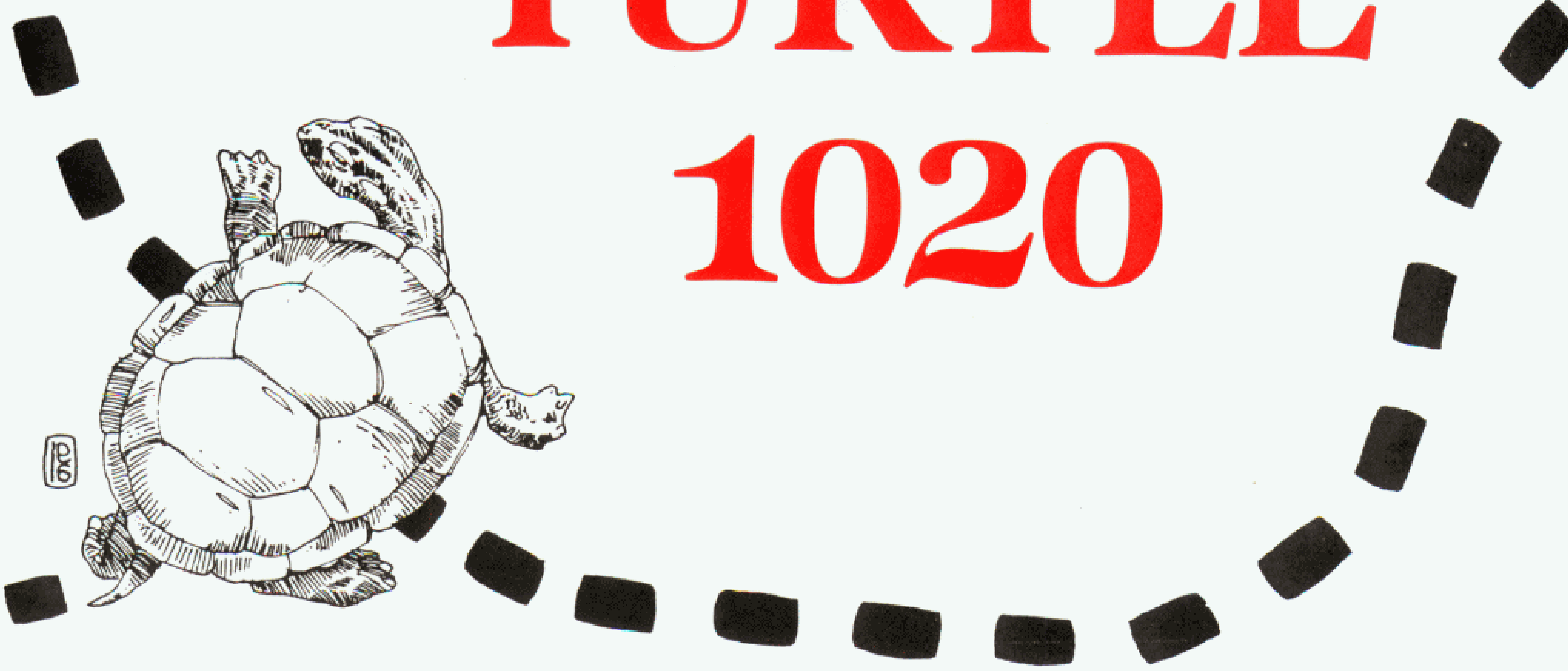
24K Cassette or
32K Disk

TURTLE 1020



by Jason Leigh

The Atari 1020 printer/plotter is an excellent Atari product as shown by Tom Hudson's **Solid States** in **ANALOG Computing**, issue 16.

To an aspiring young programmer, the plotter is a marvelous drawing tool. However, unless one knows a little trigonometry, it can be difficult to produce those intriguing circular patterns created by such languages as Logo and Pilot.

The turtle interpreter.

Turtle 1020 is an interpreted language written in Atari BASIC. The interpreter is a mixture of Pilot, Logo and BASIC commands—TURN from Pilot, FORWARD from Logo and variable use as in BASIC.

The editor.

Turtle 1020 uses the standard Atari editor, in that you can edit a line by cursoring up and over to the desired line. All the usual INSERT and DELETE functions still work, except each turtle line is limited to 40 characters in length. The editor assumes that any command without a line number is an immediate command, and that those preceded by a line number from 1 to 199 are program lines.

The commands.

All commands are three characters long, with the exception of LOAD and SAVE.

LISxx — Lists lines xx to the end of the program (xx is optional).

RUN — Executes the turtle program in memory. This will blank off the screen to speed up the drawing.

SAVE — Saves the program in memory to cassette or disk. You will be prompted for a filename, of which C selects cassette and *D:filename* selects disk.

