

# le Rubik's cube ? un jeu d'enfant avec votre PC-1211 !

**Apprenez à votre ordinateur de poche PC-1211 à remettre en ordre le fameux Rubik's cube. Voici un programme qui utilise toutes les ressources de votre petite machine favorite et qui vous permettra, de plus, d'améliorer vos performances !**

sance apparaîtra clairement à cette occasion.

Avant de commenter ce programme, rappelons quelques faits simples concernant le cube lui-même (que l'on se rassure cependant, une personne manipulant le cube pour la première fois pourrait fort bien utiliser le présent programme, sans avoir à en saisir tous les détails).

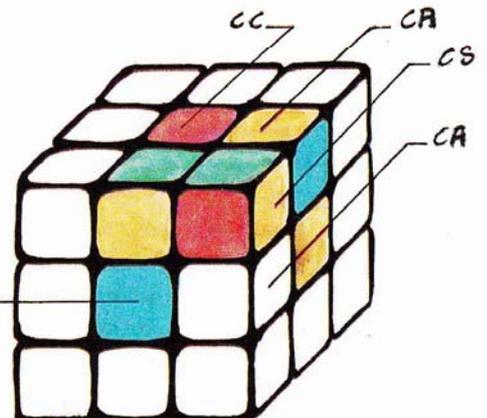
Ce cube est théoriquement formé de vingt-sept petits cubes : six n'ont qu'une facette visible (ce sont les cubes centraux ou CC), douze ont deux facettes visibles (ce sont les cubes arêtes ou CA), huit ont trois facettes visibles (ce sont les cubes sommets ou CS) et le vingt-septième (à l'intérieur) n'existe pas. Les CC ne quittent jamais leur place, et servent ainsi de repère : la « face rouge » du cube sera celle dont la facette centrale est rouge.



Si vous êtes un amateur du Rubik's cube et si vous possédez une Sharp PC-1211 avec interface cassette et imprimante, le programme présenté permettra à votre machine de remonter le cube, en partant d'une position quelconque.

Sur un ordinateur individuel, un tel programme ne présente pas de difficulté, mais pour votre ordinateur de poche c'est une performance méritoire. Hélas, n'espérez pas battre avec votre PC-1211 un record de vitesse : il lui faudra, en moyenne, une demi-heure pour remonter le cube — exploit qui lui assurerait la dernière place au championnat de Rubik's cube ! En tout cas, si vous épluchez ce programme, vous vous rendrez compte des possibilités de votre machine et vous approfondirez peut-être vos connaissances sur le cube...

Le programme en question nécessite 4 247 pas (en comptant pas de programme et mémoires de données). Cela dépasse évidemment la capacité du Sharp-TRS de poche, aussi ce programme est-il découpé en trois parties, qui sont enchaînées l'une à l'autre grâce à l'instruction CHAIN dont la puis-



Le programme des pages suivantes comprend en fait un fichier de données (baptisé CUBE) et trois programmes nommés respectivement CUBE 1, CUBE 2 et CUBE 3. Il nous faut d'abord écrire, puis enregistrer sur cassette ces quatre parties dans l'ordre : fichier CUBE, CUBE 1, CUBE 2 et CUBE 3. C'est là un travail long et fastidieux !

Pour vous aider, quelques conseils. Tout d'abord, si votre lecteur de cassette a un compteur, n'oubliez pas d'inscrire la place exacte de chaque partie. Pour vérifier qu'une partie a bien été écrite, le plus simple est d'en imprimer la liste et de la comparer à la liste ci-jointe. Enfin, une vérification très simple (dont l'efficacité vaut bien celle de la preuve par neuf !) consiste à voir si le programme que vous venez d'écrire occupe bien le nombre de pas requis ! A cet égard, les renseignements ci-après peuvent être utiles.

. Le fichier CUBE comprend les mémoires souples A(35) à A(112) ; ainsi A(35) = 12, etc.

