

Reflections...from page 9

learn a recipe (1. Understand the problem, 2. Decide on a strategy to solve the problem 3. Carry out the strategy, etc.). So unless the teacher engages the students in examples that bring out the spirit of the problem solving experience, this chapter will just be business as usual for most students.

Even Logo can turn into textbook-like, rule-driven exercises by well-intentioned teachers. For example, I have seen teachers teach the Total Turtle Trip Theorem as if it were just another *thing* for children to learn. I was reminded of this during a Logo contest in Newark, New Jersey that I recently participated in as a judge. One of the teams had difficulty drawing a rectangle because they were relying on a "rule" that they did not understand. They were a slave to the rule and couldn't adapt. The students who did well started off by typing in a few tool procedures like POLY, ARCR, ARCL and then used these tools to draw the graphics challenges. The development of this kind of ability to use problem solving heuristics is at the heart of what the *Standards* are all about.

I want to acknowledge the teachers in Newark who are contributing to the well being of many children by teaching them with the help of Logo how to use problem solving strategies effectively.

On Keeping Logo in the Council for Logo in Mathematics Education

In previous writings I have mentioned the possibility that this organization should be supporting other ways of using computers that reflect the spirit of Logo (as well as the spirit of the *Standards*.) I've noticed in recent issues of the *Logo Exchange* that the editorial staff and some columnists have been thinking about the same thing. But most of the mail I've received (See Letters to the Editor, p. 2) have encouraged me *not* to do so. This input has helped me to see that changing the name of our organization might give people the impression that the heyday of Logo is over, and it's time to move on to something different. I definitely do not want anyone to get that impression. So I declare that as long as there are people out there who are interested in using and sharing Logo, this organization will not change its focus.

**Problem... from page 6**

```
TAB 2 PRINT1 "FREQUENCY
TAB 4 PRINT1 "LETTER
TAB 2 PRINT1 "FREQUENCY PRINT []
PR.TALLY 65
END
```

```
TO COUNT.LET.IN.WORD :AWORD
IF EMPTY? :AWORD STOP
MAKE WORD "C ASCII FIRST :AWORD ( THING WORD
"C ASCII FIRST :AWORD ) + 1
COUNT.LET.IN.WORD BF :AWORD
END
```

The main procedure is COUNT.LETTERS. Here's how you use it. Type:

```
COUNT.LETTERS [Now is the time for all good men and
women to come to the aid of their country.]
```

•The procedure will first turn all the letters into upper case. (Note the primitive UPPERCASE which turns the input into all uppercase letters.)

•Next it will initialize variables C65 to C90 to be equal to 0. Note the use of WORD to link the letter C with the numbers.

•Then it will count the letters one word at a time with TALLY and COUNT.LET.IN.WORD. In the latter procedure, if the word is NOW the procedure will increment C78 (78 is ASCII for N) which initially was set to 0. So every time an N appears, C78 gets incremented. When all the letters in the word are completed, control is returned to TALLY which gets the next word checking first to see if another word exists. After the last word is tallied, control is past to DISPLAY.RESULTS which produces this frequency table.

LETTER	FREQUENCY	LETTER	FREQUENCY
A	3	N	5
B	0	O	10
C	2	P	0
D	3	Q	0
E	7	R	3
F	2	S	1
G	1	T	7
H	3	U	1
I	4	V	0
J	0	W	2
K	0	X	0
L	2	Y	1
M	4	Z	0

Some variations:

This program will work only as long as you have letters of the alphabet. If you include a period at the end of the sentence or a comma the program will print an error message. Modify the program to allow for other symbols. Include a total letter and word count. Print the letter frequency in percentages. Also include average word length if the fancy strikes you.

If your version has a way of reading data off the disk, write a program that reads the file and gives an analysis of the text.

If your version allows for the reading of text on the screen, write procedures that will analyze the text in that way.

For LogoWriter users:

UPPERCASE is not a primitive in LogoWriter. How would you write such a procedure? (Hint: The ASCII number for upper case A is 65 and the lower case is 97.)