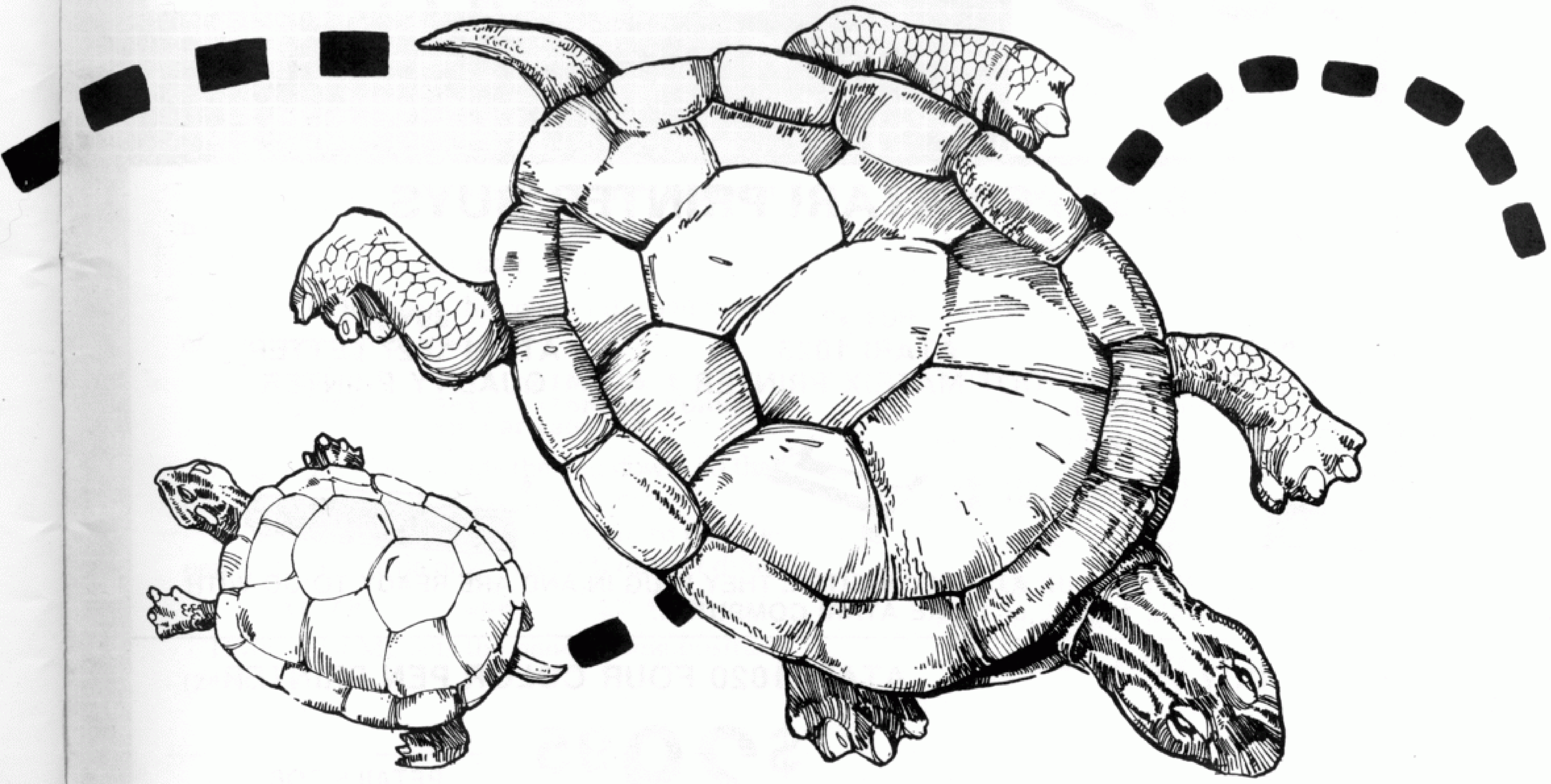
LOAD — Loads the program from cassette or disk. LOAD only loads **Turtle 1020** files and no other. Each turtle program is headed by 2 header bytes of 1s as an identifier. LOAD is operated in the same manner as SAVE.

NEW — Clears all variables and erases the turtle source code.

LIP — Lists the entire turtle program to the printer/plotter.

TON — Stands for Trace ON, which allows you to see each line as it is processed.

TOF — Turns the Trace function OFF.



MEM — Displays the number of 40-character program lines free. The command takes some time to compute, so please be patient.

SEE — Displays the X and Y position of the pen/turtle, as well as the color and angle at which the turtle is facing.

CAT — This was included for the benefit of disk users. It will allow you to see the disk directory without going to the Disk Utility Package. Variables for **Turtle 1020** are limited to single-character names from *A* to *Z*. The contents of the variables can be displayed by pressing the appropriate key and pressing RETURN.

DEL — Deletes a range of line numbers. After typing *DEL* and pressing RETURN, you will be prompted to enter certain parameters, such as starting and ending line to delete.

Be aware that the above commands can only be used in immediate mode. This may make **Turtle 1020** seem rather limited, however it was written primarily to allow easy creation of pictures on the 1020 plotter and not as competition for Action! in any way.

Turtle instructions.

These instructions are all accessible in both immediate and program modes.

