obvious definition is:

```
TO COPYDEF :NEW :OLD
  DEFINE :NEW TEXT :OLD
END
```

The trouble with this definition is that if OLD does not exist then the procedure simply goes ahead and defines NEW as nothing. It would be better to pick up this problem and report on it. So a better definition of COPYDEF would be:

```
TO COPYDEF :NEW :OLD
  IF NOT PROCEDURE? :OLD THEN ( PRINT
    [THERE IS NO PROCEDURE] :OLD ) STOP
  DEFINE :NEW TEXT :OLD
END
```

This uses a procedure called PROCEDURE?, which outputs TRUE if its input is a procedure, and FALSE otherwise. PROCEDURE? and its counterpart, PRIMITIVE?, are very useful tests, but unfortunately they do not exist in MIT LOGO. So we've developed versions of PROCEDURE? and PRIMITIVE? that will work on both the Apple and the Commodore versions of LOGO:

```
TO PROCEDURE? :NAME
  IF NUMBER? :NAME THEN OUTPUT "FALSE
  IF LIST? :NAME THEN OUTPUT "FALSE
  TEST WORD? :NAME
  IFTRUE IF WORD? TEXT :NAME THEN OUTPUT
    "FALSE ELSE IF NOT (TEXT :NAME = [] ) THEN
    OUTPUT "TRUE
  OUTPUT "FALSE
END

TO PRIMITIVE? :NAME
  IF NUMBER? :NAME THEN OUTPUT "FALSE
  IF LIST? :NAME THEN OUTPUT "FALSE
  TEST WORD? :NAME
  IFTRUE IF WORD? TEXT :NAME THEN OUTPUT
    "TRUE ELSE OUTPUT "FALSE
END
```

AFTERWORD

We have now dealt with all the major features of standard LOGO, and have covered a wide area of possible applications. If you want to read more about the language, here are four suggested books:

- *Learning with Logo* by Daniel Watt (McGraw-Hill) is a wonderful introductory book, and is ideal for using with children.
- *Logo* by Harold Abelson (McGraw-Hill) is the 'standard' book on the language.
- *Turtle Geometry* by Harold Abelson and Andrea diSessa (MIT Press) takes a serious look at turtle geometry. The maths involved is at sixth form and college level — one of the later chapters develops a simulator for General Relativity in LOGO!
- *Thinking about [TLC] Logo* by John R. Allen, Ruth E. Davis and John F. Johnson (Holt Sanders International Editions) uses the rather idiosyncratic TLC LOGO, but the book is valuable for its investigation of artificial intelligence themes using LOGO.

Logo's Virtues

- It is interpreted, like BASIC, so it is easy to run and to modify programs.
- It is structured, with true procedures, unlike most versions of BASIC.
- It is extensible, like FORTH – i.e. it is possible to define new words that then become part of the computer's vocabulary.
- It has list processing, like LISP, so is useful for the exploration of areas such as artificial intelligence.

LOGO is not really designed for implementing fully worked out perfect algorithms, but is ideal for 'hacking'. We have written most of the programs in this series by starting with a simple procedure to perform just part of the task. We then alter the procedure in various ways, and as a result improve it and develop it as our understanding of the problem grows. At the end of the process we are left with a well-designed perfect algorithm.

Logo's Vices

- The main problems remain limited workspace and slow execution speed.
- Other features would be useful, in particular LOGO lacks error-trapping facilities, arrays and file handling. Some LOGOs do have these features but they are not common to all.

Logo Flavours

On all LCSI versions use:
NUMBERP for NUMBER?
LISTP for LIST?
WORDP for WORD?
EMPTYP for EMPTY?

For the different IF syntax, see page 815.

Spectrum LOGO has COPYDEF as a primitive, as well as PRIMITIVEP (corresponding to our PRIMITIVE?) and DEFINEDP (corresponding to our PROCEDURE?).

On the Atari use: PC for PENCOLOR, SE for SENTENCE, and HT for HIDETURTLE. DEFINE and TEXT do not exist in Atari LOGO, though the manual gives a way of defining them.

The pen colours used will differ from machine to machine. PC –1 is used here to erase lines, but some LOGO versions have this as a primitive PE (standing for 'Pen Erase')