. CUBE 1 nécessite 718 pas de programme ; si vous appuyez sur MEM Enter, il doit donc apparaître :

706 Steps 88 Memories.

### **Prenez le risque ! Mélangez soigneusement votre cube...**

Ce programme utilise 86 mémoires souples :

A(27) à A(112)

ce qui laissera 18 pas disponibles.

. CUBE 2 nécessite 733 pas de programme ; vous devrez lire :

691 Steps 86 Memories.

Ce programme a besoin des mêmes mémoires souples que CUBE 1, vous laissant 3 pas disponibles !

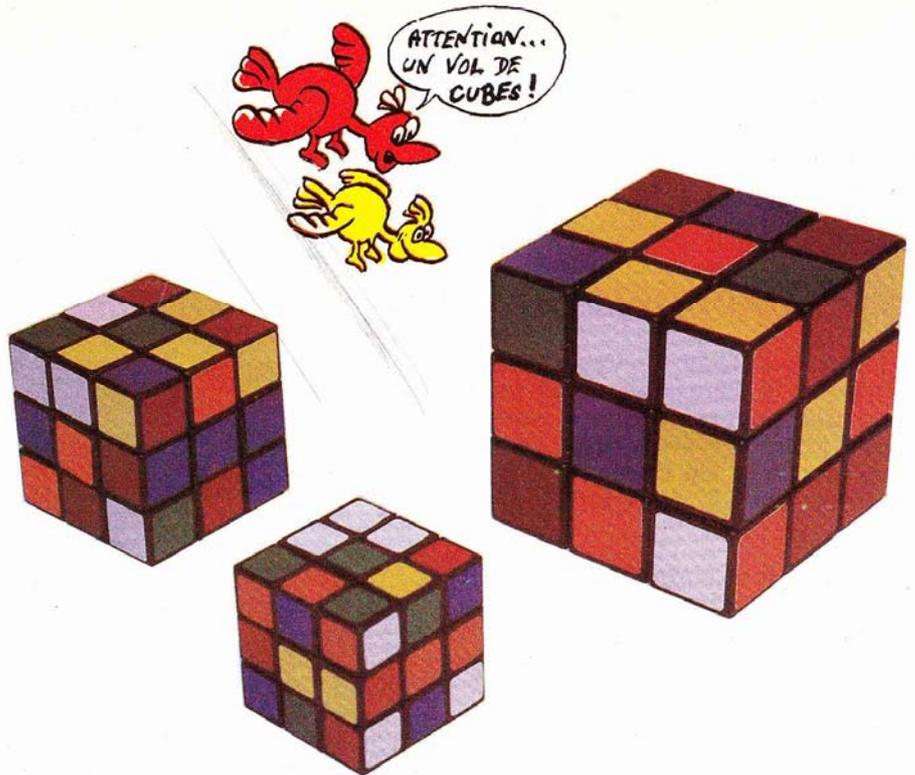
. CUBE 3 nécessite 1 140 pas de programme ; vous devez lire :

284 Steps 35 Memories.

Ce programme emploie 35 mémoires souples [ A(27) à A(61) ], vous laissant 4 pas disponibles.

. Le fichier CUBE ne sera utilisé que dans les programmes CUBE 1 et CUBE 2.

Vous-avez terminé l'enregistrement ? Vous voici prêt à commencer ! Placez votre cassette au début du fichier (faites NEW, c'est plus prudent), vérifiez vos branchements, en particulier l'imprimante doit être en mode



PRINT : n'oubliez pas d'appuyer deux fois sur ON, réglez le niveau d'écoute au maximum, enclenchez la touche PLAY, et en route ! Voici ce que vous avez à faire.

1) Transférez le fichier CUBE grâce aux instructions suivantes (en mode RUN) :

INPT # CUBE Enter.

2) En mode PRO écrivez le programme :

1 CHAIN CUBE 1,

puis, en mode RUN, faites RUN 1 Enter. Le programme CUBE 1 est alors transféré de la cassette à l'ordinateur. Ce transfert effectué, l'exécution commence (automatiquement).

