so the procedure outputs NTH 1 BUTFIRST :WORDS. This ignores the first list element and takes the first word from the remainder of the list — FATHER.

So our procedure to print a random word from our limited vocabulary would be:

```
TO GETRANDOM :LIST
  OUTPUT NTH ( ( RANDOM 9 ) + 1 ) :LIST
END
```

To use this, type GETRANDOM :WORDS.

Our procedure is restricted to lists of nine items. We could improve on this if we could determine how many items there are in a given list. Here is a procedure that does this:

```
TO LENGTH :LIST
  IF EMPTY? :LIST THEN OUTPUT 0
  OUTPUT 1 + LENGTH BUTFIRST :LIST
END
```

To see how this works try: LENGTH [SCIENCE FICTION]. As the list contains some words the first condition fails, so the procedure outputs 1 + LENGTH [FICTION]. Now LENGTH [FICTION] outputs 1 + LENGTH []. On calling LENGTH with an input of [], the condition in line 1 is true, so the procedure outputs 0. Now LENGTH [FICTION] outputs 0 + 1 = 1 and, finally, LENGTH [SCIENCE FICTION] outputs 1 + 1 = 2. So a more general procedure for getting random words from a list of any length is:

```
TO GETRANDOM :LIST
  OUTPUT NTH ( ( RANDOM LENGTH :LIST ) + 1 )
  :LIST
END
```

In many versions of LOGO there is a primitive, ITEM, which does precisely what NTH does, and a primitive called COUNT that does the same as LENGTH. Using these we can rewrite the procedure:

```
TO GETRANDOM :LIST
  OUTPUT ITEM ( ( RANDOM COUNT :LIST ) + 1 )
  :LIST
END
```

To print a selection of 10 comments to keep your psychoanalyst listening attentively, simply type:

```
REPEAT 10 [PRINT GETRANDOM :WORDS]
```

There is a pattern to these list processing programs that was shared by many of our recursive turtle graphics procedures. The pattern is:

- If the task to be performed is extremely simple then do it and stop.
- Otherwise do a small part of the task.
- Then pass the rest of the task onto another procedure (often a copy of the original procedure).

This is a highly successful strategy, which we will encounter repeatedly in list processing programs. Compare this polygon drawing program:

```
TO POLY :N
  IF :N = 0 THEN STOP
  FD 30 RT ( 360 / :N )
  POLY :N — 1
END
```

with the version of PRINTOUT given earlier. The structure of the two procedures is identical.

RANDOM POETRY

Having failed to impress our psychoanalyst, we now turn our hand to poetry. Here, we will want to produce whole sentences rather than single words.

```
TO POEM1 :LENGTH
  IF :LENGTH = 0 THEN PRINT "STOP
  ( PRINT1 "' ' GETRANDOM :WORDS )
  POEM1 :LENGTH - 1
END
```

PRINT1 "' ' is included to print a space between words. To use this procedure, type POEM1 6 for a six-word sentence.

It would be useful to be able to extend our

Abbreviations

BUTFIRST	BF
BUTLAST	BL
SENTENCE	SE

SIR JOHN TENNIEL

Mock Turtle Sings
An extract from the Mock Turtle's song from 'Alice in Wonderland' by Lewis Carroll. The metric pattern, a little difficult to duplicate in a computer poem, is adapted from an old folk song, and is also utilised by Mary Howitt in the classic poem, 'The Spider and the Fly'

'Will you walk a little faster?' said a whiting to a snail,
'There's a porpoise close behind us, and he's treading on my tail.
See how eagerly the lobsters and the turtles all advance!
They are waiting on the shingle — will you come and join the dance?
Will you, won't you, will you, won't you, will you join the dance?
Will you, won't you, will you, won't you, won't you join the dance?'

'You can really have no notion how delightful it will be
When they take us up and throw us, to the lobsters, out to sea!'
But the snail replied 'Too far, too far! and gave a look askance —
Said he thanked the whiting kindly, but he would not join the dance.
Would not, could not, would not, could not, would not join the dance.
Would not, could not, would not, could not, could not join the dance.