Auto Match VI

Goal

Use a While Loop and an iteration terminal and pass data through a tunnel.

Description

Create a VI that continuously generates random numbers between 0 and 1000 until it generates a number that matches a number selected by the user. Determine how many random numbers the VI generated before the matching number.

Design

Table 1. Inputs and Outputs

Type	Name	Properties
Input	`Number to Match`	Double-precision, floating-point between 0 and 1000, coerce to nearest whole number, default value = 50
Output	`Current Number`	Double-precision, floating-point
Output	`# of Iterations`	Integer

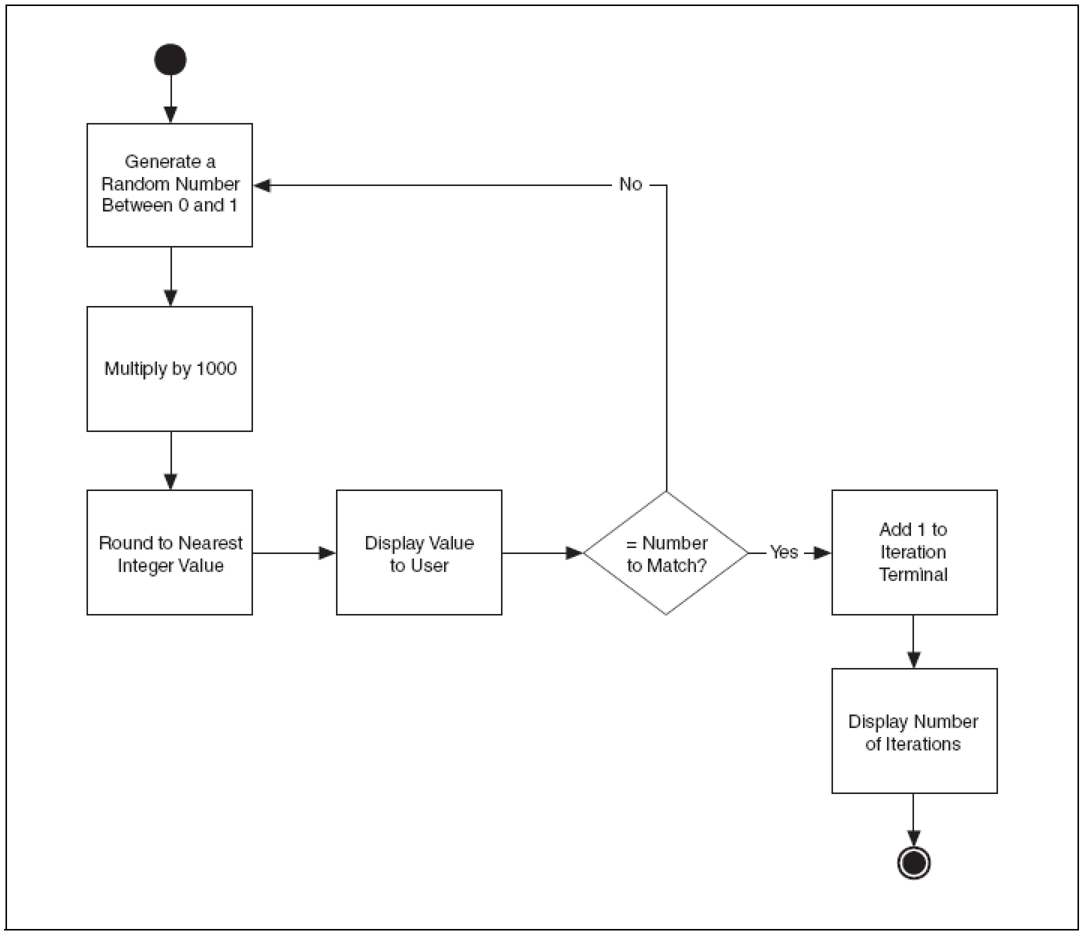
Generate a Random Number Between 0 and 1
Multiply by 1000
Round to Nearest Integer Value
Display Value to User
= Number to Match?
No
Yes
Add 1 to Iteration Terminal
Display Number of Iterations

Figure 1. Auto Match Flowchart

Implementation

The files that you need to complete this exercise are here: `<NI eLearning>\LV Core 1\While Loops Repeating the Code\Exercise`.

