CLAD Preparation Guide Using LabVIEW

This preparation guide prepares you to take the CLAD exam using LabVIEW.

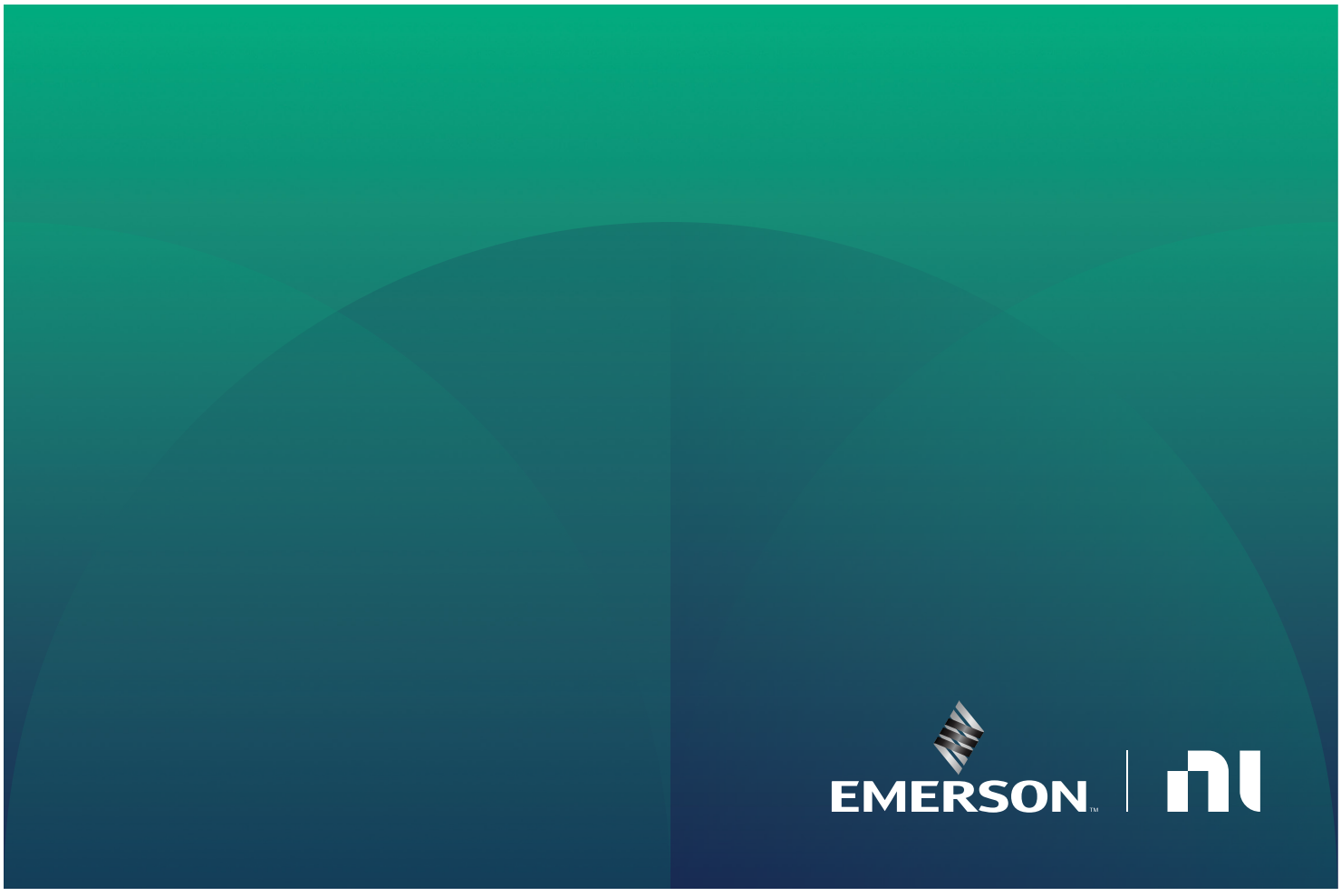


Table of Contents

Introduction: The Scope and Basis of the CLAD Exam 2
Preparation Resources 2
The Logistics of the CLAD Exam 2
Format 2
Scoring 2
Scheduling and Taking the Exam 2
KSAs: Knowledge, Skills and Abilities tested by the CLAD 3
Topics List 4
Hardware (10% of exam questions) 4
LabVIEW Programming Environment (25% of exam questions) 4
LabVIEW Programming Fundamentals (50% of exam questions) 4
Programming Best Practices (15% of exam questions) 6
Example Questions & Resources 6
Note on Using these Example Questions 6
Hardware 6
LabVIEW Programming Environment 13
LabVIEW Programming Fundamentals 18
Programming Best Practices 28
APPENDIX I: SIMULATED DAQ FOR CLAD PREP 32
Create Simulated Hardware 32
DAQmx VIs Included in CLAD Exam 33
APPENDIX II: CORRECT RESPONSES FOR EXAMPLE QUESTIONS 34

Introduction: The Scope and Basis of the CLAD Exam

The Certified LabVIEW Associated Developer (CLAD) is the first level of professional certification for LabVIEW programming. The certification is anchored in three common areas where LabVIEW is used on the job:

- Automated test
- High-channel-count data acquisition
- Measurement and data logging for domain experts

The CLAD represents a level of mastery of LabVIEW at which a person with minimal oversight can use LabVIEW to do the following:

- Acquire and interpret data
- Create small VIs
- Edit medium-sized VIs
- Contribute elements to large VIs and projects

Preparation Resources

The best preparation for this exam is LabVIEW programming experience applying the concepts presented in LabVIEW Core 1 and Core 2 courses. Class attendance alone will not be sufficient. Typical CLAD candidates have taken NI LabVIEW Core 1 and Core 2 courses (or equivalent) and have at least 6 to 9 months of experience using LabVIEW.

This prep guide provides an overview of the exam, including exam logistics and coverage. The relevant knowledge, skills and abilities (KSAs) tested by the CLAD exam are listed in the KSAs: Knowledge, Skills and Abilities tested by the CLAD section. A detailed Topic List table follows the KSA descriptions.

This prep guide also includes example questions. **This is not a sample exam.** The questions are included as a study aid. They reflect the scope of the CLAD exam and familiarize the test-taker with NI's approach to testing the KSAs relevant to a Certified LabVIEW Associate Developer.

The Logistics of the CLAD Exam

Format

The CLAD Exam is a 40-question multiple-choice exam. For each question, you will choose only one of the four options presented. If more than one answer appears correct, choose the best one. That is, the response which either provides an additional benefit or prevents problems that other responses might create.

Scoring

A score of 70% or higher is required to pass the exam.

Scheduling and Taking the Exam

To schedule the exam, complete the following steps:

1. Click **Products**, and under **SERVICES**, select **Training Courses** on `ni.com`.
2. In the **Certification Program** section, under **Certifications**, click **Learn more**.
3. From the **Available Certifications** section, select **Certified LabVIEW Associate Developer (CLAD)**.
4. In the **Schedule Your Exam** section, choose your preferred registration language.
5. Create your Pearson account, or sign in if you already have one.

Note: Make sure to use the same email address associated with your `ni.com` profile.
You can find additional helpful resources for test-takers here.

You cannot use LabVIEW or any other resources during the exam. The CLAD exam provides screenshots from the LabVIEW environment and LabVIEW Help where appropriate. Refer to the Example Questions & Resources section for examples.

To take the exam, you must agree to a Non-Disclosure Agreement (NDA). The NDA states that you will not copy, reproduce, or communicate any section of the exam in any form, including verbal or electronic. Failure to comply with the NDA will result in penalties ranging from a failed exam to a lifetime ban from LabVIEW Certification.

KSAs: Knowledge, Skills and Abilities tested by the CLAD

The CLAD exam centers around jobs requiring programming in professional settings, including Automated Test, High-Channel-Count Data Acquisition, or Domain Support. Most of these jobs include collecting and/or processing data in the form of signals from sensors. To verify the relevant knowledge, skills and abilities, the CLAD uses an NI-DAQmx system as representative hardware. NI-DAQmx was chosen because it can be simulated on all versions of LabVIEW. If you have access to LabVIEW, you have the resources to prepare for the exam. You do not need to purchase specific hardware.

Appendix I of the document provides instructions for setting up simulated hardware sufficient for exam preparation. The appendix also includes a list of the DAQmx functions possibly used in the exam. The exam does not test DAQmx function-specific settings. The questions use NI-DAQmx to test knowledge and skills necessary for common data acquisition tasks, such as calculating sample rates, determining the correct order of operations, and programming basic file I/O.

A person using LabVIEW at the Associated Developer level will be able to:

- Use software architectures from a single VI to a simple State Machine or Event-Driven UI Handler.
- Collect data from sensors using NI Hardware.
- Use Array functions extensively to extract and manipulate a single channel of data from multiple-channel data represented by a 1D waveform array or a 2D numeric array.
- Use loops to run a test a set number of times or until a condition is met, to establish a voltage ramp, or conduct other repeated tasks.
- Create and modify SubVIs, clusters, and Type Defs to simplify their code and contribute to larger projects.

Topics List

A certified LabVIEW Associate Developer should be familiar with the following topics.

