autrecexpl1	=BILACTI1.XLS!autrcr-autrechrhexpl1
B17	autrechrhexpl1	
B18	chconstavhexpl1	
B19	prdconstavhexpl1	
B20	ICNE1	
B21	impothbenef1	
B22	autrdettexpl1	=BILPASS.XLS!autrdett1-autrdetthexpl1
B23	autrdetthexpl1	
B24	EcCActCrEx1	=BILACTI1.XLS!EcConvAct-SOMME(B25:B27)
B25	EcCActCrHEX1	
B26	EcCActDetEx1	
B27	EcCActDetHEX1	
B28	EcCPasCrEx1	=BILPASS.XLS!EcConvPass1-SOMME(B29:B31)
B29	EcCPasCrHEX1	
B30	EcCPasDetEx1	
B31	EcCPasDetHEX1	
B36		=BILACTI1.XLS!totactif
B37		=BILPASS.XLS!totpass1
B38		=BILPASS.XLS!result1
B39		=RESULTAT.XLS!result1
B40		=BILFO.XLS!totactif1
B41		=BILFO.XLS!totpass1

Les zones grisées indiquent les zones de saisies.

Formules du Compte de Résultat, de l'actif et du passif des Bilans

Les Comptes de Résultat et Bilans (actif et passif) qui suivent sont les documents que les entreprises remettent annuellement à l'Administration fiscale à

l'issue de leur clôture d'exercice (inventaire). Ce sont donc des documents de saisie, mais indispensables aux calculs des documents d'analyse.

Formules de calcul de la feuille du Compte de Résultat

Refe- rences cellules	Nom cellules	Formule
B76	Result1	
B3		=INFOD.XLS!exercice1
B7		=SOMME(:B4:B6)
B13		=SOMME(B7:B12)
B29		=SOMME(B15:B28)
B31		=B13-B29
B41		=SOMME(B35:B40)
B47		=SOMME(B43:B46)
B49		=B41-B47
B51		=B13-B29+B33-B34+B41-B47
B56		=B3
B60		=SOMME(B57:B59)
B65		=SOMME(B62:B64)
B67		=B60-B65
B72		=B13+B33+B41+B60
B74		=B29+B34+B47+B65+B69+B70
B76		=B72-B74

	A	B	C	D
1	Compte de résultat de l'exercice - en liste			
2	ORDIMAGE S.A.			
3		1991	1992	1993
4	Ventes de marchandises.....			
5	Production vendue (biens).....	115 000	381 910	330 100
6	Production vendue (services).....			
7	Chiffre d'affaires net	115 000	381 910	330 100
8	Production stockée.....	80 000	-20 000	3 000
9	production immobilisée.....	45 990	46 910	105 100
10	Subvention d'exploitation.....			
11	Reprises sur amortissements et provisions.....			
12	Autres produits (1).....	8 900	1 688	5 933
13	Total des produits d'exploitation (2) (I)	249 890	410 508	444 133
14				
15	Achats de marchandises (y compris droits de douane).....	1 500	685	5 933
16	Variation de stock (marchandises).....			
17	Achats de M.P. et autres approv. (avec douane).....	140 495	140 955	190 105
18	Variation de stock (M.P. et approv.).....	20 000	-10 000	50 000
19	Autres achats et charges externes (3).....	10 000	11 000	12 000
20	Impôts, taxes et versements assimilés.....	12 000	13 000	14 000
21	Salaires et traitements.....	40 600	42 000	42 500
22	Charges sociales.....	20 300	21 000	21 500
23	Dotations d'exploitation :			
24	- sur immobilisations (amortissements).....	54 550	58 950	75 400
25	- sur immobilisations (provisions).....		15 000	3 990
26	- sur actif circulant (provisions).....			
27	- pour risques et charges (provisions).....			
28	Autres charges.....			
29	Total des charges d'exploitation (4) (II)	299 445	292 590	415 428
30				
31	1-Résultat d'exploitation (I-II)	-49 555	117 918	28 705
32				
33	Bénéfice attribué ou perte transférée (III)			
34	Perte supportée ou bénéfice transféré (IV)			
35	Produits financiers de participation (5).....			
36	Produits des autres valeurs mobilières et créances (5).....			
37	Autres intérêts et produits assimilés (5).....			
38	Reprises sur provisions et transfert de charges.....			
39	Différences positives de change.....			
40	Produits nets sur cessions de valeurs mobilières de plac.....			
41	Total des produits financiers (V)	0	0	0
42				
43	Dotations financières aux amortissements et provisions.....			
44	Intérêts et charges assimilées (6).....		12 500	45 000
45	Différences négatives de change.....			
46	Charges nettes sur cessions de valeurs mob. de plac.....			
47	Total des charges financières (VI)	0	12 500	45 000
48				
49	2-Résultat financier (V-VI)	0	-12 500	-45 000
50				
51	3-Résultat courant avant impôts (I+III-IV+V-VI)	-49 555	105 418	-16 295
52				

	A	B	C	D
54	Compte de résultat de l'exercice - en liste (suite)			
55	ORDIMAGE S.A.			
56		1991	1992	1993
57	Produits exceptionnels sur opérations de gestion.....	12 700		
58	Produits exceptionnels sur opérations de en capital.....		5 000	
59	Reprises sur provisions et transferts de charges.....			
60	Total des produits exceptionnels (VII)	12 700	5 000	0
61				
62	Charges exceptionnelles sur opérations de gestion.....	100	35 003	8 721
63	Charges exceptionnelles sur opérations en capital.....			
64	Dotations exceptionnelles aux amortissements et prov.....			
65	Total des charges exceptionnelles (VIII)	100	35 003	8 721
66				
67	3-Résultat exceptionnel (VII-VIII)	12 600	-30 003	-8 721
68				
69	Participation des salariés aux résultats de l'entreprise (IX)			
70	Impôts sur les bénéfices (X)			
71				
72	Total des produits (I+III+V+VII)	262 590	415 508	444 133
73				
74	Total des charges (II+IV+VI+VIII+IX+X)	299 545	340 093	469 149
75				
76	4-Bénéfice ou perte (produits - charges)	-36 955	75 415	-25 016
77				
78	RENOIS :			
79				
80	(1) Dont produits nets partiels sur opérations à long terme.....			
81	(2) Dont produits d'exploitation afférents à des ex. antér.....			
82	(3) Dont :			
83	- crédit-bail mobilier.....			
84	- crédit-bail immobilier.....			
85	(4) Dont charges d'exploitation afférents à des ex. antér.....			
86	(5) Dont produits concernant les entreprises liées.....			
87	(6) Dont intérêts concernant les entreprises liées.....			
88	(7) Détail des produits et charges exceptionnels :			
89	Créances irrécouvrables.....			
90	Amende.....			
91	Procès.....			
92	Annulation chèques cautions.....			
93	Annulation dettes fournisseurs.....			
94	Pénalités de retard.....			
95				
96				
97	(8) Détail des produits et charges sur exercices antérieurs :			
98	RAN fournisseurs non justifiés.....			
99				
100				
101				
102				
103				

