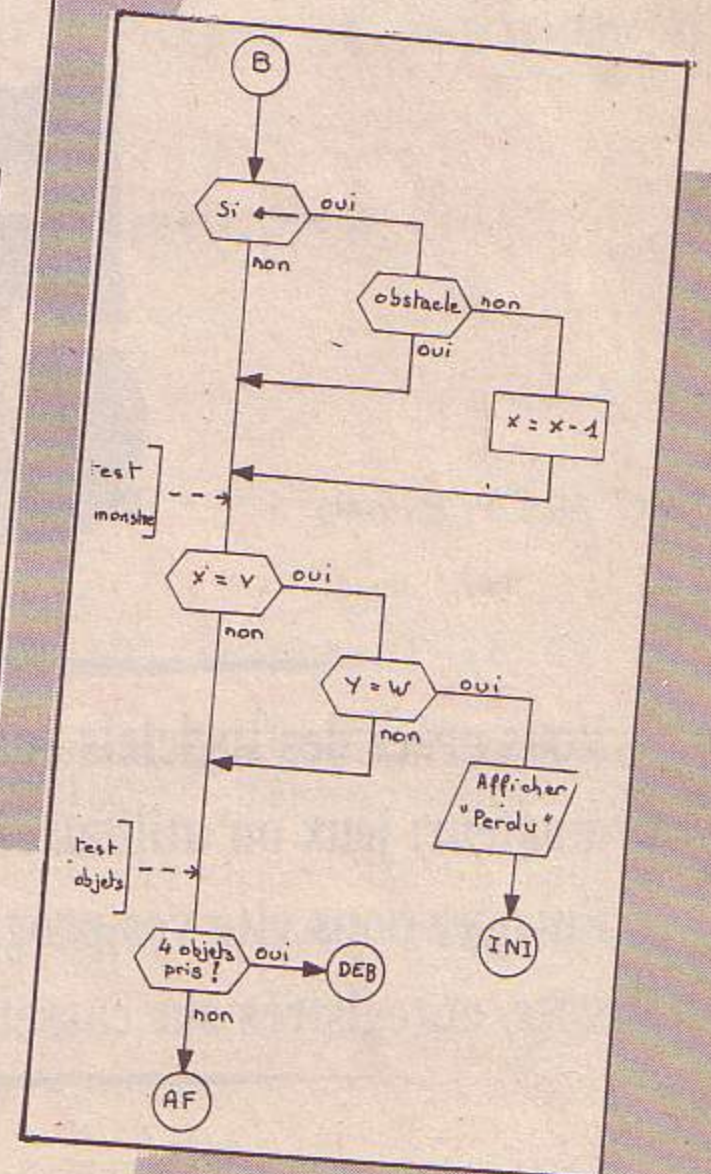
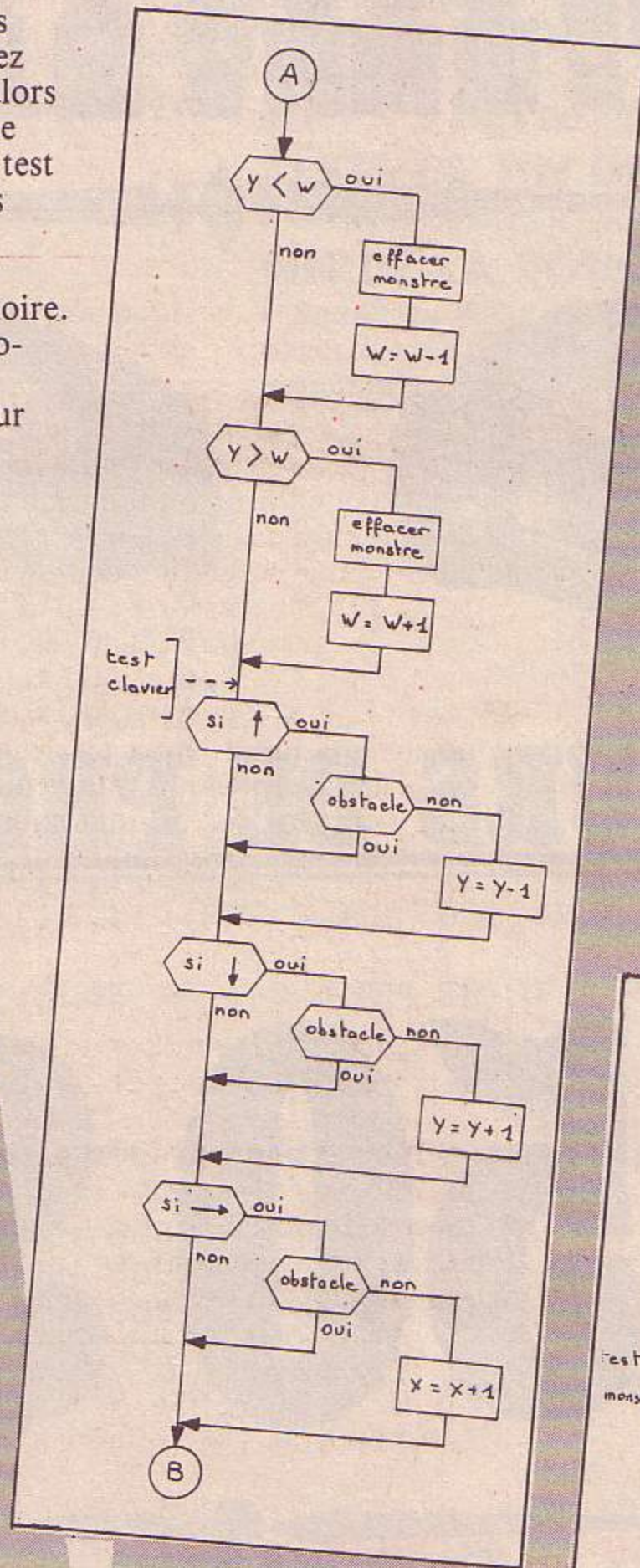