Build the following front panel and modify the controls and indicators as shown on the front panel in Figure 2 and described in the following steps.

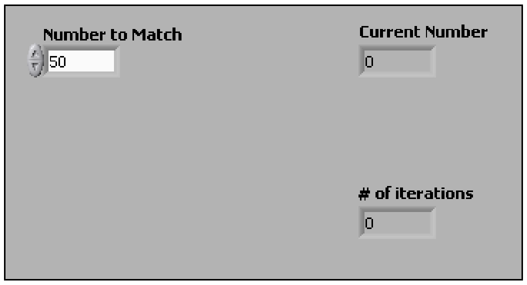

Figure 2. Auto Match VI Front Panel

1. Open a blank VI.
2. Save the VI as `Auto Match.vi` in the `<Exercise>` directory.
3. Create the **Number to Match** input.

   ❑ Add a numeric control to the front panel window.

   ❑ Label the control `Number to Match`.

4. Set the default value for the **Number to Match** control.

   ❑ Set the **Number to Match** control to `50`.

   ❑ Right-click the Number to Match control and select **Data Operations»Make Current Value Default**.

5. Set the properties for the **Number to Match** control so that the data range is from 0 to 1000, the increment value is 1, and the digits of precision is 0.

   ❑ Right-click the **Number to Match** control and select **Data Entry** from the shortcut menu. The Data Entry page of the Numeric Properties dialog box appears.

   ❑ Disable the **Use Default Limits** checkbox.

   ❑ Set the **Minimum** value to `0` and select **Coerce** from the **Response to value outside limits** pull-down menu.

   ❑ Set the **Maximum** value to `1000` and select **Coerce** from the **Response to value outside limits** pull-down menu.

   ❑ Set the **Increment** value to `1` and select **Coerce to Nearest** from the **Response to value outside limits** pull-down menu.

   ❑ Select the **Display Format** tab.

   ❑ Select **Floating Point** and change the Precision Type from **Significant digits** to **Digits of precision**.

   ❑ Enter `0` in the **Digits** text box and click the **OK** button.

6. Create the **Current Number** output.

   ❑ Add a numeric indicator to the front panel window.

   ❑ Label the indicator `Current Number`.

7. Set the digits of precision for the Current Number output to 0.

   ❑ Right-click the **Current Number** indicator and select **Display Format** from the shortcut menu. The Display Format page of the Numeric Properties dialog box appears.

   ❑ Select **Floating Point** and change the **Precision Type** to **Digits of precision**.

   ❑ Enter `0` in the Digits text box and click the **OK** button.

8. Create the **# of iterations** output.

   ❑ Place a numeric indicator on the front panel.

   ❑ Label the indicator `# of iterations`.

9. Set the representation for the **# of iterations** output to a long integer.

   ❑ Right-click the **# of iterations** indicator.

   ❑ Select **Representation»I32** from the shortcut menu.

Create the following block diagram. Refer to the following steps for instructions.

