Unit 2: Assigning Values to Variables Application: A Mystery

In this Application for Unit 2, you will write a program that uses some simple assignment statements and arithmetic to perform some coding 'magic.' Skill Builders for Unit 2 should be completed prior to this activity.

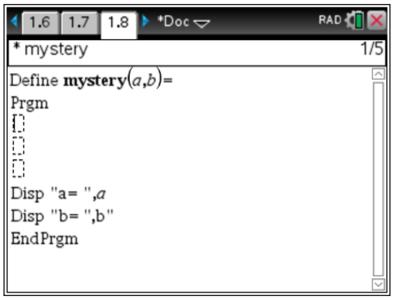
Objectives:

- Write a program from given instructions
- Use the assignment statement
- Discover the mystery inside a program

Program Mystery

Write a program that takes in two arguments, *a* and *b*, and then:

- Adds **a** to **b** and stores the result in **b**
- Subtracts **a** from **b** and stores the result in **a**
- Subtracts **a** from **b** and stores the result in **b**
- Displays **a** and **b**



```
Define mystery(a,b)=
Prgm
□
□
□
Disp "a= ",a
Disp "b= ",b
EndPrgm
```

Store and test your program. What is the effect? Does it work with *any* two numbers?

At first glance it appears that the same value (see the subtraction instructions) is being stored in *a* and *b*. But, is it?

Teacher Tip: The mystery program swaps the values of the two variables. Here's how it works step-by-step:

Statement	Values after Statement		Examples	
start	a	b	a = 5	b = 8
b:=a + b	a	a + b	a = 5	b = 5 + 8 or 13
a:=b - a	a + b – a = b	a + b	a = 13 – 5 or 8	b = 13
b:=b - a	b	a + b - b = a	a = 8	a = 13 – 8 or 5

You can use the statement *Disp a,b* to display both values in one line or, better, use

Disp "a=",a
Disp "b=",b

at the beginning and the end of the program. For more detail, add *Disp* statements after each assignment statement.

This program illustrates the power of the *sequential* nature of programming. Each step is processed one at a time, and the resulting values have a profound impact on the subsequent statements.