	A	B	C	D	E	F
1	Bilan - Actif					
2	ORDIMAGE S.A.					
3	1991					
4	ACTIF					
5				BRUT	Amortiss. et Provisions	NET
6						
7	Capital souscrit non appelé.....					
8						
9	IMMOBILISATIONS INCORPORELLES :					
10	Frais d'établissement.....			60 000	20 000	40 000
11	Frais de recherche et développement.....			45 990	8 550	37 440
12	Concessions, brevets, licences, marques, procédés, droits.....					
13	Fonds commercial (1).....					
14	Autres immobilisations incorporelles.....					
15	Avances et acomptes sur immo. incorp.....					
16	IMMOBILISATIONS CORPORELLES :					
17	Terrains.....					
18	Constructions.....			160 000	9 000	171 000
19	Installations techniques, outillages, et matériels industriels.....					
20	Autres immobilisations corporelles.....			190 000	17 000	173 000
21	Immobilisations corporelles en cours.....					
22	Avances et acomptes.....					
23	IMMOBILISATIONS FINANCIERES :					
24	Part. selon méthode équivalence.....					
25	Autres participations.....					
26	Créances rattachées à participations.....					
27	Autres titres immobilisés.....					
28	Prêts.....					
29	Autres immobilisations financières.....					
30						
31						
32	TOTAL (I)			475 990	54 550	421 440
33						
34	STOCKS ET EN-COURS :					
35	Matières premières, approvisionnements.....			20 000		20 000
36	En cours de production de biens.....					
37	Biens et services.....					
38	Produits intermédiaires et finis.....			80 000		80 000
39	Marchandises (vendues en l'état).....					
40	AVANCES ET ACOMPTES VERSES SUR COMMANDES D'EXPLOITATION					
41	CREANCES :					
42	Créances clients et comptes rattachés (3).....			90 000		90 000
43	Autres créances (3).....					
44	Capital souscrit, appelé non versé.....					
45	VALEURS MOBILIERES DE PLACEMENT					
46	(dont act.prop.:).....					
47	DISPONIBILITES.....			58 005		58 005
48	CHARGES CONSTATEES D'AVANCE (3).....			1 500		1 500
49	TOTAL (II)			249 505	0	249 505
50	Chargés à répartir sur exercices (III).....					
51	Finmes de remb. des obligations (IV).....					
52	Ecart de conversion - actif (V).....					
53	TOTAL GENERAL (I à V)			725 495	54 550	670 945
54	Renvoi: (1) dont droit au bas:					
55	(2) part à plus d'un an des immobilisations financières nettes:					
56	(3) part à plus d'un an:					

Formules de calcul de la feuille de l'Actif du Bilan

Références cellules	Nom cellules	Formule
D7	KSousNApp	
D33	total I	=SOMME(D7:D31)
D36: D40	stock	
D42	AvAcpte	
D44	crelt	
D45	autrer	
D48	VMP	
D49	dispo	
D50	chconstav	
D51		=SOMME(D36:D50)
D52	ChARep	
D53	PrmRembOblig	
D54	EcConvAct	
D55		=D33+D51+SOMME(D52:D54)
E33		=SOMME(E7:E31)
E51		=SOMME(E36:E50)
E55	AmortProv	=E33+E51+SOMME(E52:E54)
F7		=SI(D7-E7<>0;D7-E7;"")
F33		=SOMME(F7:F31)
F51		=SOMME(F36:F50)
F55	totactif	=F33+F51+SOMME(F52:F54)

Les zones grisées indiquent les zones de saisies.

	A	B	C	D
1	Bilan avant répartition - Passif			
2				
3	ORDIMAGE S.A.			
4				
5	PASSIF	1991	1992	1993
6				
7	CAPITAUX PROPRES :			
8	Capital social ou individuel (1) (dont versé :).....	250 000	380 000	380 000
9	Primes d'émission, fusion, d'apport.....			
10	Écarts de réévaluation (2) (dont écart d'équiv. :).....			
11	Réserve légale (3).....			3 771
12	Réserves statutaires ou contractuelles.....			
13	Réserves réglementées (3) (4).....			
14	Autres réserves.....			
15	Report à nouveau.....		-36 955	
16	Résultat de l'exercice (bénéfice ou perte).....	-36 955	75 415	-25 016
17	Subventions d'investissement.....			
18	Provisions réglementées.....			
19				
20	TOTAL (I)	213 045	418 460	358 755
21				
22	AUTRES FONDS PROPRES :			
23				
24	Produit des émissions de titres participatifs.....			
25	Avances conditionnées.....			
26				
27	TOTAL (II)	0	0	0
28				
29	Provisions pour risques.....			
30	Provisions pour charges.....			
31				
32	TOTAL (III)	0	0	0
33				
34	DETTES :			
35				
36	Emprunts obligataires convertibles.....			
37	Autres emprunts obligataires.....			
38	Emprunts et dettes auprès des établis. de crédit (6).....	350 000	175 000	195 000
39	Emprunts et dettes financières divers (7).....			
40	Avances et acomptes reçus sur commandes en cours..	10 000	17 500	18 000
41	Dettes fournisseurs et comptes rattachés.....	15 000	28 000	54 000
42	Dettes fiscales et sociales.....	82 900	98 500	115 900
43	Dettes sur immobilisations et comptes rattachés.....			
44	Autres dettes.....			
45				
46	Produits constatés d'avance (5).....			
47				
48	TOTAL (IV)	457 900	319 000	382 900
49				
50	Ecarts de conversion - passif (V).....			
51	TOTAL GENERAL (I à V)	670 945	737 460	741 655
52	ORDIMAGE S.A.			