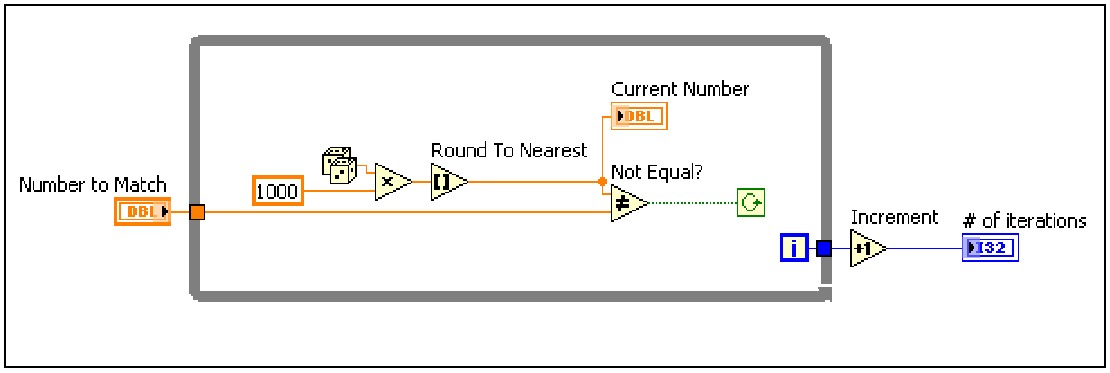

Figure 3. Auto Match VI Block Diagram

10. Generate a random number integer between 0 and 1000.

   ❑ Add the **Random Number (0-1)** function to the block diagram. The Random Number (0-1) function generates a random number between 0 and 1.

   ❑ Add the **Multiply** function to the block diagram. The Multiply function multiples the random number the y input to produce a random number between 0 and y.

❑ Wire the output of the Random Number function to the x input of the Multiply function.

❑ Right-click the y input of the Multiply function, select **Create» Constant** from the shortcut menu, enter `1000`, and press the `<Enter>` key to create a numeric constant.

❑ Add the **Round to Nearest** function to the block diagram. This function rounds the random number to the nearest integer.

❑ Wire the output of the Multiply function to the input of the Round To Nearest function.

❑ Wire the output of the Round To Nearest function to the Current Number indicator.

11. Compare the randomly generated number to the value in the Number to Match control.

❑ Add the **Not Equal?** function to the block diagram. This function compares the random number with the Number to Match and returns True if the numbers are not equal; otherwise, it returns False.

❑ Wire the output of the Round To Nearest function to the x input of the Not Equal? function.

12. Repeat the algorithm until the Not Equal? function returns False.

❑ Add a **While Loop** from the Structures palette to the block diagram.

❑ Right-click the conditional terminal and select **Continue if True** from the shortcut menu.

❑ Wire the Number to Match numeric control to the border of the While Loop. An orange tunnel appears on the While Loop border.

❑ Wire the orange tunnel to the y input of the Not Equal? function.

❑ Wire the output of the Not Equal? function to the conditional terminal.

13. Display the number of random numbers generated to the user by adding one to the iteration terminal value.

❑ Wire the iteration terminal to the border of the While Loop. A blue tunnel appears on the While Loop border.

Tip Each time the loop executes, the iteration terminal increments by one. You must wire the iteration value to the Increment function because the iteration count starts at 0. The iteration count passes out of the loop upon completion.

- ❑ Add the **Increment** function to the block diagram. This function adds 1 to the While Loop count.
- ❑ Wire the blue tunnel to the Increment function.
- ❑ Wire the Increment function to the # of iterations indicator.

14. Save the VI.

Test

1. Display the front panel.
2. Change the number in **Number to Match** to a number that is in the data range, which is 0 to 1000 with an increment of 1.
3. Right-click the Current Number indicator and select **Advanced» Synchronous Display**.

Note If synchronous display is enabled, then every time the block diagram sends a value to the Current Number indicator, the block diagram will stop executing until the front panel has updated the value of the indicator. In this exercise, you enable the synchronous display, so you can see the Current Number indicator get updated repeatedly on the front panel. Typically, the synchronous display is disabled to increase execution speed since you usually do not need to see every single updated value of an indicator on the front panel.

4. Run the VI.
5. Change the **Number to Match** and run the VI again. Current Number updates at ever iteration of the loop because it is inside the loop. # of iterations updates upon completion because it is outside the loop.
6. To see how the VI updates the indicators, enable execution highlighting.
   - ❑ On the block diagram toolbar, click the **Highlight Execution** button to enable execution highlighting. Execution highlighting shows the movement of data on the block diagram from one node to another so you can see each number as the VI generates it.
7. Run the VI and observe the data flow.

8. Try to match a number that is outside the data range.

   - ❑ Change **Number to Match** to a number that is out of the data range.
   - ❑ Run the VI. LabVIEW coerces the out-of-range value to the nearest value in the specified data range.

9. Close the VI.

End of Exercise

Notes