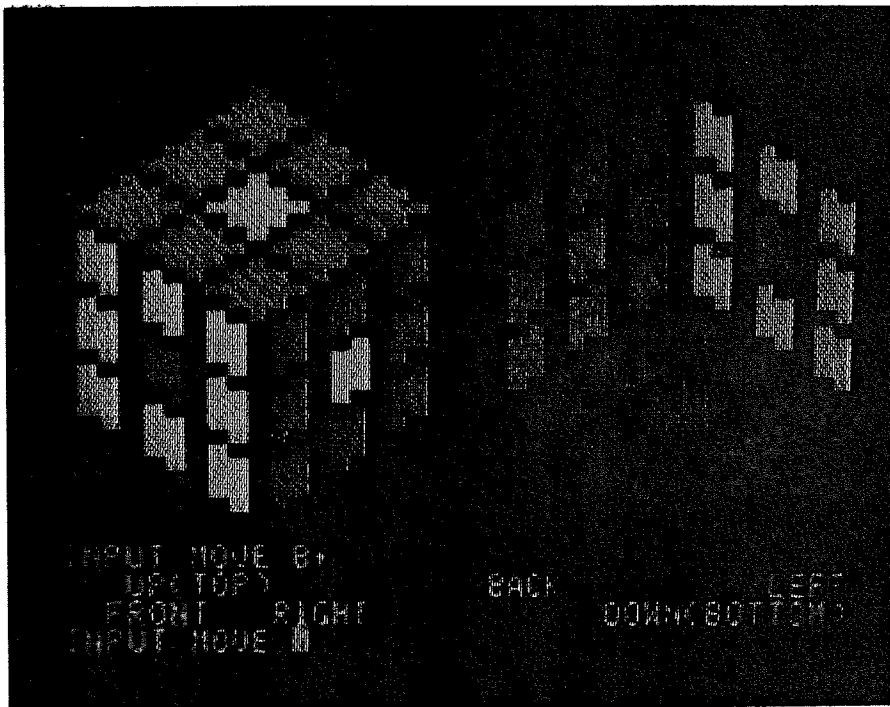


# Rubik's Cube Demystified

By Curtis and Lillian J. Cooper



Screen display of the Applesoft Rubik's Cube program. (Photo by Harold Nelson)

This program, in Applesoft BASIC using low-resolution graphics, simulates Rubik's Cube. The problem is to take any arrangement of the cube and restore it to its pristine state.

In solving Rubik's Cube, each face on the cube can turn clockwise or counterclockwise. In addition, different views of the cube are obtained by rotating the cube about axes through the center squares of the top and bottom faces, right and left faces, and front and back faces.

## Program Notes

The program uses an array R, dimensioned to 5 by 5 by 5, as its representation for Rubik's Cube. Colors are stored numerically as follows:

Magenta 1  
Orange 9  
Blue 2  
Green 4  
White 15  
Yellow 13.

The F face (see photo for face identification) is stored in the middle 3 by 3 squares where  $x=1$ . The R face is stored in the middle 3 by 3 squares where  $y=1$ . The D face is stored in the middle 3 by 3 squares where  $z=1$ . Similarly, the B face is stored in the middle 3 by 3 squares where  $x=5$ ; the L face is stored in the middle 3 by 3 squares where  $y=5$ ; and the U face is stored in the middle 3 by 3 squares where  $z=5$ .