	A	B	C	D
53	ORDIMAGE S.A.			
54	RENVOIS du passif du bilan :	1991	1992	1993
55	(1) Ecart de réévaluation incorporé au capital.....			
56	(2) Dont :			
57	Réserve spéciale de réévaluation (1959).....			
58	Ecart de réévaluation libre.....			
59	Réserve de réévaluation (1976).....			
60	(3) Dont réserve réglementée plus-values à lg. terme.....			
61	(4) Dont réserve relative à l'achat d'œuvres			
62	originaires d'artistes vivants.....			
63	(5) Dettes et produits constatés d'avance à - d'un an.....			
64	(6) Dont concours bancaires et soldes créditeurs de			
65	banques et CCP.....			
66	(7) Dont emprunts participatifs.....			
67	ORDIMAGE S.A.			

Formules de calcul de la feuille du Passif du Bilan

Références cellules	Nom cellules	Formule
B16	resultI	
B20	total II	=SOMME(B7:B18)
B27	total III	=SOMME(B22:B25)
B32	total III1	=SOMME(B29:B30)
B36:	empruntI	
B39		
B40	AvAcptel	
B41	DettFourI	
B41:	DettFourFiscI	
B42		
B43	DettImmI	
B44	autrdentI	
B46	prdeconstavI	
B48		=SOMME(B34:B46)
B50	EcConvPassI	
B51	totpassI	=B20+B27+B32+B48+B50
B65	CCBI	

Les zones grises indiquent les zones de saisies.

Formules du tableau de retraitement du Crédit-Bail

Ce tableau récapitule les immobilisations que l'entreprise loue en crédit-bail. En vue du retraitement des bilans fiscaux, on calcule des amortissements fictifs (les amortissements fictifs sont calculés sur la base du coût d'achat du bien par la société de crédit-bail (valeur d'origine), la méthode d'amortissement est celle qu'appliquerait l'entreprise si elle était propriétaire du bien, le nombre d'années doit être égal à la durée

du contrat de crédit-bail), les frais financiers fictifs (qui sont la différence entre la valeur d'origine et les amortissements fictifs). Les biens qui auront été rendus (fin de contrat) ne doivent plus figurer dans le tableau de retraitement de crédit-bail, qu'ils aient été rendus, achetés (option d'achat), ou que le contrat ait été prorogé. Dans ce dernier cas, il faudra recréer une ligne dans le tableau de retraitement.

Formules de calcul de la feuille de retraitement du Crédit-Bail

Refe- rences cellules	Nom cellules	Formule
G5		=C5-F5
C19		=SOMME(C5:C18)
E19	vocrdbail	=SOMME(E5:E18)
F19	AmortFict1	=SOMME(F5:F18)
G19	FrsFil	=SOMME(G5:G18)

Formules des Bilans Fonctionnels du tableau d'analyse des Bilans Fonctionnels et des ratios

Les feuilles de calcul précédentes ont permis la préparation des Bilans fonctionnels. Les formules des Bilans fonctionnels vont chercher les données dont

elles ont besoin directement sur les feuilles de calcul concernées. Il en est de même pour les feuilles d'analyse des Bilans fonctionnels et de ratios.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Retraitement du Crédit-Bail												
2	ORDIMAGE S.A.												
3	Eléments	Date début	Valeur	Nombre	1991			1992			1993		
4		de contrat	d'origine	d'années :	Valeur d'origine	Amortissement fictifs	Frais financiers	Valeur d'origine	Amortissement fictifs	Frais financiers	Valeur d'origine	Amortissement fictifs	Frais financiers
5													
6													
7													
8													
9													
10													
11													
12													
13													
14													
15													
16													
17													
18													
19	TOTAL												
20	(*) voiture commerciale : TVA taux normal (18,6%) déductible.												

	A	B	C	D
1	Bilans fonctionnels condensés			
2	ORDIMAGE S.A.			
3	ACTIF	1991	1992	1993
4	Emplois stables			
5	actif immobilisé d'exploitation.....	475 990	476 910	475 100
6	actif immobilisé hors exploitation.....	0	0	0
7	<i>TOTAL I</i>	475 990	476 910	475 100
8	Emplois circulants			
9	stocks.....	100 000	88 000	141 000
10	créances d'exploitation.....	117 032	114 121	126 073
11	créances hors exploitation.....	0	0	0
12	trésorerie active.....	58 005	228 865	200 307
13	<i>TOTAL II</i>	275 037	430 986	467 380
14		751 027	907 896	942 480
15				
16	ORDIMAGE S.A.			
17	PASSIF	1991	1992	1993
18	Ressources stables			
19	fonds propres.....	267 595	546 960	506 645
20	dettes financières.....	343 750	152 500	163 200
21	<i>TOTAL I</i>	611 345	699 460	669 845
22	Ressources circulantes			
23	dettes d'exploitation.....	107 900	144 000	187 900
24	dettes hors exploitation.....	6 250	22 500	31 800
25	trésorerie passive.....	25 532	41 936	52 935
26	<i>TOTAL II</i>	139 682	208 436	272 635
27		751 027	907 896	942 480
28				

