VCU College of Engineering

Can you understand Ladder Logic? A Survey

CAN YOU UNDERSTAND LADDER LOGIC?

This survey aims to evaluate the readability and writability of Ladder Logic, a programming language used in the development of software for Programmable Logic Controllers (PLCs). It is part of a study made by researchers from the College of Engineering at the Virginia Commonwealth University (VCU), and it is divided into three sections:

Section 1: A short video will introduce you to the Ladder Logic programming language, addressing basic concepts and examples of this technology. Respondents without prior experience in Ladder Logic can use this video as a preparation to answer the following sections.

Section 2: Ten multiple-choice questions containing different Ladder Logic problems will be presented for you in Section 2. You will be asked to select for each problem a diagram or a short description of a diagram that correctly answers each question.

Section 3: In the last section, we would like to ask you to evaluate the usability of the Ladder Logic programming language using a matrix rating scale. You don't need to have previous experience in Ladder Logic to answer this or any other questions of this survey.

Interested in participating?

Here is some important information before starting:

- The estimated time to complete this survey is **less than 30 minutes**.
- Your contribution is voluntary and your answers will be used to enhance the quality of our study.
- No personal data will be collected and all your responses will be stored anonymously.

If you have any questions, feel free to contact us:

- David C. Shepherd, shepherdd@vcu.edu, 804-873-4270
- Felipe Fronchetti, fronchettl@vcu.edu, 804-298-0694

Please, to determine your eligibility for this study, answer the questions below:

Please select the role most applicable to you:

○ Engineer ○ Technician ○ Software Developer

○ Researcher ○ Engineering Student

○ Other (Please specify)

How many years of previous programming experience do you have?

○ No programming experience ○ < 1 Year ○ 1-2 Years

○ 3-5 Years ○ 5+ Years

Do you have any prior experience with Programmable Logic Controllers?

○ No experience ○ < 1 Year ○ 1-2 Years

○ 3-5 Years ○ 5+ Years

Which programming languages do you use to write code for Programmable Logic Controllers?

□ Ladder diagram (LD) □ Function block diagram (FBD) □ Structured text (ST)

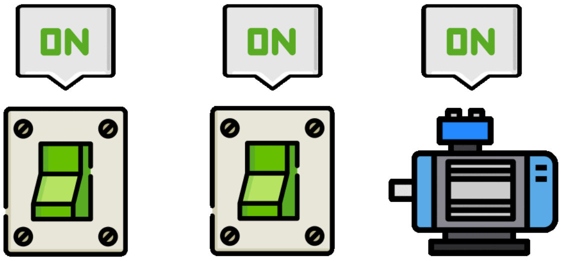
□ Instruction list (IL) □ Sequential function chart (SFC)

□ Other (Please specify)

The basic understanding of Ladder Logic will be necessary to answer the questions of the following sections. If you already know Ladder Logic, feel free to skip this section by clicking on the Next button. If you don't have any prior experience on Ladder Logic, please watch the short tutorial below to better comprehend how these technologies work:

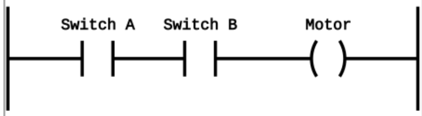
Click on the Next button when you feel comfortable to continue.

* Suppose the following automation task containing two switches and one motor, where to turn the motor ON, both switches must be ON. The possible states for the elements of this task can be described by the table and illustration below:

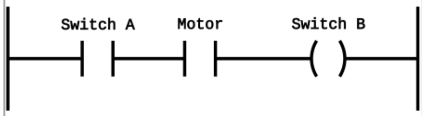


Switch A	Switch B	Motor
ON	ON	ON
ON	OFF	OFF
OFF	ON	OFF
OFF	OFF	OFF

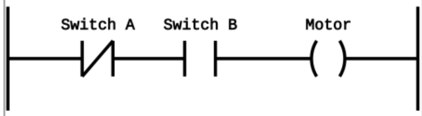
Using your understanding of Ladder Logic, select the diagram that correctly solves this task:


Switch A
Switch B
Motor

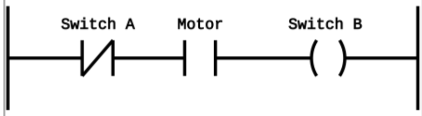
○


Switch A
Motor
Switch B

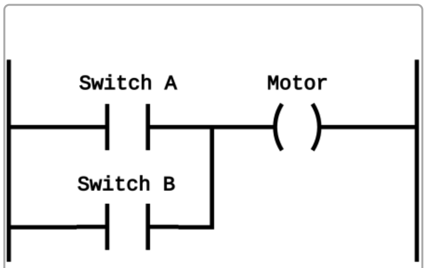
○


Switch A
Switch B
Motor

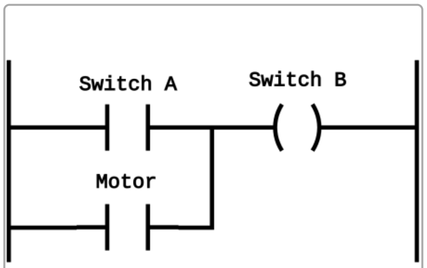
○


Switch A
Motor
Switch B

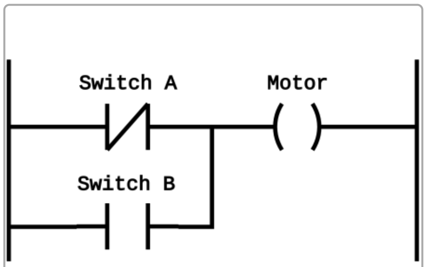
○


Switch A
Motor
Switch B

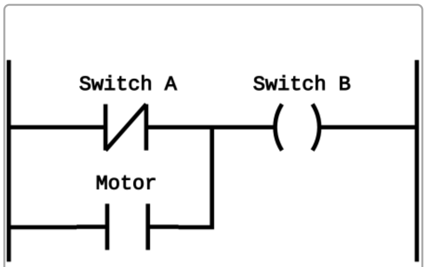
○


Switch A
Switch B
Motor

○

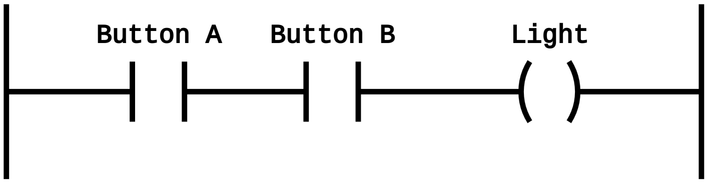

Switch A
Motor
Switch B

○


Switch A
Switch B
Motor

○

* Using your understanding of Ladder Logic, look at the diagram below and select the answer that correctly describes this diagram:



○ Light will only be turned ON when both Button A and Button B are ON.

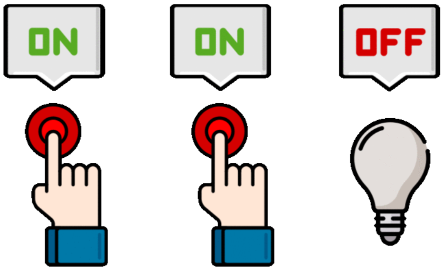
○ Light will be turned ON when at least Button A or Button B is ON.

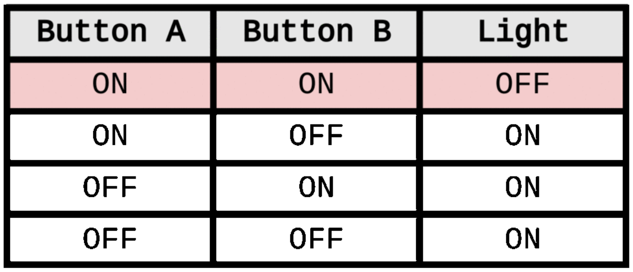
○ Light will only be turned ON when both Button A and Button B are OFF.

○ Light will be turned ON when at least Button A or Button B is OFF.

○ Light will never be turned ON, regardless of the state of Button A and Button B.