Hardware (10% of exam questions)

- Connecting Hardware: Sensors, DAQ, Devices under test (DUT).
- Acquiring and validating a signal
- Processing signals
- Using appropriate sample rates

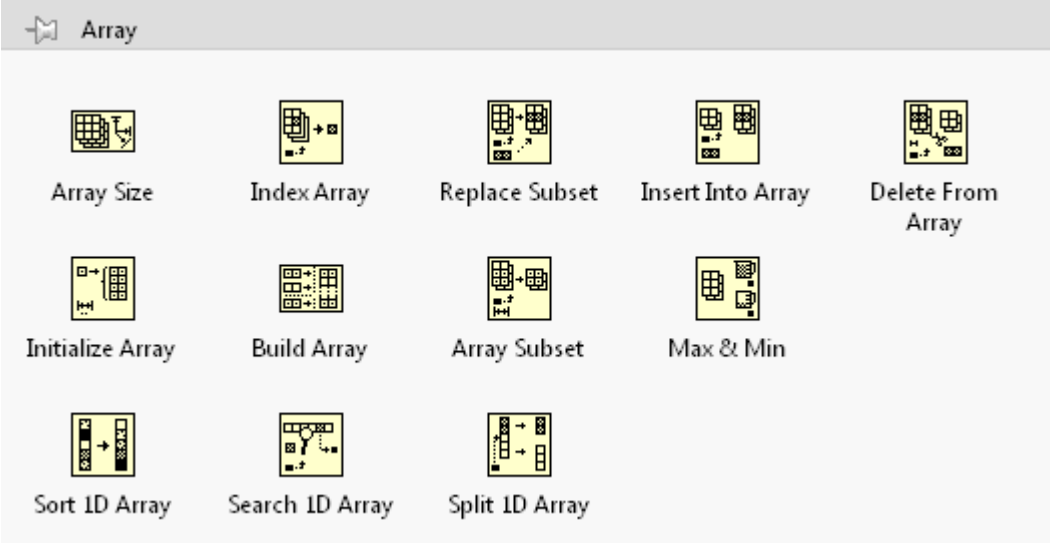
LabVIEW Programming Environment (25% of exam questions)

- Setting up and using a LabVIEW Project to:
    - Add, delete, and move elements
    - Use libraries and appropriate types of folders
    - Avoid cross-linking
- Data Types:
    - Recognize data types on the front panel
    - Recognize data types on the block diagram from terminals and wires
    - Choose appropriate controls, indicators, data types, & functions for a given scenario
- Predicting order-of-execution and behavior of
    - A non-looping VI
    - A Simple State Machine
    - An Event-driven UI Handler
    - Parallel Loops (without queues)
- Using basic functions to create a simple Acquire-Analyze-Visualize application
- Troubleshooting by identifying and correcting the cause of a broken arrow or incorrect data
- Error handling using error clusters and merge error functions to ensure errors are handled well
- Navigating LabVIEW help to get more information about inputs, outputs, and functions

LabVIEW Programming Fundamentals (50% of exam questions)

- Loops
    - Create continuous HW acquisition or generation loop by applying a Open-Configure-Perform Operation-Close model.
    - Retain data in shift registers
    - Use input and output terminals effectively, including:
        - Determining the last value output
        - Indexing input and output terminals
        - Concatenating output
        - Using conditional output
        - Using shift registers, both initialized and uninitialized.
    - Use timing of loops appropriately, including:
        - Software timing
        - Hardware timing
    - Use For Loops and While Loops appropriately

- Arrays
  - View data from an n-channel HW acquisition VI (using the DAQmx Read VI) using a waveform graph, waveform chart, or numeric/waveform array indicator.
  - Extract a single channel of data (waveform or 1D Array) from a:
    - 1D waveform array representing acquired data from multiple channels
    - 2D numeric array representing acquired data from multiple channels
    - 1D numeric array representing single measurement from multiple channels
  - Use a For Loop with auto-indexing and conditional tunnels to:
    - Iterate through an array
    - Iterate processing code on each channel of data in a 1D waveform array
    - Generate an array of data that meets required conditions
  - Identify by sight and be able to use and predict the behavior of the following array functions and VIs:
    - Array Size
    - Index Array
    - Replace Subset
    - Insert Into Array
    - Delete From Array
    - Initialize Array
    - Build Array
    - Array Subset
    - Max & Min
    - Sort 1D Array
    - Search 1D Array
    - Split 1D Array



- Writing conditional code to perform an action based on the value of a user input or a measurement result.
- Reading and Writing data to a file
  - Use Open/Act/Close model for file I/O
  - Write data to a text file using high-level file I/O functions
  - Continuously stream data to a text file or a TDMS file
  - Append data to an existing data file
  - Log data using simple VIs

- Acquire data from DAQmx functions
- Display data on a graph
- Save data to a CSV file
- Choose single measurement/multiple channel and single channel/multiple measurements configurations appropriately

Programming Best Practices (15% of exam questions)

- SubVIs – Reusing Code
  - Create SubVIs to increase readability and scalability of VIs
  - Configure the subVI connector pane using best practices
  - Choose appropriate code as a SubVI source
- Clusters – Grouping Data of Mixed Data Types
  - Create, manipulate, analyze, and use cluster data in common scenarios
  - Group related data by creating a cluster to improve data organization and VI readability
- Type Defs – Propagate Data Type Changes
  - Create Type Defs and use Type Defs in multiple places
  - Update Type Defs to propagate changes to all instances of the Type Def

Example Questions & Resources

Note on Using these Example Questions
This section is not a sample exam. It is a set of questions provided as a study guide. The best use of these questions is to:

1. Consider the question, and record what you believe to be the correct answer.
2. Create any VIs represented in a question.
3. Use the VI to confirm your answer.
4. Change the parameters and other aspects of the VIs to make sure you understand the topic and area of knowledge represented by the question.

Hardware

Sub-Topics

- Connecting Hardware: Sensors, DAQ, Devices under test (DUT).
- Acquiring and validating a signal
- Processing signals
- Using appropriate sample rates

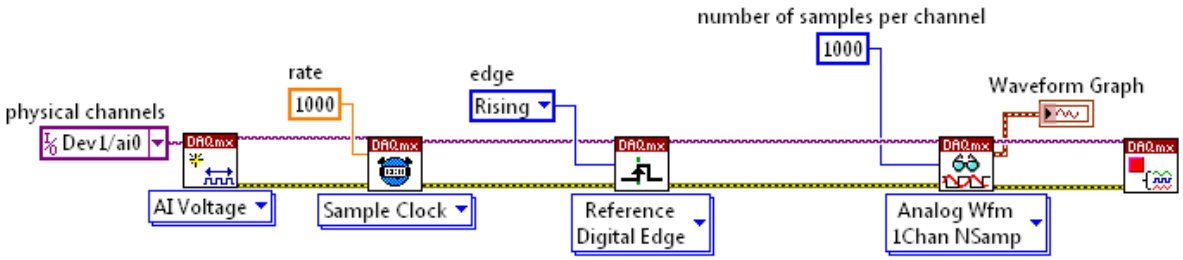
Resources

- LabVIEW Help -- *Taking Measurements*
- LabVIEW Help -- *Aliasing*
- NI-DAQmx Help
  - *NI-DAQmx Key Concepts*
  - *Measurement Fundamentals*
- NI White Papers on sample rate and aliasing
- NI White Paper "How to Set Up an Academic Laboratory for Data Acquisition"

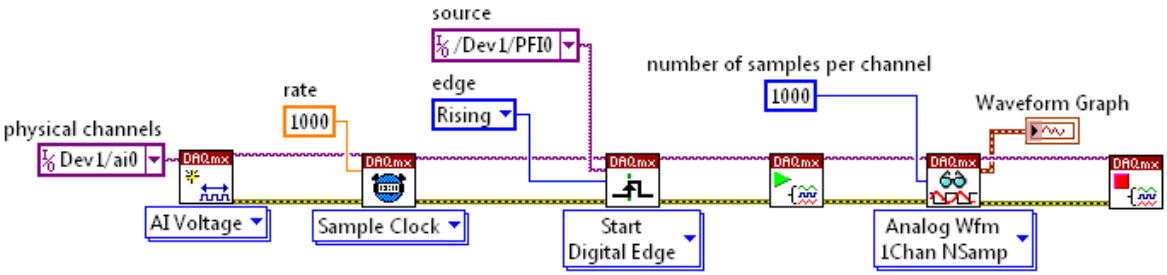
Example Questions

Q1: Which VI will acquire 1000 samples immediately after PFI0 transitions from low to high?

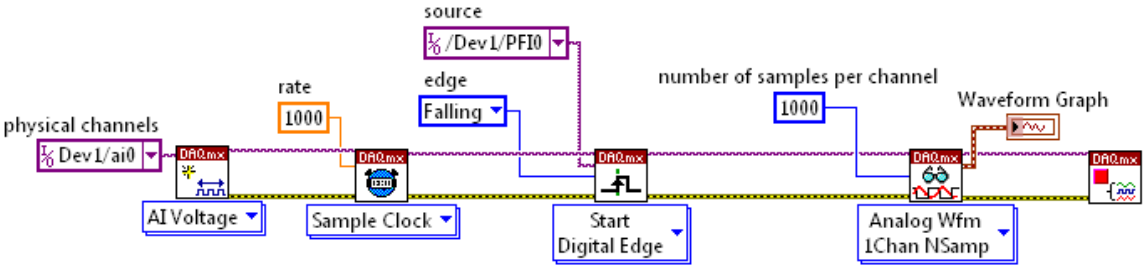
A:



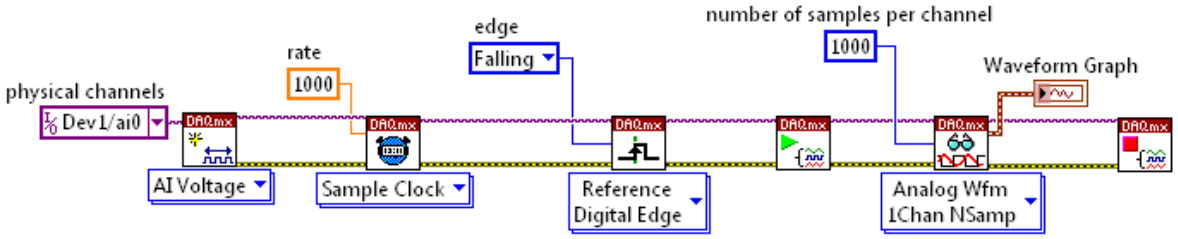
B:



C:



D:



