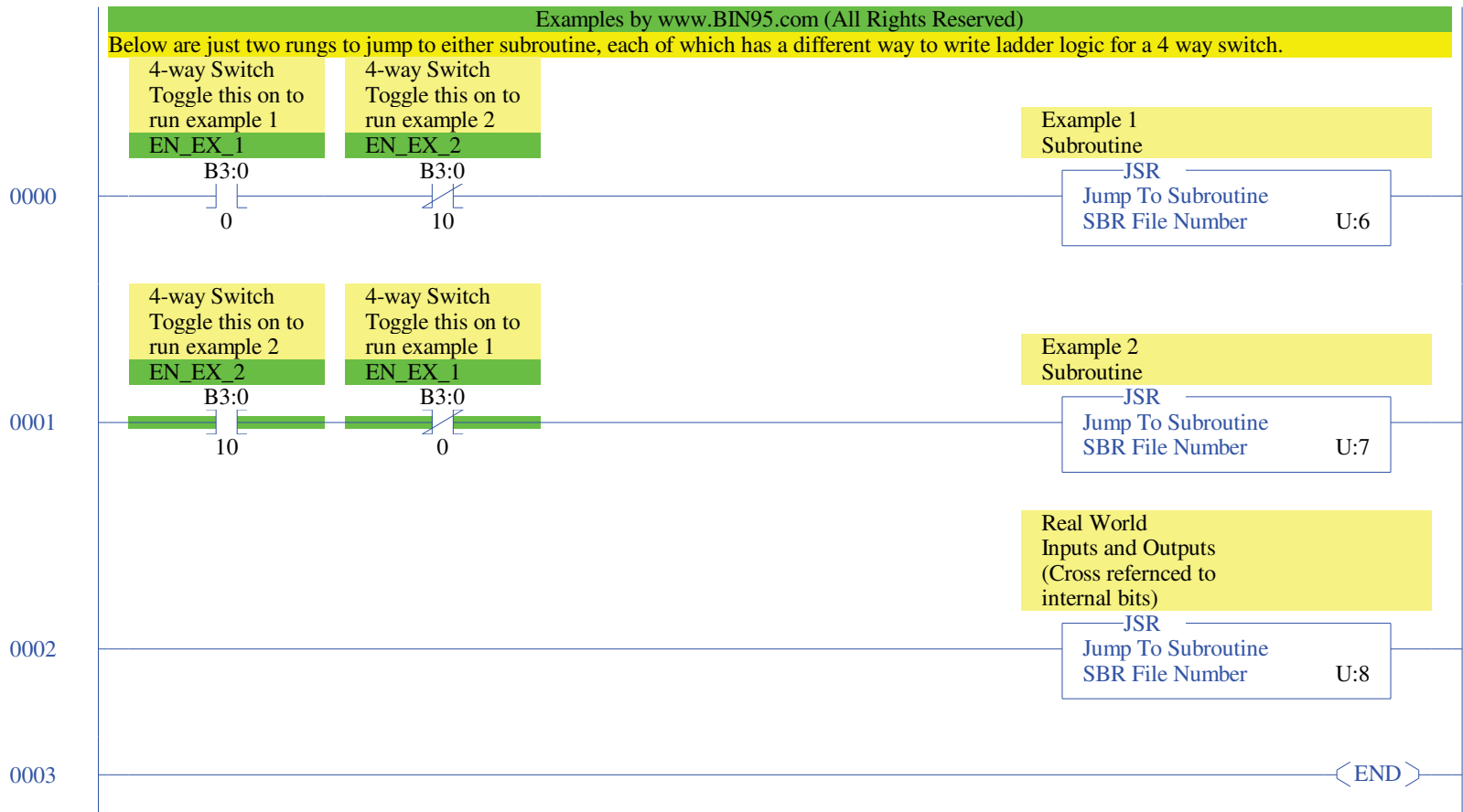


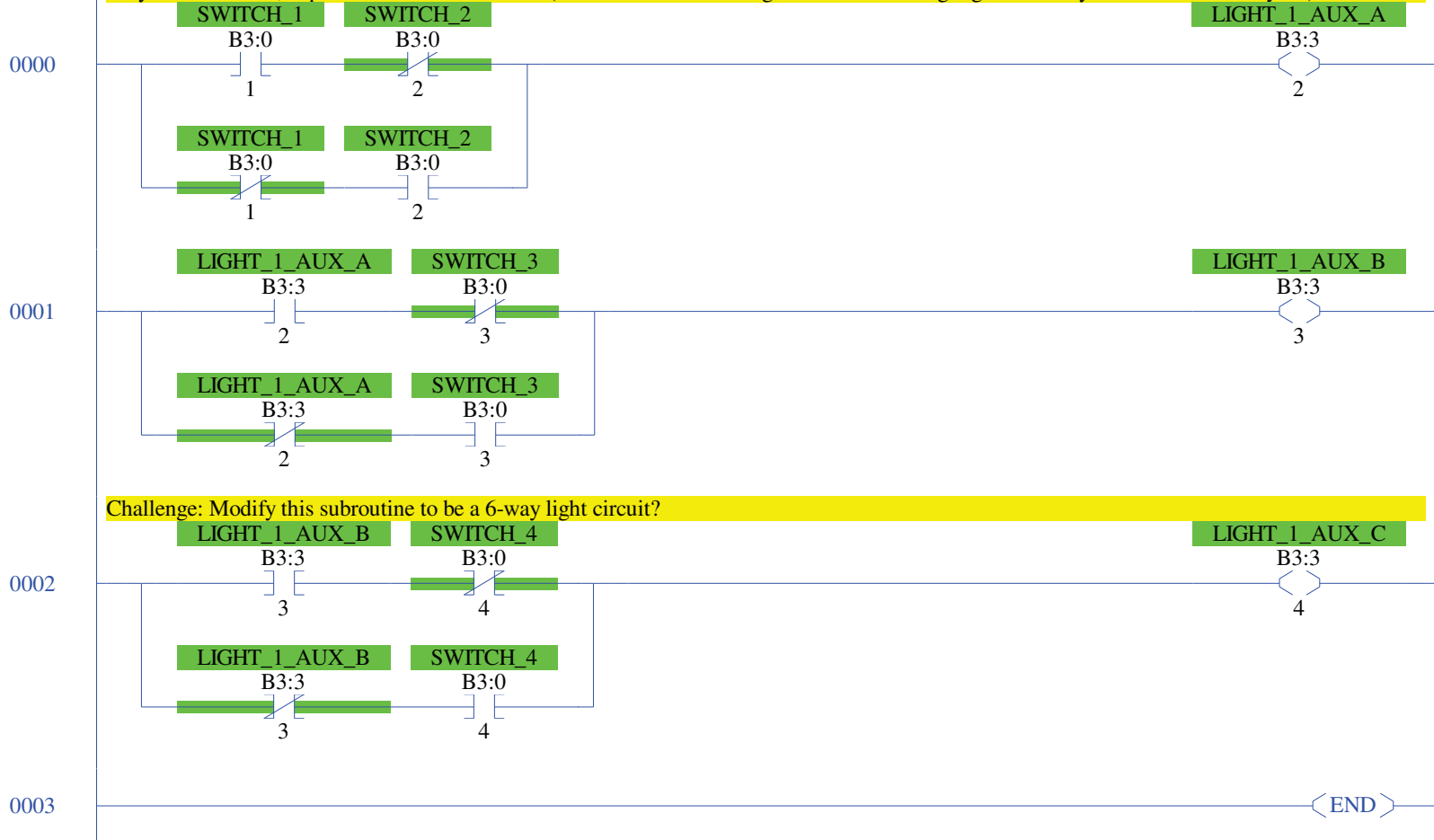
Examples by www.BIN95.com (All Rights Reserved)

Below are just two rungs to jump to either subroutine, each of which has a different way to write ladder logic for a 4 way switch.



BIN95.com Example 1 (Best Way)

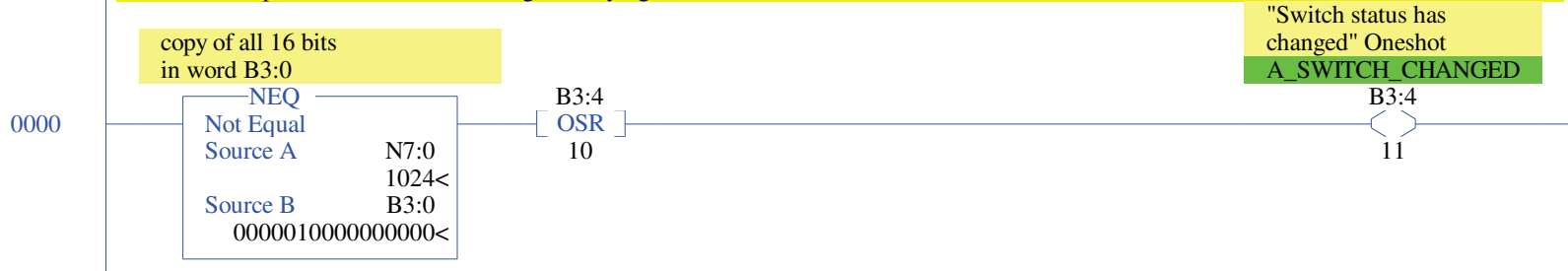
Rungs 0-3 below demonstrate the best way to write a 4-way light circuit.
 "BEST" being defined as "Easiest for the layman to understand, troubleshoot and modify."
 If you needed a 2-way light circuit, you put the real world output on rung zero, for 3-way place real world output on rung one, for 4-way we place real world output on rung 2. This way if someone needed a 5-way light circuit, they need only copy rung 2, create a LIGHT_1_AUX_D output, modify rung 3 with AUX_D.
 Easy to understand, expand on and troubleshoot. (the least amount of rungs method of writing logic is usually not the "BEST" way. :-)



Challenge: Modify this subroutine to be a 6-way light circuit?

