

9/9.4.1

Un poids deux mesures pour connaître les quatre opérations

Le bambin, qui utilisait le programme d'éveil (voir Partie 9, chapitre 9.4), il y a quelques années, a maintenant atteint l'âge de raison, et s'est lancé dans les études primaires.

Nous vous proposons ici encore de le motiver grâce à votre fidèle AMSTRAD CPC, en vous réalisant un programme qui lui fera découvrir et aimer, nous l'espérons, le calcul mental à l'aide des quatre opérations.

UN PROGRAMME SIMPLE

Bien sûr, il est possible de réaliser facilement ce genre de programme éducatif, en proposant des calculs aléatoires à réaliser, tel le programme suivant, que vous pourrez éventuellement modifier pour le rendre plus attrayant.

```
10 REM *****
20 REM * TRAVAIL SUR LES *
30 REM * QUATRE OPERATIONS *
40 REM *****
50 REM
60 REM *** ATTENTE CHOIX ***
70 MODK 1
80 PRINT "Type d'operation"
90 PRINT
100 PRINT TAB(10) "1 = addition"
110 PRINT TAB(10) "2 = soustraction"
120 PRINT TAB(10) "3 = multiplication"
130 PRINT TAB(10) "4 = division"
140 PRINT
150 PRINT TAB(5) "votre choix";
160 INPUT choix
170 REM
180 IF choix >5 OR choix <1 THEN RUN
190 ON choix GOTO 260,430,600,770
200 RUN
210 REM
220 REM *****
230 REM
240 REM *** ADDITION ***
250 REM
260 nombre1 = INT(RND(10) * 100)
270 nombre2 = INT(RND(10) * 100)
280 solution = nombre1 + nombre2
290 PRINT
```

```
300 PRINT STR$(nombre1) + CHR$(32) +
      CHR$(43) + STR$(nombre2) +
      CHR$(61) + CHR$(32);
310 PRINT "... " + CHR$(8) + CHR$(8)
      + CHR$(8);
320 INPUT resultat
330 IF solution = resultat THEN GOTO 370
340 PRINT "Erreur, la solution est";
350 PRINT SOLUTION
360 GOTO 260
370 PRINT "Bravo !"
380 GOTO 260
390 REM *****
400 REM
410 REM *** SOUSTRACTION ***
420 REM
430 nombre1 = INT(RND(10) * 100)
440 nombre2 = INT(RND(10) * 100)
450 solution = nombre1 - nombre2
460 PRINT
470 PRINT STR$(nombre1) + CHR$(32) +
      CHR$(45) + STR$(nombre2) +
      CHR$(61) + CHR$(32);
480 PRINT "... " + CHR$(8) + CHR$(8)
      + CHR$(8);
490 INPUT resultat
500 IF solution = resultat THEN GOTO 540
510 PRINT "Erreur, la solution est";
520 PRINT solution
530 GOTO 430
540 PRINT "Bravo !"
550 GOTO 260
560 REM *****
570 REM
580 REM *** MULTIPLICATION ***
590 REM
600 nombre1 = INT(RND(10) * 100)
610 nombre2 = INT(RND(10) * 100)
620 solution = nombre1 * nombre2
630 PRINT
640 PRINT STR$(nombre1) + CHR$(32) +
      CHR$(42) + STR$(nombre2) +
      CHR$(61) + CHR$(32);
650 PRINT "... " + CHR$(8) + CHR$(8)
      + CHR$(8);
660 INPUT resultat
670 IF solution = resultat THEN GOTO 710
680 PRINT "Erreur, la solution est";
690 PRINT solution
700 GOTO 600
710 PRINT "Bravo !"
720 GOTO 600
730 REM *****
740 REM
750 REM *** DIVISION ***
760 REM
770 nombre1 = INT(RND(10) * 100)
780 solution = INT(RND(10) * 100)
790 nombre2 = nombre1 * solution
800 PRINT
```

```
810 PRINT STR$(nombre2) + CHR$(32) +  
      CHR$(47) + STR$(nombre1) +  
      CHR$(61) + CHR$(32);  
820 PRINT "... " + CHR$(8) + CHR$(8)  
      + CHR$(8);  
830 INPUT resultat  
840 IF solution = resultat THEN GOTO 710  
850 PRINT "Erreur, la solution est";  
860 PRINT solution  
870 GOTO 770  
880 PRINT "Bravo !"  
890 GOTO 770
```

Ce programme se compose d'une structure de choix portant sur chacune des opérations $+$, $-$, $*$, et $/$.

De présentation simple, pour chacune des opérations, il recherche deux nombres aléatoires compris entre 0 et 99, qu'il propose, et demande le résultat de l'opération entre ces deux nombres.

Le mécanisme de la division est légèrement différent, car suite à la génération de deux nombres, l'un divisé par l'autre peut produire un résultat décimal, difficilement résolvable par la petite tête blonde. Aussi, l'un de ces nombres est considéré comme étant la solution, l'autre, le diviseur, il suffit de rechercher le dividende en multipliant le diviseur par la solution.

