Title: **Introduction to PLC Final** Test: Final

Course: Introduction to Automation Unit: Introduction to PLC CLO: 4

Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Grade \_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Objectives**

1. Student shall calculate the correct number conversion base from a number of a different number base system.
2. Student shall demonstrate their knowledge of introductory PLC principles.
3. Student shall draw the output of an instruction given its input(s).

**Assessment**

Students shall demonstrate a comprehension of the objectives listed above by scoring a minimum of 75% on this Test. Grading shall be based on the answer key.

**Instructions**

Select the best answer to each multiple-choice question below.

1. How many instructions can the MicroLogix 1100 PLC perform at any one time?
   1. 4
   2. 3
   3. 2
   4. 1
2. What does the three-letter acronym, PLC mean?
   1. Programmable Ladder Computer
   2. Proportional Logic Controller
   3. Programmable Logic Controller
   4. Programmable Ladder Controller
3. Between inputs and outputs, which items can actually be controlled?
   1. Inputs
   2. Outputs
   3. Both
   4. Neither
4. How are the inputs to a PLC protected?
   1. Through a fuse
   2. Through a resettable circuit breaker
   3. Through an LED and photo transistor
   4. Through a current limiting zener
5. What component is used to relay an output signal?
   1. A coil
   2. A dry contact
   3. An LED and photo transistor
   4. A current limiting Zener
6. A collection of \_\_\_\_\_ makes up a word.
   1. Places
   2. Binaries
   3. Bits
   4. Variables
7. A word has \_\_\_\_ bits.
   1. 1
   2. 4
   3. 8
   4. 16
8. The binary number system has how many digits?
   1. 1
   2. 2
   3. 8
   4. 10
9. Why is the binary number system used for motor control/ladder logic?
   1. It’s simple
   2. It has only two states
   3. It’s not used for motor control logic
   4. None of the above
10. List the order of operation of a PLC.
    1. Scan inputs, write outputs, evaluate ladder logic
    2. Evaluate each rung reading inputs and driving outputs that are in the rung.
    3. Scan input, evaluate ladder logic, write outputs
    4. Inputs, logic and outputs are all read/evaluated/written in parallel at the same time.
11. Input addresses start with the letter \_\_\_\_
12. Output addresses start with the letter \_\_\_\_
13. Internal memory bits start with the letter \_\_\_\_

Match the address type with its function

1. I \_\_\_\_\_ A. Sets a set of contacts to manipulate a final control element
2. O \_\_\_\_\_ B. Tied to the value of switches and contacts from the field
3. B \_\_\_\_\_ C. Stores a decision inside the PLC
4. Describe the problem with the ladder logic below.

|  |  |
| --- | --- |
|  | 1. When input I/0 is pressed, all the outputs are energized. 2. When input I/0 is pressed, outputs O/1 and O/2 are energized. 3. When input I/3 is pressed, all the outputs are energized. 4. O/0 will never be energized. |

1. A number that represents a whole number.
   1. Binary
   2. Integer
   3. Float
   4. Word
2. A binary number can be converted to an integer and vise-versa.
   1. True
   2. False
3. A number that can have a fractional component.
   1. Binary
   2. Integer
   3. Float
   4. Word
4. Integers can be negative numbers.
5. True
6. False
7. A byte has \_\_\_ bits and can represent an integer up to \_\_\_\_\_?
8. 8,255
9. 16, 65535
10. 4, 128
11. None of the above
12. Two \_\_\_\_\_\_\_ make a word?
13. bits
14. nibbles
15. bytes
16. None of the above
17. A \_\_\_\_\_\_\_ can be broken into four words?
18. nibble
19. bytes
20. DWORD
21. QWORD
22. This rung has rung continuity.

|  |  |
| --- | --- |
| * 1. True   2. False |  |

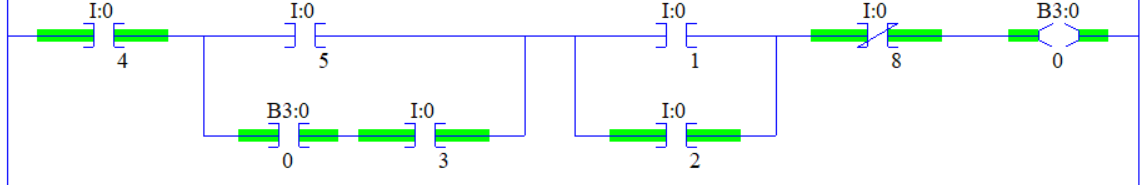
1. An XIC would be used for? (Choose the best answer)
2. Checking the status of a normally open contact
3. Testing a set of normally closed contacts
4. Checking to see if an address is true.
5. Is used to test for a 0
6. An XIO would be used for? (Choose the best answer)
7. Checking the status of a normally open contact
8. Testing a set of normally closed contacts
9. Checking to see if an address is true.
10. Is used to test for a 0
11. What does the instruction TON stand for?
12. Timer over network
13. Test for ON
14. On-Delay Timer
15. Table Output Numbers
16. Fill in the state of each timer bit (0 or 1).