Formules de calcul de la feuille des Bilans fonctionnels

Références cellules	Nom cellules	Formule
B3		=INFOD.XLS!exercice1
B5		=(BILACTI1.XLS!total_1+CRDBAIL.XLS!vocrd Bail1+BILACTI1.XLS!ChARep)*INFOD.XLS!immexpl1-BILACTI1.XLS!PrmRembOblig-BILACTI1.XLS!KSousNApp
B6		=(BILACTI1.XLS!total_1+CRDBAIL.XLS!vocrd Bail1+BILACTI1.XLS!ChARep)*INFOD.XLS!immhexpl1
B7	empstabl	=SOMME(B5:B6)
B9	stock1	=SOMME(BILACTI1.XLS!stock)
B10	crexpl1	=BILACTI1.XLS!AvAcpte+BILACTI1.XLS!crcit+INFOD.XLS!autrecrexpl1+INFOD.XLS!EENE1+(BILACTI1.XLS!chconstav-INFOD.XLS!chconstavhexpl1)+(INFOD.XLS!EcCAActCrEx1-INFOD.XLS!EcCPasCrEx1)
B11	crhexpl1	=INFOD.XLS!autrechrhexpl1+BILACTI1.XLS!VMP+INFOD.XLS!chconstavhexpl1+(INFOD.XLS!EcCAActCrHEX1-INFOD.XLS!EcCPasCrHEX1)
B12	tresoract1	=BILACTI1.XLS!dispo
B13	empcircl	=SOMME(B9:B12)
B14	totacrif1	=B7+B13
B19	fdsprp1	=BILPASS.XLS!total_11+BILPASS.XLS!total_111+BILPASS.XLS!total_1111+BILACTI1.XLS!AmortProv+CRDBAIL.XLS!AmortFict1-BILACTI1.XLS!KSousNApp
B20	dettfi1	=SOMME(BILPASS.XLS!emprunt1)-BILPASS.XLS!CCB1-INFOD.XLS!ICNE1+CRDBAIL.XLS!FrsFi1-BILACTI1.XLS!PrmRembOblig
B21	ressstabl	=SOMME(B19:B20)
B23	dettexpl1	=BILPASS.XLS!AvAcpte1+SOMME(BILPASS.XLS!DettFourFisc1)-INFOD.XLS!impotbenef1+INFOD.XLS!autrdettexpl1+(BILPASS.XLS!prdconstav1-INFOD.XLS!prdconstavhexpl1)+(INFOD.XLS!EcCPasDetEx1-INFOD.XLS!EcCAActDetEx1)
B24	detthexpl1	=BILPASS.XLS!DettImm1+INFOD.XLS!autrdettexpl1+INFOD.XLS!impotbenef1+(BILPASS.XLS!prdconstav1-INFOD.XLS!prdconstavhexpl1)+INFOD.XLS!ICNE1+(INFOD.XLS!EcCPasDetHEX1-INFOD.XLS!EcCAActDetHEX1)
B25	tresorpass1	=BILPASS.XLS!CCB1+INFOD.XLS!EENE1
B26	resscircl	=SOMME(B23:B25)

	A	B	C	D
1	L'analyse des bilans fonctionnels			
2	ORDIMAGE S.A.			
3		1991	1992	1993
4	Ressources stables.....	611 345	699 460	669 845
5	Emplois stables.....	475 990	476 910	475 100
6				
7	F.R.N.G.			
8	(Fonds de Roulement Net Global)	135 355	222 550	194 745
9				
10	Actif circulant d'exploitation.....	217 032	202 121	267 073
11	Passif circulant d'exploitation.....	107 900	144 000	187 900
12				
13	B.F.R.E.			
14	(Besoin en Fonds de Roulement d'Exploitation)	109 132	58 121	79 173
15				
16	Actif circulant hors exploitation.....	0	0	0
17	Passif circulant hors exploitation.....	6 250	22 500	31 800
18				
19	B.F.R.H.E.			
20	(Besoin en Fonds de Roulement Hors Exploitation)	-6 250	-22 500	-31 800
21				
22	Besoin en fonds de roulement d'exploitation.....	109 132	58 121	79 173
23	Besoin en fonds de roulement hors exploitation...	-6 250	-22 500	-31 800
24				
25	B.F.R.			
26	(Besoin en Fonds de Roulement)	102 882	35 621	47 373
27				
28	Trésorerie active.....	58 005	228 865	200 307
29	Trésorerie passive.....	25 532	41 936	52 935
30				
31				
32	Trésorerie nette	32 473	186 929	147 372
33				
34	ORDIMAGE S.A.			
35	Contrôle	1991	1992	1993
36	F.R.N.G.....	135 355	222 550	194 745
37	B.F.R.....	102 882	35 621	47 373
38	Excédent (ou insuffisance) net(te) de trésorerie....	32 473	186 929	147 372
39	Trésorerie nette.....	32 473	186 929	147 372
40	ORDIMAGE S.A.			

Formules de calcul de la feuille d'analyse des Bilans fonctionnels

Références cellules	Nom cellules	Formule
A2		=INFOD.XLS!RaisonSoc
B3		=INFOD.XLS!exercice1
B4		=BILFO.XLS!ressstab1
B5		=BILFO.XLS!empstab1
B8	fng1	=B4-B5
B10		=BILFO.XLS!stock1+BILFO.XLS!crexp1
B11		=BILFO.XLS!dettexp1
B14	bfre1	=B10-B11
B16		=BILFO.XLS!crhexp1
B17		=BILFO.XLS!detthexp1
B20	bfrhe1	=B16-B17
B22		=bfre1
B23		=bfrhe1
B26	bfr1	=SOMME(B22:B23)
B28		=BILFO.XLS!tresoract1
B29		=BILFO.XLS!tresorpass1
B32	tresornett1	=B28-B29
B36		=fng1
B37		=bfr1
B38		=B36-B37
B39		=tresornett1

1	A						G				
	B		C		D			E		F	
	1991		1992		1993			1992		1993	
	Entreprise	Secteur	Entreprise	Secteur	Entreprise	Secteur	Entreprise	Secteur	Entreprise	Secteur	
2	ORDIMAGE S.A.										
3	(résultats en pourcentage sauf indication contraire)										
4	Financement des investissements.....	128,44		146,66		140,99					
5	Couverture des capitaux investis.....	104,48	105,20	130,73	102,90	120,85				N.C.	
6	Ratio d'endettement.....	114,63	59,30	28,79	57,40	33,02				N.C.	
7	Ratio de crédit-clients (en jours).....	279,53	94,00	81,55	87,00	99,41				N.C.	
8	Ratio de crédit-fournisseurs (en jours).....	139,12	86,00	265,60	81,00	834,74				N.C.	
9	Répartition V.A. globale : Personnel.....	61,15	75,80	22,67	75,70	25,80				N.C.	
10	Répartition V.A. globale : Etat.....	0,00	1,70	0,00	1,70	0,00				N.C.	
11	Bénéfice net d'I.S./chiffre d'affaires.....	32,13		19,75		7,58					
12	N.C. - Résultats encore imputés par la Banque de France										
13	Les zones grisées correspondent aux ratios non calculés par la Banque de France										
14	ORDIMAGE S.A.										
15	Chiffre d'affaires H.T.....		115 000		381 910		330 100				
16	Chiffre d'affaires T.T.C.....		136 390		452 945		381 499				
17	Valeur ajoutée.....		99 595		277 865		248 095				
18	Achats et charges externes T.T.C.....		11 860		13 046		14 232				
19	Charges de personnel (salaires et ch. soc.).....		60 900		63 000		64 000				