Le PC-1211 imprime : ANNONCEZ LES COULEURS, puis FACE AVANT ? et s'arrête, attendant un renseignement.

3) Mélangez votre cube, puis posez-le devant vous, de façon arbitraire. Indiquez alors la couleur de la face avant. Ainsi, dans l'exemple de la figure 1, faites : BLEU Enter.

### **Votre ordinateur de poche enregistre les six couleurs de face**

L'OP imprime ensuite FACE DROITE ?, faites alors : JAUNE Enter, et opérez ainsi pour les six faces. Votre OP enregistrera ainsi les six couleurs de face (mémoires A\$,..., F\$) ; cette phase est nécessaire, car la répartition des couleurs des faces change d'un cube à l'autre.

4) Cela fait, votre machine vous demande de lui indiquer (succes-

sivement) les positions des douze CA. Ainsi lorsqu'elle imprimera JAUNE VERT (cf. figure page précédente) et affichera : ?, faites : BLEU Enter, puis ROUGE Enter.

Cela signifie que la facette jaune du CA « jaune vert » se trouve sur la face bleue du cube et que sa facette verte est sur la face rouge du cube. Noter que l'ordre est important : dans l'exemple de la figure de la page précédente, pour le CA JAUNE ROUGE, vous devrez faire :

ROUGE Enter puis JAUNE Enter.

### **Suspense, l'OP indique : « début de la solution » !**

5) L'étape 4 achevée, le PC-1211 vous demande de lui indiquer les positions des huit CS, suivant le même principe que précédemment ; lorsque l'on imprimera (par exemple) :

VERT JAUNE ROUGE, et affichera ?, faites :

ROUGE Enter (affichage : ?),

puis :

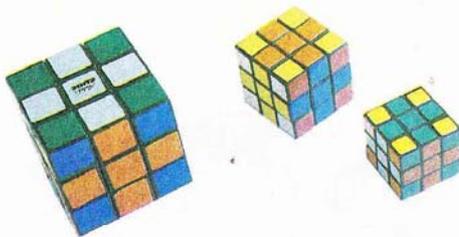
JAUNE Enter (affichage : ?), et

enfin :

BLEU Enter.

Opérez de même pour les huit CS.

6) Votre OP connaît maintenant l'état exact de votre cube, cela lui suffit. Il imprime alors : « début de la solution », il enchaîne avec CUBE 2, le transfère à l'ordinateur puis en démarre l'exécution. A la fin de CUBE 2, il enchaînera avec CUBE 3, le transfèrera et l'exécutera. Cela fait, après quelques BEEP, il s'arrêtera en vous exprimant (imprimant) sa satisfaction.



7) Après l'étape 5) vous n'avez plus rien à faire : absentez-vous une demi-heure. A votre retour, vous trouverez imprimées toutes les rotations de faces qui vous permettront de remonter le cube.

Ainsi :  
1 BLEU (ou simplement BLEU) vous dit de faire subir à la face bleue du cube un quart de tour, dans le sens des aiguilles d'une montre (à main droite) ;  
- 1 BLEU désigne le même quart de tour, mais en sens inverse.

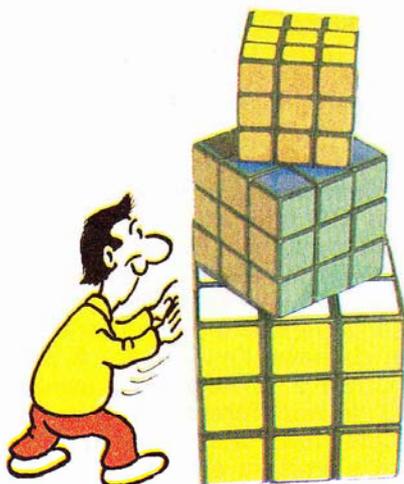
### Attention ! Une erreur de manipulation et l'on repart à zéro

Pour des raisons de place, il n'est pas possible d'analyser dans cet article ce programme en détail. Voici simplement quelques indications.

