Unit 2: Using Variables and Expressions | **Skill Builder 3: Expressions and Storing**

In this third lesson for Unit 2 you will learn about using expressions and storing values in variables within programs.

Objectives:

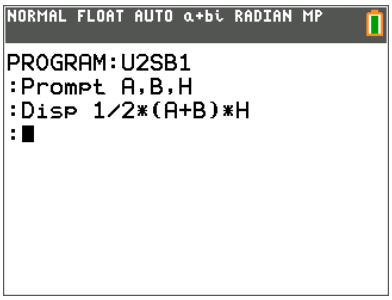
- Learn about programming mathematical expressions.
- Understand order of operations.
- Realize the difference between mathematical variables and computer program variables.
- Evaluate **expressions.**
- **Store** the results of expressions in variables.

Expressions

Items such as $\mathbf{A^2}$ and **A+B** are called *expressions*. Expressions can be found in mathematical formulas. For example, the **formula** for the area of a triangle is $A = \frac{1}{2} \times B \times H$. The **expression** is $\frac{1}{2} \times B \times H$.

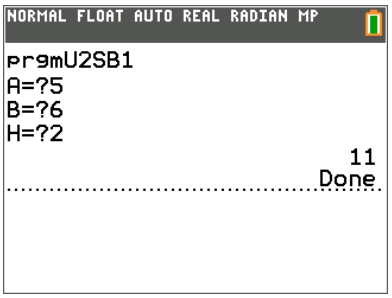
A program evaluates an expression using the current values of all variables and gives the result as a numeric value. Expressions are evaluated using the *algebraic order of operations*.

Try the program to the right which computes the area of a *trapezoid* with bases **A** and **B** and height **H**.



You cannot use variables such as **B1** and **B2**. The calculator computes these as the expressions $B \times 1$ and $B \times 2$ and may cause an error when used incorrectly.

You also cannot use variables with more than one letter such as **AB**. As stated earlier, this means $A \times B$. This is called *implied* multiplication because the multiplication sign between the variables is 'implied' or assumed.



Mathematical Expressions and Computer Expressions

While there are many similarities in the *appearance* of expressions in mathematics and computer programs there are also important differences. The most significant difference is that in a mathematical expression the variables stand for 'unknown' numbers and are replaced with numbers when needed. In a computer expression the variables are *names* for numbers.

In mathematics we use formulas to state a relationship between things such as area and lengths. In programs we use the expressions to do the calculations and the variables' values are used to compute a result. When we *write* the program we type the expression but when we *run* the program the expression is computed and creates a result to be used later.

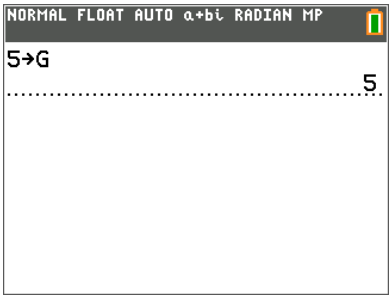
Teacher Tip: One of the more confusing statements in a program for beginners is x+1→x. It's pretty clear in the syntax of TI-Basic that 1 is being added to the variable x and then being stored in variable x, so the x on the left and the x on the right represent different values. This statement is known as a 'counter' because each time it is processed (in a loop) it increases x by 1. But in other languages such as B.A.S.I.C and Lua (and many more) the statement appears as x=x+1 which is clearly a false assertion in a mathematical sense but is perfectly good in a program!

Storing Values in Variables: The Assignment Statement

The STO▶ operator is used to **store** (assign) the result of an expression into a variable.

Pressing the STO▶ key always displays the symbol →.

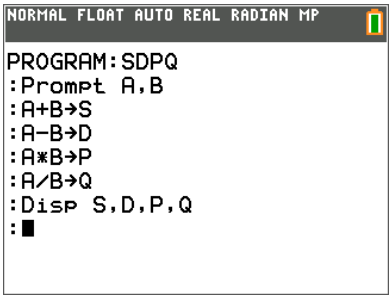
After pressing ENTER, the HOME screen displays the result of the expression *and* the variable **G** now contains the value **5**. In front of the arrow must be a **value** or an **expression** that produces a **value**. This is called the *assignment statement* because it assigns a value to a variable. The symbol after the arrow must be a **variable**.



Programming with Assignment Statements

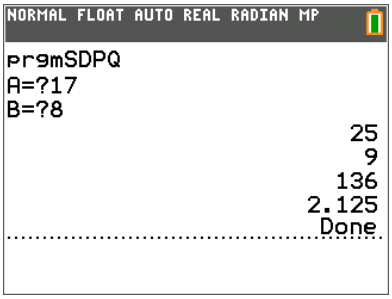
Let's write a program that asks you to enter two numbers, stores their sum, difference, product, and quotient in four variables and then displays the results.

1. Start a new program. Our program name is **SDPQ.**
2. Prompt for two variables **A** and **B.**
3. Store the four expressions in four *other* variables **S**, **D**, **P**, and **Q.**
4. Display **S**, **D**, **P** and **Q.**
5. Run the program.



Enter a value for **A** and another value for **B**.

Be sure to use numbers for which you can confirm that the calculations were performed correctly! This is known as 'testing' the program.



Note: TI Basic's assignment statement is unique in that the expression comes first, then the 'store' operator, then the variable. This makes it easy to read from left to right. In most other languages the order is the opposite, such as S=A+B. This is backwards because the computer evaluates the expression on the right first and then stores the result in the variable on the left.

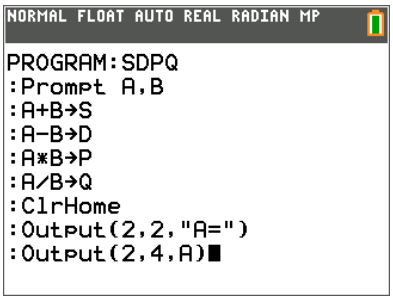
Teacher Tip: There are many advantages to storing values in variables. Using variables makes it easier to display the result. It also allows the variable to be used in subsequent calculations. If you find you are using the same number in a lot of different places in your program then you can store the number in a variable and use it everywhere that it occurs in your program.

Making Improvements

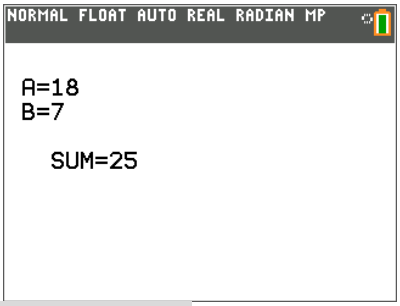
You can improve the program using **Output(** rather than **Disp** to show the original values entered *and* the four results *properly labeled.* You try it!

To the right is a snippet of code and a screen of the <u>partially</u> completed program running. There's still some work to do here so we'll leave it to you to complete the program.

Remember to include a **Pause** statement at the end of the **Output** statements to prevent the 'Done' message from spoiling the display.



Remember that the **TI-84 Plus** and the **TI-84 Plus C/CE** have different HOME screen sizes (as well as different GRAPH screen sizes) so plan accordingly. The TI-84 Plus HOME has 16 characters per line and 8 lines, while TI-8 Plus C/CE HOME has 26 characters per line and 10 lines.



Teacher Tip: Challenge students to experiment with other mathematical formulas and use assignment statements to store the result to make it easier to **Display** or **Output** the results. In the Application for this unit students are given three mathematical formulas to compute.