De présentation ingrate et de fonctionnement très simple, ce programme risque de ne pas motiver longtemps notre écolier, sauf s'il possède vraiment la bosse des maths.

UN PROGRAMME PLUS ATTRAYANT

Après réflexion, nous avons envisagé de modifier le programme, tant au niveau de la façon d'interroger sur le résultat, que sur la présentation.

Deux pierres d'un coup

Nous avons décidé, non pas pour piéger notre génie en herbe, mais pour motiver sa réflexion, de réaliser deux opérations en une.

En effet, pour une opération quelconque, au lieu de demander le résultat de la composition de deux nombres pris au hasard, nous interrogeons sur le nombre qui manque pour réaliser une opération exacte entre un nombre proposé, et le résultat de la dite opération.

Rien de machiavélique là-dedans (essayez donc le programme pour l'adopter), juste un stade supplémentaire dans la réflexion, puisqu'il faut d'abord rechercher la solution par l'opération inverse, et bien sûr, la vérifier en recomposant l'opération proposée.

La présentation

Pour augmenter l'intérêt du programme, et le transformer en jeu, nous vous proposons de réaliser un équilibre sur une balance de type Roberval.

Dès qu'une proposition de chiffre sera validée, la balance indiquera par son aiguille le résultat :

- si la solution proposée réalise une opération plus élevée que le résultat attendu, la balance penche à gauche ;
- si la solution proposée réalise une opération moins élevée, la balance penche du côté du résultat attendu ;
- une solution correcte équilibre, bien entendu, la balance.

Pour jouer, notre écolier devra sélectionner les chiffres un à un parmi les poids proposés, à l'aide des touches curseur droit ($->$) ou gauche ($<-$). La prise en main d'un poids s'effectue à l'aide de la touche $<ESPACE>$, et le dépôt de tous les poids choisis est validé par $<RETURN>$.

Lors de la sélection, les chiffres choisis sont placés en rotation sur trois chiffres, il est ainsi possible de corriger une erreur, à temps, en sélectionnant d'autres chiffres (par exemple, si 123 est affiché, la sélection du chiffre 4 provoque l'affichage de 234).

Nous avons aussi pensé aux passionnés du joystick, ce qui évitera les frappes rageuses sur le clavier, après des essais infructueux.

Il est ainsi possible d'équilibrer la balance avec le manche habituellement réservé aux jeux d'arcades.

- Le joystick à gauche déplace la flèche de choix à gauche.
- Le joystick à droite déplace vers la droite la flèche de choix.
- Le joystick vers le haut permet de prendre en main le poids pointé.
- Un appui sur le bouton de FEU valide la composition réalisée.

Nous avons même prévu un concours entre matheux, ou pourquoi pas un défi de toute la famille, en affichant le score le plus élevé pour chacune des opérations.

Par opération sélectionnée, dix propositions seront affichées, chaque proposition comportant cinq essais (qui sont comptabilisés).

Vous pourrez à loisirs modifier le nombre d'essais permis, car nous vous fournirons les explications nécessaires sur le programme.

L'algorithme général

Nous vous proposons ici l'algorithme général du programme éducatif de calcul.

- DEBUT
 - TANTQUE qu'un exercice est choisi
 - PROCEDURE AFFICHAGE MENU
 - Initialiser les variables
 - TANTQUE le nombre d'opérations exécutées est inférieur à 10
 - SELONCAS
 - "CHOIX 1" : PROCEDURE ADDITION
 - "CHOIX 2" : PROCEDURE SOUSTRACTION
 - "CHOIX 3" : PROCEDURE MULTIPLICATION
 - "CHOIX 4" : PROCEDURE DIVISION
 - FINCAS
 - SI le nombre d'essais maximal n'a pas été atteint
 - ALORS
 - afficher la justesse du résultat
 - Mettre à jour les scores
 - FINSI
 - Incrémenter le nombre de coups
 - Afficher les scores
 - FINTANTQUE
 - FINTANTQUE
 - FIN

Cet algorithme contient une première procédure d'affichage du menu, que nous n'étudierons pas dans le détail, à cause de sa simplicité.

Nous nous attacherons aux procédures d'addition, soustraction, multiplication et division. Ces quatre procédures, mise à part la différence dans le type d'opérations, se ressembleront assez, aussi, l'algorithme de l'addition sera détaillé, le parallèle étant facilement réalisable pour les trois autres opérations.