|  |  |
| --- | --- |
|  | EN \_\_\_\_\_  TT \_\_\_\_\_  DN \_\_\_\_\_ |

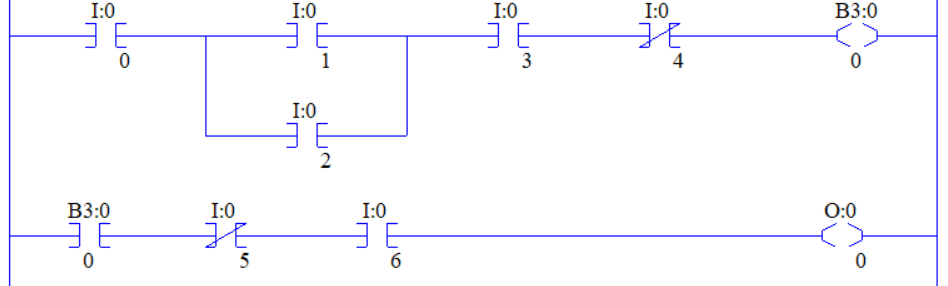
1. What does the instruction TOF stand for?
2. Timer over force
3. Off-Delay Timer
4. Test for OFF
5. Table Output Floats
6. Fill in the state of each timer bit (0 or 1).

|  |  |
| --- | --- |
|  | EN \_\_\_\_\_  TT \_\_\_\_\_  DN \_\_\_\_\_ |

1. If you want to transfer an integer value to another location, what instruction would you use?
2. CPY
3. MOV
4. TRNS
5. All the above
6. The number 123.456 is an example of an Integer.
7. True
8. False
9. The largest number that an Integer can be in the MicroLogix 1100 is 32,767.
10. True
11. False
12. Does this rung have logical continuity? Why or why not? Explain in detail.



1. An CTU would be used to?
2. Timing a specific set of contacts
3. Be a Clock timer unit
4. Counting up to a certain Preset
5. Counting down to a certain Preset
6. A CTD would be used to?
7. Each logical continuity true transition, decrements an integer by one
8. Subtract two numbers until they equal each other
9. Counting up to a certain Preset then setting the done bit
10. Is used to test for a 0 and sets the Accum value
11. What describes an Integer?
12. Any natural number
13. Any number that doesn’t have a fraction.
14. Is negative or positive
15. Both b and c
16. All the above
17. How does a number become negative inside the PLC?
18. The first bit indicates the sign
19. The MSB equaling 1
20. All the bits are 1
21. None of the above
22. Below is a printout of a program. The current states of each input can be seen in the table below the printout. What is the value of O:0/0?

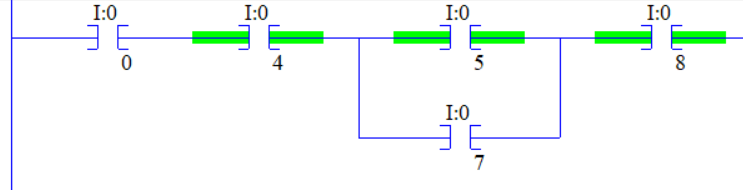




Convert the following to the indicated number base.



1. Word \_\_\_\_\_BCD
2. Byte \_\_\_\_\_16
3. Bit \_\_\_\_\_2
4. Nibble \_\_\_\_\_10
5. Does this rung have logical continuity? Why or why not? Explain in detail.



1. An EQU would be used to?
2. Test two numbers to see if they are the same
3. Set one number equal to another number
4. Write a word to an integer
5. All of the above
6. A GEQ would be used to?
7. Make sure a number is less than a specific value
8. Test for a number to be at least a given number
9. Test two number to see if they are not equal
10. None of the above
11. A OSF would be used to?
12. Check to see if a value is ON
13. Force an output value
14. Indicate that a value went to 0 this scan
15. Set if a number has been FALSE
16. A OTU would be used to?
17. Clear a bit value
18. Turn on a pilot light
19. Counting up to a certain Preset then setting the done bit
20. Is used to test for a 0
21. An ADD would be used to?
22. Set a register based on the sum of two numbers
23. Set a register based on the product of two numbers
24. Set a register based on the difference of two numbers
25. Set a register based on the dividend of two numbers
26. A SUB would be used to?
27. Set a register based on the sum of two numbers
28. Set a register based on the product of two numbers
29. Set a register based on the difference of two numbers
30. Set a register based on the dividend of two numbers
31. A DIV would be used to?
32. Set a register based on the sum of two numbers
33. Set a register based on the product of two numbers
34. Set a register based on the difference of two numbers
35. Set a register based on the dividend of two numbers
36. A MUL would be used to?
37. Set a register based on the sum of two numbers
38. Set a register based on the product of two numbers
39. Set a register based on the difference of two numbers
40. Set a register based on the dividend of two numbers

Draw the output to the following logic diagrams.