Formules de calcul de la feuille de ratios

Références cellules	Nom cellules	Formule
B5		=BILFO.XLS!ressstab1/BILFO.XLS!empstab1*100
B7		=BILFO.XLS!ressstab1/(BILFO.XLS!empstab1+ANABILFO.XLS!bfre1)*100
B9		=(BILFO.XLS!dettfi1+BILPASS.XLS!CCB1+INFOD.XLS!EENE1)/(BILFO.XLS!fdsprp1+BILACTI1.XLS!AmortProv)*100
B11		=(BILACTI1.XLS!crcit+INFOD.XLS!EENE1)/CATTTC1*INFOD.XLS!NbJActivite
B13		=(BILPASS.XLS!DettFour1-BILPASS.XLS!AvAcpte1)/AchChExtTTC1*INFOD.XLS!NbJActivite
B15		=ChPers1/VA1*100
B17		=INFOD.XLS!impotbenef1/VA1*100
B19		=BILPASS.XLS!result1/CAHT1*100
C26	CAHT1	
C27	CATTTC1	
C28	VA1	
C29	AchChExtTTC1	
C30	ChPers1	

Les zones grisees indiquent les zones de saisies.

Analyse des instructions de la macro-commande

POUR("compteur", début, fin, pas)

SUIVANT()

Cette commande fait varier la variable 'compteur' de la valeur 'début' à la valeur 'fin' avec une progression de pas 'pas'. Par exemple POUR("lig";36;41;2), fera prendre successivement les valeurs : 36, 38, 40 à la variable 'lig'.

ADRESSE(lig;col;n;b;"feuille_calcul") : forme une référence cellulaire à l'aide de ligne 'lig' et de colonne 'col' en absolu ou en relatif suivant la variable 'n' et de type 'L1C1' si b=FAUX ou 'A1' si b=VRAI. ADRESSE(lig;col;1;FAUX;"INFOD.XLS") donne pour résultat (quand lig=36 et col=2) : 'INFOD.XLS!L36C2'.

EVALUER(référence_cellulaire) : fournit le contenu de la cellule référencée 'référence_cellulaire'. EVALUER(INFOD.XLS!L36C2) renvoie '4 213 819'.

ABS (num) : donne la valeur absolue d'une valeur numérique. Ainsi dans la formule 'ABS (EVALUER(ADRESSE(lig; col; 1; FAUX; "infod.xls")) - EVALUER(ADRESSE(lig+1; col; 1; FAUX; "infod.xls")))', on demande à EXCEL de prendre la valeur absolue du résultat de la soustraction du contenu des cellules référencées par L(lig)C(col) et L(lig+1)C(col).

ALERTE("texte") : affiche des messages de l'utilisateur et/ou des résultats de calculs.

SI(condition_1)...SINON.SI(condition_2)...FIN.SI() : exécute des actions (suites d'instructions) en fonction du résultat de conditions.

RETOUR() : indique à EXCEL que la macro-commande doit prendre fin.

4 Liasse des documents obtenus

Si tout s'est bien passé, vous devriez obtenir la liasse de documents qui figure dans les pages suivantes.

5 Extensions possibles

Le calcul de deux tableaux supplémentaires pourrait être envisagé à savoir un Tableau de Financement et un Tableau des Soldes Intermédiaires de Gestion afin de compléter l'analyse. Ce sujet

fera peut-être l'objet d'une prochaine étude. On pourrait également envisager d'établir des graphiques à l'aide de l'Assistant Graphique d'Excel à partir des données de l'application.

Informations de base

RAISON SOCIALE.....	ORDIMAGE S.A.
Année premier exercice traité.....	1991
l'exercice débute le 1er avril et s'arrête au 30 mars de l'année suivante.....	1er exercice traité : 1/1/91-31/12/91
Nombre de jours d'activité (*).....	330
Période annuelle durant l'entreprise cesse ses activités.....	du 1er au 30 août

(*) Pour le plus d'informations, consultez le chapitre de page 30

baselle
Informations complémentaires nécessaires à l'élaboration du bilan fonctionnel

ORDIMAGE S.A.

	1991	1992	1993
immobilisations - "exploitation" (en % des immobilisations).....	100%	100%	100%
Immobilisations - "hors exploitation" (en %).....	0%	0%	0%
E.E.N.E (effets escomptés non échus)/affacturage.....	25 532,00	41 936,00	52 935,00
Autres créances d'exploitation.....	0	0	0
Autres créances hors exploitation.....	0	0	0
Charges constatées d'avance hors exploitation.....	0	0	0
Produits constatés d'avance hors exploitation.....	0	0	0
Intérêts courus non échus.....	6 250	22 500	31 800
Impôt sur les bénéfices.....	0	0	0
Autres dettes d'exploitation.....	0	0	0
Autres dettes hors exploitation.....	0	0	0
Ecart de conversion actif - Créance d'exploitation.....	0	0	0
Ecart de conversion actif - Créance hors exploitation.....	0	0	0
Ecart de conversion actif - Dette d'exploitation.....	0	0	0
Ecart de conversion actif - Dette hors exploitation.....	0	0	0
Ecart de conversion passif - Créance d'exploitation.....	0	0	0
Ecart de conversion passif - Créance hors exploitation.....	0	0	0
Ecart de conversion passif - Dette d'exploitation.....	0	0	0
Ecart de conversion passif - Dette hors exploitation.....	0	0	0

Synthèse permettant une vérification rapide

ORDIMAGE S.A.