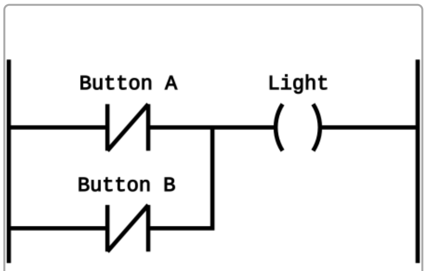
* Suppose the following automation task containing two buttons and one light bulb, where to turn the light bulb ON, at least one button must be OFF. The possible states for the elements of this task can be described by the table and illustration below:



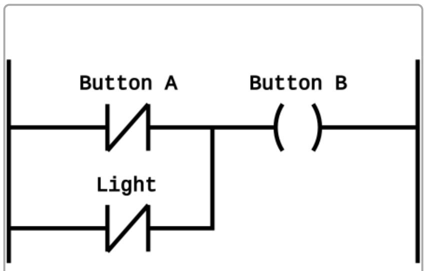


Button A	Button B	Light
ON	ON	OFF
ON	OFF	ON
OFF	ON	ON
OFF	OFF	ON

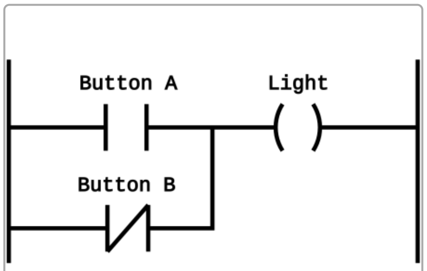
Using your understanding of Ladder Logic, select the diagram that correctly solves this task:


Button A
Light
Button B

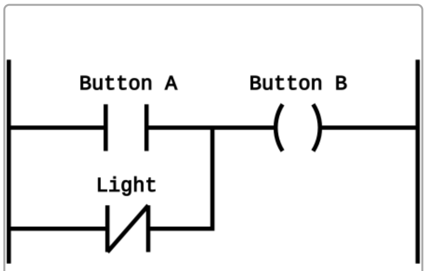
○


Button A
Button B
Light

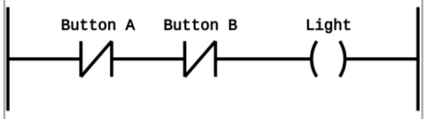
○


Button A
Light
Button B

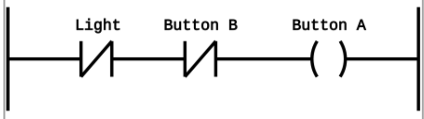
○


Button A
Button B
Light

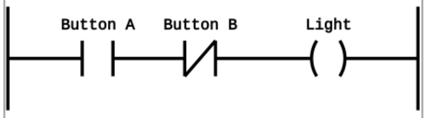
○


Button A
Button B
Light

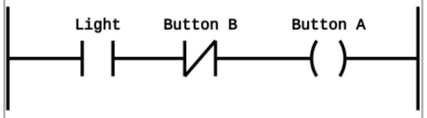
○


Light
Button B
Button A

○

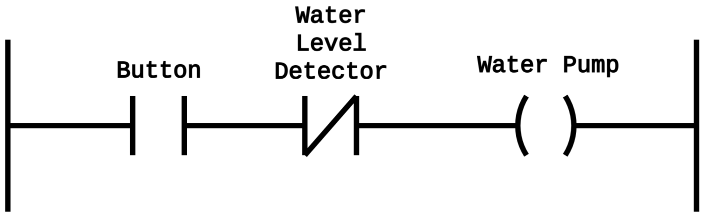

Button A
Button B
Light

○


Light
Button B
Button A

○

* Using your understanding of Ladder Logic, look at the diagram below and select the answer that correctly describes this diagram:



○ Water Pump will be turned ON when the Button is ON and the Water Level Detector is OFF.

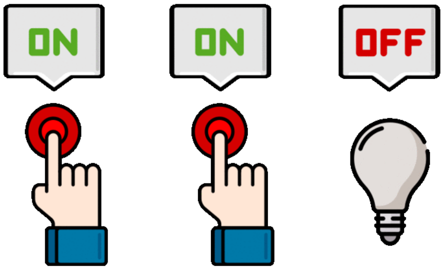
○ Water Pump will be turned ON when the Button is OFF and the Water Level Detector is ON.