## Program listing. Rubik's Cube simulation in Applesoft BASIC.

```
10 PRINT "THIS PROGRAM SIMULATES RUBIK'S CUBE"
20 REM
30 REM INITIALIZE RUBIK'S CUBE
40 REM
50 DIM R(5,5,5),R1(5,5,5),A(5,5),B(5,5),C(5,5),D(5,5)
60 FOR I = 1 TO 5: FOR J = 1 TO 5: FOR K = 1 TO 5
70 R(I,J,K) = 0
80 NEXT K: NEXT J: NEXT I
90 FOR I = 2 TO 4: FOR J = 2 TO 4
100 R(1,I,J) = 1:R(1,1,J) = 15:R(I,J,1) = 4
110 R(5,I,J) = 9:R(I,5,J) = 2:R(I,J,5) = 13
120 NEXT J: NEXT I
130 REM
140 REM MIX RUBIK'S CUBE
150 REM
160 INPUT "INPUT NUMBER OF MIXES";N
170 Z$ = ""
180 FOR I = 1 TO N
190 X = INT (6 * RND (1)):Y = INT (3 * RND (1))
200 IF X < > 0 THEN 220
210 X$ = "R": GOTO 310
220 IF X < > 1 THEN 240
230 X$ = "F": GOTO 310
```

More →

Address correspondence to Curtis and Lillian J. Cooper, 803 E. Clark, Warrensburg, MO 64093.

The program contains two big sub-routines. One subroutine is used to rearrange the cube. The following notation, similar to that in James F. Nourse's *The Simple Solution to Rubik's Cube*, is used by the program to change the cube.

### Summary of Moves

- R+—Turn R face one quarter turn clockwise
- R—Turn R face one quarter turn counterclockwise
- R2—Turn R face one half turn
- F+—Turn F face one quarter turn clockwise
- F—Turn F face one quarter turn counterclockwise
- F2—Turn F face one half turn
- L+—Turn L face one quarter turn clockwise
- L—Turn L face one quarter turn counterclockwise
- L2—Turn L face one half turn
- D+—Turn D face one quarter turn clockwise
- D—Turn D face one quarter turn counterclockwise
- D2—Turn D face one half turn
- U+—Turn U face one quarter turn clockwise
- U—Turn U face one quarter turn counterclockwise
- U2—Turn U face one half turn
- B+—Turn B face one quarter turn clockwise
- B—Turn B face one quarter turn counterclockwise
- B2—Turn B face one half turn.
- MFR# where # is 1, 2, or 3—Rotate the cube about the axis passing through the center squares of the up (top) and down (bottom) faces. Move F to R face 1, 2, or 3 times.
- MFU# where # is 1, 2, or 3—Rotate the cube about the axis passing through the center squares of the right and left faces. Move F to U face 1, 2, or 3 times.
- MUR# where # is 1, 2, or 3—Rotate the cube about the axis passing through the center squares of the front and back faces. Move U to R face 1, 2, or 3 times.

Several moves can be performed on the cube by concatenating together any of the above moves.

The second subroutine draws the cube. Two views are displayed on the screen. The first view shows the corner formed by the up, right and front faces as the corner closest to the viewer. The second view has the opposite corner (formed by the down, left and back faces) closest to the viewer. The faces are each labeled (see photo).

The program begins by initializing the cube and asking how many moves you want it to make to mix up the cube. It is then randomly mixed the number of times specified and the resulting cube is displayed. The program asks you to input your move or moves. Invalid move entries are rejected and you are asked to reenter your move. If S is input, the program stops. Otherwise the resulting cube is displayed and another move is re-

quested.

Enjoy exploring this color-graphics version of Rubik's Cube. ■

### References

- Nourse, James G. *The Simple Solution to Rubik's Cube* (New York: Bantam Books, 1981).
- Singmaster, David. *Notes on Rubik's 'Magic Cube'* (Hillside, New Jersey: Enslow Publishers, 1980).