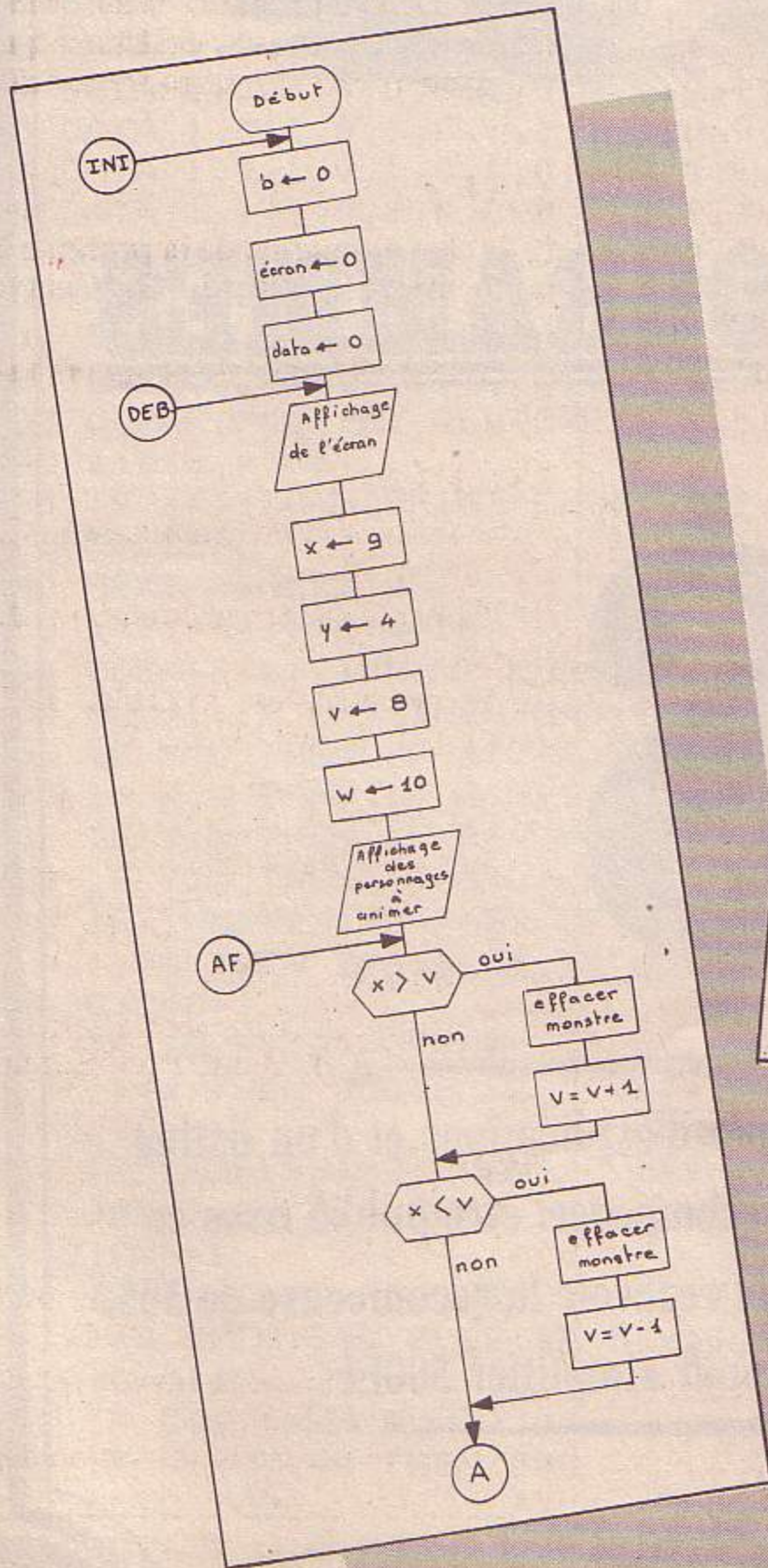