- *Procédure addition*

- DEBUT
 - Générer deux nombres aléatoires
 - Calculer la solution de la somme de ces deux nombres
 - Afficher le questionnaire
 - PROCEDURE ACQUERIR LES POIDS

- SI le résultat est trop grand
 - ALORS
 - PROCEDURE PENCHER LA BALANCE A GAUCHE
- FINSI
- SI le résultat est trop petit
 - ALORS
 - PROCEDURE PENCHER LA BALANCE A DROITE
 - SINON
 - PROCEDURE EQUILIBRER LA BALANCE
- FINSI
- FIN

Dans cet algorithme, nous ne détaillerons pas les procédures concernant le mouvement de la balance, celles-ci étant purement graphiques. Nous nous attacherons plutôt à l'acquisition des poids dans l'algorithme suivant :

- *Procédure acquérir les poids*

- DEBUT
 - PROCEDURE AFFICHER LES POIDS
 - PROCEDURE DESSINER L'AIGUILLE DE LA BALANCE
 - PROCEDURE DESSINER LES PLATEAUX
 - Initialiser les variables de calcul
 - TANTQUE le nombre maximal d'essais n'est pas atteint et que la réponse n'est pas trouvée
 - Initialiser les variables
 - TANTQUE le poids n'est pas validé
 - PROCEDURE PRISE EN MAIN DES POIDS
 - FINTANTQUE
 - FINTANTQUE
- FIN

Ici encore, quatre procédures graphiques sur l'écran du CPC, pour gérer la balance et les poids.

Nous ne développerons pas plus les algorithmes du reste du programme, beaucoup de procédures graphiques étant utilisées, ainsi que celles de positionnement de textes dans l'affichage des scores.

Le programme

Le programme ainsi développé sera le suivant :

```

10 REM *****
20 REM *      PROGRAMME  EDUCATIF      *
30 REM *      DE CALCUL SUR LES      *
40 REM *      QUATRE OPERATIONS      *
50 REM *      CONSISTANT A TROUVER    *
60 REM *      LE NOMBRE MANQUANT     *
70 REM *      POUR EQUILIBRER        *
80 REM *      UNE BALANCE             *
90 REM *****
100 REM
110 BORDER 1
120 MODE 1
130 REM
140 INK 0,0:INK 1,26
150 INK 2,6:INK 3,5
160 REM
170 DIM scores(4)
180 REM
190 REM ***** MENU *****
200 REM
210 choix = 8
220 WHILE choix <> 5
230   CLS
240   LOCATE 14,10
250   PRINT "* MENU *"
260   LOCATE 10,12
270   PRINT"1 : ADDITION"
280   LOCATE 10,14
290   PRINT"2 : SOUSTRACTION"
300   LOCATE 10,16
310   PRINT"3 : MULTIPLICATION"
320   LOCATE 10,18
330   PRINT"4 : DIVISION"
340   LOCATE 10,20
350   PRINT"5 : FIN DE CALCUL"
360   LOCATE 12,22
370   REM
380   REM *****
390   REM
400   INPUT "VOTRE CHOIX : ",CHOIX
410   CLS
420   IF CHOIX = 0 THEN choix = 5
430   IF CHOIX > 5 THEN choix = 5
440   IF choix <> 5 THEN GOSUB 4410
450   REM
460   nombrecoup = 1
470   nombrepoint = 0
480   nombressai = 0
490   WHILE nombrecoup < 10 AND
      choix <> 5
500     LOCATE 37,7
510     PRINT nombrecoup
520     ON choix GOSUB 4650,4900,5220,
      5510

```

```

530         IF (nombressai = 6 AND
              solution) = result THEN
              GOTO 630
540         LOCATE 1,16
550         PRINT "Bon resultat : ";
560         PRINT solution
570         CALL &BB1E
580         FOR attente = 1 TO 700
590         NEXT attente
600         CALL &BB06
610         LOCATE 1,16
620         PRINT SPACES(20)
630         nombrecoup = nombrecoup + 1
640         result = 0
650         LOCATE 7,14
660         PRINT SPACES(19)
670         couleur = 0
680         GOSUB 2630
690         IF scores(choix) >=
              nombrepoint THEN GOTO 730
700         scores(choix) = nombrepoint
710         LOCATE 37,13
720         PRINT scores(choix)
730     WEND
740 WEND
750 REM
760 END : REM FIN DE CALCUL
770 REM *****
780 REM
790 REM *** AFFICHAGE RANGEE NOMBRE ***
800 REM
810 couleur = 2
820 abscisse = 63
830 GOSUB 2630
840 REM
850 REM affichage rangee de nombre
860 REM
870 GOSUB 2950
880 LOCATE 1,23
890 PEN 1
900 FOR I = 0 TO 9
910     PRINT CHR$(32);
920     PRINT I;
930 NEXT
940 PEN 1
950 REM
960 REM *****
970 REM
980 REM cadre balance
990 REM
1000 GOSUB 5960
1010 REM
1020 REM plateaux balance
1030 REM
1040 GOSUB 3190
1050 REM
1060 REM *** METTRE MENU ***
1070 REM
1080 REM CHOIX ADDITION, SOUSTRACTION
1090 REM     MULTIPLICATION, DIVISION