A typical turtle line begins with a line number, a three-letter instruction and any other data required by the instruction, like this:

```
10 SET20,20
```

Note that no spaces are allowed between an instruction and its parameters. Spaces are allowed between two different instructions and between a line number and its instruction. Each line can only hold one instruction, except when a comparison of true and false states is involved.

```
IFT GT01
```

This reads *IF TRUE GOTO Line 1.* The two instructions are *IF TRUE* and *GOTO.* The interpreter understands the following turtle instructions:

REM — This instruction acts as in BASIC. It's there to allow insertion of comments and is not executed by the turtle program.

```
10 REM My first program
```

TXT — The TeXT instruction puts text onto the printer in standard 40-column print.

```
10 TXT How to program in TURTLE
```

Turtle 1020 *continued*

PFN — The Print FunctioN allows use of the standard printer commands as described in the Atari 1020 manual.

```
10 PFN€+    Sets printer for 80 columns
```

GRH — GRapHics sets the printer to plotter mode.

```
10 GRH
```

COL — COLor sets the color of the pen to any of the four pen colors.

```
10 COL1     Sets color to 1
10 COLA     Sets color to number
            in variable A
```

PND — PeN Down puts the pen down, so that the plotter will actually draw.

PNU — PeN Up pulls the pen up, so that the pen cartridge/turtle only moves; it doesn't draw.

```
10 PND     and     10 PNU
```

HME — HoME sets the pen to home position (240,0).

```
10 HME
```

ORG — This instruction sets the pen to 0,0.

SET — This places the pen at a specified location. If the pen is down when SET is issued, a line will be drawn to this new position.

```
10 SET1,2   Sets the pen to location
            1 across and 2 down.
10 SETA,B   Sets the pen to location
            A across and B down.
```

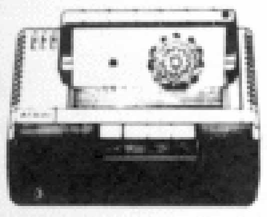
Note that turtle uses the Cartesian plane when drawing, so a positive Y-value is up and a negative is down.

TRN — TuRN causes the angle to increase or decrease, depending on the sign.

```
10 TRN10    Turn 10 degrees clockwise.
10 TRN-10   Turn 10 degrees counter-
            clockwise.
10 TRNX     Turn X degrees.
```

TTO — Turn TO makes the turtle point at an exact direction.

```
10 TTO20    and    TTOF
```



Turtle 1020 *continued*

FOR — FORward moves the pen forward a specified number of steps.

```
          10 FOR20    and    FORC
```

BAC — BACkward operates in the same way as FOR, except in the opposite direction.

GTO — GoTO places program control to a certain line number.

```
            10 GTO20    Goto line 20
            10 GTOB     Goto line B
```

CLR — CLeaR sets all variables to 0.

END — This simply returns control to the user, terminating the turtle program.

ACC — ACCept acts like INPUT as in BASIC. It puts your numerical input into a variable.

```
  10 ACCX      Your entry will be stored
               in variable X
```

TST — TeST checks whether the following comparison is true or false.

```
10 TSTA>B   Test if A is greater than B
10 TSTC=D   Test if C equals D
```

The tests allowed are >, < and =.

IFT — IF True allows the following instruction on the same line to be executed if the TST was successful.

```
           10 TSTA=0
           20 IFT TXT A equals zero
```

IFF — IF False acts in the same way as IFT, except the instruction is executed only if the TST condition was false.

REP — The REPeat instruction allows a certain range of lines to be repeatedly executed. The instruction requires a variable and a number to specify the number of repeats.

```
10 REPY10   Repeat 10 times; variable
            Y is used to keep track
            of which loop is executing.
```

ELP — End LooP closes the above loop.

```
 20 ELPY    Close the repeat loop
            opened by Y in line 10.
```

Math pack.

Turtle 1020 supports: addition (+), subtraction (−), multiplication (∗), division (/) and exponentiation (∧). These are used in the same way as in BASIC (e.g., A = 2 + 3∗4). However, it does not follow true algebraic logic and, hence, performs the operator as it detects it. In the above example, 2 will be added to 3 before being multiplied by 4. Further examples: 10 A = B∧C; and 20 Z = A + B − C∗D/E.

To generate a random number, use RND and a number (e.g., X = RND10). This yields a random number between 1 and 10, to be stored in variable X.

(continued on next page)

EXPAND
YOUR UNIVERSE
ATARI SOFTWARE PROTECTION TECHNIQUES
GEORGE MORRISON
ADVANCED ATARI PROTECTION TECHNIQUES
THE BOOK I
THE BOOK II
Actually done with Magniprint
Print Giant Posters · Up to 6 feet
NEW FEATURES

Turtle error messages.

All errors are translated into English, except for the input/output errors from disk or cassette. You'll have to consult your BASIC manual for the definition of these errors.

The errors which are translated into English are:

Command error — This occurs when you attempt to use an unknown command or instruction.

Illegal variable — This occurs when you attempt to use a variable other than those between A and Z.

Printer error — This occurs if you execute a printer command or turtle program without the printer being in its proper state of operation.

Illegal function — You will encounter this error if you use a mathematical function not in the categories mentioned earlier.

Line too long — This occurs when your turtle program line length exceeds 40 characters.

Overflow — Any variable holding a number either too large for BASIC to handle or divided by 0 will cause this error.

Illegal input — This occurs when you attempt to enter a non-numerical input.