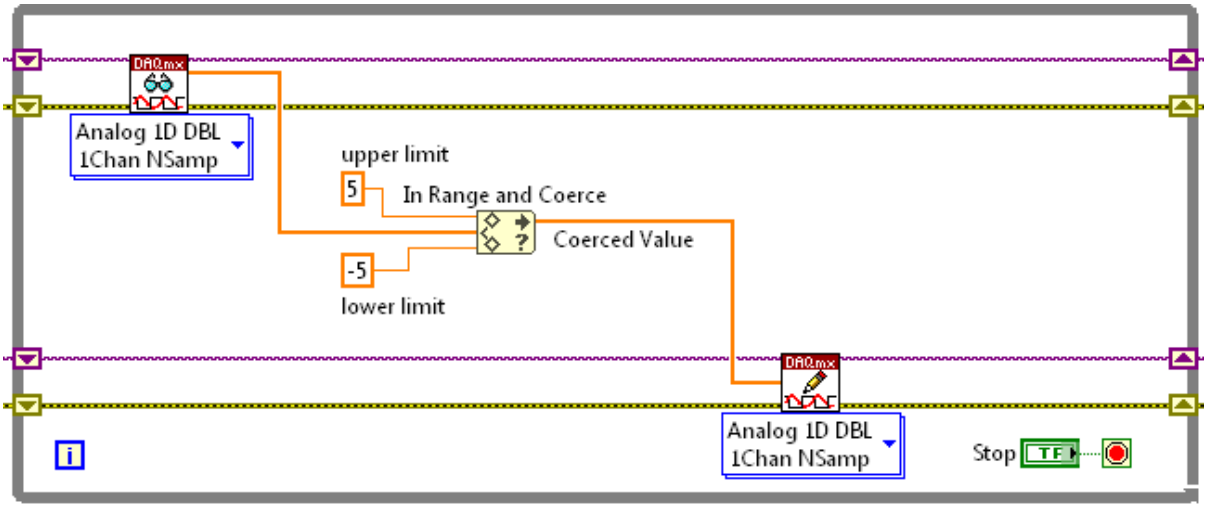
Q2: The DAQmx Read Task is configured to continuously read between -10V and +10V.

The DAQmx Write Task is configured to continuously write between -5V and +5V without regeneration.

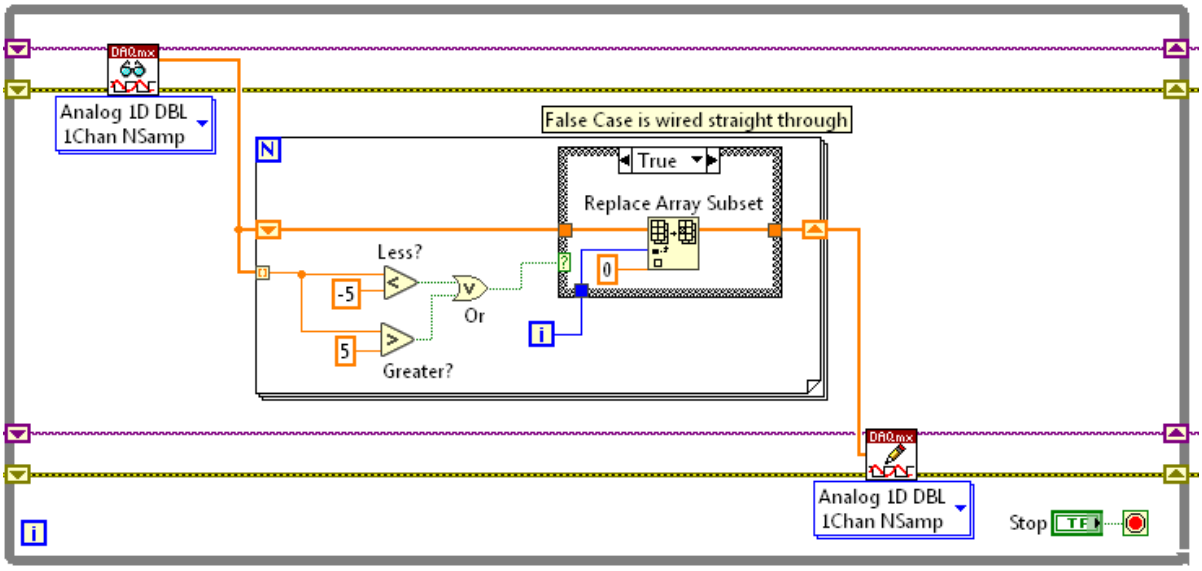
Both Read and Write tasks are sampling at 1 KHz, and regeneration is set to "none".

If the signal read has samples along the full +/-10V range, which VI Snippet will eventually cause a buffer underrun error?

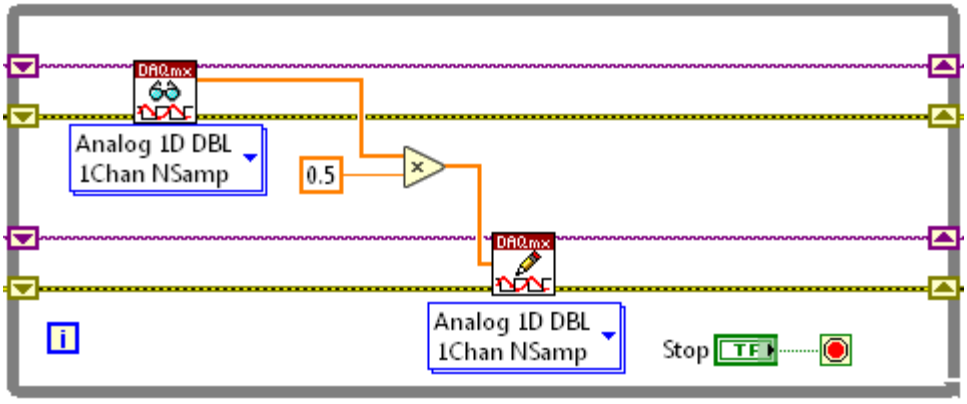
A:



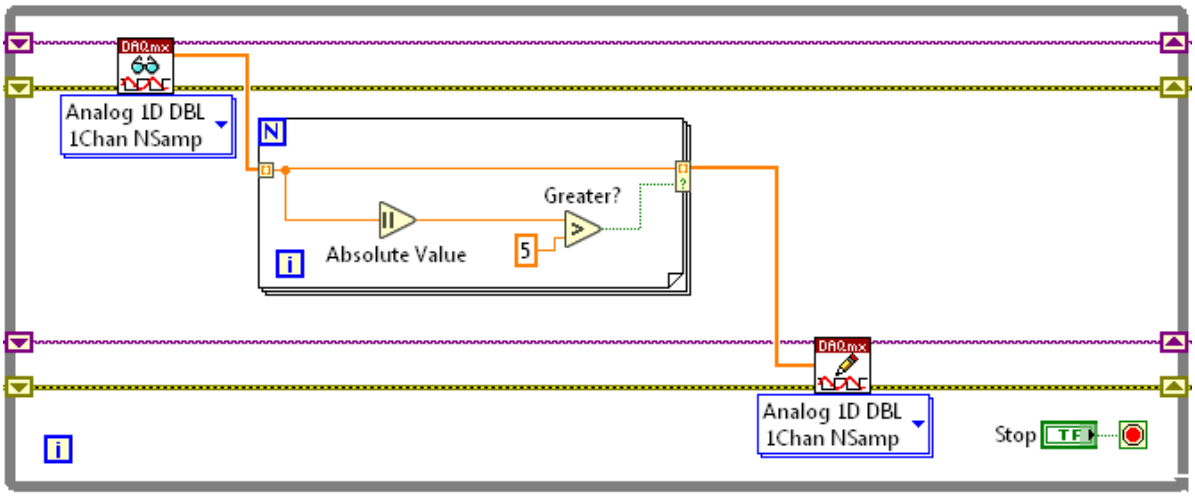
B:



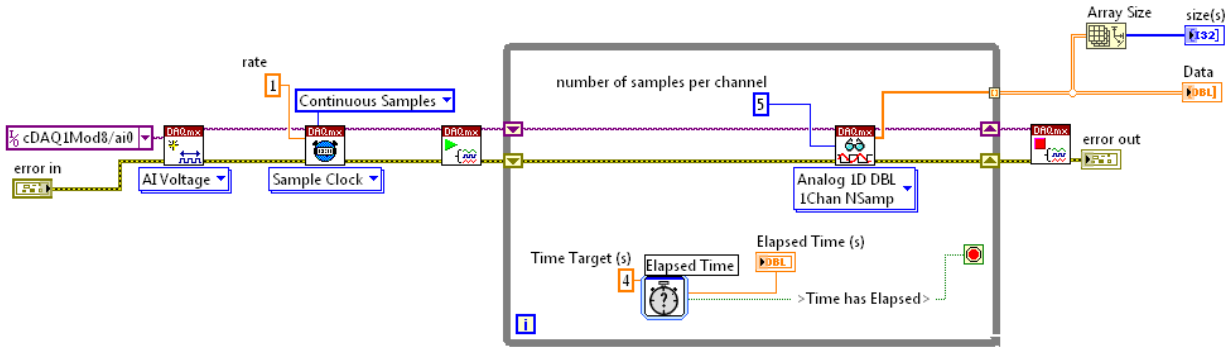
C:


Analog 1D DBL
1Chan NSamp
0.5
Analog 1D DBL
1Chan NSamp
Stop
TF

D:


Analog 1D DBL
1Chan NSamp
N
Greater?
Absolute Value
5
Analog 1D DBL
1Chan NSamp
Stop
TF

Q3: After this VI runs, what will be the size of the array in the Data indicator?



A: 5 rows and 4 columns.