```



```

1100 REM
1110 pointeur = 1
1120 unites = 0
1130 dizaines = unites
1140 centaines = unites
1150 select = 0
1160 nombressai = 1
1170 WHILE (nombressai <= 5 AND
           result <> solution)
1180     LOCATE 1,16
1190     PRINT"Appuyez sur une touche"
1200     FOR attente = 1 TO 1500
1210     NEXT attente
1220     CALL &BB06
1230     LOCATE 1,16
1240     PRINT SPACES(23)
1250     LOCATE 1,14
1260     PRINT SPACES(20)
1270     centaines = 0
1280     dizaines = centaines
1290     unites = centaines
1300     GOSUB 5810
1310     LOCATE 37,10
1320     PRINT nombressai
1330     a$ = " rien "
1340     WHILE (a$ <> CHR$(13) AND
           JOY(0) <> 16)
1350         IF (a$ <> CHR$(242) AND
           JOY(0) <> 4) THEN
           GOTO 1430
1360         couleur = 0
1370         GOSUB 2630
1380         couleur = 2
1390         abscisse = abscisse
           - 64
1400         select = select - 1
1410         GOSUB 2840
1420         GOSUB 2630
1430         IF (a$ <> CHR$(243) AND
           JOY(0) <> 8) THEN
           GOTO 1510
1440         couleur = 0
1450         GOSUB 2630
1460         couleur = 2
1470         abscisse = abscisse
           + 64
1480         select = select + 1
1490         GOSUB 2840
1500         GOSUB 2630
1510         a$ = INKEY$
1520         WHILE a$ = " " OR
           JOY(0) = 1
1530         FOR attente = 1 TO 500
1540         NRXT attente
1550             IF pointeur <> 1
           THEN GOTO 1610
1560             unites = select
1570             LOCATE 32,10
1580             GOSUB 5810

```

```

1590         pointeur = 2
1600         GOTO 1740
1610         IF pointeur <> 2
                THEN GOTO 1680
1620         dizaines = unites
1630         unites = select
1640         LOCATE 32,10
1650         GOSUB 5810
1660         pointeur = 3
1670         GOTO 1740
1680         IF pointeur <> 3
                THEN GOTO 1740
1690         centaines =
                dizaines
1700         dizaines =
                unites
1710         unites = select
1720         LOCATE 32,10
1730         GOSUB 5810
1740         a$ = ""
1750         WEND
1760     WEND
1770     REM
1780     REM EFFACEMENT DROITE/GAUCHE
1790     REM
1800     GOSUB 4130
1810     IF choix <> 1 THEN GOTO 1940
1820     calcul = result + nombre1
1830     PRINT calcul
1840     IF result >= solution THEN
        GOTO 1870
1850     GOSUB 3680
1860     GOTO 1940
1870     IF result <= solution THEN
        GOTO 1900
1880     GOSUB 3570
1890     GOTO 1940
1900     nombrepoint = nombrepoint + 1
1910     LOCATE 37,4
1920     PRINT nombrepoint
1930     GOSUB 3410
1940     IF choix <> 3 THEN GOTO 2080
1950     calcul = result * nombre1
1960     LOCATE 7,14
1970     PRINT calcul
1980     IF result >= solution THEN
        GOTO 2010
1990     GOSUB 3680
2000     GOTO 2080
2010     IF result <= solution THEN
        GOTO 2040
2020     GOSUB 3570
2030     GOTO 2080
2040     nombrepoint = nombrepoint + 1
2050     LOCATE 37,4
2060     PRINT nombrepoint
2070     GOSUB 3410
2080     IF choix <> 2 THEN GOTO 2220
2090     calcul = nombre1 - result

```

```

2100      LOCATE 7,14
2110      PRINT calcul
2120      IF result <= solution THEN
           GOTO 2390
2130      GOSUB 3680
2140      GOTO 2220
2150      IF result >= solution THEN
           GOTO 2180
2160      GOSUB 3570
2170      GOTO 2220
2180      nombrepoint = nombrepoint + 1
2190      LOCATE 37,4
2200      PRINT nombrepoint
2210      GOSUB 3410
2220      IF choix <> 4 THEN
           GOTO 2390
2230      IF result <> 0 THEN
           GOTO 2260
2240      GOSUB 2430
2250      GOTO 2390
2260      calcul = nombre1 / result
2270      LOCATE 7,14
2280      PRINT calcul
2290      IF result <= solution THEN
           GOTO 2320
2300      GOSUB 3680
2310      GOTO 2390
2320      IF result >= solution THEN
           GOTO 2350
2330      GOSUB 3570
2340      GOTO 2390
2350      nombrepoint = nombrepoint + 1
2360      LOCATE 37,4
2370      PRINT nombrepoint
2380      GOSUB 3410
2390      nombressai = nombressai + 1
2400 WEND
2410 RETURN
2420 REM
2430 REM * MESSAGE DIVISION PAR ZERO *
2440 REM
2450 FOR i=1 TO 10
2460     LOCATE 1,3
2470     PRINT CHR$(7);
2480     PRINT "pas de division par 0"
2490     LOCATE 1,3
2500     PRINT SPACES(21)
2510 NEXT
2520 RETURN
2530 REM *****
2540 REM
2550 REM * affichage rangee de nombre *
2560 REM
2570 LOCATE 20,12
2580 FOR I= 0 TO 9
2590     PRINT ,SPACES(1);
2600     PRINT i
2610 NEXT
2620 RETURN