	1991	1992	1993
Bilan - total ACTIF.....	670 945,00	737 460,00	741 655,00
Bilan - total PASSIF.....	670 945,00	737 460,00	741 655,00
Résultat figurant au bilan.....	-36 955,00	75 415,00	-25 016,00
Résultat figurant au compte de résultat.....	-36 955,00	75 415,00	-25 016,00
Bilan fonctionnel - total ACTIF.....	751 027,00	907 896,00	942 480,00
Bilan fonctionnel - total PASSIF.....	751 027,00	907 896,00	942 480,00

Compte de résultat de l'exercice - en liste

ORDIMAGE S.A.

	1991	1992	1993
Ventes de marchandises.....			
Production vendue (biens).....	115 000	381 910	330 100
Production vendue (services).....			
Chiffre d'affaires net	115 000	381 910	330 100
Production stockée.....	80 000	-20 000	3 000
production immobilisée.....	45 990	46 910	105 100
Subvention d'exploitation.....			
Reprises sur amortissements et provisions.....			
Autres produits (1).....	8 900	1 688	5 933
Total des produits d'exploitation (2) (I)	249 890	410 508	444 133
Achats de marchandises (y compris droits de douane).....	1 500	685	5 933
Variation de stock (marchandises).....			
Achats de M.P. et autres approv. (avec douane).....	140 495	140 955	190 105
Variation de stock (M.P. et approv.).....	20 000	-10 000	50 000
Autres achats et charges externes (3).....	10 000	11 000	12 000
Impôts, taxes et versements assimilés.....	12 000	13 000	14 000
Salaires et traitements.....	40 600	42 000	42 500
Charges sociales.....	20 300	21 000	21 500
Dotations d'exploitation :			
- sur immobilisations (amortissements).....	54 550	58 950	75 400
- sur immobilisations (provisions).....			
- sur actif circulant (provisions).....		15 000	3 990
- pour risques et charges (provisions).....			
Autres charges.....			
Total des charges d'exploitation (4) (II)	299 445	292 590	415 428
1-Résultat d'exploitation (I-II)	-49 555	117 918	28 705
Bénéfice attribué ou perte transférée (III)			
Perte supportée ou bénéfice transféré (IV)			
Produits financiers de participation (5).....			
Produits des autres valeurs mobilières et créances (5).....			
Autres intérêts et produits assimilés (5).....			
Reprises sur provisions et transfert de charges.....			
Différences positives de change.....			
Produits nets sur cessions de valeurs mobilières de plac.....			
Total des produits financiers (V)	0	0	0
Dotations financières aux amortissements et provisions.....			
Intérêts et charges assimilées (6).....		12 500	45 000
Différences négatives de change.....			
Charges nettes sur cessions de valeurs mob. de plac.....			
Total des charges financières (VI)	0	12 500	45 000
2-Résultat financier (V-VI)	0	-12 500	-45 000
3-Résultat courant avant impôts (I-II+III-IV+V-VI)	-49 555	105 418	-16 295

Compte de résultat de l'exercice - en liste (suite)

ORDIMAGE S.A.

	1991	1992	1993
Produits exceptionnels sur opérations de gestion.....	12 700		
Produits exceptionnels sur opérations de en capital.....		5 000	
Reprises sur provisions et transferts de charges.....			
Total des produits exceptionnels (VII)	12 700	5 000	0
Charges exceptionnelles sur opérations de gestion.....	100	35 003	8 721
Charges exceptionnelles sur opérations en capital.....			
Dotations exceptionnelles aux amortissements et prov.....			
Total des charges exceptionnelles (VIII)	100	35 003	8 721
3-Résultat exceptionnel (VII-VIII)	12 600	-30 003	-8 721
Participation des salariés aux résultats de l'entreprise (11)			
Impôts sur les bénéfices (X)			
Total des produits (I+III+V+VII)	262 590	415 508	444 133
Total des charges (II+IV+VI+VIII+IX+X)	299 545	340 093	469 149
4-Bénéfice ou perte (produits - charges)	-36 955	75 415	-25 016
RENVois :			
(1) Dont produits nets partiels sur opérations à long terme.....			
(2) Dont produits d'exploitation afférents à des ex. antér.....			
(3) Dont :			
- crédit-bail mobilier.....			
- crédit-bail immobilier.....			
(4) Dont charges d'exploitation afférents à des ex. antér.....			
(5) Dont produits concernant les entreprises liées.....			
(6) Dont intérêts concernant les entreprises liées.....			
(7) Détail des produits et charges exceptionnels :			
Créances irrécouvrables.....			
Amende.....			
Procès.....			
Annulation chèques cautions.....			
Annulation dettes fournisseurs.....			
Pénalités de retard.....			
(8) Détail des produits et charges sur exercices antérieurs :			
RAN fournisseurs non justifiés.....			

Bilan - Actif

ORDIMAGE S.A.

ACTIF	1991		
	BRUT	Amortiss. et Provisions	NET
Capital souscrit non appelé.....			
IMMOBILISATIONS INCORPORELLES :			
Frais d'établissement.....	60 000	20 000	40 000
Frais de recherche et développement.....	45 990	8 550	37 440
Concessions, brevets, licences, marques, procédés, droits.....			
Fonds commercial (1).....			
Autres immobilisations incorporelles.....			
Avances et acomptes sur immo. incorp.....			
IMMOBILISATIONS CORPORELLES :			
Terrains.....			
Constructions.....	180 000	9 000	171 000
Installations techniques, outillages, et matériels industriels.....			
Autres immobilisations corporelles.....	190 000	17 000	173 000
Immobilisations corporelles en cours.....			
Avances et acomptes.....			
IMMOBILISATIONS FINANCIERES :			
Part. selon méthode équivalence.....			
Autres participations.....			
Créances rattachées à participations.....			
Autres titres immobilisés.....			
Prêts.....			
Autres immobilisations financières.....			
TOTAL (I)	475 990	54 550	421 440
STOCKS ET EN-COURS :			
Matières premières, approvisionnements.....	20 000		20 000
En cours de production de biens.....			
Biens et services.....			
Produits intermédiaires et finis.....	80 000		80 000
Marchandises (vendues en l'état).....			
AVANCES ET ACOMPTES VERSES SUR COMMANDES D'EXPLOITATION.....			
CREANCES :			
Créances clients et comptes rattachés (3).....	90 000		90 000
Autres créances (3).....			
Capital souscrit, appelé non versé.....			
VALEURS MOBILIERES DE PLACEMENT (dont act.prop.:).....			
DISPONIBILITES.....	58 005		58 005
CHARGES CONSTATEES D'AVANCE (3).....	1 500		1 500
TOTAL (II)	249 505	0	249 505
Charges à répartir sur exercices (III)			
Primes de remb. des obligations (IV)			
Ecart de conversion - actif (V)			
TOTAL GENERAL (I à V)	725 495	54 550	670 945

Rendus: (1) dont doit être...