Illegal line number — If your line number exceeds 199 or is less than 0, this error message will be issued.

All errors will be signaled by a bell, and the line at which the error occurred will be returned if in program execution mode.

Final words.

That's **Turtle 1020** in its entirety. If for any reason there's a program error or you pressed the BREAK key, you may resume **Turtle 1020** by typing:

```
GOTO 90
```

If this doesn't help, you'll have to RUN the BASIC program from the start.

I've included two turtle programs for you to try out using **Turtle 1020**. The first performs a TO SQUIRAL, a procedure recognized by many Logo and Pilot users. The second is a kaleidoscope generator, in which each design generated is completely different. It will prompt you for an input to represent the number of loops it's to perform. □

Jason Leigh graduated from King George V School in Hong Kong two years ago and is now a Computer Science student at the University of Utah. He's been working enthusiastically with the Atari since he was a pupil at Kowloon Junior School.

Listing 1.
BASIC listing.

```
10 REM TURTLE 1020 By Jason Leigh
20 REM
30 DEG :READ BEEP,PROMPT,CR,BELL,K1,K2
,K3,K4,K27,K40,K65:DATA 1080,90,155,25
3,1,2,3,4,27,40,65
40 DIM PROG$(8000):PROG$=CHR$(CR):PROG
$(8000)=PROG$:PROG$(K2)=PROG$
50 DIM LINE$(255),DAT$(255),NO$(K4),A(
26),B(26),C(26),A$(K4),C$(39),D$(128)
60 C$=CHR$(CR):C$(39)=C$:C$(K2)=C$
70 FOR T=K0 TO 26:A(T)=K0:B(T)=K0:C(T)
=K0:NEXT T
80 ? CHR$(125);"TURTLE 1020":?
90 EXE=K0:GOSUB 1110:POKE 82,K1:? CHR$
(30);CHR$(K27);CHR$(31);:POKE 82,K2:IN
PUT #16,LINE$
100 IF  NOT LEN(LINE$) THEN GOTO PROMP
T
110 IF LEN(LINE$)=K1 THEN A=ASC(LINE$)
:IF A>47 AND A<58 THEN 140
120 TRAP 540:IF LEN(LINE$)=K1 THEN A=A
(ASC(LINE$)-K65):? LINE$;"=";A:GOTO PR
OMPT
130 IF LINE$="DEL" THEN 1900
140 IF LINE$="CAT" OR LINE$="DIR" THEN
 1810
150 IF LINE$="TON" THEN TR=K1:TRON=34:
GOTO BEEP
160 IF LINE$="TOF" THEN TR=K0:TRON=K0:
GOTO BEEP
170 IF LINE$="NEW" THEN RUN
180 IF LINE$="MEM" THEN 1650
190 IF LINE$="LOAD" THEN 1440
200 IF LINE$="SAVE" THEN SA=K1:GOTO 14
40
210 A=ASC(LINE$):IF A>47 AND A<58 THEN
 730
220 IF LINE$="RUN" THEN 850
230 IF LINE$(K1,K3)="LIS" THEN A=K0:GO
SUB 560:GOTO BEEP
240 IF LINE$(K1,K3)="LIP" THEN TRAP 62
0:CLOSE #K3:OPEN #K3,8,K0,"P:":A=K3:GO
SUB 560:GOTO BEEP
250 IF LINE$="SEE" THEN ? "X=";INT(X);
" Y=";INT(Y);" COLOR=";C;" ANGLE=";ANG
LE:GOTO BEEP
260 DAT$=LINE$:NO$=DAT$(K1,K3):TRAP 62
0
270 IF NO$="REM" THEN GOTO BEEP
280 IF NO$="TXT" THEN CLOSE #K2:OPEN #
K2,8,K0,"P:":? #K2;DAT$(K4):? #K2;CHR$
(K27);CHR$(7):GOTO BEEP
290 IF NO$="PFN" THEN ? #K2;DAT$(K4):G
OTO BEEP
300 IF NO$="GRH" THEN ? #K2;CHR$(K27);
CHR$(7):GOTO BEEP
310 IF NO$(K2,K2)="=" THEN 1120
320 TRAP 940:IF NO$="COL" THEN C=VAL(D
AT$(K4)):TRAP 620:? #K2;"C";C:GOTO BEE
P
330 IF NO$="PND" THEN P=K1:GOTO BEEP
340 IF NO$="PNU" THEN P=K0:GOTO BEEP
350 TRAP 620:IF NO$="HME" THEN X=240:Y
=K0:ANGLE=K0:? #K2;"M";X;",";Y:GOTO BE
EP
360 IF NO$="SET" THEN 990
370 TRAP 950:IF NO$="TRN" THEN ANGLE=A
NGLE+VAL(DAT$(K4)):GOTO BEEP
380 TRAP 960:IF NO$="TTO" THEN ANGLE=V
AL(DAT$(K4)):GOTO BEEP
390 TRAP 970:IF NO$="BAC" THEN 630
400 TRAP 980:IF NO$="FOR" THEN 680
410 TRAP 520:IF NO$="GTO" THEN EXE=K1:
TRAP 1890:I=(VAL(DAT$(K4))-K1)*K40:GOT
O BEEP
```

```
420 IF NO$="REP" THEN E=(ASC(DAT$(K4))
-K65):C(E)=VAL(DAT$(5)):B(E)=I:A(E)=K0