RECREATION

L'organigramme et le programme suivants vous permettront de réaliser un petit jeu. Vous devrez aider le personnage à ramasser quatre objets, alors qu'il est poursuivi par un monstre. C'est le type même du jeu comportant un grand nombre de test de collision, comme on peut le trouver dans les programmes de jeux d'arcades. Etudiez l'organigramme, l'algorithme et le programme. Regardez le résultat après l'avoir entré en mémoire. Essayez enfin d'écrire dans votre propre pseudo-langage le programme, de refaire un organigramme, ainsi qu'une documentation pour l'utilisateur. Bonne récréation.



EXPLICATIONS:

DEBUT :

- Initialisation des variables de l'écran et du tableau.

AFFICHAGES :

- Affichage de l'écran et initialisation des variables de position des différents objets (x et y pour le petit personnage ainsi que v et w pour le monstre).

AFFICHAGE ET ANIMATION DES PERSONNAGES :

- En premier lieu test de position afin de réaliser une poursuite. Les variables du monstre sont incrémentées ou décré- mentées en fonction de la position du petit bonhomme.

TEST CLAVIER :

- Le test est de la forme : si action alors vérifier s'il n'y a pas d'obstacle à la position d'arrivée oui, alors saut vers b non, actualiser x et y.

TESTS DE COLLISION :

- Ce test est de la forme si $x = y$, alors vérifier que $y = w$ et si $y = w$ alors fin du jeu et retour à d.

TESTS DE FIN DE TABLEAUX

- Si quatre objets sont pris alors le tableau est fini, retour à c.

FIN DE LA BOUCLE :

- Retour au début, c'est-à-dire vers l'affichage des person- nages à animer (l'affichage est réactualisé en fonction des variables).

```
1 REM listing amstrad
10 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
20 DATA 1,2,0,0,0,0,0,0,0,0,0,0,0,1,1,2,1
30 DATA 1,0,1,1,1,1,0,1,1,1,1,1,0,0,0,0,1
40 DATA 1,0,1,0,0,0,0,0,0,0,0,1,1,1,1,0,1
50 DATA 1,0,0,0,1,0,1,1,1,1,0,0,0,0,0,0,1
60 DATA 1,0,1,0,1,0,1,0,0,0,0,1,1,0,1,0,1
70 DATA 1,1,1,0,1,0,0,0,1,0,1,1,1,0,1,0,1
80 DATA 1,0,0,0,0,0,1,0,1,0,0,0,0,0,1,0,1
90 DATA 1,0,1,1,1,1,1,0,1,1,1,1,1,1,1,0,1
100 DATA 1,2,0,0,0,0,0,4,0,0,0,0,0,0,0,2,1
110 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
120 CLEAR:MODE 0: RESTORE:DIM b(17,11):FOR y =1 TO 11:FOR x=1 TO 17:LOCATE x,y:R
EAD a:b(x,y)=a
130 IF a=1 THEN PRINT CHR$(206):ELSE IF a=0 THEN PRINT " ":ELSE IF a=2 THEN PRIN
T CHR$(230)
140 NEXT x:NEXT y
150 x=9:y=4:v=8:w=10
160 LOCATE x,y:PRINT CHR$(249):LOCATE v,w:PRINT CHR$(225):B(v,w)=4
170 U=U+1:IF U=2 THEN U=0:ELSE GOTO 220
180 IF X>V THEN IF B(v+1,w)=0 THEN GOSUB 290:v=v+1
190 IF X<V THEN IF B(v-1,w)=0 THEN GOSUB 290:v=v-1
200 IF Y<W THEN IF B(v,w-1)=0 THEN GOSUB 290:w=w-1
210 IF Y>W THEN IF B(v,w+1)=0 THEN GOSUB 290:w=w+1
220 IF INKEY(2)=0 THEN IF b(x,y+1)=0 THEN LOCATE x,y:PRINT " ": y=y+1:ELSE IF b(
x,y+1)=4 THEN 290:ELSE IF b(x,y+1)=2 THEN b(x,y+1)=0
:A=50:C=C+1
230 IF INKEY(0)=0 THEN IF b(x,y-1)=0 THEN LOCATE x,y:PRINT " ":y=y-1 :ELSE IF b(
x,y-1)=4 THEN 300:ELSE IF b(x,y-1)=2 THEN b(x,y-1)=0
:A=50:C=C+1
240 IF INKEY(8)=0 THEN IF b(x-1,y)=0 THEN LOCATE x,y:PRINT " ":x=x-1:ELSE IF b(x
-1,y)=4 THEN 300:ELSE IF b(x-1,y)=2 THEN b(x-1,y)=0:
A=50:C=C+1
250 IF INKEY(1)=0 THEN IF b(x+1,y)=0 THEN LOCATE x,y:PRINT " ":x=x+1:ELSE IF b(x
+1,y)=4 THEN 300:ELSE IF b(x+1,y)=2 THEN B(x+1,y)=0:
A=50:C=C+1
260 IF C=4 THEN RUN
270 IF B(X,Y)=4 THEN 300
280 GOTO 160
290 LOCATE v,w:PRINT " ":B(v,w)=0:RETURN
300 FOR X=100 TO 200:SOUND 1,X,2:SOUND 2,X,2:NEXT:LOCATE 5,25:PRINT"PERDU":FOR X
=1 TO 200:NEXT :RUN
```