BIN95.com Example 2 (NOT the Best Way)

This example may use less rungs, no matter how many switches you have, but is difficult for the layman to understand (assumes everyone is well versed in binary coding and data table change) and because it uses a one-shot, difficult to troubleshoot as change only occurs for one scan cycle. So this is the non-preferred method of writing a 4-way light circuit.



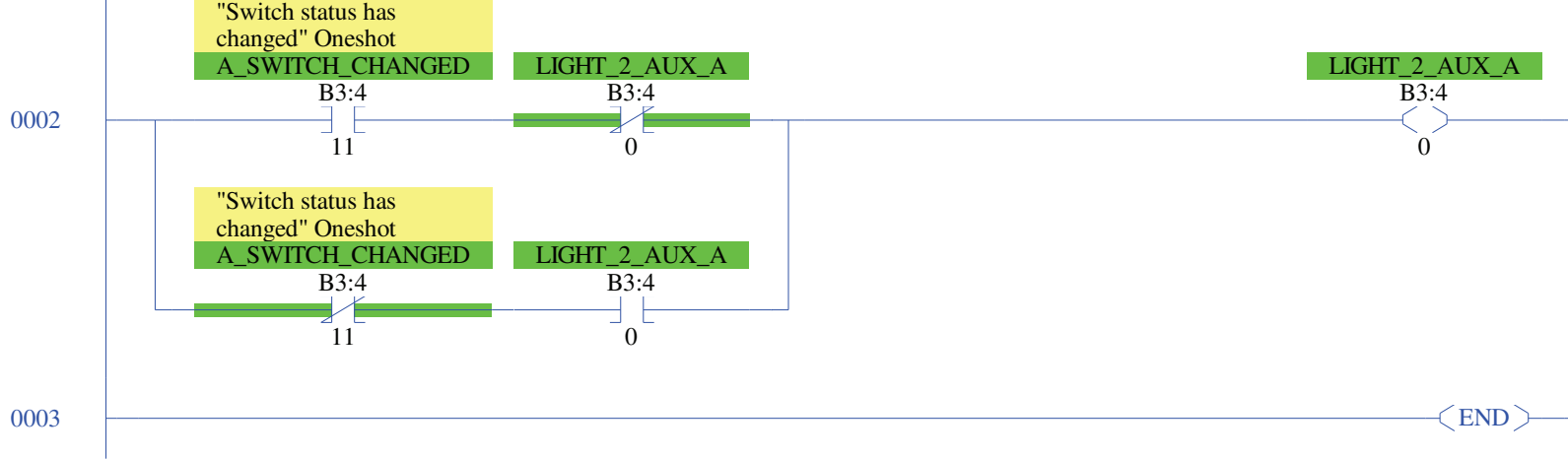
Note: Rung zero above would normally (logically) be switched with this rung 1, but we purposely switched the two rungs to capitalize on the scan cycle working with the One-Shot (which only fires for one scan cycle.)

Sequence of events...

PLC scans Real World Inputs and Rung 0 which now reads new switch pattern in word B:0. If the NEQ functions see a different pattern (binary number) in B:0 than it finds in OLD switch pattern N7:0 (recorded last scan cycle), Then the NEQ fires the One-Shot which in turn makes B3/75 turn on for one scan cycle. Rung 1 sees the true condition for one scan cycle and toggles the light output B3/64. (Toggles=if it's on, turn it off, and vice versa.)



Challenge: How do you make this an 8-way light circuit?



BIN95.com Example of XREF sub (An experienced programmer always uses these. :)

This is real world input and output cross reference subroutine. It should always be unconditionally scanned. It is a non-documented standard to place all your real world I/O in one subroutine so if you ever need to change the terminal a real world input or output is wired to, you only have to change one place in the program, instead of a dozen places in the program. It also decreases the scan time. But more importantly it decreases troubleshooting time, because you always start out in the real world with a symptom, work your way through the program, and wind up back out in the real world with what I/O is the cause of the problem. so this is one quick place to relate the real world with the program.