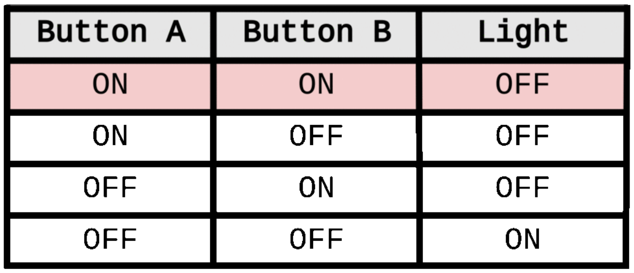
○ Water Pump will be turned ON when the Button is ON or the Water Level Detector is ON.

○ Water Pump will be turned ON when the Button is ON or the Water Level Detector is OFF.

○ Water Pump will never be turned ON, regardless the state of the input elements (Button and Water Level Detector).

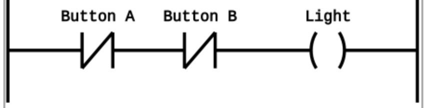
*Suppose the following automation task containing two buttons and one light bulb, where to turn the light ON both buttons must be OFF. The possible states for the elements of this task can be described by the table and illustration below:



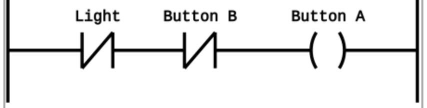


Button A	Button B	Light
ON	ON	OFF
ON	OFF	OFF
OFF	ON	OFF
OFF	OFF	ON

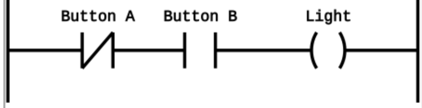
Using your understanding of Ladder Logic, select the diagram that correctly solves this task:


Button A
Button B
Light

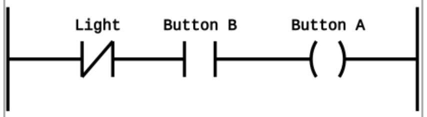
○


Light
Button B
Button A

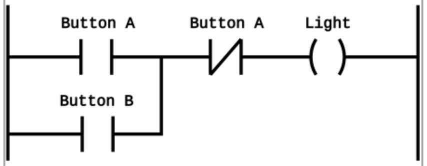
○


Button A
Button B
Light

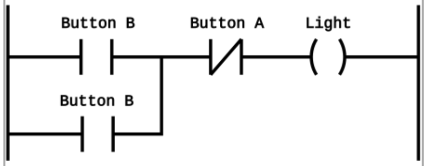
○


Light
Button B
Button A

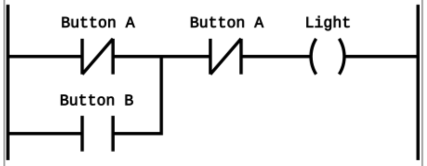
○


Button A
Button A
Light
Button B

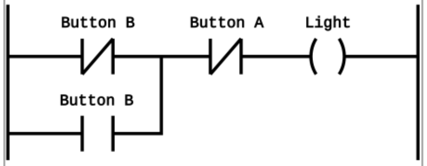
○


Button B
Button A
Light
Button B

○

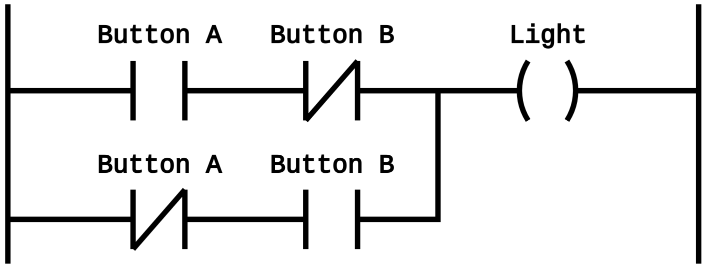

Button A
Button A
Light
Button B

○


Button B
Button A
Light
Button B

○

* Using your understanding of Ladder Logic, look at the diagram below and select the answer that correctly describes this diagram:



○ Light will be turned ON when Button A and Button B have the same state (ON and ON or OFF and OFF)

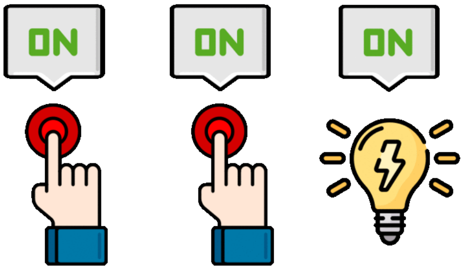
○ Light will be turned ON when Button A and Button B have different states (ON and OFF or OFF and ON)