Listing continued.

```

240 IF X < > 2 THEN 260
250 X$ = "L": GOTO 310
260 IF X < > 3 THEN 280
270 X$ = "U": GOTO 310
280 IF X < > 4 THEN 300
290 X$ = "B": GOTO 310
300 X$ = "D"
310 IF Y < > 0 THEN 330
320 Y$ = "+": GOTO 360
330 IF Y < > 1 THEN 350
340 Y$ = "-": GOTO 360
350 Y$ = "2"
360 Z$ = Z$ + X$ + Y$
370 NEXT I
380 GOSUB 500
390 GOSUB 2050
400 REM
410 REM CHANGE RUBIK'S CUBE
420 REM
430 INPUT "INPUT MOVE "; Z$
440 GOSUB 500
450 GOSUB 2050
460 GOTO 430
470 REM
480 REM PERFORM MOVES
490 REM
500 FOR I = 1 TO 5: FOR J = 1 TO 5: FOR K = 1 TO 5
510 R1(I,J,K) = R(I,J,K)
520 NEXT K: NEXT J: NEXT I
530 IF Z$ = "" THEN RETURN
540 X$ = MID$(Z$,1,1)
550 IF X$ = "S" THEN 3100
560 IF X$ = "M" THEN 1250
570 REM
580 REM MOVE FACES
590 REM
600 Y$ = MID$(Z$,2,1)
610 IF X$ < > "R" THEN 710
620 FOR I = 1 TO 5: FOR J = 1 TO 5
630 A(I,J) = R(I,1,J):B(I,J) = R(I,2,J)
640 NEXT J: NEXT I
650 GOSUB 1720
660 IF E = 1 THEN 1670
670 FOR I = 1 TO 5: FOR J = 1 TO 5
680 R(I,1,J) = A(I,J):R(I,2,J) = B(I,J)
690 NEXT J: NEXT I
700 GOTO 1200
710 IF X$ < > "F" THEN 810
720 FOR I = 1 TO 5: FOR J = 1 TO 5
730 A(I,J) = R(1,1,J):B(I,J) = R(2,1,J)
740 NEXT J: NEXT I
750 GOSUB 1800
760 IF E = 1 THEN 1670
770 FOR I = 1 TO 5: FOR J = 1 TO 5
780 R(1,1,J) = A(I,J):R(2,1,J) = B(I,J)
790 NEXT J: NEXT I
800 GOTO 1200
810 IF X$ < > "L" THEN 910
820 FOR I = 1 TO 5: FOR J = 1 TO 5
830 A(I,J) = R(I,5,J):B(I,J) = R(I,4,J)
840 NEXT J: NEXT I
850 GOSUB 1800
860 IF E = 1 THEN 1670
870 FOR I = 1 TO 5: FOR J = 1 TO 5
880 R(I,5,J) = A(I,J):R(I,4,J) = B(I,J)
890 NEXT J: NEXT I
900 GOTO 1200
910 IF X$ < > "B" THEN 1010
920 FOR I = 1 TO 5: FOR J = 1 TO 5
930 A(I,J) = R(5,1,J):B(I,J) = R(4,1,J)
940 NEXT J: NEXT I
950 GOSUB 1720
960 IF E = 1 THEN 1670
970 FOR I = 1 TO 5: FOR J = 1 TO 5
980 R(5,1,J) = A(I,J):R(4,1,J) = B(I,J)
990 NEXT J: NEXT I
1000 GOTO 1200
1010 IF X$ < > "U" THEN 1110
1020 FOR I = 1 TO 5: FOR J = 1 TO 5
1030 A(I,J) = R(I,J,5):B(I,J) = R(I,J,4)
1040 NEXT J: NEXT I
1050 GOSUB 1720
1060 IF E = 1 THEN 1670
1070 FOR I = 1 TO 5: FOR J = 1 TO 5

```

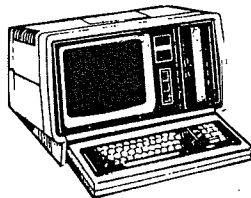
More



\$ SAVE \$

# Software at SUPER DISCOUNT PRICES

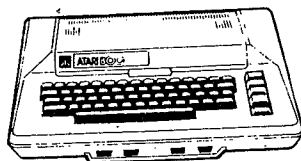
the best available on the  
market today for



## TRS-80



apple computer



## ATARI

GAMES \* UTILITIES \* BUSINESS  
for the budget minded individual

ONE OF THE WORLD'S  
LARGEST INVENTORIES

We have all the latest  
software—ASK US!