:GOTO BEEP
430 IF NO$="ELP" THEN E=(ASC(DAT$(K4))
-K65):A(E)=A(E)+K1:IF A(E)<C(E) THEN I
=B(E):GOTO BEEP
440 IF NO$="ELP" THEN GOTO BEEP
450 TRAP 620:IF NO$="ORG" THEN X=K0:Y=
K0:? #K2;"H":GOTO BEEP
460 IF NO$="CLR" THEN FOR T=K0 TO 26:A
(T)=K0:B(T)=K0:C(T)=K0:NEXT T:GOTO BEE
P
470 IF NO$="TST" THEN 1670
480 IF NO$="IFF" THEN 1750
490 IF NO$="IFT" THEN 1780
500 IF NO$="END" THEN EXE=K0:GOTO BEEP
510 TRAP 550:IF NO$="ACC" THEN POKE 55
9,34:A=ASC(DAT$(K4))-K65:INPUT B:TRAP
540:A(A)=B:POKE 559,TRON:GOTO BEEP
520 ? CHR$(BELL);"Command error":GOTO
1420
530 ? CHR$(BELL);"Illegal line number"
:GOTO 1420
540 ? CHR$(BELL);"Illegal variable":GO
TO 1420
550 IF PEEK(195)=8 THEN ? CHR$(BELL);"
Illegal input":GOTO 1420
560 TRAP BEEP:? #A:T=K0:IF LEN(LINE$)>
K3 THEN T=(VAL(LINE$(K4))-K1)*K40
570 T=T+K40:IF ASC(PROG$(T))<>CR THEN
600
580 GOSUB 1400:IF T>7960 OR T>PRO THEN
 RETURN
590 GOTO 570
600 ? #A;T/K40;" ";:FOR R=T TO T+39:IF
 ASC(PROG$(R))<>CR THEN NEXT R:? #A;PR
OG$(T,T+39):GOTO 580
610 ? #A;PROG$(T,R-K1):GOTO 580
620 ? CHR$(BELL);"Printer error":GOTO
PROMPT
630 LONG=VAL(DAT$(K4))
640 TRAP 620:DEG :Y1=COS(ANGLE)*LONG:X
1=SIN(ANGLE)*LONG
650 IF P THEN ? #K2;"D";X-X1;",";Y-Y1
660 IF  NOT P THEN ? #K2;"M";X-X1;",";
Y-Y1
670 X=X-X1:Y=Y-Y1:GOTO BEEP
680 LONG=VAL(DAT$(K4))
690 TRAP 620:DEG :Y1=COS(ANGLE)*LONG:X
1=SIN(ANGLE)*LONG
700 IF P THEN ? #K2;"D";X+X1;",";Y+Y1
710 IF  NOT P THEN ? #K2;"M";X+X1;",";
Y+Y1
720 X=X+X1:Y=Y+Y1:GOTO BEEP
730 IF LEN(LINE$)<K3 THEN 750
740 IF LEN(LINE$)>39 THEN ? CHR$(BELL)
;"Line too long":GOTO PROMPT
750 L=LEN(LINE$):LINE$(L+K1)=CHR$(CR)
760 FOR T=K1 TO L:A=ASC(LINE$(T)):IF A
>47 AND A<58 THEN NEXT T
770 LINE=VAL(LINE$(K1,T)):L1=T
780 IF LINE>199 OR LINE<K1 THEN ? CHR$
(BELL);"Illegal line number":GOTO PROM
PT
790 PROG$(LINE*K40,LINE*K40+39)=C$
800 IF L<L1 THEN GOTO PROMPT
810 IF ASC(LINE$(L1))=32 THEN L1=L1+K1
:GOTO 810
820 IF LEN(LINE$)-L1<K3 THEN 520
830 PROG$(LINE*K40,LINE*K40+L-L1+K1)=L
INE$(L1):IF LINE*K40>PRO THEN PRO=LINE
*K40
840 GOTO PROMPT
850 TRAP 620:CLOSE #K2:OPEN #K2,8,K0,"
P:":POKE 559,TRON
860 I=K0:EXE=K1:ANGLE=K0:X=K0:Y=K0:FOR
 T=K0 TO 26:A(T)=K0:B(T)=K0:C(T)=K0:NE
XT T:C=K0:P=K0
870 TRAP 530:LINE$="":E=K0:I=I+K40:IF
ASC(PROG$(I))<>CR THEN 900
880 IF I>7960 OR I>PRO THEN EXE=K0:GOT
O PROMPT
890 GOTO 870
900 FOR R=I TO I+39:IF ASC(PROG$(R))<>
CR THEN E=E+K1:LINE$(E,E)=PROG$(R):S=K
1:NEXT R:GOTO 920
910 IF S THEN S=K0:GOTO 920
920 GOSUB 1400:IF TR THEN ? I/K40;" ";
LINE$
930 GOTO 260
940 TRAP 620:C=A(ASC(DAT$(K4))-K65):?
#K2;"C";C:GOTO BEEP
950 TRAP 520:ANGLE=ANGLE+A(ASC(DAT$(K4
))-K65):GOTO BEEP
960 TRAP 520:ANGLE=A(ASC(DAT$(K4))-K65
):GOTO BEEP
970 TRAP 520:LONG=A(ASC(DAT$(K4))-K65)
:GOTO 640
980 TRAP 520:LONG=A(ASC(DAT$(K4))-K65)
:GOTO 690
990 T1=K1:A=ASC(DAT$(K4))-K65:IF A>=K0
 AND A<K27 THEN 1040
1000 FOR T=K4 TO LEN(DAT$):IF DAT$(T,T
)<>"," THEN NEXT T
1010 T1=T+K1:X=VAL(DAT$(K4,T-K1))
1020 A=ASC(DAT$(T1))-K65:IF A>=K0 AND
A<K27 THEN 1050
1030 Y=VAL(DAT$(T1)):GOTO 1060
1040 X=A(A):T1=6:GOTO 1020
1050 Y=A(A)
1060 TRAP 620:IF P THEN ? #K2;"D";X;",
";Y
1070 IF  NOT P THEN ? #K2;"M";X;",";Y
1080 IF ANGLE>360 OR ANGLE<-360 THEN A
=INT(ANGLE/360):ANGLE=ANGLE-A*360
1090 IF EXE THEN 880
1100 ? :? "Okay":POKE 752,K0:GOTO PROM
PT
1110 FOR T=10 TO K0 STEP -K1:SOUND K1,
240,10,T:NEXT T:POKE 559,34:RETURN
1120 TRAP 1130:IF DAT$(K3,5)="RND" THE
N 1850
1130 V=ASC(NO$)-K65:ACC=K0