(2) par à dire, en des immobilisations financières nettes

(3) par à dire, en...

Bilan - Actif

ORDIMAGE S.A.

ACTIF	1992		
	BRUT	Amortiss. et Provisions	NET
Capital souscrit non appelé.....			
IMMOBILISATIONS INCORPORELLES :			
Frais d'établissement.....	60 000	40 000	20 000
Frais de recherche et développement.....	46 910	21 500	25 410
Concessions, brevets, licences, marques, procédés, droits.....			
Fonds commercial (1).....			
Autres immobilisations incorporelles.....			
Avances et acomptes sur immo. incorp.....			
IMMOBILISATIONS CORPORELLES :			
Terrains.....			
Constructions.....	180 000	18 000	162 000
Installations techniques, outillages, et matériels industriels.....			
Autres immobilisations corporelles.....	190 000	34 000	156 000
Immobilisations corporelles en cours.....			
Avances et acomptes.....			
IMMOBILISATIONS FINANCIERES :			
Part. selon méthode équivalence.....			
Autres participations.....			
Créances rattachées à participations.....			
Autres titres immobilisés.....			
Prêts.....			
Autres immobilisations financières.....			
TOTAL (I)	476 910	113 500	363 410
STOCKS ET EN-COURS :			
Matières premières, approvisionnements.....	10 000		10 000
En cours de production de biens.....			
Biens et services.....			
Produits intermédiaires et finis.....	78 000		78 000
Marchandises (vendues en l'état).....			
AVANCES ET ACOMPTES VERSES SUR COMMANDES D'EXPLOITATION.....			
CREANCES :			
Créances clients et comptes rattachés (3).....	70 000	15 000	55 000
Autres créances (3).....			
Capital souscrit, appelé non versé.....			
VALEURS MOBILIERES DE PLACEMENT (dont act.prop.:).....			
DISPONIBILITES.....	228 865		228 865
CHARGES CONSTATEES D'AVANCE (3).....	2 185		2 185
TOTAL (II)	389 050	15 000	374 050
Charges à répartir sur exercices (III).....			
Primes de remb. des obligations (IV).....			
Ecart de conversion - actif (V).....			
TOTAL GENERAL (I à V)	865 960	128 500	737 460

Remarques: (1) fait état du bail

(2) part à l'un en des immobilisations financières nettes

(3) part à plus d'un en

Bilan - Actif

ORDIMAGE S.A.

ACTIF	1993		
	BRUT	Amortiss. et Provisions	NET
Capital souscrit non appelé.....			
IMMOBILISATIONS INCORPORELLES			
Frais d'établissement.....			
Frais de recherche et développement.....	105 100	38 900	66 200
Concessions, brevets, licences, marques, procédés, droits.....			
Fonds commercial (1).....			
Autres immobilisations incorporelles.....			
Avances et acomptes sur immo. incorp.....			
IMMOBILISATIONS CORPORELLES :			
Terrains.....			
Constructions.....	180 000	27 000	153 000
Installations techniques, outillages, et matériels industriels.....			
Autres immobilisations corporelles.....	190 000	63 000	127 000
Immobilisations corporelles en cours.....			
Avances et acomptes.....			
IMMOBILISATIONS FINANCIERES :			
Part. selon méthode équivalence.....			
Autres participations.....			
Créances rattachées à participations.....			
Autres titres immobilisés.....			
Prêts.....			
Autres immobilisations financières.....			
TOTAL (I)	475 100	128 900	346 200
STOCKS ET EN-COURS :			
Matières premières, approvisionnements.....	60 000		60 000
En cours de production de biens.....			
Biens et services.....			
Produits intermédiaires et finis.....	81 000		81 000
Marchandises (vendues en l'état).....			
AVANCES ET ACOMPTES VERSES SUR COMMANDES D'EXPLOITATION			
CREANCES			
Créances clients et comptes rattachés (3).....	65 000	18 990	46 010
Autres créances (3).....			
Capital souscrit, appelé non versé.....			
VALEURS MOBILIERES DE PLACEMENT (dont act.prop.:).....			
DISPONIBILITES	200 307		200 307
CHARGES CONSTATEES D'AVANCE (3)	8 138		8 138
TOTAL (II)	414 445	18 990	395 455
Charges à répartir sur exercices (III).....			
Primes de remboursement obligations (IV).....			
Ecart de conversion - actif (V).....			
TOTAL GENERAL (I à V)	889 545	147 890	741 655

Remarque (1) : considérer au bilan

(2) par la date d'acquisition des immobilisations financières nettes

(3) par la date d'échéance

Bilan avant répartition - Passif

ORDIMAGE S.A.

PASSIF	1991	1992	1993
CAPITAUX PROPRES			
Capital social ou individuel (1) (dont versé :).....	250 000	380 000	380 000
Primes d'émission, fusion, d'apport.....			
Ecarts de réévaluation (2) (dont écart d'équiv. :).....			3 771
Réserve légale (3).....			
Réserves statutaires ou contractuelles.....			
Réserves réglementées (3) (4).....			
Autres réserves.....			
Report à nouveau.....		-36 955	
Résultat de l'exercice (bénéfice ou perte).....	-36 955	75 415	-25 016
Subventions d'investissement.....			
Provisions réglementées.....			
TOTAL (I)	213 045	418 460	358 755
AUTRES FONDS PROPRES			
Produit des émissions de titres participatifs.....			
Avances conditionnées.....			
TOTAL (II)	0	0	0
Provisions pour risques.....			
Provisions pour charges.....			
TOTAL (III)	0	0	0
DETTES			
Emprunts obligataires convertibles.....			
Autres emprunts obligataires.....			
Emprunts et dettes auprès des établis. de crédit (6).....	350 000	175 000	195 000
Emprunts et dettes financières divers (7).....			
Avances et acomptes reçus sur commandes en cours.....	10 000	17 500	18 000
Dettes fournisseurs et comptes rattachés.....	15 000	28 000	54 000
Dettes fiscales et sociales.....	82 900	98 500	115 900
Dettes sur immobilisations et comptes rattachés.....			
Autres dettes.....			
Produits constatés d'avance (5).....			
TOTAL (IV)	457 900	319 000	382 900
Ecarts de conversion - passif (V).....			
TOTAL GENERAL (I à V)	670 945	737 460	741 655

ORDIMAGE S.A.