A-BARGAIN Distributors  
COMPUTER GAMES SPECIALISTS

P.O. BOX 24121 OAKLAND, CA 94607

Listing continued.

```

1080 R(I,J,5) = A(I,J):R(I,J,4) = B(I,J)
1090 NEXT J: NEXT I
1100 GOTO 1200
1110 IF X$ < > "D" THEN 1670
1120 FOR I = 1 TO 5: FOR J = 1 TO 5
1130 A(I,J) = R(I,J,1):B(I,J) = R(I,J,2)
1140 NEXT J: NEXT I
1150 GOSUB 1800
1160 IF E = 1 THEN 1670
1170 FOR I = 1 TO 5: FOR J = 1 TO 5
1180 R(I,J,1) = A(I,J):R(I,J,2) = B(I,J)
1190 NEXT J: NEXT I
1200 Z$ = MID$(Z$,3)
1210 GOTO 530
1220 REM
1230 REM CHANGE VIEWING CORNERS
1240 REM
1250 X$ = MID$(Z$,2,1):Y$ = MID$(Z$,3,1)
1260 S$ = MID$(Z$,4,1)
1270 IF S$ < > "1" AND S$ < > "2" AND S$ < > "3" THEN 1670
1280 N = VAL(S$)
1290 IF X$ < > "F" THEN 1540
1300 IF Y$ < > "R" THEN 1420
1310 Z$ = MID$(Z$,5)
1320 FOR K = 1 TO N
1330 Z$ = "U-D+" + Z$
1340 FOR I = 1 TO 5: FOR J = 1 TO 5
1350 A(I,J) = R(I,J,3)
1360 NEXT J: NEXT I
1370 FOR I = 1 TO 5: FOR J = 1 TO 5
1380 R(6 - J, I, 3) = A(I,J)
1390 NEXT J: NEXT I
1400 NEXT K
1410 GOTO 530
1420 IF Y$ < > "U" THEN 1670
1430 Z$ = MID$(Z$,5)
1440 FOR K = 1 TO N
1450 Z$ = "R+L-" + Z$
1460 FOR I = 1 TO 5: FOR J = 1 TO 5
1470 A(I,J) = R(I,3,J)
1480 NEXT J: NEXT I
1490 FOR I = 1 TO 5: FOR J = 1 TO 5
1500 R(J,3,6 - I) = A(I,J)
1510 NEXT J: NEXT I
1520 NEXT K
1530 GOTO 530
1540 IF X$ < > "U" THEN 1670
1550 IF Y$ < > "R" THEN 1670
1560 Z$ = MID$(Z$,5)
1570 FOR K = 1 TO N
1580 Z$ = "F+B-" + Z$
1590 FOR I = 1 TO 5: FOR J = 1 TO 5
1600 A(I,J) = R(3,I,J)
1610 NEXT J: NEXT I
1620 FOR I = 1 TO 5: FOR J = 1 TO 5
1630 R(3,6 - J, I) = A(I,J)
1640 NEXT J: NEXT I
1650 NEXT K
1660 GOTO 530
1670 PRINT "INVALID MOVE, TRY AGAIN."
1680 FOR I = 1 TO 5: FOR J = 1 TO 5: FOR K = 1 TO 5
1690 R(I,J,K) = R1(I,J,K)
1700 NEXT K: NEXT J: NEXT I
1710 RETURN
1720 E = 0
1730 IF Y$ < > "+" THEN 1750
1740 GOSUB 1950: RETURN
1750 IF Y$ < > "-" THEN 1770
1760 GOSUB 1880: RETURN
1770 IF Y$ < > "2" THEN 1790
1780 GOSUB 1880: GOSUB 1880: RETURN
1790 E = 1: RETURN
1800 E = 0
1810 IF Y$ < > "+" THEN 1830
1820 GOSUB 1880: RETURN
1830 IF Y$ < > "-" THEN 1850
1840 GOSUB 1950: RETURN
1850 IF Y$ < > "2" THEN 1870
1860 GOSUB 1880: GOSUB 1880: RETURN
1870 E = 1: RETURN
1880 FOR I = 1 TO 5: FOR J = 1 TO 5
1890 C(I,J) = A(I,J):D(I,J) = B(I,J)
1900 NEXT J: NEXT I
1910 FOR I = 1 TO 5: FOR J = 1 TO 5
1920 A(6 - J, I) = C(I,J):B(6 - J, I) = D(I,J)
1930 NEXT J: NEXT I
1940 RETURN
1950 FOR I = 1 TO 5: FOR J = 1 TO 5
1960 C(I,J) = A(I,J):D(I,J) = B(I,J)
1970 NEXT J: NEXT I
1980 FOR I = 1 TO 5: FOR J = 1 TO 5
1990 A(J,6 - I) = C(I,J):B(J,6 - I) = D(I,J)
2000 NEXT J: NEXT I
2010 RETURN
2020 REM
2030 REM PRINT RUBIK'S CUBE
2040 REM
2050 TEXT
2060 GR
2070 COLOR= 0
2080 X1 = 9:Y1 = 39:X2 = 6:Y2 = 36:X3 = 3:Y3 = 33:X4 = 0:Y4 = 30:W2 = 12:W3 = 15
      :W4 = 18
2090 FOR I = 0 TO 15
2100 PLOT X1,Y1 - I: PLOT X2,Y2 - I: PLOT X3,Y3 - I: PLOT X4,Y4 - I: PLOT W2,Y2
      - I: PLOT W3,Y3 - I: PLOT W4,Y4 - I
2110 NEXT I