Le programme CUBE 1 a pour seul but d'indiquer à l'ordinateur la position de départ.

Le programme CUBE 2 s'occupe de remettre successivement à leurs places les douze CA (bien entendu l'ordinateur doit calculer aussi les déplacements des CS). Pour y arriver, on utilise une succession de sept rotations de faces que l'on appellera générateur.

Deux remarques importantes :  
- les générateurs sont séparés



#### FICHER CUBE

```

35.:12.
36.:13.
37.:14.
38.:15.
39.:23.
40.:24.
41.:26.
42.:35.
43.:36.
44.:45.
45.:46.
46.:56.
47.:363537013.
48.:353738132.
49.:383635201.
50.: -353941045.
51.: -403539104.
52.:394140451.
53.: -364243264.
54.: -433936402.
55.: -443740315.
56.: -404544573.
57.: -464238623.
58.:373836320.
59.:383635201.
60.:393642026.
61.:374045157.
62.:394140451.
63.: -364243264.
64.: -433936402.
65.:384446376.
66.:454143546.
67.: -464238623.
68.:373836320.
69.:394140451.
70.:374045157.
71.:414035510.
72.:423844237.
73.:424339640.
74.: -443740315.
75.: -404544573.
76.: -444642762.
77.: -433936402.
78.: -404544573.
79.:414346467.
80.:423844237.
81.:424339640.
82.:384446376.
83.:454437731.
84.: -444642762.
85.:414035510.
86.:394140451.
87.:424339640.
88.: -433936402.
89.:454437731.
90.:454143546.
91.:434645675.
92.:414035510.
93.:424339640.
94.:414346467.
95.:454437731.
96.: -404544573.
97.:464541754.
98.:424339640.
99.:414346467.
100.:454437731.

```

```

101.:454143546.
102.:464541754.
103.:424339640.
104.: -444642762.
105.:464541754.
106.: -464238623.
107.: -444642762.
108.:464541754.
109.:434645675.
110.:454437731.
111.: -444642762.
112.:464541754.

```

\*\*\*\*\*

#### CUBE 1

```

2:PRINT "ANNON
CEZ LES COUL
EURS:"
4:INPUT "FACE
AVANT?";A#
6:INPUT "FACE
DROITE?";B#
8:INPUT "FACE
DU HAUT?";C#
10:INPUT "FACE
DU BAS?";D#
12:INPUT "FACE
GAUCHE?";E#
14:INPUT "FACE
ARRIERE?";F#
16:FOR W=35 TO 4
6
18:G=INT (A(W)/
10),H=A(W)-1
OG:PRINT A#(
G),A#(H):
INPUT U#,T#
20:GOSUB "A":I=
Q:U#=T#:
GOSUB "A":V=
1:IF I>QLET
V=-1,P=I,I=Q
,Q=P
22:T=I*(9-I)/2+
Q+29,T=T+(I=
5)-(I<4)*(Q+
I>7),A(W)=VT
:NEXT W
26:FOR W=0 TO 7
28:G=INT (W/4),
H=INT (W/2-2
G),I=W-46-2H
,G=1+5G,H=2+
3H,I=3+I
30:PRINT A#(G);
" ";A#(H);"
";A#(I)
32:INPUT U#,T#,
S#:GOSUB "A"
:Q=Q:U#=T#:
GOSUB "A":P=
Q

```

```

34:U#=S#:GOSUB
"A":U=0
36:L=0:IF L>3
LET L=7-L,U=
2^(Q-4)
38:M=P:IF M>3
LET M=7-M,U=
U+2^(P-4)
40:N=Q:IF N>3
LET N=7-N,U=
U+2^(Q-4)
42:Z=SGN ((40-5
L)*(4P-5M)*(
4Q-5N)*(M-L)
*(N-M)*(N-L)
)
44:V=Z*((N=1)-(
M=1)),V=V-3*
INT (V/3)
46:A(2T+W)=U+V/
10:NEXT W:
PRINT "DEBUT
DE LA SOLUT
ION:"
48:CHAIN "CUBE
2"
50:"A":FOR X=1
TO 6
52:IF A#(X)=U#
LET Q=X
54:NEXT X:
RETURN