1140 IF V<K0 THEN 540
1150 Z=K3:GOSUB 1290:ACC=N:T2=T1:NEG=K
0
1160 Z=T1+K1:GOSUB 1290:N2=N:T3=T1:NEG
=K0
1170 A$=DAT$(T2,T2)
1180 TRAP 1250
1190 IF A$="+" THEN ACC=ACC+N2:GOTO 12
60
1200 IF A$="-" THEN ACC=ACC-N2:GOTO 12
60
1210 IF A$="/" THEN ACC=ACC/N2:GOTO 12
60
1220 IF A$="*" THEN ACC=ACC*N2:GOTO 12
60
1230 IF A$="^" THEN ACC=ACC^N2:GOTO 12
60
1240 ? CHR$(BELL);"Illegal function":G
OTO PROMPT
1250 ? CHR$(BELL);"Overflow":GOTO PROM
PT
1260 T2=T3:T1=T3
1270 IF S THEN A(V)=ACC:GOTO BEEP
1280 GOTO 1160
1290 TRAP 1390:A=ASC(DAT$(Z))-K65:IF A
>=K0 AND A<K27 THEN 1360
1300 NEG=K0:A$=DAT$(Z,Z):IF A$="-" OR
A$="+" THEN Z=Z+K1:NEG=K1:A=ASC(DAT$(Z
))-K65:IF A>=K0 AND A<K27 THEN 1360
1310 S=K0:FOR T=Z TO LEN(DAT$):A=ASC(D
AT$(T)):IF A=46 OR A>47 AND A<59 THEN
NEXT T
1320 IF NEG THEN Z=Z-K1:NEG=K0
```

```
1330 N=VAL(DAT$(Z,T-K1)):T1=T
1340 IF T>=LEN(DAT$) THEN S=K1
1350 RETURN
1360 V1=A:IF NEG AND A$="-" THEN N=-A(
V1):GOTO 1380
1370 N=A(V1)
1380 T1=Z+K1:T=T1-K1:GOTO 1340
1390 S=K1:GOTO 1270
1400 IF PEEK(764)<>255 THEN POKE 764,2
55:? "Break":GOTO 1420
1410 POKE 764,255:RETURN
1420 IF EXE THEN ? "at line #";I/K40:E
XE=K0
1430 GOTO PROMPT
1440 ? "Enter Filename->";:INPUT #16,D
AT$
1450 TRAP 1550:CLOSE #K3:IF SA THEN SA
=K0:GOTO 1570
1460 OPEN #K3,K4,K0,DAT$:A=K0
1470 GET #K3,I:GET #K3,K
1480 IF I<>K1 OR K<>K1 THEN ? :? CHR$(
BELL);"Not a TURTLE 1020 file":GOTO BE
EP
1490 GET #K3,PRO:PRO=PRO*K40
1500 GET #K3,LINE
1510 GET #K3,D:IF D=255 THEN 1540
1520 PROG$(LINE*K40+A,LINE*K40+A)=CHR$
(D):A=A+K1
1530 GOTO 1510
1540 A=K0:GOTO 1500
1550 IF PEEK(195)<>136 THEN ? :? CHR$(
BELL);"I/O error # ";PEEK(195)
1560 GOTO BEEP
1570 OPEN #K3,8,K0,DAT$:A=K0:T=K0
1580 PUT #K3,K1:PUT #K3,K1
1590 PUT #K3,PRO/K40
1600 T=T+K40:IF ASC(PROG$(T))<>CR THEN
 1630
1610 IF T>7960 OR T>PRO THEN CLOSE #3:
GOTO BEEP
1620 GOTO 1600
1630 PUT #K3,T/K40:FOR R=T TO T+39:IF
ASC(PROG$(R))<>CR THEN PUT #K3,ASC(PRO
G$(R)):S=K1:NEXT R:GOTO 1610
1640 IF S THEN S=K0:PUT #K3,255:GOTO 1
610
1650 R=K0:FOR T=K40 TO 8000 STEP K40:I
F ASC(PROG$(T))=CR THEN R=R+K1
1660 NEXT T:? R;" LINES FREE":GOTO BEE
P
1670 TRAP 520:A=ASC(DAT$(K4))-K65:A=A(
A)
1680 B=ASC(DAT$(6))-K65:IF B>-K1 AND B
<26 THEN B=A(B):GOTO 1700
1690 B=VAL(DAT$(6))
1700 C=ASC(DAT$(5)):STA=K0
1710 IF C=60 THEN STA=A<B:GOTO BEEP
1720 IF C=62 THEN STA=A>B:GOTO BEEP
1730 IF C=61 AND A=B THEN STA=K1:GOTO
BEEP
1740 GOTO BEEP
1750 TRAP 520:T=K4:IF STA THEN GOTO BE
EP
1760 IF DAT$(T,T)=" " THEN T=T+K1:GOTO
 1760
1770 LINE$="":LINE$=DAT$(T):GOTO 260
1780 TRAP 520:T=K4:IF  NOT STA THEN GO
TO BEEP
1790 IF DAT$(T,T)=" " THEN T=T+K1:GOTO
 1790
1800 LINE$="":LINE$=DAT$(T):GOTO 260
1810 ? :? "   DISK CATALOG":?
1820 TRAP 1550:CLOSE #K3:OPEN #K3,6,K0
,"D:*.*":TRAP 1840
1830 INPUT #K3,D$:? D$:GOTO 1830
1840 ? :GOTO BEEP
1850 TRAP 520
1860 V=ASC(DAT$(6))-K65:IF V>=K0 AND V
<K27 THEN A=A(V):GOTO 1880
1870 A=VAL(DAT$(6))
1880 V=ASC(NO$)-K65:A(V)=INT(RND(K0)*A
)+K1:GOTO BEEP
1890 V=ASC(DAT$(K4))-K65:I=(A(V)-K1)*K
40:GOTO BEEP
1900 ? "DELETE FROM LINE #->";:INPUT L
:? "TO LINE #->";:INPUT L1
1910 IF L>=L1 THEN ? CHR$(BELL);"Illeg
al values":GOTO PROMPT
1920 IF L1>199 OR L<K1 THEN L=K1:L1=K0
:GOTO 1910
1930 ? "DELETE LINE ";L;" TO ";L1;" (Y
/N)";:INPUT LINE$
1940 IF LINE$(K1,K1)="Y" THEN ? "DELET
ING":GOTO 1960
1950 GOTO PROMPT
1960 FOR T=L*K40 TO L1*K40 STEP K40:PR
OG$(T,T)=CHR$(CR):NEXT T:GOTO BEEP
```