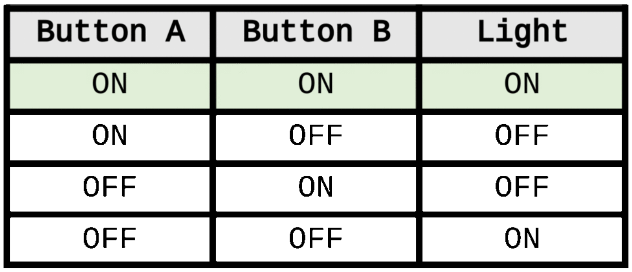
○ Light will be turned ON only when Button A and Button B are OFF.

○ Light will be turned ON only when Button A is ON and button B is OFF.

○ Light will never be turned ON, regardless the state of the input elements (Button A and Button B).

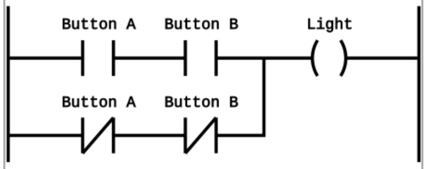
* Suppose the following automation task containing two buttons and one light bulb, where for the light to be ON, the two buttons must keep in the same state. The possible states for the elements of this task can be described by the table and illustration below:



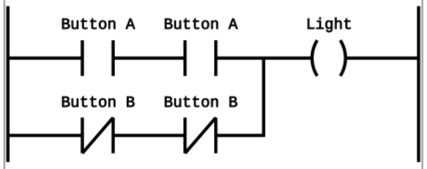


Button A	Button B	Light
ON	ON	ON
ON	OFF	OFF
OFF	ON	OFF
OFF	OFF	ON

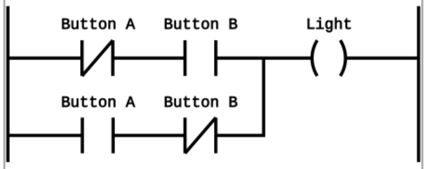
Using your understanding of Ladder Logic, select the diagram that correctly solves this task:


Button A
Button B
Light
Button A
Button B

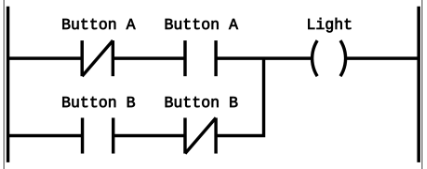
○


Button A
Button A
Light
Button B
Button B

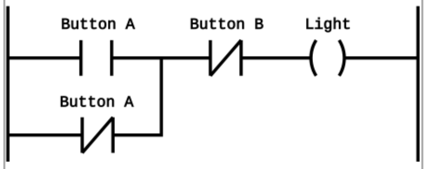
○


Button A
Button B
Light
Button A
Button B

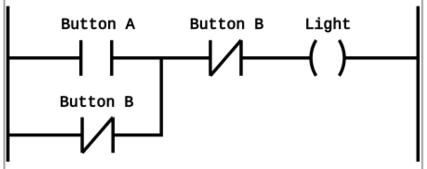
○


Button A
Button A
Light
Button B
Button B

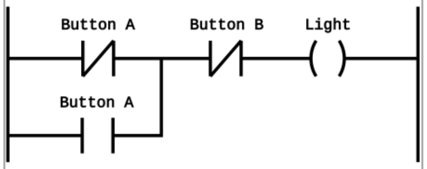
○


Button A
Button B
Light
Button A

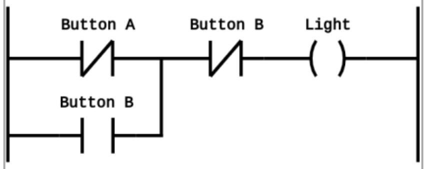
○


Button A
Button B
Light
Button B

○

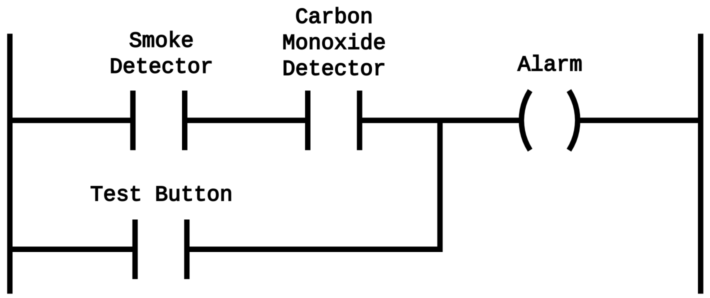

Button A
Button B
Light
Button A

○


Button A
Button B
Light
Button B

○

* Using your understanding of Ladder Logic, look at the diagram below and select the answer that correctly describes this diagram:



○ Alarm will be turned ON when both the Smoke and Carbon Monoxide detectors are ON, or when the Test Button is ON.

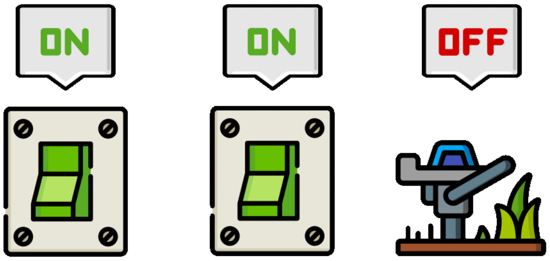
○ Alarm will be turned ON when at least one of the following elements is ON: Smoke Detector, Carbon Monoxide Detector or Test Button.

○ Alarm will be turned ON when both the Smoke and Carbon Monoxide detectors are ON, and the Test Button is OFF.

○ Alarm will only be turned ON when the three input elements are OFF at the same time: Smoke Detector, Carbon Monoxide Detector and Test Button.

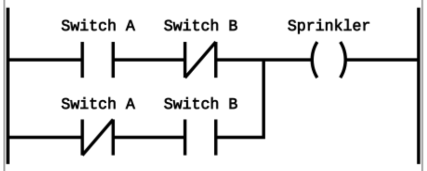
○ Alarm will never be turned ON, regardless the state of the input elements (Smoke Detector, Carbon Monoxide Detector or Test Button).

*Suppose the following automation task containing two switches and one sprinkler, where to turn the sprinkler ON, the two switches must be in opposite states. In other words, while one switch is ON, the other must be OFF to activate the sprinkler. The possible states for the elements of this task can be described by the table and illustration below:

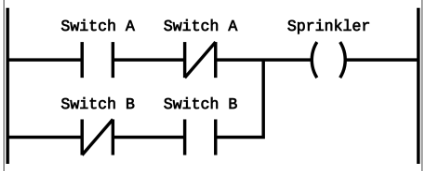


Switch A	Switch B	Sprinkler
ON	ON	OFF
ON	OFF	ON
OFF	ON	ON
OFF	OFF	OFF

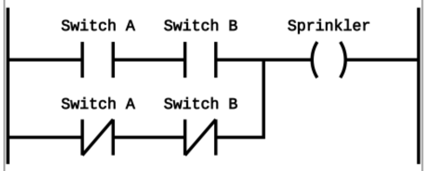
Using your understanding of Ladder Logic, select the diagram that correctly solves this task:


Switch A
Switch B
Sprinkler
Switch A
Switch B

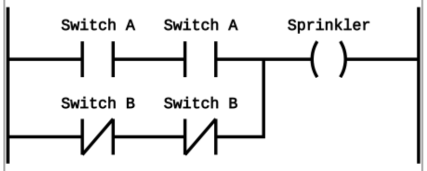
○


Switch A
Switch A
Sprinkler
Switch B
Switch B

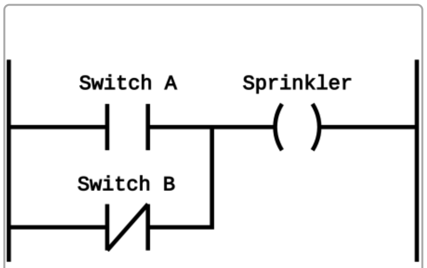
○


Switch A
Switch B
Sprinkler
Switch A
Switch B

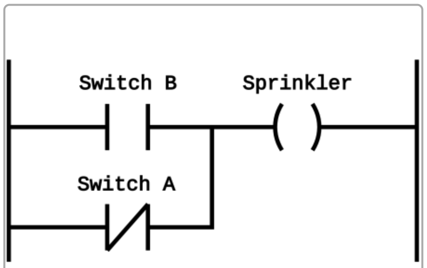
○


Switch A
Switch A
Sprinkler
Switch B
Switch B

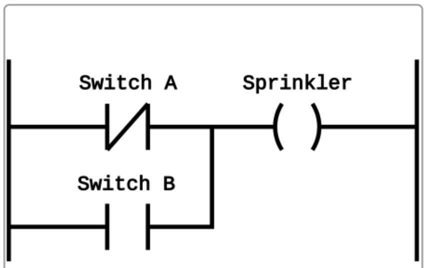
○


Switch A
Sprinkler
Switch B

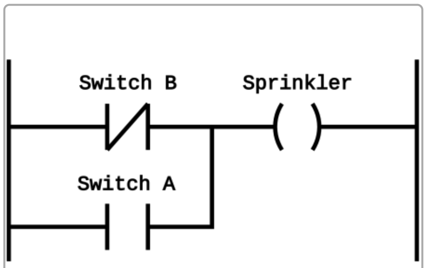
○


Switch B
Sprinkler
Switch A

○

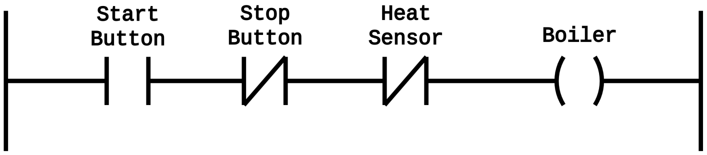

Switch A
Sprinkler
Switch B

○


Switch B
Sprinkler
Switch A

○

* Using your understanding of Ladder Logic, look at the diagram below and select the answer that correctly describes this diagram:



○ Boiler will be turned ON when at least one of the input elements is ON: Start Button, Stop Button or Heat Sensor.

○ Boiler will be turned ON when the Start Button and Stop Button are OFF, or when the Heat Sensor is ON.

○ Boiler will be turned ON when the Start Button is ON, and the Stop Button and Heat Sensor are OFF.

○ Boiler will be turned ON when the Start Button is OFF, and the Stop Button and Heat Sensor are OFF.

○ Boiler will never be turned ON, regardless the state of the input elements (Start Button, Stop Button and Heat Sensor).

In this section, we would like to ask your opinion about the overall simplicity of Ladder Logic. It is valid to remember that Ladder Logic is a programming language used to write software for Programmable Logic Controllers and its development occurs through the usage of diagrams such as the following example:

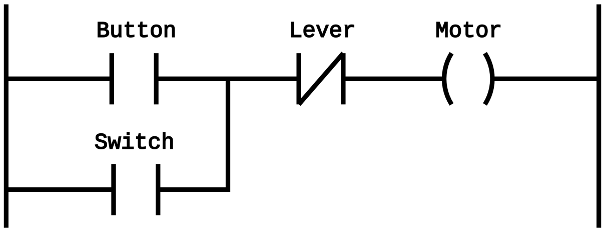


Figure 1. Example of a diagram in Ladder Logic

For each one of the following statements, mark the box that best describes your reactions to the Ladder Logic programming language:

	Strongly Disagree				Strongly Agree
I think that I would like to use Ladder Logic frequently.	1	2	3	4	5
I found Ladder Logic unnecessarily complex.	1	2	3	4	5
I thought that Ladder Logic was easy to use.	1	2	3	4	5
I think that I would need assistance to be able to use Ladder Logic.	1	2	3	4	5
I found the various functions in Ladder Logic were well integrated.	1	2	3	4	5
I thought there was too much inconsistency in Ladder Logic.	1	2	3	4	5
I would imagine that most people would learn to use Ladder Logic very quickly.	1	2	3	4	5
I found Ladder Logic very cumbersome/awkward to use.	1	2	3	4	5

I felt very confident using Ladder Logic.	1	2	3	4	5
I needed to learn a lot of things before I could get going with Ladder Logic.	1	2	3	4	5

What is your opinion about Ladder Logic?

Feel free to provide any thoughts about this language or click on the 'Submit' button to submit your answer.