```

More

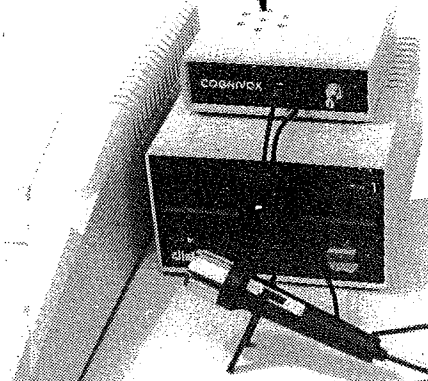
```

= 9:Z1 = 39:Z2 = 34:Z3 = 29:Z4 = 24:Y2 = 21:Y3 = 18:Y4 = 15
OR I = 0 TO 9
LOT X1 + I,Z1 - I: PLOT X1 + I,Z2 - I: PLOT X1 + I,Z3 - I: PLOT X1 + I,Z4
LOT X1 - I,Z1 - I: PLOT X1 - I,Z2 - I: PLOT X1 - I,Z3 - I: PLOT X1 - I,Z4
LOT X2 + I,Y2 - I: PLOT X3 + I,Y3 - I: PLOT X4 + I,Y4 - I
LOT W2 - I,Y2 - I: PLOT W3 - I,Y3 - I: PLOT W4 - I,Y4 - I
NEXT I
= 30:X2 = 27:X3 = 24:X4 = 21:Y1 = 21:Y2 = 24:Y3 = 27:Y4 = 30:W2 = 33:W3
= 39
OR I = 0 TO 15
LOT X1,Y1 - I: PLOT X2,Y2 - I: PLOT X3,Y3 - I: PLOT X4,Y4 - I: PLOT W2,Y2
LOT W3,Y3 - I: PLOT W4,Y4 - I
NEXT I
= 30:Z1 = 6:Z2 = 11:Z3 = 16:Z4 = 21
OR I = 0 TO 9
LOT X1 - I,Z1 + I: PLOT X1 - I,Z2 + I: PLOT X1 - I,Z3 + I: PLOT X1 - I,Z4
LOT X1 + I,Z1 + I: PLOT X1 + I,Z2 + I: PLOT X1 + I,Z3 + I: PLOT X1 + I,Z4
LOT X2 + I,Y2 + I: PLOT X3 + I,Y3 + I: PLOT X4 + I,Y4 + I
LOT W2 - I,Y2 + I: PLOT W3 - I,Y3 + I: PLOT W4 - I,Y4 + I
NEXT I
PRINT " UP(TOP)          BACK          LEFT"
PRINT " FRONT RIGHT     DOWN(BOTTOM)"
I = 8:X2 = 7:Y1 = 37:Y2 = 36
FOR J = 2 TO 4: FOR K = 2 TO 4
= 1
GOSUB 2990
FOR I = 0 TO 3
PLOT X1 - 3 * (J - 2),Y1 - I - 5 * (K - 2) - 3 * (J - 2)
PLOT X2 - 3 * (J - 2),Y2 - I - 5 * (K - 2) - 3 * (J - 2)
NEXT I
NEXT K: NEXT J
I = 10:X2 = 11:Y1 = 37:Y2 = 36
FOR I = 2 TO 4: FOR K = 2 TO 4
= 1
GOSUB 2990
FOR J = 0 TO 3
PLOT X1 + 3 * (I - 2),Y1 - J - 5 * (K - 2) - 3 * (I - 2)
PLOT X2 + 3 * (I - 2),Y2 - J - 5 * (K - 2) - 3 * (I - 2)
NEXT J
NEXT K: NEXT I
I = 32:X2 = 31:Y1 = 22:Y2 = 21
FOR J = 4 TO 2 STEP - 1: FOR K = 2 TO 4
I = 5
GOSUB 2990
FOR J = 0 TO 3
PLOT X1 - 3 * (4 - I),Y1 - J - 5 * (K - 2) + 3 * (4 - I)
PLOT X2 - 3 * (4 - I),Y2 - J - 5 * (K - 2) + 3 * (4 - I)
