

PROGRAMMING IN LABVIEW MATHSCRIPT-SIMPLE FOR LOOP EXERCISES*

Anthony Antonacci
Darryl Morrell

Based on *Programming in MATLAB-Simple For Loop Exercises*[†] by
Darryl Morrell

This work is produced by OpenStax-CNX and licensed under the
Creative Commons Attribution License 2.0[‡]

Abstract

This module provides several simple exercises designed to test and increase your understanding of for loops in LabVIEW MathScript.

1 Some For Loop Exercises

Exercise 1

(Solution on p. 4.)

Loop Indices

How many times will this program print "Hello World"?

```
for a=0:50
    disp('Hello World')
end
```

Exercise 2

(Solution on p. 4.)

Loop Indices II

How many times will this program print "Guten Tag Welt"?

```
for a=-1:-1:-50
    disp('Guten Tag Welt')
end
```

*Version 1.1: Aug 3, 2006 12:44 am +0000

[†]<http://cnx.org/content/m13276/1.3/>

[‡]<http://creativecommons.org/licenses/by/2.0/>

Exercise 3*(Solution on p. 4.)***Loop Indices III**

How many times will this program print "Bonjour Monde"?

```
for a=-1:1:-50
    disp('Bonjour Monde')
end
```

Exercise 4*(Solution on p. 4.)***Nested Loops**

How many times will this program print "Hola Mundo"?

```
for a=10:10:50
    for b=0:0.1:1
        disp('Hola Mundo')
    end
end
```

Exercise 5*(Solution on p. 4.)***A tricky loop**

What sequence of numbers will the following for loop print?

```
n = 10;
for j = 1:n
    n = n-1;
    j
end
```

Explain why this code does what it does.

Exercise 6*(Solution on p. 4.)***Nested Loops II**

What value will the following program print?

```
count = 0;
for d = 1:7
    for h = 1:24
        for m = 1:60
            for s = 1:60
                count = count + 1;
            end
        end
    end
end
end
count
```

What is a simpler way to achieve the same results?

Exercise 7*(Solution on p. 4.)***Multiple Hypotenuses**

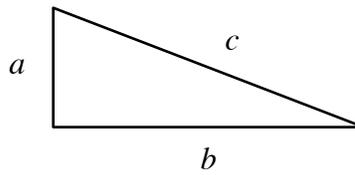


Figure 1: Sides of a right triangle.

Consider the right triangle shown in Figure 1. Suppose you wish to find the length of the hypotenuse c of this triangle for several combinations of side lengths a and b ; the specific combinations of a and b are given in Table 1: Side Lengths. Write a LABVIEW MATHSCRIPT's program to do this.

Side Lengths

a	b
1	1
1	2
2	3
4	1
2	2

Table 1

Solutions to Exercises in this Module

Solution to Exercise (p. 1)

The code `0:50` creates a vector of integers starting at 0 and going to 50; this vector has 51 elements. "Hello World" will be printed once for each element in the vector (51 times).

Solution to Exercise (p. 1)

The code `-1:-1:-50` creates a vector of integers starting at -1 and going backward to -50; this vector has 50 elements. "Guten Tag Welt" will be printed once for each element in the vector (50 times).

Solution to Exercise (p. 2)

The code `-1:1:-50` creates an empty vector with no elements. Since this is an empty vector the following error occurs: "Error in function range at line 1. You cannot specify a step size of zero for a range."

Solution to Exercise (p. 2)

The outer loop (the loop with `a`) will be executed five times. Each time the outer loop is executed, the inner loop (the loop with `b`) will be executed eleven times, since `0:0.1:1` creates a vector with 11 elements. "Hola Mundo" will be printed 55 times.

Solution to Exercise (p. 2)

In the first line, the value of `n` is set to 10. The code `1:n` creates a vector of integers from 1 to 10. Each iteration through the loop sets `j` to the next element of this vector, so `j` will be sent to each value 1 through 10 in succession, and this sequence of values will be printed. Note that each time through the loop, the value of `n` is decreased by 1; the final value of `n` will be 0. Even though the value of `n` is changed in the loop, the number of iterations through the loop is not affected, because the vector of integers is computed once before the loop is executed and does not depend on subsequent values of `n`.

Solution to Exercise (p. 2)

The `d` loop will be executed seven times. In each iteration of the `d` loop, the `h` loop will be executed 24 times. In each iteration of the `h` loop, the `m` loop will be executed 60 times. In each iteration of the `m` loop, the `s` loop will be executed 60 times. So the variable `count` will be incremented $7 \times 24 \times 60 \times 60 = 604800$ times.

A simpler way to achieve the same results is the command

```
7*24*60*60
```

Solution to Exercise (p. 2)

This solution was created by Heidi Zipperian:

```
a=[1 1 2 4 2]
b=[1 2 3 1 2]
for j=1:5
    c=sqrt(a(j)^2+b(j)^2)
end
```

A solution that does not use a for loop was also created by Heidi:

```
a=[1 1 2 4 2]
b=[1 2 3 1 2]
c=sqrt(a.^2+b.^2)
```