```

```
2630 ordonnee = 115
2640 PLOT abscisse,ordonnee,couleur
2650 ordonnee = ordonnee - 20
2660 DRAW abscisse,ordonnee,couleur
2670 abscisse = abscisse + 25
2680 DRAW abscisse,ordonnee
2690 abscisse = abscisse - 50
2700 ordonnee = ordonnee - 30
2710 DRAW abscisse,ordonnee
2720 abscisse = abscisse - 50
2730 ordonnee = ordonnee + 30
2740 DRAW abscisse,ordonnee
2750 abscisse = abscisse + 25
2760 DRAW abscisse,ordonnee
2770 ordonnee = ordonnee + 20
2780 DRAW abscisse,ordonnee
2790 abscisse = abscisse + 50
2800 DRAW abscisse,ordonnee
2810 RETURN
2820 REM *****
2830 REM
2840 REM ** TEST DEPASSEMENT ECRAN **
2850 REM
2860 IF abscisse <= 639 THEN GOTO 2890
2870 abscisse = 63
2880 select = 0
2890 IF abscisse >= 63 THEN GOTO 2920
2900 abscisse = 639
2910 select = 9
2920 RETURN
2930 REM *****
2940 REM
2950 REM * EFFET TOURNANT SUR CHIFFRE *
2960 REM
2970 ordchiffre = 40
2980 FOR abschiffre = 19 TO 600 STEP 64
2990     PLOT abschiffre,b,3
3000     abschiffre = abschiffre + 20
3010     ordchiffre = ordchiffre + 20
3020     DRAW abschiffre,ordchiffre
3030     abschiffre = abschiffre + 20
3040     ordchiffre = ordchiffre - 20
3050     DRAW abschiffre,ordchiffre
3060     abschiffre = abschiffre - 20
3070     ordchiffre = ordchiffre - 20
3080     DRAW abschiffre,ordchiffre
3090     abschiffre = abschiffre - 20
3100     ordchiffre = ordchiffre + 20
3110     DRAW abschiffre,ordchiffre
3120     MOVE abschiffre + 10,
           ordchiffre+2
3130     FILL 3
3140     ORIGIN 0,0
3150 NEXT abschiffre
3160 RETURN
3170 REM *****
3180 REM
3190 REM *** CADRE DE LA BALANCE ***
3200 REM
```

```
3210 absbalance = 200
3220 ordbalance = 250
3230 inkbalance = 1
3240 PLOT absbalance, ordbalance,
      inkbalance
3250 absbalance = absbalance - 30
3260 ordbalance = ordbalance + 60
3270 DRAW absbalance, ordbalance
3280 absbalance = absbalance + 30
3290 ordbalance = ordbalance + 30
3300 DRAW absbalance, ordbalance
3310 absbalance = absbalance + 30
3320 ordbalance = ordbalance - 30
3330 DRAW absbalance, ordbalance
3340 absbalance = absbalance - 30
3350 ordbalance = ordbalance - 60
3360 DRAW absbalance, ordbalance
3370 RETURN
3380 REM
3390 REM ***** AIGUILLE *****
3400 REM
3410 REM egalite
3420 REM
3430 absbalance = 200
3440 ordbalance = 250
3450 inkbalance = 1
3460 PLOT absbalance, ordbalance,
      inkbalance
3470 ordbalance = ordbalance + 60
3480 DRAW absbalance, ordbalance
3490 FOR i=1 TO 5
3500     LOCATE 1,10
3510     PRINT "resultat exact"
3520     LOCATE 1,10
3530     PRINT SPACES(15)
3540 NEXT i
3550 RETURN
3560 REM
3570 REM a gauche
3580 REM
3590 absbalance = 200
3600 ordbalance = 250
3610 inkbalance = 1
3620 PLOT absbalance, ordbalance,
      inkbalance
3630 absbalance = absbalance - 15
3640 ordbalance = ordbalance + 60
3650 DRAW absbalance, ordbalance
3660 RETURN
3670 REM
3680 REM a droite
3690 REM
3700 absbalance = 200
3710 ordbalance = 250
3720 inkbalance = 1
3730 PLOT absbalance, ordbalance,
      inkbalance
3740 absbalance = absbalance + 15
3750 ordbalance = ordbalance + 60
```

```
3760 DRAW absbalance,ordbalance
3770 RETURN
3780 REM
3790 REM ***** EFFACE *****
3800 REM
3810 REM efface si egalite
3820 REM
3830 absbalance = 200
3840 ordbalance = 250
3850 inkbalance = 0
3860 PLOT absbalance,ordbalance,
      inkbalance
3870 ordbalance = ordbalance + 60
3880 DRAW absbalance,ordbalance
3890 RETURN
3900 REM
3910 REM efface a gauche
3920 REM
3930 absbalance = 200
3940 ordbalance = 250
3950 inkbalance = 0
3960 PLOT absbalance,ordbalance,
      inkbalance
3970 absbalance = absbalance - 15
3980 ordbalance = ordbalance + 60
3990 DRAW absbalance,ordbalance
4000 RETURN
4010 REM
4020 REM efface a droite
4030 REM
4040 absbalance = 200
4050 ordbalance = 250
4060 inkbalance = 0
4070 PLOT absbalance,ordbalance,
      inkbalance
4080 absbalance = absbalance + 15
4090 ordbalance = ordbalance + 60
4100 DRAW absbalance,ordbalance
4110 RETURN
4120 REM
4130 REM *** EFFACE DROITE GAUCHE ***
4140 REM
4150 absbalance = 200
4160 ordbalance = 250
4170 inkbalance = 0
4180 PLOT absbalance,ordbalance,
      inkbalance
4190 ordbalance = ordbalance + 60
4200 DRAW absbalance,ordbalance
4210 absbalance = 200
4220 ordbalance = 250
4230 inkbalance = 0
4240 PLOT absbalance,ordbalance,