```

\*\*\*\*\*

#### CUBE 2

```

2:W=35
4:IF W=46GOTO
28
6:U=ABS (A(W))
:IF U=MLET M
=W+1:GOTO 4
7:GOSUB "D":
GOTO 6
8:"D":V=A(W*(9
1-W)/2+U-969
),G=ABS V#
GOSUB "A"
10:P=10P+Q,N=10
N+M,L=10L+K
12:T=35,G=P:
GOSUB "B":S=
Y,G=N:GOSUB
"B":R=Y,G=L:
GOSUB "B":Q=
Y
14:A(Q)=-A(Q),A
(S)=N*SGN (V
A(S)),A(R)=-
P*SGN (VA(R)
)

```

```

16: T=27, G=J:
   GOSUB "B": S=
   Y, G=1: GOSUB
   "B": R=Y, G=H:
   GOSUB "B": Q=
   Y
8: FOR Y=8 TO 10
20: T=A(A*Y+9):
   (10-Y)/10, V=
   INT T, Z=10*(
   T-V), Z=Z-3*
   INT (Z/3), A(
   A*Y+9)=V+Z/
   10: NEXT Y
22: A(S)=A(S)+H-
   J, A(Q)=A(Q)+
   J-H, T=7-H-J:
   GOSUB "C": Q=
   S, T=J-I:
   GOSUB "C": R=
   S, T=I-H
24: GOSUB "C":
   BEEP 2: PRINT
   A#(Q): PRINT
   -1; A#(R):
   PRINT -1; A#(
   Q): PRINT -1;
   A#(S)
26: PRINT A#(Q):
   PRINT A#(S):
   PRINT A#(R):
   PRINT "
   *****":
   RETURN
28: W=35
30: IF W=46 GOTO
   38
32: U=W+1: IF A(W
   )LET W=W+1:
   GOTO 30
34: IF A(U)LET U
   =U+1: GOTO 34
36: GOSUB "D":
   GOSUB "D":
   GOTO 32
38: FOR Y=35 TO 1
   12
40: A(Y)=0: NEXT
   Y: CHAIN "CUB
   E 3"
42: "A": FOR X=8
   TO 16
44: A(X)=G-10*
   INT (G/10), G
   =INT (G/10):
   NEXT X:
   RETURN
48: "B": Y=T
50: IF INT (ABS
   (A(Y)))=G
   GOTO 54
52: Y=Y+1: GOTO 5
   0
54: RETURN

```

```

56: "C": S=3.5-
   SGN (T)*(0.5
   +LOG (ABS (T
   ))/LOG 2):
   RETURN
   CUBE 3
2: G=142011, H=3
   60262, I=1232
   01, J=240450,
   K=235100, L=3
   16402, M=6446
   71, N=-122312
4: O=131321, P=2
   14512, Q=4631
   52, R=546230,
   S=427310, T=R
   , U=-314621, V
   =415732
6: W=322640, X=5
   17622, Y=N:
   FOR Z=7 TO 25
8: A(Z+28)=A(Z)
   : NEXT Z: G=A(
   50), H=-51673
   1, I=533760, J
   =-646452, K=A
   (41), L=63754
   1
10: M=626751, N=-
   635762: FOR Z
   =7 TO 14
12: A(Z+47)=A(Z)
   : NEXT Z: FOR
   W=27 TO 32
16: V=INT (A(W))
   : IF V=W-27
   GOTO 32
18: G=A(W*(67-W)
   /2+V-506), V=
   SGN G, G=VG:
   GOSUB "A"
20: G=7: GOSUB "B
   ": P=Y, G=J:
   GOSUB "B": Q=
   Y, G=K: GOSUB
   "B": R=Y
22: A(P)=A(P)+J-
   I+H/10, A(Q)=
   A(Q)+K-J+(3-
   H-V)/10, A(R)
   =A(R)+I-K+(3
   +V)/10
24: FOR Y=16 TO 1
   8
26: S=A(A*Y), T=
   INT S, U=10*(
   S-T), U=U-3*
   INT (U/3), A(
   A*Y)=T+U/10
   : NEXT Y
28: Y=7-L: BEEP 2
   : PRINT V; A#(
   M): PRINT V; A
   #(L): PRINT -
   V; A#(N):
   PRINT -V; A#(
   Y)