●

CHECKSUM DATA.

(see page 18)

```
10 DATA 957,253,195,882,49,155,153,876
,710,297,922,100,778,659,363,7349
160 DATA 355,981,801,498,336,188,580,2
94,362,345,22,283,542,21,54,5662
310 DATA 818,364,699,716,888,318,591,8
06,428,476,368,387,271,276,527,7933
460 DATA 690,188,113,136,883,768,53,15
4,613,543,757,280,363,747,61,6349
610 DATA 89,628,112,518,546,97,805,127
,533,521,76,784,581,369,964,6750
760 DATA 834,77,432,575,93,655,916,296
,382,969,625,133,251,759,786,7783
910 DATA 657,561,728,944,344,549,786,7
99,385,870,141,218,803,118,341,8244
1060 DATA 875,560,822,456,401,166,45,1
72,689,153,197,617,682,19,21,5875
1210 DATA 30,11,220,540,981,237,256,72
3,233,128,534,989,755,819,793,7249
1360 DATA 117,603,840,688,273,397,706,
545,789,530,423,1,12,33,243,6200
1510 DATA 115,961,721,672,106,927,61,2
86,573,520,40,722,211,55,322,6292
1660 DATA 139,984,561,608,202,594,599,
286,931,424,540,670,938,546,666,8688
1810 DATA 588,639,775,235,868,217,611,
814,157,119,645,901,835,632,562,8598
1960 DATA 110,110
```