      inkbalance
4250 absbalance = absbalance - 15
4260 ordbalance = ordbalance + 60
4270 DRAW absbalance,ordbalance
4280 absbalance = 200
4290 ordbalance = 250
```

```
4300 inkbalance = 0
4310 PLOT absbalance,ordbalance,
      inkbalance
4320 absbalance = absbalance + 15
4330 ordbalance = ordbalance + 60
4340 DRAW absbalance,ordbalance
4350 RETURN
4360 REM
4370 REM ***** MODULE SCORE *****
4380 REM
4390 REM premier cadre
4400 REM
4410 PLOT 402,390,2
4420 DRAW 402,130
4430 DRAW 634,130
4440 DRAW 634,390
4450 DRAW 402,390
4460 REM
4470 REM deuxieme cadre
4480 REM
4490 PLOT 406,386
4500 DRAW 406,134
4510 DRAW 630,134
4520 DRAW 630,386
4530 DRAW 406,386
4540 LOCATE 27,4
4550 PRINT "SCORE" + SPACES(4) +
      CHR$(58)+CHR$(32)
4560 LOCATE 27,7
4570 PRINT"NB COUPS : "
4580 LOCATE 27,10
4590 PRINT"NB ESSAI : "
4600 LOCATE 27,13
4610 PRINT"MEILLEUR :";scores(choix)
4620 REM
4630 RETURN
4640 REM
4650 REM ***** addition *****
4660 REM
4670 nombre2 = INT(RND * 100)
4680 nombre1 = INT(RND * nombre2)
4690 solution = nombre2 - nombre1
4700 LOCATE 3,12
4710 PRINT nombre1
4720 LOCATE 7,12
4730 PRINT CHR$(43)
4740 LOCATE 9,12
4750 PRINT CHR$(32) + CHR$(63) +
      CHR$(32)
4760 LOCATE 13,12
4770 PRINT CHR$(61)
4780 LOCATE 16,12
4790 PRINT nombre2
4800 GOSUB 790
4810 IF result >= solution THEN
      GOTO 4840
4820 GOSUB 3680
4830 GOTO 4880
```

```
4840 IF result <= solution THEN
      GOTO 4870
4850 GOSUB 3570
4860 GOTO 4880
4870 GOSUB 3410
4880 RETURN
4890 REM
4900 REM **** MODULE SOUSTRACTION ****
4910 REM
4920 nombre1 = 1:nombre2 = 2
4930 WHILE nombre1 < nombre2 OR k = 0

4940 nombre1 = INT(RND * 100)
4950 nombre2 = INT(RND * 100)
4960 WEND
4970 PRINT nombre1
4980 PRINT nombre2
4990 PRINT nombre1 - nombre2
5000 CALL &BB06
5010 solution = nombre1 - nombre2
5020 LOCATE 3,12
5030 PRINT nombre1
5040 LOCATE 7,12
5050 PRINT "-"
5060 LOCATE 9,12
5070 PRINT CHR$(32) + CHR$(63) +
      CHR$(32)
5080 LOCATE 13,12
5090 PRINT CHR$(61)
5100 LOCATE 16,12
5110 PRINT nombre2
5120 GOSUB 790
5130 IF result <= solution THEN
      GOTO 5160
5140 GOSUB 3680
5150 GOTO 5200
5160 IF result >= solution THEN
      GOTO 5190
5170 GOSUB 3570
5180 GOTO 5200
5190 GOSUB 3410
5200 RETURN
5210 REM
5220 REM *** MODULE MULTIPLICATION ***
5230 REM
5240 nombre1 = 0
5250 solution = 0
5260 WHILE nombre1 = 0 OR solution = 0
      OR nombre2 = 0
5270 nombre1 = INT(RND * 10)
5280 solution = INT(RND * 10)
5290 nombre2 = solution * nombre1
5300 WEND
5310 LOCATE 4,12
5320 PRINT nombre1
5330 LOCATE 7,12
5340 PRINT CHR$(42)
5350 LOCATE 9,12
5360 PRINT CHR$(32) + CHR$(63) +
      CHR$(32)
```



```
5370 LOCATE 13,12
5380 PRINT CHR$(61)
5390 LOCATE 16,12
5400 PRINT nombre2
5410 GOSUB 790
5420 IF result <= solution THEN
    GOTO 5450
5430 GOSUB 3570
5440 GOTO 5490
5450 IF result >= solution THEN
    GOTO 5480
5460 GOSUB 3680
5470 GOTO 5490
5480 GOSUB 3410
5490 RETURN
5500 REM
5510 REM **** MODULE DIVISION ****
5520 REM
5530 nombre2 = 0
5540 solution = 0
5550 nombre1 = 0
5560 WHILE nombre2 = 0 OR solution = 0
    OR nombre1 = 0
5570 nombre2 = INT(RND * 10)
5580 solution = INT(RND * 10)
5590 nombre1 = solution * nombre2
5600 WEND
5610 LOCATE 3,12
5620 PRINT nombre1
5630 LOCATE 7,12
5640 PRINT CHR$(47) : REM ou chr$(58)
5650 LOCATE 9,12
5660 PRINT CHR$(63)
5670 LOCATE 13,12
5680 PRINT CHR$(61)
5690 LOCATE 16,12
5700 PRINT nombre2
5710 GOSUB 790
5720 IF result >= solution THEN
    GOTO 5750
5730 GOSUB 3570
5740 GOTO 5790
5750 IF result <= solution THEN
    GOTO 5780
5760 GOSUB 3680
5770 GOTO 5790
5780 GOSUB 3410
5790 RETURN
5800 REM
5810 REM *** affichage saisie ***
5820 REM
5830 unites$ = STR$(unites)
5840 dizaines$ = STR$(dizaines)
5850 centaines$ = STR$(centaines)
5860 result$ = centaines$ + dizaines$
    + unites$
5870 result = VAL(result$)
5880 LOCATE 8,12
5890 PRINT SPACES(4)
```

```
5900 LOCATE 8,12
5910 PRINT result
5920 RETURN
5930 REM
5940 REM *** PLATEAUX BALANCE ***
5950 REM
5960 ORIGIN 84,258
5970 DRAW 40,0
5980 ORIGIN 126,260
5990 DRAW 5,5
6000 ORIGIN 82,260
6010 DRAW -5,5
6020 ORIGIN 274,258
6030 DRAW 40,0
6040 ORIGIN 316,260
6050 DRAW 5,5
6060 ORIGIN 272,260
6070 DRAW -5,5
6080 LOCATE 7,9
6090 PRINT CHR$(244)
6100 LOCATE 19,9
6110 PRINT CHR$(244)
6120 ORIGIN 0,0
6130 RETURN
```