NEXT J
NEXT K: NEXT I
I = 28:X2 = 29:Y1 = 22:Y2 = 21
FOR J = 4 TO 2 STEP - 1: FOR K = 2 TO 4
I = 5
GOSUB 2990
FOR I = 0 TO 3
PLOT X1 - 3 * (4 - J),Y1 - I - 5 * (K - 2) + 3 * (4 - J)
PLOT X2 - 3 * (4 - J),Y2 - I - 5 * (K - 2) + 3 * (4 - J)
NEXT I
NEXT K: NEXT J
X1 = 7:X2 = 8:X3 = 9:X4 = 10:X5 = 11
Y1 = 21:Y2 = 22:Y3 = 23:Y4 = 22:Y5 = 21
FOR I = 2 TO 4: FOR J = 2 TO 4
K = 5
GOSUB 2990
PLOT X1 + 3 * (I - 2) - 3 * (J - 2),Y1 - 3 * (J - 2) - 3 * (I - 2)
PLOT X5 + 3 * (I - 2) - 3 * (J - 2),Y5 - 3 * (J - 2) - 3 * (I - 2)
FOR K = 0 TO 2
PLOT X2 + 3 * (I - 2) - 3 * (J - 2),Y2 - K - 3 * (J - 2) - 3 * (I - 2)
PLOT X4 + 3 * (I - 2) - 3 * (J - 2),Y4 - K - 3 * (J - 2) - 3 * (I - 2)
NEXT K
FOR K = 0 TO 4
PLOT X3 + 3 * (I - 2) - 3 * (J - 2),Y3 - K - 3 * (J - 2) - 3 * (I - 2)
NEXT K
NEXT J: NEXT I
X1 = 28:X2 = 29:X3 = 30:X4 = 31:X5 = 32
Y1 = 24:Y2 = 23:Y3 = 22:Y4 = 23:Y5 = 24
FOR I = 4 TO 2 STEP - 1: FOR J = 4 TO 2 STEP - 1
K = 1
GOSUB 2990
PLOT X1 + 3 * (4 - I) - 3 * (4 - J),Y1 + 3 * (4 - J) + 3 * (4 - I)
PLOT X5 + 3 * (4 - I) - 3 * (4 - J),Y5 + 3 * (4 - J) + 3 * (4 - I)
FOR K = 0 TO 2
PLOT X2 + 3 * (4 - I) - 3 * (4 - J),Y2 + K + 3 * (4 - J) + 3 * (4 - I)
PLOT X4 + 3 * (4 - I) - 3 * (4 - J),Y4 + K + 3 * (4 - J) + 3 * (4 - I)
NEXT K
FOR K = 0 TO 4
PLOT X3 + 3 * (4 - I) - 3 * (4 - J),Y3 + K + 3 * (4 - J) + 3 * (4 - I)
NEXT K
NEXT J: NEXT I
RETURN
IF R(I,J,K) < > 1 THEN 3010
COLOR= 1: RETURN
IF R(I,J,K) < > 15 THEN 3030
COLOR= 15: RETURN
IF R(I,J,K) < > 4 THEN 3050
COLOR= 4: RETURN
IF R(I,J,K) < > 9 THEN 3070
COLOR= 9: RETURN
IF R(I,J,K) < > 2 THEN 3090
COLOR= 2: RETURN
COLOR= 13: RETURN
PRINT "THANK YOU FOR PLAYING."
END