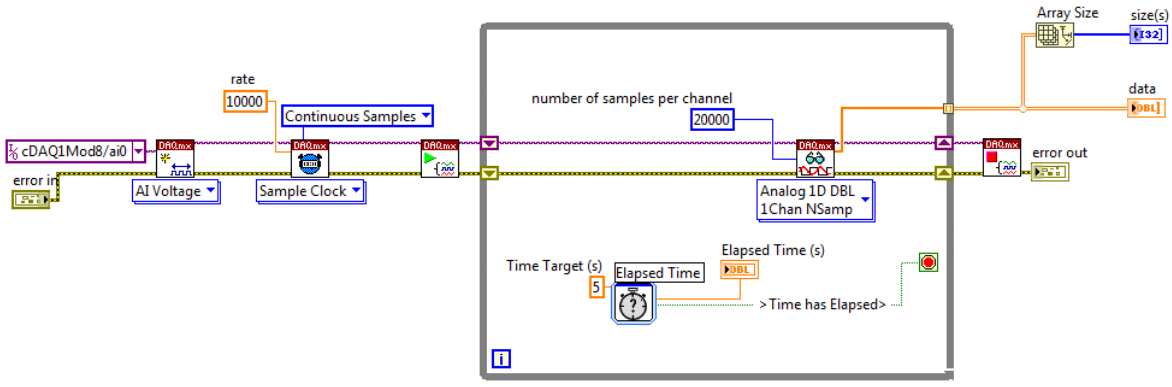
B: 2 rows and 5 columns.

C: 4 rows and 5 columns.

D: 1 row and 5 columns.

Note: Try different combinations of values for **rate**, **number of samples per channel**, and **Time Target**.

Q4: After this VI runs, what will be the size of the Data indicator array?



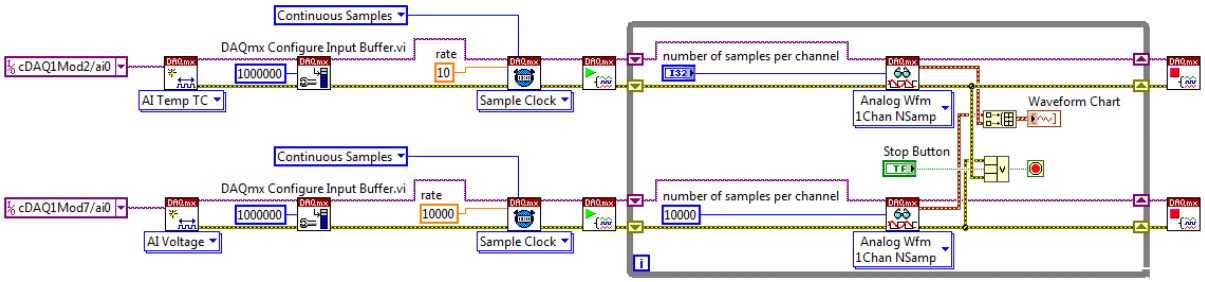
A: 2 rows, 20,000 columns

B: 3 rows, 20,000 columns

C: 4 rows, 20,000 columns

D: 5 rows, 20,000 columns

Q5: What value must be entered in the **number of samples per channel** control for this VI to execute continuously until the **Stop Button** is pressed?



A: 10

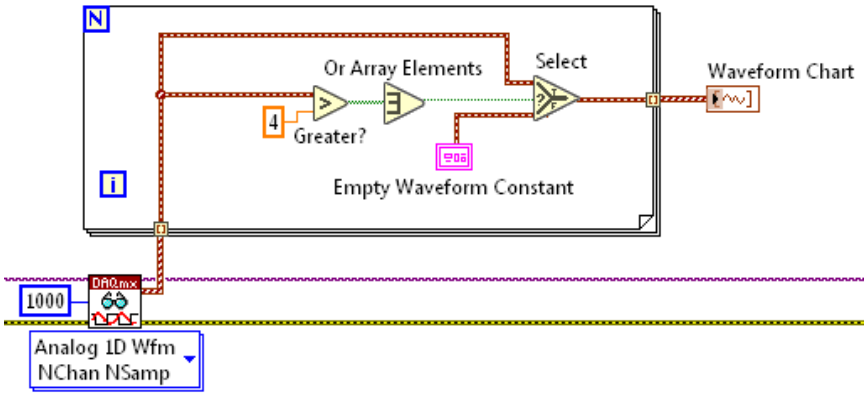
B: 1000

C: 10000

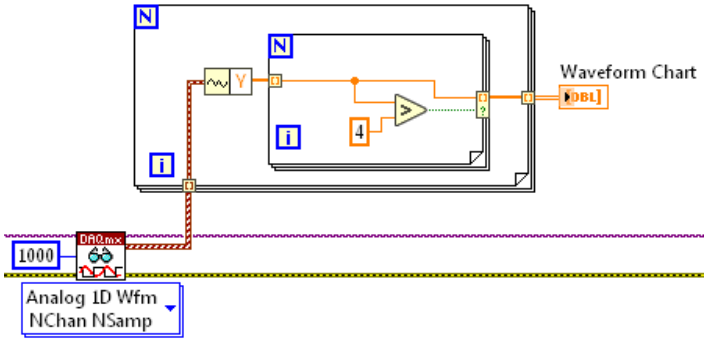
D: 1000000

Q6: Which VI will only output all channels that have a value greater than 4 in them?

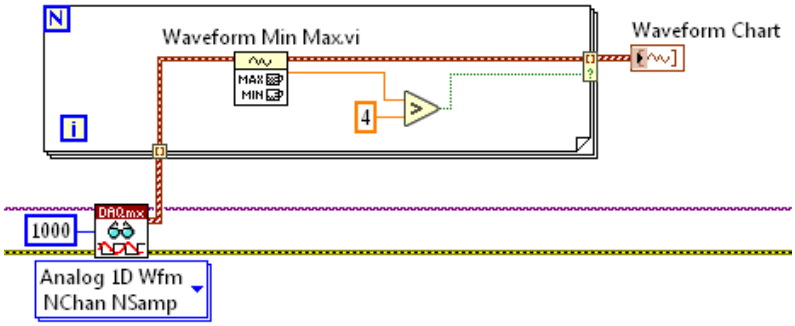
A:



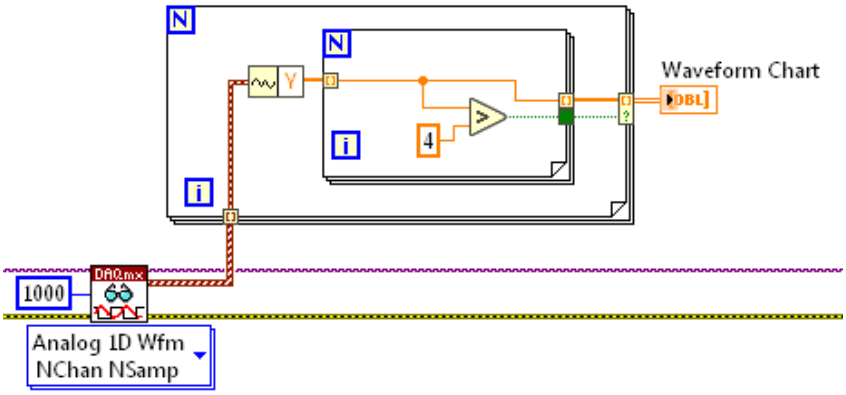
B:



C:



D:



LabVIEW Programming Environment

Topics

- Setting up and using a LabVIEW Project
- Data Types
- Predicting order-of-execution and behavior of a VI
- Using basic functions to create a simple Acquire-Analyze-Visualize application
- Troubleshooting by identifying and correcting the cause of a broken arrow or incorrect data
- Error handling using error clusters and merge error functions to ensure errors are handled well
- Navigating LabVIEW help to get more information about inputs, outputs, and functions

Resources

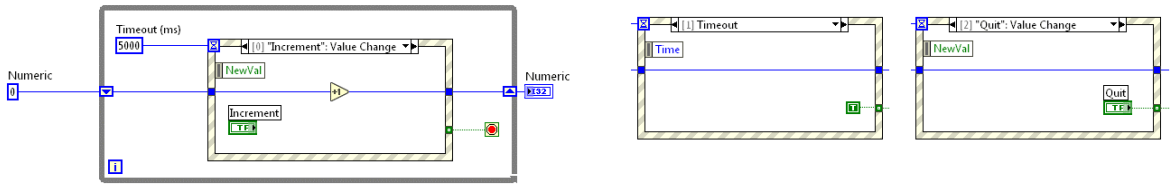
- LabVIEW Help
  - *Numeric Data Types Table*
  - *Boolean Controls and Indicators*
  - *Block Diagram Data Flow*
  - *Types of Graphs and Charts*

Example Questions

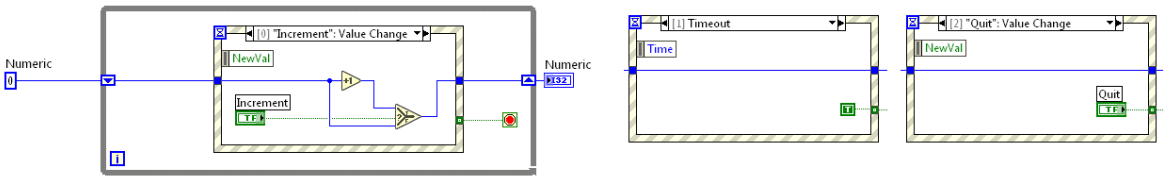
Q1: Your VI must count the number of times the Increment button is pushed between when the Run arrow and the Quit button are pressed.

All Event Cases are shown. Which code best meets these requirements?