Vous retrouverez dans ce programme suffisamment de commentaires pour retrouver les algorithmes présentés, et éventuellement procéder à des modifications graphiques, pour amuser votre élève.

MODIFICATIONS POUR CPC 464

Ce programme a été développé sur CPC 6128, et doit fonctionner sans problème sur le CPC 664. Par contre, la version du Basic étant moins performante sur le CPC 464, quelques modifications doivent être apportées pour que le programme fonctionne sur ce dernier.

SPACES

Commençons par l'instruction **PRINT SPACES**(nombre), que vous devrez remplacer par **PRINT " "**, le nombre d'espaces compris entre les guillemets étant égal au nombre spécifié dans les parenthèses.

Un problème, n'apparaissant pas dans le programme de calcul, peut arriver dans certains programmes : si le nombre spécifié n'est pas connu à l'avance, mais est le résultat d'un savant calcul effectué auparavant dans le programme. Dans ce cas, il faut créer une boucle itérative d'affichage du type :

```
10 FOR I = 1 TO nombre
20 PRINT " ";
30 NEXT I
```

FILL

Une instruction intéressante des CPC 664 et 6128 n'existe pas sur CPC 464, il s'agit de l'instruction de remplissage FILL couleur, qui remplit une surface fermée avec la couleur du bord de cette surface.

Nous vous proposons de la réaliser par simulation, à l'aide d'un sous-programme Basic.

Il suffit de positionner le point de départ du remplissage dans la surface à remplir, de tester les différents points horizontaux par rapport à ce point (à droite et à gauche de celui-ci), jusqu'au bord de la figure.

On trace ensuite une ligne, à la couleur du bord, entre ces deux extrêmes.