ORDIMAGE S.A.

RENVOIS du passif du bilan	1991	1992	1993
(1) Écart de réévaluation incorporé au capital.....			
(2) Dont			
Réserve spéciale de réévaluation (1959).....			
Écart de réévaluation libre.....			
Réserve de réévaluation (1976).....			
(3) Dont réserve réglementée plus-values à lg. terme.....			
(4) Dont réserve relative à l'achat d'oeuvres originales d'artistes vivants.....			
(5) Dettes et produits constatés d'avance à - d'un an.....			
(6) Dont concours bancaires et soldes créditeurs de banques et CCP.....			
(7) Dont emprunts participatifs.....			

ORDIMAGE S.A.

Retraitement du Crédit-Bail

ORDIMAGE S.A.

Eléments	Date début de contrat	Valeur d'origine	Dotation d'amortissement	1991		1992		1993												
				Valeur d'origine	Amortissement	Finis financiers	Valeur d'origine	Amortissement	Finis financiers	Valeur d'origine	Amortissement	Finis financiers								
TOTAL																				

(*) voiture commerciale : TVA taux normal (18,6%) déductible.

Bilans fonctionnels condensés

ORDIMAGE S.A.

ACTIF	1991	1992	1993
Emplois stables			
actif immobilisé d'exploitation.....	475 990	476 910	475 100
actif immobilisé hors exploitation.....	0	0	0
<i>TOTAL I</i>	475 990	476 910	475 100
Emplois circulants			
stocks.....	100 000	88 000	141 000
créances d'exploitation.....	117 032	114 121	126 073
créances hors exploitation.....	0	0	0
trésorerie active.....	58 005	228 865	200 307
<i>TOTAL II</i>	275 037	430 986	467 380
	751 027	907 896	942 480

ORDIMAGE S.A.

PASSIF	1991	1992	1993
Ressources stables			
fonds propres.....	267 595	546 960	506 645
dettes financières.....	343 750	152 500	163 200
<i>TOTAL I</i>	611 345	699 460	669 845
Ressources circulantes			
dettes d'exploitation.....	107 900	144 000	187 900
dettes hors exploitation.....	6 250	22 500	31 800
trésorerie passive.....	25 532	41 936	52 935
<i>TOTAL II</i>	139 682	208 436	272 635
	751 027	907 896	942 480

L'analyse des bilans fonctionnels

ORDIMAGE S.A.

	1991	1992	1993
Ressources stables.....	611 345	699 460	669 845
Emplois stables.....	475 990	476 910	475 100
F.R.N.G. (Fonds de Roulement Net Global)	135 355	222 550	194 745
Actif circulant d'exploitation.....	217 032	202 121	267 073
Passif circulant d'exploitation.....	107 900	144 000	187 900
B.F.R.E. (Besoin en Fonds de Roulement d'Exploitation)	109 132	58 121	79 173
Actif circulant hors exploitation.....	0	0	0
Passif circulant hors exploitation.....	6 250	22 500	31 800
B.F.R.H.E. (Besoin en Fonds de Roulement Hors Exploitation)	-6 250	-22 500	-31 800
Besoin en fonds de roulement d'exploitation.....	109 132	58 121	79 173
Besoin en fonds de roulement hors exploitation....	-6 250	-22 500	-31 800
B.F.R. (Besoin en Fonds de Roulement)	102 882	35 621	47 373
Trésorerie active.....	58 005	226 865	200 307
Trésorerie passive.....	25 532	41 936	52 935
Trésorerie nette	32 473	186 929	147 372

ORDIMAGE S.A.

Contrôle	1991	1992	1993
F.R.N.G.....	135 355	222 550	194 745
B.F.R.....	102 882	35 621	47 373
Excédent (ou insuffisance) net(te) de trésorerie....	32 473	186 929	147 372
Trésorerie nette.....	32 473	186 929	147 372

ORDIMAGE S.A.

Ratios issus des bilans fonctionnels, d'éléments du compte de résultat, et du D.S.9.G

ORDIMAGE S.A.

	1991		1992		1993	
	Entreprise	Secteur	Entreprise	Secteur	Entreprise	Secteur
Résultats en pourcentifs à part report et retraités	128,44		146,66		140,93	
Financement des investissements.....						
Couverture des capitaux investis.....	104,48	105,20	130,73	102,90	120,85	N.C.
Ratio d'endettement.....	114,63	59,30	28,79	57,40	33,02	N.C.
Ratio de crédit-clients (en jours).....	279,53	94,00	81,55	87,00	99,41	N.C.
Ratio de crédit-fournisseurs (en jours).....	139,12	88,00	265,60	81,00	834,74	N.C.
Répartition V.A. globale : Personnel.....	61,15	75,80	22,67	75,70	25,80	N.C.
Répartition V.A. globale : Etat.....	0,00	1,70	0,00	1,70	0,00	N.C.
Bénéfice net d'I.S. / chiffre d'affaires.....	-92,13		19,75		-7,58	

N.C. : Résultats encore inconnus par la Banque de France

Les zones grisées correspondent aux ratios non calculés par la Banque de France

ORDIMAGE S.A.

	1991		1992		1993	
	Entreprise	Secteur	Entreprise	Secteur	Entreprise	Secteur
Chiffre d'affaires H.T.....	115 000		381 910		330 100	
Chiffre d'affaires T.T.C.....	136 390		452 845		391 499	
Valeur ajoutée.....	99 595		277 865		248 095	
Achats et charges externes T.T.C.....	11 860		13 046		14 232	
Charges de personnel (salaires et ch. soc.).....	60 900		63 000		64 000	