

BEST PRACTICES

Comments can be added to programs simply by inserting them in a pair of quotes so that they become strings. Mathcad ignores all lines containing only strings. Adding comments to programs will aide other users who might use your worksheet.

```
locate(x, V) := "This program searches a vector V for a scalar value x"
               "the index (location) of the element is returned when found"
               "If the element is not found, -1 is returned"
               "Initialize the counter to 1"
               i ← 1
               "Iterate over the elements of V"
               for v ∈ V
               | "Break out of the program once x is found"
               | (break) if v = x
               | return (-1) if i = length(V)
               | i ← i + 1 otherwise
               return i
```

$$\text{locate} \begin{bmatrix} 1, \\ 3 \\ 5 \end{bmatrix} = -1$$

WARNING:
You cannot break up an if-otherwise pair using any line, even if it only includes a comment.

Notice the same program below without the comments:

```
locate(x, V) := i ← 1
               for v ∈ V
               | (break) if v = x
               | return (-1) if i = length(V)
               | i ← i + 1 otherwise
```

Instead of comments, you may also use overlapping text regions to describe your program.

When creating a program in Mathcad, the following are critical issues to consider:

- Use comments by inserting a pair of quotes.
- Use units, but make sure you use them consistently.
- Use built-in functions like rows and cols to make your programs more general and reusable.
- Use *on error* to trap errors and the *error* function to generate custom error messages.
- Use *trace error* to pinpoint problems in your programs and for added convenience in debugging.
- Use parentheses with *if* statements to keep the logic clear.
- Use local definitions at the top of your programs to allocate space for arrays and to allocate fewer arrays.
- Do not use := assignments inside programs. Use the local assignment operator instead.
- Do not use the local assignment operator to define functions inside of programs. Instead, define functions outside and pass them in as arguments, or call them as you normally would.
- Do not use the same name for loop indices and local variables within the same program.
- Do not type programming operators in by hand. Instead, use the Programming toolbar or keyboard shortcuts.
- Do not call components, such as Excel or File Read/Write from within a program.

Mathcad provides a variety of functions that are useful when creating programs and that allow for type checking of values. The list includes:

- rows(M) - Returns the number of rows in the array M.
- cols(M) - Returns the number of columns in the array M.
- IsArray(M) - Returns 1 if M is an array, 0 otherwise.
- IsScalar(M) - Returns 1 if M is a real or complex scalar, 0 otherwise.
- IsString(M) - Returns 1 if M is a string, 0 otherwise.
- length(v) - Returns the number of elements in a vector v.
- last(v) - Returns the index of the last element in vector v.
- mod(x,y) - Returns the remainder on dividing x by y. Useful in determining if a number is even or odd.

Functions like rows and cols allow the program to adapt to the arrays being passed in rather than hard coding values which will only work for arrays of a fixed dimension.

EDITING TIPS

Programs are built using the Add Line operator on the Programming toolbar. In general, Mathcad reads programs line-by-line, starting at the top. Each line contains a separate calculation. You have the ability to add lines to your program at any time, however, you must be aware of the blue editing lines.

If the blue vertical line is to the right, the new line will be added below.



If the blue vertical line is to the left, the new line will be added above.



To add a line below and outside of a program loop:

1. Select the entire loop with the editing lines.
2. Click Add Line from the Programming toolbar



NOTE:

Programming basics is covered in the "Mathcad Essentials" class, T2124. "Using Advanced Programming Techniques", T2161, is a class that covers more advanced tips and tricks in Mathcad. See www.tristar.com for more details.