Twice Upon a Time

Judi Harris

College of William and Mary

Follow this and additional works at: https://scholarworks.wm.edu/educationpubs

Part of the Education Commons

Recommended Citation
Twice Upon a Time  
by Judi Harris

"Anytwo for elevennis?"

This sentence has suffered Logo inflation. Last year, it might have read,

"Anyone for tennis?"

Next year, if things keep going up, it may read,

"Anythree for twelvehnis?"

Now that holiday gift bills are starting to arrive, perhaps inflation is the last thing that you want to remember. Yet it can inspire enjoyable classroom exploits with syllabication, sequencing, and homophones.

Rising to the Occasion

Once Logo inflation hits, "I ate a tenderloin with my fork" becomes "I nined an elevenderloin with my fived." "Four-score and seven years ago, our forefathers brought forth" reads, instead: "Fivescore and eight years ago, our fivedathers brought fiftth." And so on and so fifth.

Danish comedian Victor Borge first introduced the notion of inflationary words in an effort to match language to economic trends. He reminds us that English "is your language; I'm just trying to use it." Borge suggests that we inflate words as a proactive measure, since inflation (like taxation) is inevitable.

Getting a Rise Out of Them

This presents an interesting Logo challenge. The sound of the first step toward solution is a homophonic one. How many different ways are there to spell each of the number words, 1 through 10? Your students will probably be glad to list the possibilities.

one two three four five ....
won to for
too fore
tu

Now, form a list from these homonyms, output by a procedure called PREINFLATION.

TO PREINFLATION
OUTPUT [ONE WON JUAN TWO TO TOO TU
THREE FOUR FOR FORE FIVE SIX
SICKS SICS SEVEN EIGHT NINE
ATE ATE AN ELEVEN]
END

An accompanying list of the same length can output inflated "values" for each of the words, in order.

TO POSTINFLATION
OUTPUT [TWO TWO TWO THREE THREE
THREE THREE FOUR FIVE FIVE
SIX SEVEN SEVEN SEVEN EIGHT NINE
NINE NINE TEN TEN TEN ELEVEN]
END

Inflated Ergo

An INFLATE command can be written to output corresponding inflated list elements.

TO INFLATE :WORD.PART
IF MEMBERP :WORD.PART PREINFLATION
[OUTPUT ITEM (ELEMENT :WORD.PART PREINFLATION)
POSTINFLATION] [OUTPUT :WORD.PART]
END

INFLATE uses an adaptation of Alison Birch's ELEMENT subprocedure, which is the opposite of the primitive ITEM.

TO ELEMENT :ITEM :OBJECT
IF :ITEM = FIRST :OBJECT [OUTPUT 1]
OUTPUT 1 + ELEMENT :ITEM
BUTFIRST :OBJECT
END

The superprocedure INFLATED uses these four subprocedures to output more expensive words.

TO INFLATED :LIST
IF EMPTY :LIST [OUTPUT "]
OUTPUT WORD ( INFLATE FIRST :LIST )
INFLATED BUTFIRST :LIST
END

Students must supply syllabicated words as input to INFLATE. For example, if a user types

PRINT INFLATED [WON DER FUL]
the computer will return:

TWODERFUL .

PRINT INFLATED [BE FORE]
yields

BEFIVE .

PREINFLATION and POSTINFLATION resultant lists can, of course, be adjusted to predict inflation at any rate. Who knows? Don Juan may someday be Don Eight. Why not adjust the fable now?
Leaping to Conclusions with Spreadsheets
by Glen Bull and Gina Bull

This column is about connections between Logo and other kinds of hardware, software, and concepts. Ordinarily we might discuss how data from Logo can be transferred to a spreadsheet, or vice versa. However, this month we would like to discuss how similar concepts can find expression in both Logo and spreadsheets. We have chosen spreadsheets as our basis for comparison because of the (dare we say it?) widespread familiarity with them, but other software such as Hypercard would serve as well.

If a thing is worth doing, it is worth doing poorly.

Learning new things is often only possible through a series of successive approximations. Rarely is a skill perfected the first try. Thousands of tennis buffs enjoy their inexpert weekend games just as much as Bjorn Borg or Jimmy Conners. Millions enjoy chess matches with their friends even though they have not achieved even the lowest national ranking. There are two important ideas here. Often activities worth doing can be enjoyed even if done inexpertly. And, more importantly, most experts begin as novices.

This rule also applies to problem solving activities. If it is not possible to solve a problem completely, solving part of the problem may be a good way to begin. Recently we met a teacher who wanted to calculate the ages of children in her classroom. She had just acquired a spreadsheet and wanted to use it to create a template to do the calculations. The initial format of the spreadsheet that she set up looked like this:

<table>
<thead>
<tr>
<th>Date of Birth</th>
<th>Current Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>Month</td>
</tr>
<tr>
<td>John 1979</td>
<td>1</td>
</tr>
<tr>
<td>Sally 1978</td>
<td>10</td>
</tr>
<tr>
<td>Sam 1979</td>
<td>3</td>
</tr>
</tbody>
</table>

She wanted to know how to create a formula to perform the computations. Although the problem looks easy, it is a nontrivial task for a novice. To see why, let's look at the calculations involved for John and Sam. John was born on January 17, 1979. Let's assume that today's date is February 20, 1989. The calculation would look like this:

\[
\begin{align*}
\text{Year} & \quad \text{Month} & \quad \text{Day} \\
1989 & \quad 2 & \quad 20 \\
1979 & \quad 1 & \quad 17 \\
10 & \quad 1 & \quad 3
\end{align*}
\]

Through a matter of three separate subtractions we would determine that John is 10 years, 1 month, and 3 days old. Sam's case is a bit more complicated. Sam was born in March.

References


Judi Harris, 621F Madison Avenue
Charlottesville, VA 22903