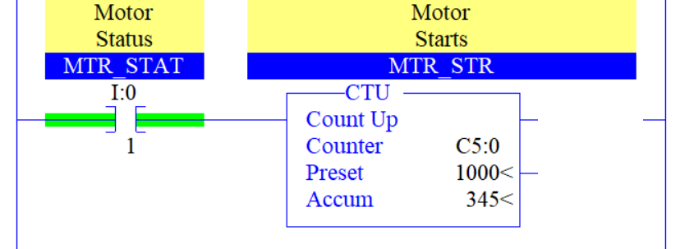




Calculate the following number to the new number base system given the value from another number base system.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. 000101102 \_\_\_\_\_\_\_10 2. 000010012 \_\_\_\_\_\_\_10 3. 4610 \_\_\_\_\_\_ \_\_\_\_\_\_2 4. 2210 \_\_\_\_\_\_ \_\_\_\_\_\_2 5. 368 \_\_\_\_\_\_10 6. 168 \_\_\_\_\_\_10 7. 3610 \_\_\_\_\_\_\_8 8. 1610 \_\_\_\_\_\_\_8 | |  |  |  |  |  | | --- | --- | --- | --- | --- | | Truth Table | | | | | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |  |  |  |  |  | |

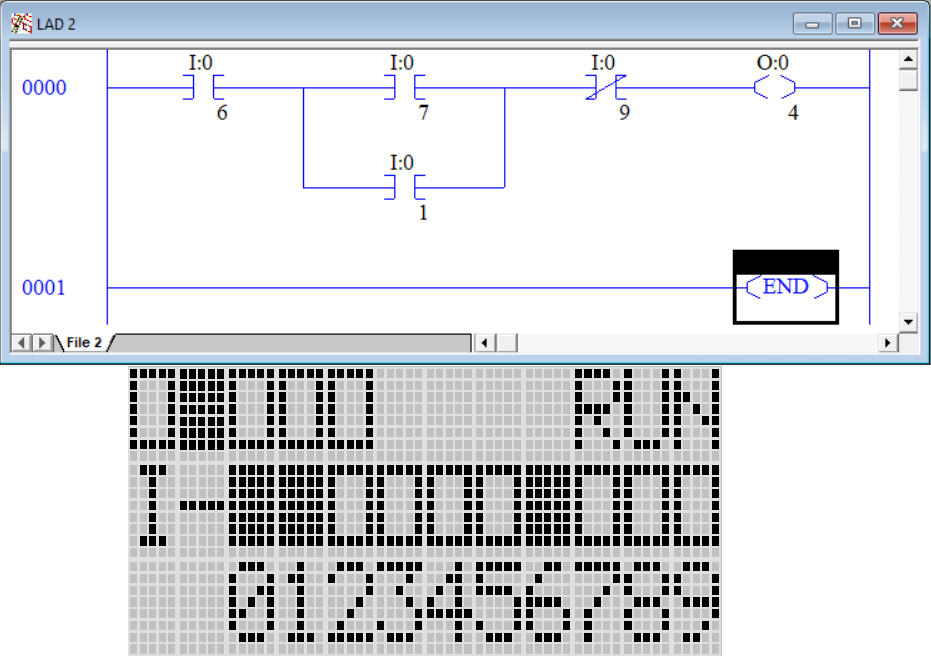
|  |  |
| --- | --- |
| 1. 2258 \_\_\_\_\_\_ \_\_\_\_\_\_2 | 1. 1678 \_\_\_\_\_\_ \_\_\_\_\_\_2 |
| 1. 010010112 \_\_\_\_\_\_8 | 1. 110010012 \_\_\_\_\_\_8 |
| 1. D2B416 \_\_\_\_ \_\_\_\_ \_\_\_\_ \_\_\_\_2 | 1. E17A16 \_\_\_\_ \_\_\_\_ \_\_\_\_ \_\_\_\_2 |
| 1. 1000 0101 1101 10102 \_\_\_\_\_\_16 | 1. 1110 1000 0000 10012 \_\_\_\_\_\_16 |
| 1. 1000 0101 1001 1000BCD \_\_\_\_10 | 1. 0001 0011 0000 0100BCD \_\_\_\_10 |
| 1. 271910 \_\_\_\_ \_\_\_\_ \_\_\_\_ \_\_\_\_BCD | 1. 768110 \_\_\_\_ \_\_\_\_ \_\_\_\_ \_\_\_\_BCD |

1. Determine the state of the output bits.

CU \_\_\_\_

DN \_\_\_\_

OV \_\_\_\_

1. Determine the state of the output of this **offline** program given the running PLC status screen.
   1. True
   2. False