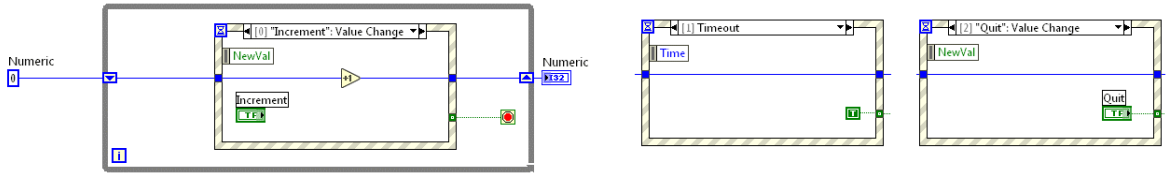
A:



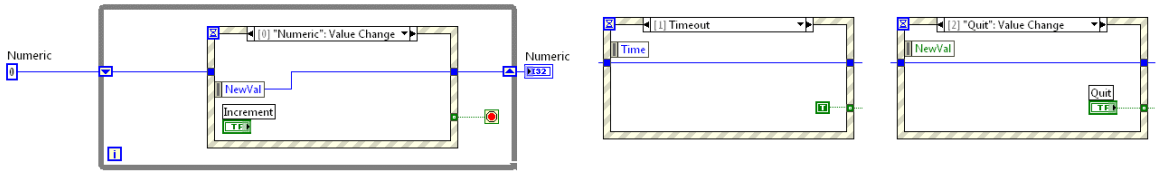
B:



C:

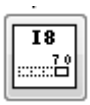


D:



Q2: Which data type would be appropriate to display angle in degrees, from -180 to +180 degrees, rounded to the nearest whole number?

A: 

B: 

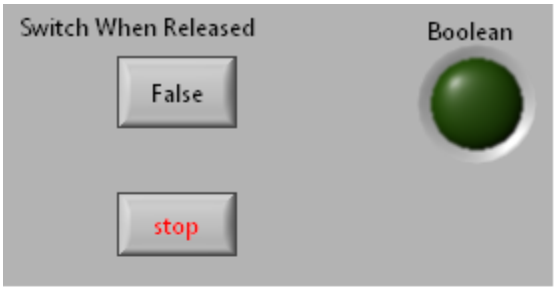
C: 

D: 

Q3: Before this VI is run, **Switch When Released** and **Boolean** are both False.

Consider the VI and user actions.

What value will be displayed in the **Boolean** indicator at 12 seconds and then at 18 seconds?





Time in Seconds	User Action
0	Clicks **Run** Arrow
5	Clicks and holds mouse on the **Switch When Released** button
8	Moves mouse away from the **Switch When Released** button
11	Releases mouse
15	Clicks and holds mouse on the **Switch When Released** button
17	Releases mouse while still over the **Switch When Released** button
20	Clicks the **Stop** button

A: TRUE, TRUE

B: TRUE, FALSE

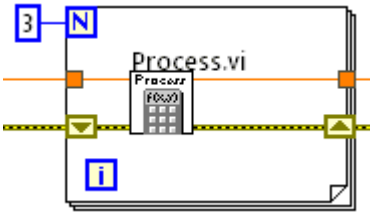
C: FALSE, TRUE

D: FALSE, FALSE

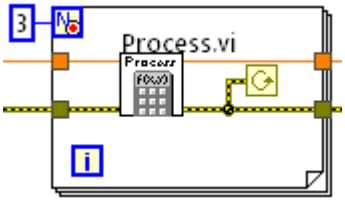
Q4: If SubVI Process.vi generates an error, your VI should clear the error and retry the SubVI two more times to give it a chance to succeed.

Which of the following VIs best meets these requirements?

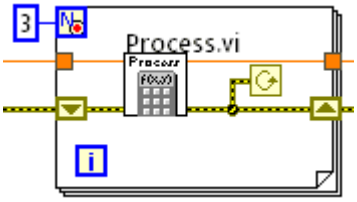
A:



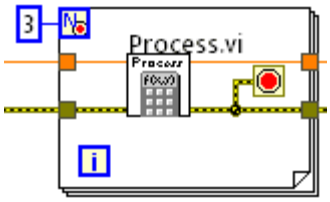
B:



C:



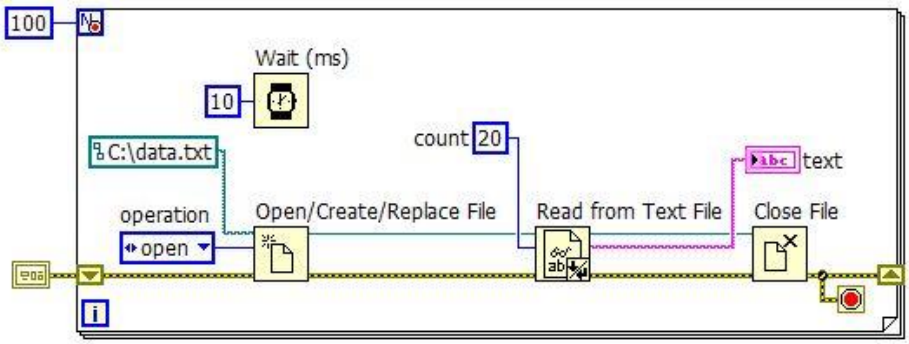
D:



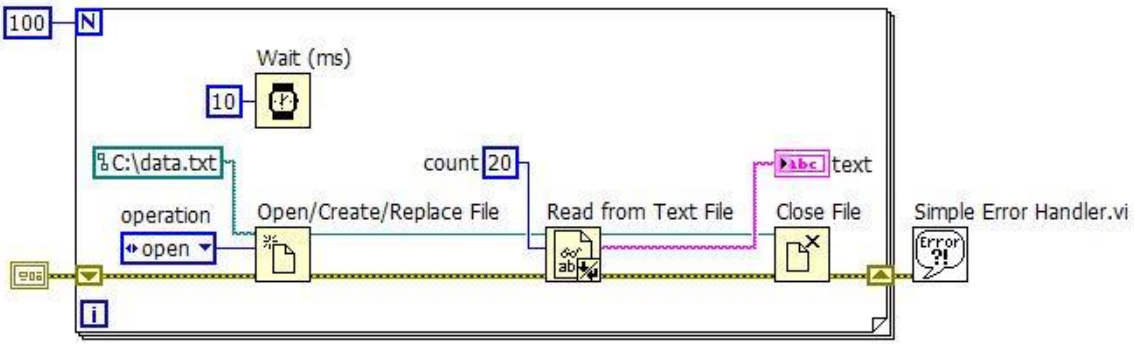
Q5: The file C:\data.txt does not exist. Automatic error handling is disabled.

Which diagram will report the error and stop within 100 ms?

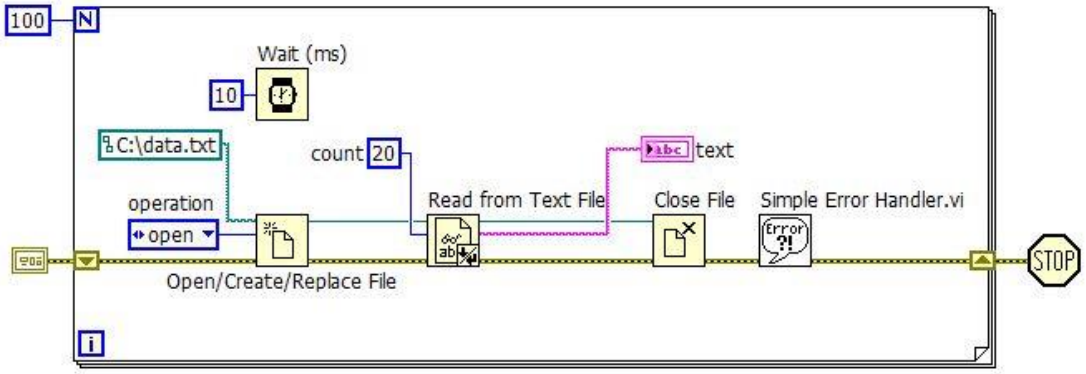
A:



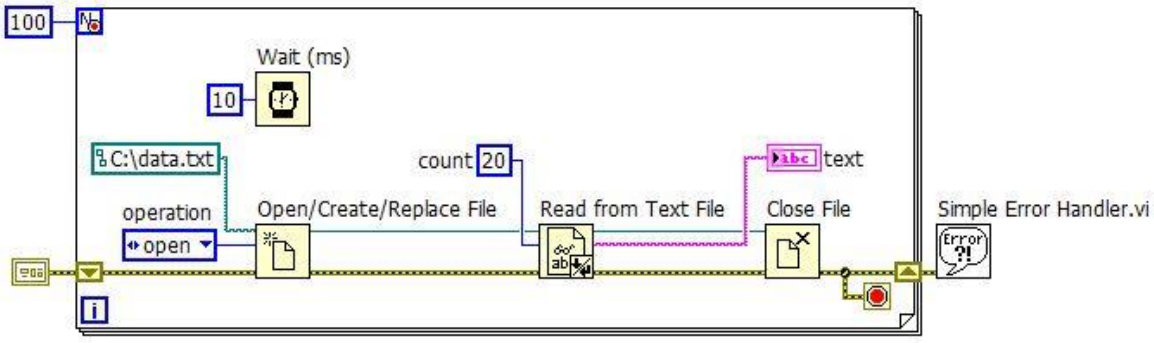
B:



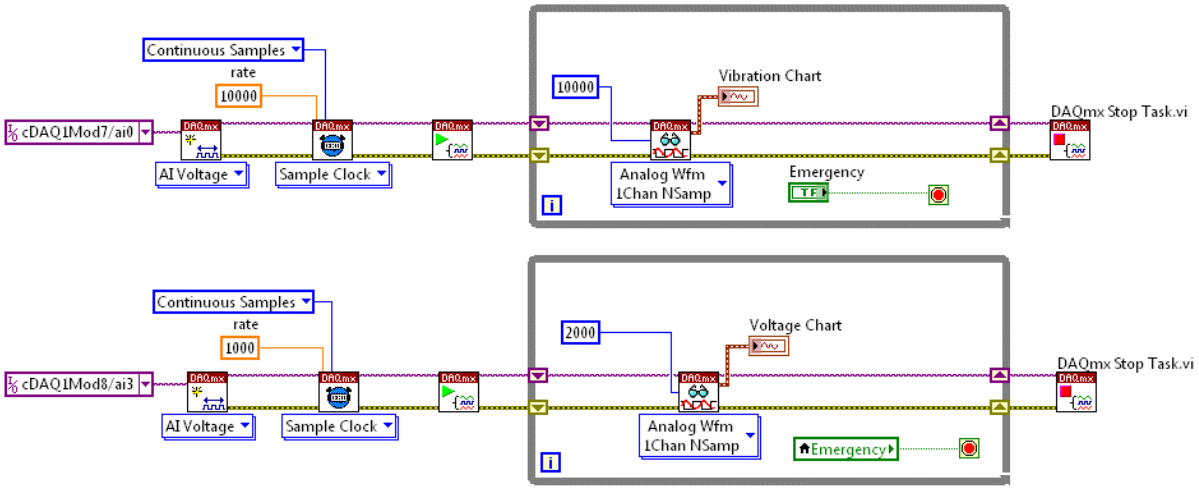
C:



D:



Q6: After the chart has begun updating, the Emergency button is pressed. The VI will:



A: Stop after a maximum of 1 second.

