3-1988

After Before Comes...?

Judi Harris
College of William and Mary

Recommended Citation

Harris, J. (1988). After before comes...?, Logo Exchange, 6(7), 22-23.

Logo LinX

"After Before Comes...?"

by Judi Harris

Stanley Kubrick's 1968 space odyssey, "2001," introduced HAL, an almost-perfect computer that appeared to make some fatal mistakes. After "2001" came "2010," during which HAL's mistakes were (of course) discovered to be the result of human misjudgement. But what comes after HAL?

The answer: IBM.

> I comes after **H**,
> B comes after **A**,
> and M comes after **L**.

Clever nomenclature, isn't it? It's also an interesting Logo exploration that can help to reinforce notions of alphabetic sequence and phonetic analysis.

Position is (Almost) Everything

What are some other "afterwords"? What spelling patterns (if any) do they follow? In this "afterworld," all vowels become consonants:

> A becomes B, E becomes F,
> I becomes J, O becomes P,
> U becomes V, Y becomes Z.

These consonants become vowels:

> Z becomes A, D becomes E,
> H becomes I, N becomes O,
> T becomes U, X becomes Y.

Many consonants become other consonants.

> B becomes C, C becomes D,
> F becomes G, J becomes K,
> K becomes L, L becomes M,
> M becomes N, ...and so on.

Afterimage

Two-letter "afterword" pairs mirror each other with regard to letter type.

OH becomes PI (Vowel-consonant becomes consonant-vowel.)
> (V C becomes C V.)
TO becomes UP
> (Consonant-vowel becomes vowel-consonant.)
> (C V becomes V C.)

Three-letter "afterword" pairs appear in several configurations. Since all vowels become consonants, a V C C word can only become a C V V or a C V C word.

> END becomes FOE (V C C becomes C V V.)
> ADS becomes BET (V C C becomes C V C.)

Some consonants become other consonants, but a V C V word can only become a C V C word, since all vowels become consonants, and no word can be spelled without a vowel.

> AHA becomes BIB (V C V becomes C V C.)

The same reasoning applies to a C V V word, which must become a V C C word. The only other possibility for it to become is a C C C word, which is impossible.

> NEE becomes OFF (C V V becomes V C C.)

V V C words that become C C V words are impossible. Can you figure out why? Here's a hint: you may have to make a list of all of the possible V V combinations and their "afterimages."

Aftermath

How many three-letter "afterword" sequences are possible? Which are linguistically feasible? To boggle the mind even further, how many four-letter successions can occur? Here's just one example:

> ANTS becomes BOUT (V C C C becomes C V V C.)

Would you predict that there are more two-letter, three-letter, or four-letter "afterwords"? Are five, six, or more-letter "afterwords" possible? If you could generate all possible English "afterwords," would the frequencies (based on length) constitute a mathematical pattern? If so, can you postulate some of its characteristics?

As you may have already guessed, a few simple Logo procedures can serve as helpful tools in this exploration.

Each letter typed in Logo has a corresponding numeric value in ASCII code. The computer will print out this number if the ASCII command is invoked.

> PRINT ASCII "A yields the number 65.
> PRINT ASCII "B yields the number 66.
> PRINT ASCII "C yields the number 67.

Conversely,
  PRINT CHAR 65 yields A,
  PRINT CHAR 66 yields B,
  and PRINT CHAR 67 yields C.

Notice that the numbers conveniently increase by one as you type the alphabet in order. We can take advantage of that convention by putting together a procedure that will output the letter immediately following the input :LETTER. Also note that a special result is directed for an input of the letter Z.

```
TO NEXT :LETTER
 IF :LETTER = "Z [OUTPUT "A]
 OUTPUT CHAR (( ASCII :LETTER ) + 1 )
END
```

```
PRINT NEXT "A yields B,
PRINT NEXT "B yields C,
PRINT NEXT "Z yields A.
```

Afterworld

The procedure NEXT.WORD uses NEXT to recursively concatenate and output the "afterword" that corresponds to its :INPUT.

```
TO NEXT.WORD :INPUT
  IF EMPTYP :INPUT [OUTPUT :INPUT]
  OUTPUT WORD ( NEXT FIRST :INPUT )
      (NEXT.WORD BUTFIRST :INPUT )
END
```

NOTE: In Terrapin Logo 2.0 and later, substitute EMPTY? for EMPTYP.

```
PRINT NEXT.WORD "AX  yields BY.
PRINT NEXT.WORD "ADD yields BEE.
```

Finally, the superprocedure AFTERWORD prompts the user to type a "beforeword," then prints the corresponding "afterword."

```
TO AFTERWORD
  TYPE [FIRST WORD:]
  PRINT ( SENTENCE [NEXT WORD:] NEXT.WORD
      READWORD )
END
```

NOTE: In Terrapin Logo 2.0 and later, substitute PRINT1 for TYPE, and FIRST REQUEST for READWORD.

The exchange might look like this:

```
FIRST WORD :ITS
NEXT WORD :JUT
```

Afterthought

How many more "afterwords" can you and your students discover? We will publish a compiled list of all entries and participants' names, if you will mail the fruits of your "after-effects" to me by September 1, 1988. Address your output to:

Judi Harris
292 Ruffner Hall
Curry School of Education
University of Virginia
Charlottesville, VA 22903

In addition, you may want to adapt the procedures and ideas presented in this article to view the challenge from a different perspective. What is the purpose of this mission?

"To go where no [one] has gone BEFORE...."
                              —Cap't. J.T. Kirk