```

```

30: PRINT V; A#(M
   ): PRINT -V; A
   #(L): PRINT -
   V; A#(N):
   PRINT V; A#(Y
   ): PRINT "
   *****":
   GOTO 16
32: NEXT W
34: FOR W=27 TO 3
   4
36: T=A(W), A(W)=
   10*(T-INT T)
   : NEXT W
38: M=2631, N=462
   3, O=1233, P=4
   217, Q=6535, R
   =6327, S=5137
40: FOR W=27 TO 3
   3
42: V=A(W), A(W)=
   Q: IF V=0 GOTO
   54
44: G=A(W-14):
   GOSUB "A": G=
   V+A(H+27)+A(
   H+27)=G-3*(G
   /2)
46: IF V=2 LET U=
   K, K=7-J, J=7-
   U, I=7-I
48: BEEP 2: PRINT
   A#(K): PRINT
   -1; A#(J):
   PRINT -1; A#(
   K): PRINT -1;
   A#(I): PRINT
   -1; A#(J)
50: Y=7-J: PRINT
   A#(I): PRINT
   -1; A#(Y):
   PRINT -1; A#(
   I): PRINT A#(
   J): PRINT A#(
   I)
52: PRINT A#(K):
   PRINT A#(J):
   PRINT -1; A#(
   K): PRINT A#(
   Y): PRINT "
   *****"
54: NEXT W: BEEP
   8: PRINT "JE
   SUIS GENIAL,
   NON?": END
56: "B": Y=27
58: IF INT (A*Y)
   )=GGOTO 62
60: Y=Y+1: GOTO 5
   8
62: RETURN
64: "A": X=8
66: A(X)=G-10*
   INT (G/10), G
   =INT (G/10):
   IF GLET X=X+
   1: GOTO 66
68: RETURN

```

les uns des autres dans l'impression de la solution : cela en rend la lecture plus commode ;  
- à la fin de CUBE 2, les CA doivent être rangés, c'est-à-dire que chaque face doit présenter une « croix » unicolore.

. Le programme CUBE 3 ne s'occupe plus que des CS, qu'il doit successivement placer (un à un) puis ranger. Ce programme est plus simple et plus rapide que CUBE 2.

. A chaque instant, les positions des CS sont stockées dans les mémoires A (27) à A(34), celles des CA dans les mémoires A(35) à A(46) (sauf dans CUBE 3).

**Miracle ! Mais oui, c'est vrai, le cube est remonté !**

. Le fichier CUBE sert essentiellement dans CUBE 2 : à un moment donné, il permet à l'ordinateur de choisir le générateur à effectuer et fournit toutes les indications utiles concernant ce générateur.

. Une erreur de manipulation ou une erreur dans la description de l'état du cube vous obligera à repartir de zéro ! En revanche, si le transfert de l'un des programmes (par exemple CUBE 3) n'est pas effectué correctement - cela arrive ! -, ce n'est pas grave ; recommencez au début de ce programme en écrivant (mode PRO) :  
1 CHAIN "CUBE 3" Enter,  
puis en faisant (mode RUN) :  
RUN 1 Enter.

Et maintenant, bon courage !

*Emmanuel Halberstadt*