```

Hello.  
This is the APPLE  
talking. The message  
is: Don't byte your  
APPLE. Use COGNIVOX  
to speak to it!

I am now listening  
for your reply . . .



Let's face it. Voice I/O is a fascinating and efficient way to communicate with computers. And now, thanks to VOICETEK, Voice I/O peripherals are easily available, easy to use and very affordable.

If you own an APPLE II computer, COGNIVOX model VI0-1003 will enable your computer to understand your spoken commands and talk back with clear, natural sounding voice.

COGNIVOX can be trained to recognize up to 32 words or short phrases chosen by the user. To train COGNIVOX to recognize a new word, you simply repeat the word three times under the prompting of the system.

COGNIVOX will also talk with a vocabulary of 32 words or phrases chosen by the user. This vocabulary is independent of the recognition vocabulary, so a dialog with the computer is possible. The speech output is natural sounding since it is a digital recording of the user voice using a data compression algorithm.

For applications requiring more than 32 words, you can have two or more vocabularies of 32 words and switch back and forth between them. Vocabularies can also be stored on disk.

COGNIVOX VI0-1003 comes complete with microphone, power supply, software on cassette and extensive manual, ready to plug in and use. It plugs into the paddle connector and thus it leaves the valuable expansion slots free for other peripherals.

Software provided with the unit includes demonstration programs and two voice operated, talking video games! It is also very easy to incorporate voice in your own programs. A single statement from BASIC is all that is needed to either recognize or say a word.

COGNIVOX can be used as an educational tool, a data entry device when hands and/or eyes are busy, an aid to the handicapped, a foreign language translator, a sound effects generator, an intelligent telephone answering machine, a talking calculator. Using an IEEE 488 interface card you can control by voice instruments, plotters, test systems. And all these devices can talk back to you, telling you their readings, alarm conditions, even their name.

COGNIVOX VI0-1003 costs \$249 plus \$5 shipping (CA res. add 6% tax). Software on diskette (DOS 3.3) with extra features to save vocabularies on disk, \$19. Order by mail or call us at (805) 685-1854, 9AM to 5PM PST, M-F and charge it on your MASTERCARD or VISA. Foreign orders welcome, add 10% for air mail shipping and handling. COGNIVOX is backed by a 120 day limited warranty against manufacturing defects.

**VOICETEK**  
Dept. K, Box 388  
Goleta, CA 93116