●

Listing 2.

```
1 GRH
2 COL2
3 SET240,-500
4 PND
5 REPA90
6 FORS
7 TRN89
8 S=S+2
9 ELPA
```

●

Listing 3.

```
4 ACCY
6 GRH
8 C=RND4
9 S=RND60
10 C=C-1
12 COLC
14 TTO0
16 D=RND5
18 D=D-1
20 F=D*10+100
22 REPB6
24 PNU
26 SET240,-500
27 TRN60
28 FORS
30 PND
32 GTOF
36 ELPB
38 TSTZ>Y
40 IFTGTO46
42 Z=Z+1
44 GTO8
46 END
100 TRN-30
101 REPA3
102 FORS
103 TRN120
104 ELPA
105 TRN30
106 GTO36
110 TRN-45
111 REPA4
112 FORS
113 TRN90
114 ELPA
115 TRN45
116 GTO36
120 TRN-67.5
121 REPA8
122 E=S/2
123 FORE
124 TRN45
125 ELPA
126 TRN67.5
127 GTO36
130 TRN-18
131 REPA5
132 FORS
133 TRN144
134 ELPA
135 TRN18
136 GTO36
140 TRN-60
141 FORS
142 TRN60
143 FORS
144 TRN120
145 FORS
146 TRN-60
147 FORS
148 TRN120
149 FORS
150 TRN60
151 FORS
152 TRN120
153 GTO36
```

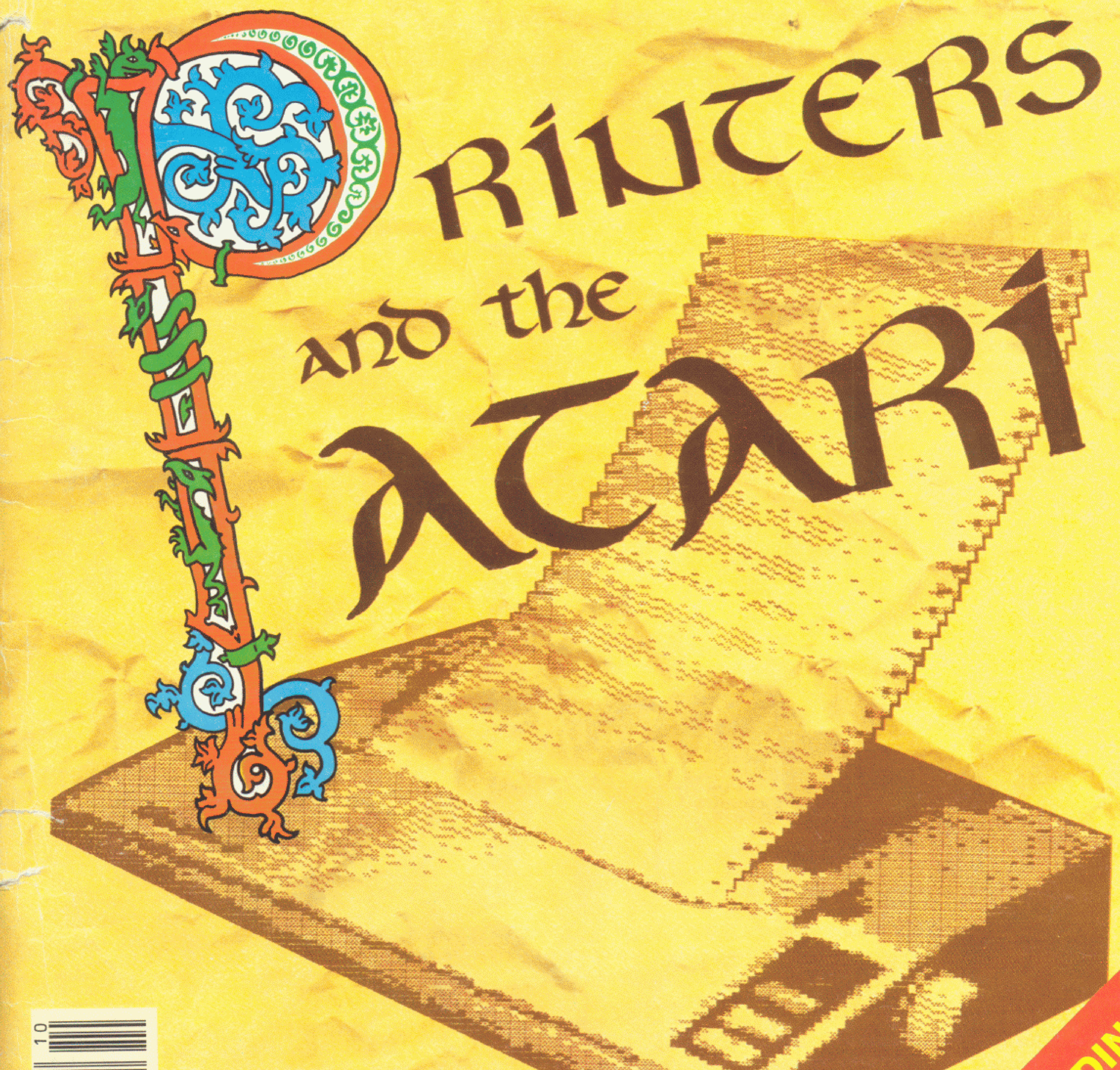
●

NO. 35
OCTOBER 1985

U.S.A. $3.00
CANADA $3.50

THE #1 MAGAZINE FOR ATARI® COMPUTER OWNERS

A.N.A.L.O.G. COMPUTING™



Printers and the Atari

INCLUDING ATARI 520ST