On procède de même en remontant de deux pixels (sachez que la définition de l'AMSTRAD CPC n'est pas de 640×400, mais de 640×200, les points verticaux étant doubles). Cette remontée s'effectue jusqu'au bord supérieur.

La même procédure est utilisée pour atteindre le bord inférieur à partir du point initial.

```

10000 REM *****
10010 REM *   SOUS-PROGRAMME BASIC   *
10020 REM * SIMULANT L'INSTRUCTION *
10030 REM *   FILL DU CPC-6128   *
10040 REM * ----- *
10050 REM * il faut positionner le *
10060 REM * point de depart par   *
10070 REM * l'instruction MOVE   *
10080 REM * puis initialiser      *
10090 REM * pour le remplissage  *
10100 REM * et appeler le        *
10110 REM * sous-programme      *
10120 REM *****
10130 REM
10140 REM POSITIONNEMENT SUR
10150 REM SITUE SUR DES COORDONNEES
10160 REM PAIRES EN ABSCISSE ET EN
10170 REM ORDONNEES
10180 REM
10190 MOV# 2*INT(XPOS/2),2*INT(YPOS/2)
10200 REM
10210 REM *****
10220 REM
10230 REM TEST SI LE POSITIONNEMENT
10240 REM CORRECT, OU SUR UN BORD
10250 REM DE LA FIGURE A REMPLIR
10260 REM
10270 IF tstr(0,0) <> couleur THEN GOTO
10340
10280 RETURN : REM ON EST SUR UN BORD
10290 REM
10300 REM *****
10310 REM
10320 REM SAUVEGARDE POSITION

```

```
10330 REM
10340 positionx = XPOS
10350 positiony = YPOS
10360 REM
10370 REM *****
10380 REM
10390 REM POSITION GAUCHE A RECUPERER
10400 positiongauche = XPOS
10410 REM
10420 REM *****
10430 REM
10440 REM RECHERCHE DU BORD GAUCHE
10450 REM
10460 WHILE TESTR(-2,0) <> couleur AND X
POS > 0
10470 WEND
10480 positiongauche = XPOS + 2
10490 REM
10500 REM *****
10510 REM
10520 REM RECHERCHE DU BORD DROIT
10530 REM
10540 WHILE TESTR(2,0)<>couleur AND XPOS
< 640
10550 WEND
10560 positiondroite = XPOS
10570 REM
10580 REM *****
10590 REM
10600 REM REMPLISSAGE D'UNE LIGNE
10610 REM
10620 DRAWR positiongauche - positiondro
ite,0
10630 REM
10640 REM *****
10650 REM
10660 REM REPOSITIONNEMENT
10670 REM
10680 MOVER positiondroite - positiongau
che,2
10690 IF YPOS > 399 THEN 10800
10700 WHILE TESTR(-2,0) = couleur AND po
sitiondroite > positiongauche
10710 positiondroite = XPOS
10720 WEND
10730 IF POSITIONNDROITE>POSITIONNGAUCHE T
HKN 10460
10740 REM
10750 REM *****
10760 REM
10770 REM REPOSITIONNEMENT EN DESSOUS
10780 REM POUR FAIRE LE BAS
10790 REM
10800 MOVE positionx,positiony-2
10810 REM
10820 REM *****
10830 REM *****
10840 REM
10850 REM RECHERCHE DU BORD GAUCHE
10860 REM
```

```

10870 WHILE TESTR(-2,0) <> couleur AND X
POS > 0
10880 WEND
10890 positiongauche = XPOS + 2
10900 REM
10910 REM *****
10920 REM
10930 REM RECHERCHE DU BORD DROIT
10940 REM
10950 WHILE TESTR(2,0)<>couleur AND XPOS
< 640
10960 WEND
10970 positiondroite = XPOS
10980 REM
10990 REM *****
11000 REM
11010 REM REMPLISSAGE D'UNE LIGNE
11020 REM
11030 DRAWR positiongauche - positiondro
ite,0
11040 REM
11050 REM *****
11060 REM
11070 REM REPOSITIONNEMENT EN LIGNE BASS
E
11080 REM
11090 MOVER positiondroite - positiongau
che,-2
11100 WHILE TESTR(-2,0) = couleur AND po
sitiondroite > positiongauche
11110 positiondroite = XPOS
11120 WEND
11130 IF YPOS<0 THEN RETURN
11140 IF positiondroite <= positiongauch
e THEN RETURN
11150 GOTO 10830
11160 REM
11170 REM *****
11180 REM

```

Nous avons fait démarrer ce sous-programme en ligne 10000, mais un simple RENUM *numérodeline* le replacera où vous le désirez.

Pour cela, l'insérer dans le programme de calcul, gardez la numérotation débutant en 10000, et remplacez la ligne :

```
3130 FILL 3
```

par

```
3130 couleur = 3 : GOSUB 10000
```

Bien sûr, cette routine est moins rapide que le FILL du CPC 6128, mais elle remplit son rôle dans bien des cas, pour les surfaces simples et fermées.