B: Stop after a maximum of 2 seconds.

C: Stop after a maximum of 4 seconds.

D: Become unresponsive until aborted.

LabVIEW Programming Fundamentals

Topics

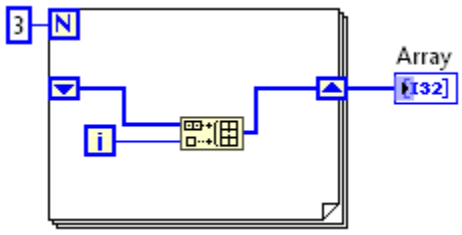
- Loops
- Arrays
- Conditional Code
- Read/Write Data (File I/O)

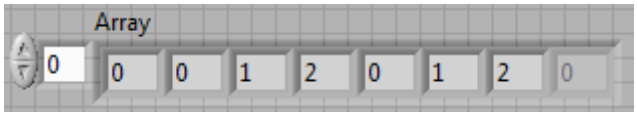
Resources

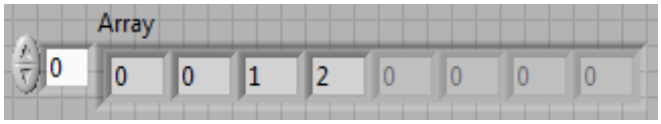
- LabVIEW Core 1 Course
- LabVIEW Help
    - *File I/O*
    - *Binary Files*
    - *Spreadsheet Files*
    - *TDM/TDMS Files*
    - *Text Files*
    - *Waveforms*
    - *For Loop*
    - *While Loop*
    - *Processing Individual Elements in an Array with a Loop*

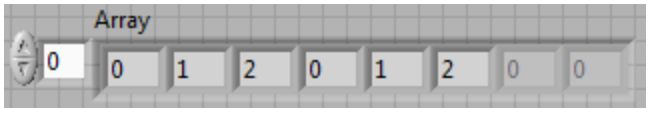
Example Questions

Q1: What does the indicator **Array** look like after this VI has run twice?



A: 

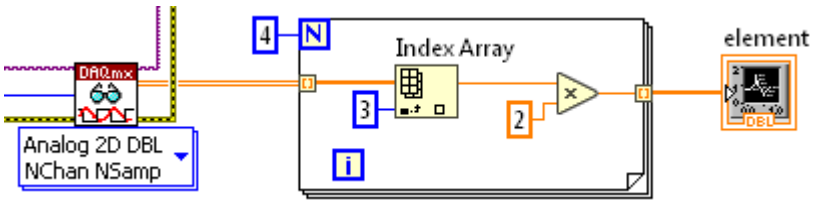
B: 

C: 

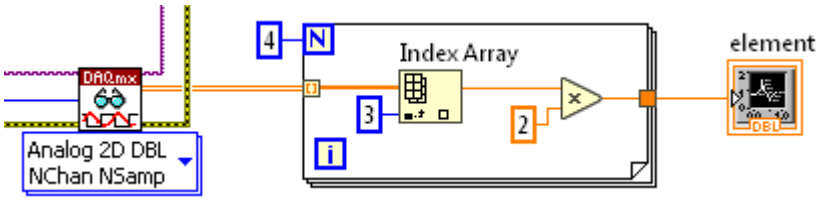
D: 

Q2: Which VI can extract the fourth channel output by the DAQmx Read function and double every value, or return an empty array if there are fewer than 4 channels?

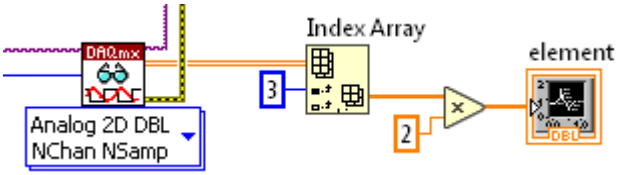
A:



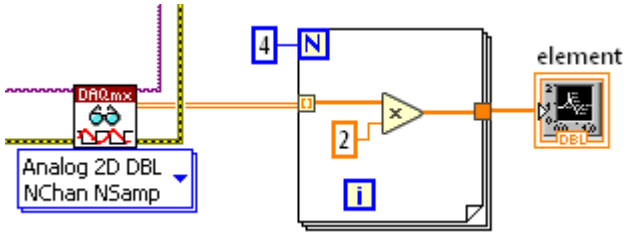
B:



C:



D:

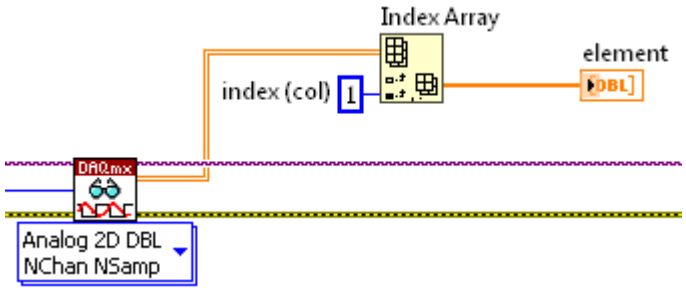


Q3: The DAQmx Read VI is set up to read these channels:

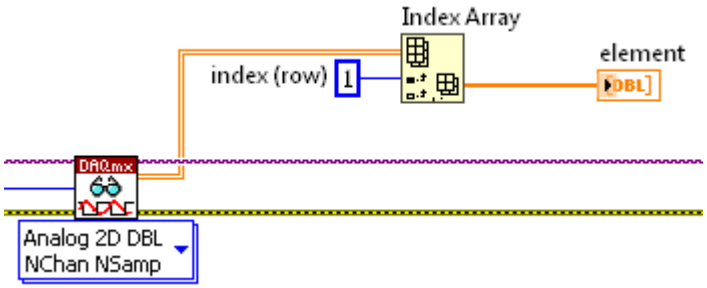


Which VI will correctly extract and display all the data from channel ai2?

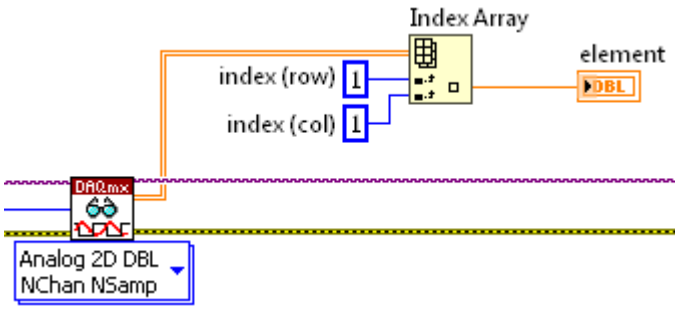
A:



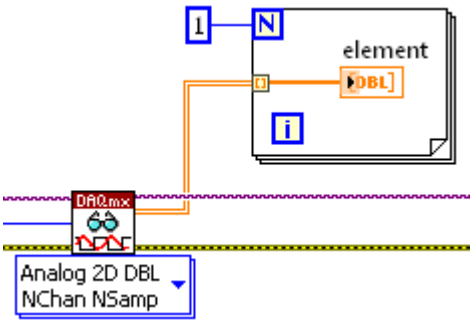
B:



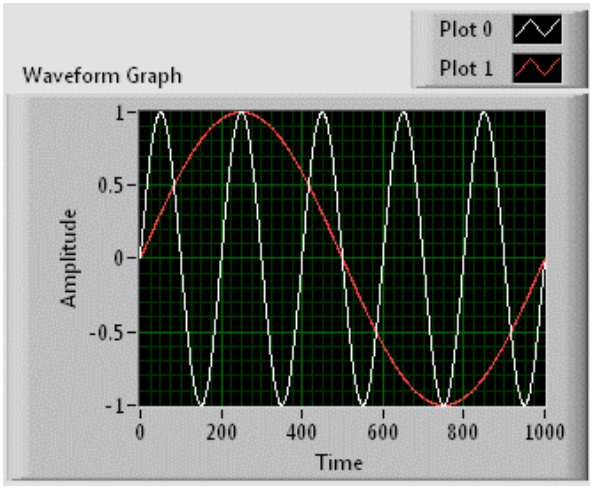
C:



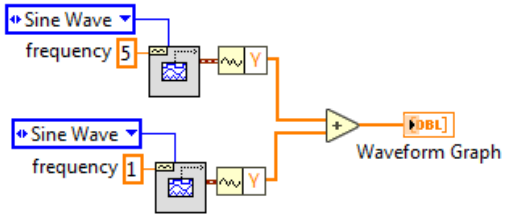
D:



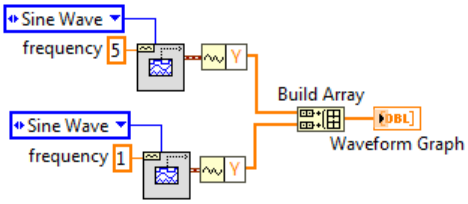
Q4: Which VI will generate this output?



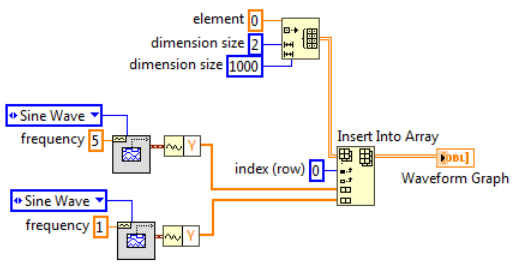
A:



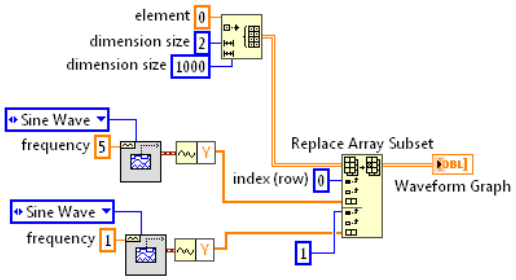
B:



C:

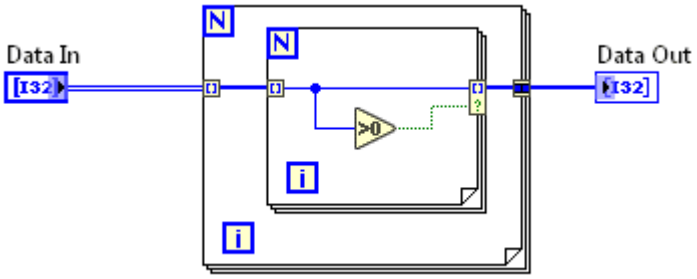


D:

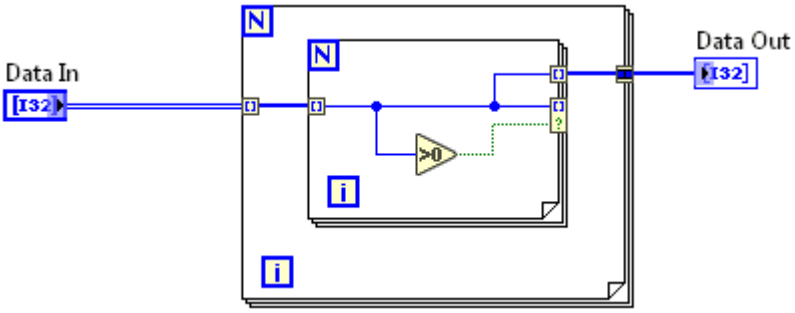


Q5: You are asked to write a VI that will take a 2D array and output all the positive values in a 1D array. Which VI satisfies these requirements?

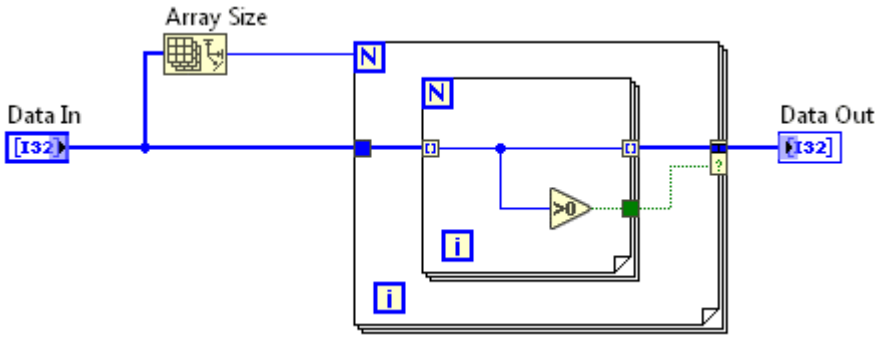
A:



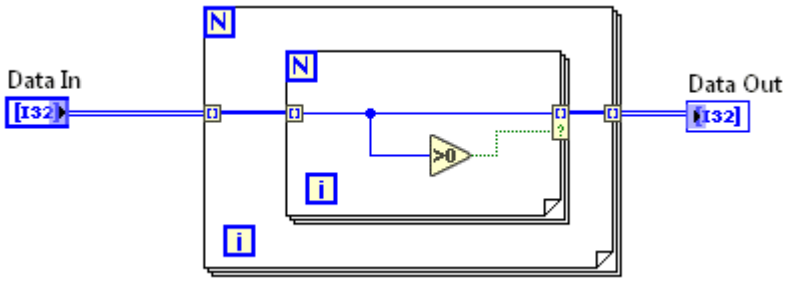
B:



C:

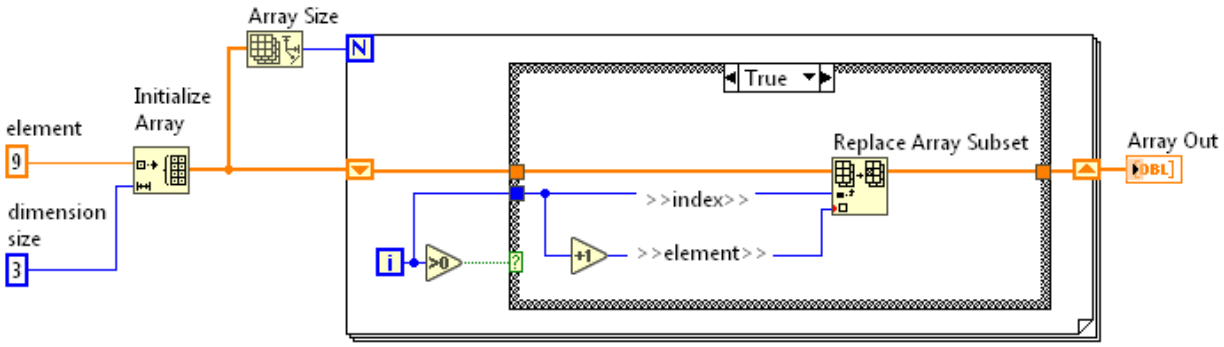


D:



Q6: What will the **Array Out** indicator look like after the VI completes execution?

The False case is wired straight through.



A: 

C: 

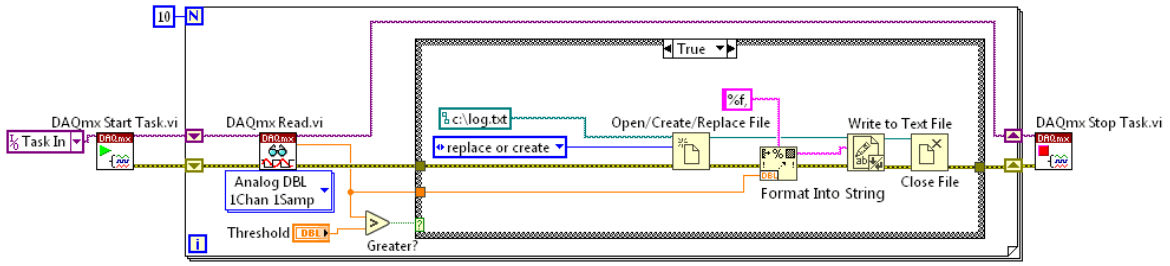
B: 

D: 

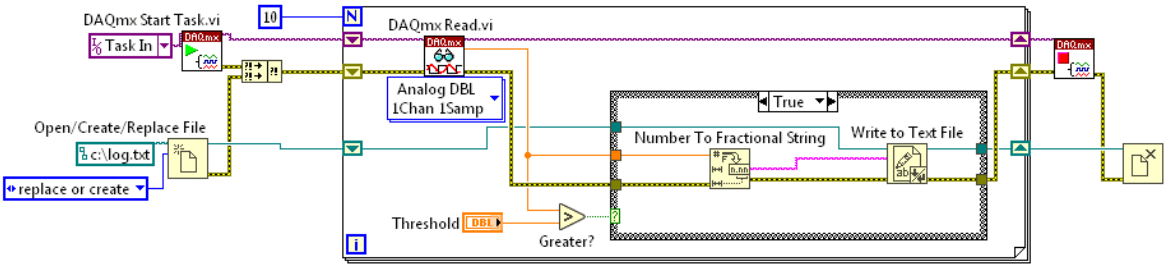
Q7: Which VI will log every sample above the Threshold value to a human-readable file?

All cases not shown are wired straight through.

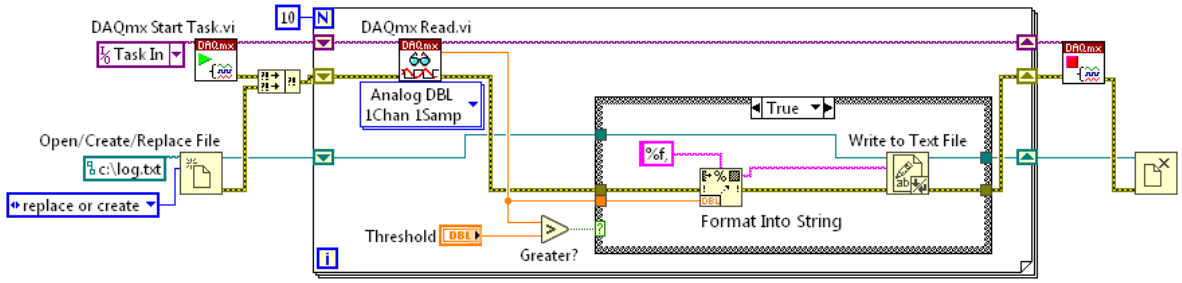
A:



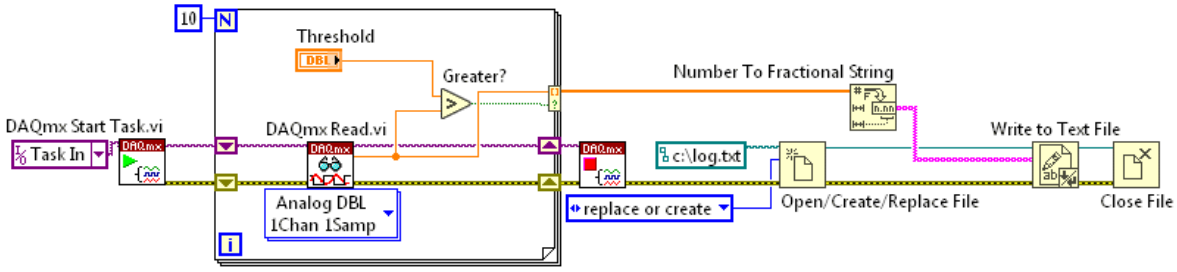
B:



C:

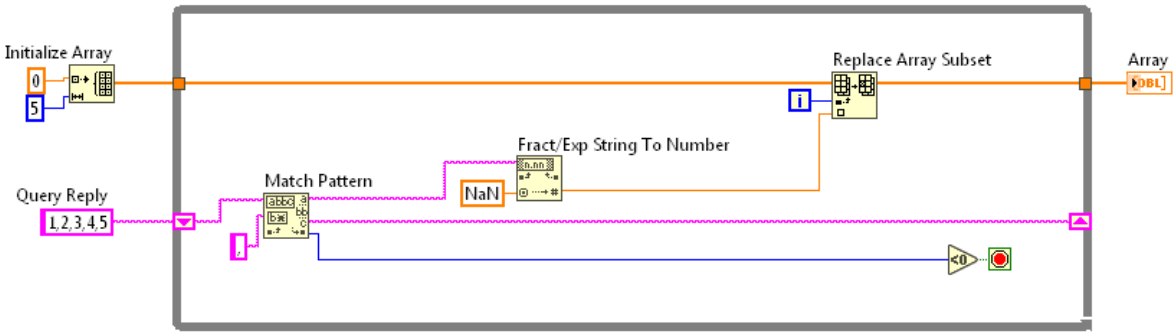


D:

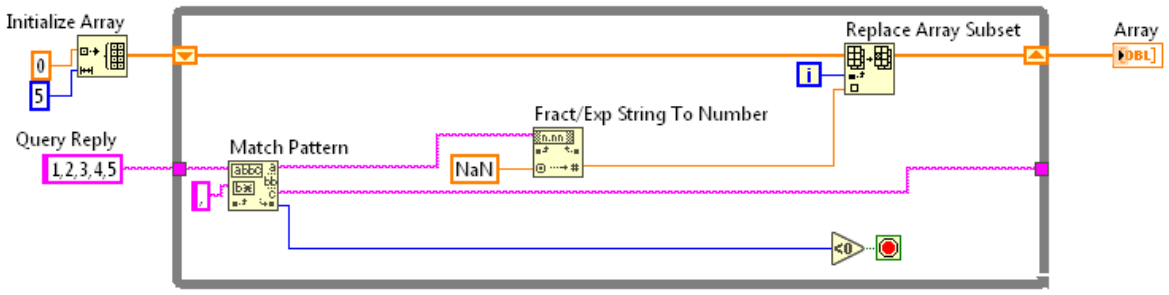


Q8: The reply from an equipment query produces a string of up to 5 comma-separated parameters. Which VI parses the reply and places the values in order in a numeric array?

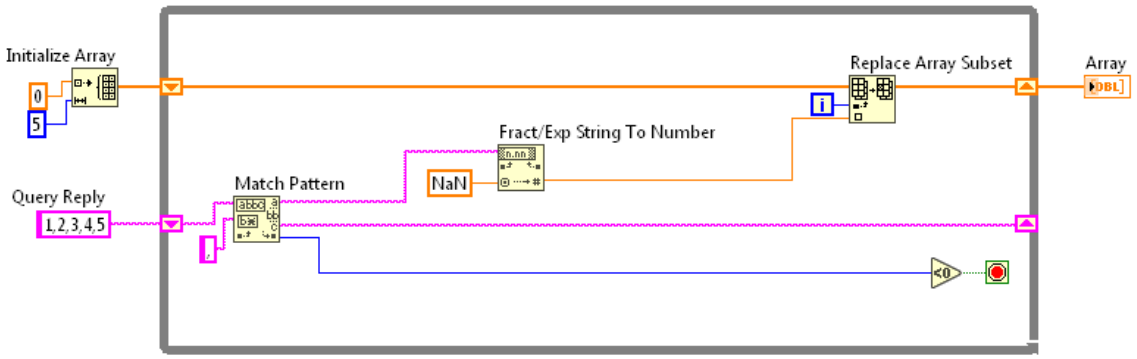
A:



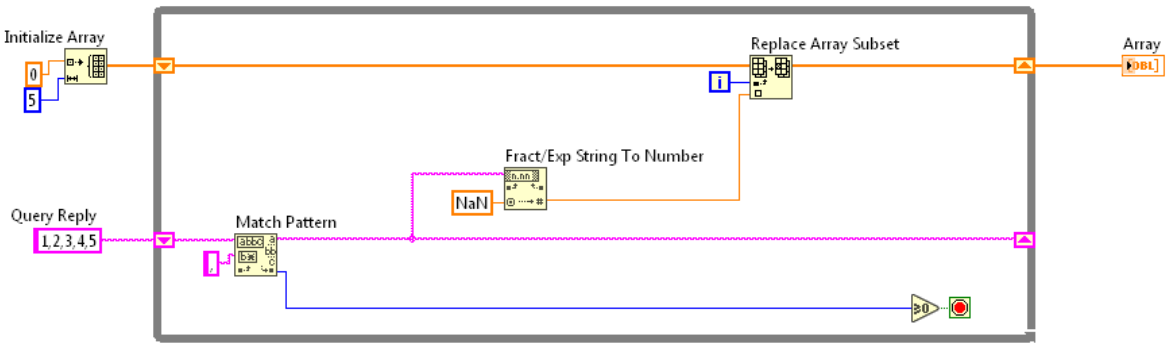
B:



C:



D:

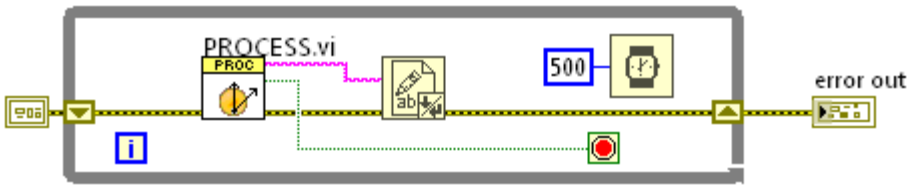


Q9: The SubVI PROCESS represents a test that outputs a string each time the test runs, with no Carriage Return or End of Line constant in the string.

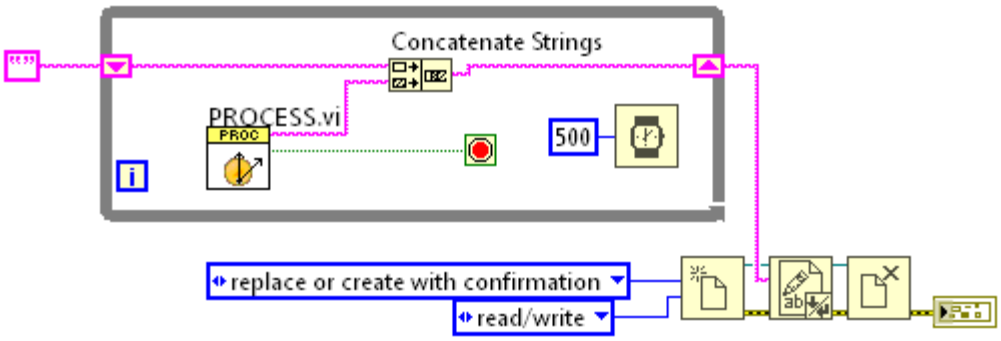
Your VI must repeat the SubVI PROCESS until the user pushes the "Stop" button, then allow the user to select the location for the log. The VI must log all the data, putting data from each run of PROCESS on a separate line.

Which of the following VIs meets these requirements?

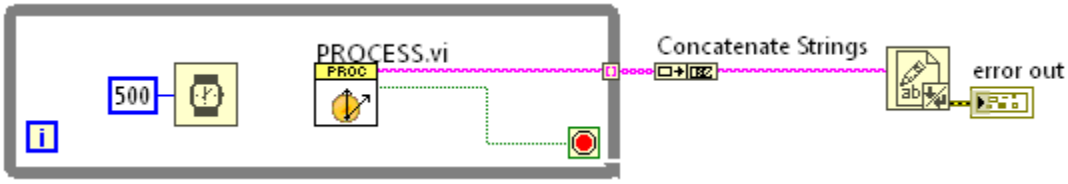
A:



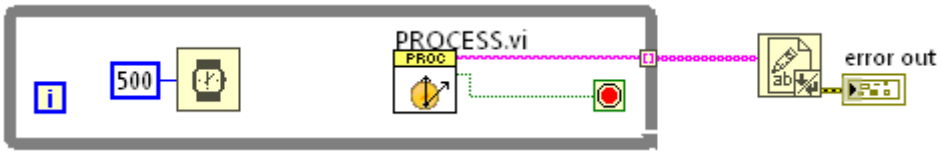
B:



C:

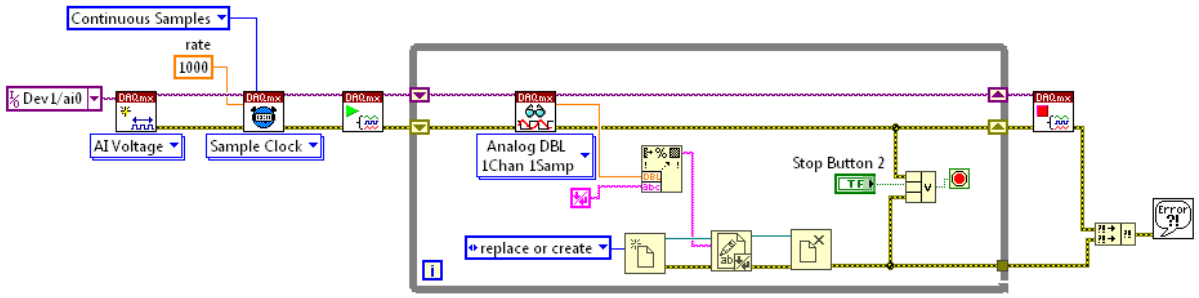


D:

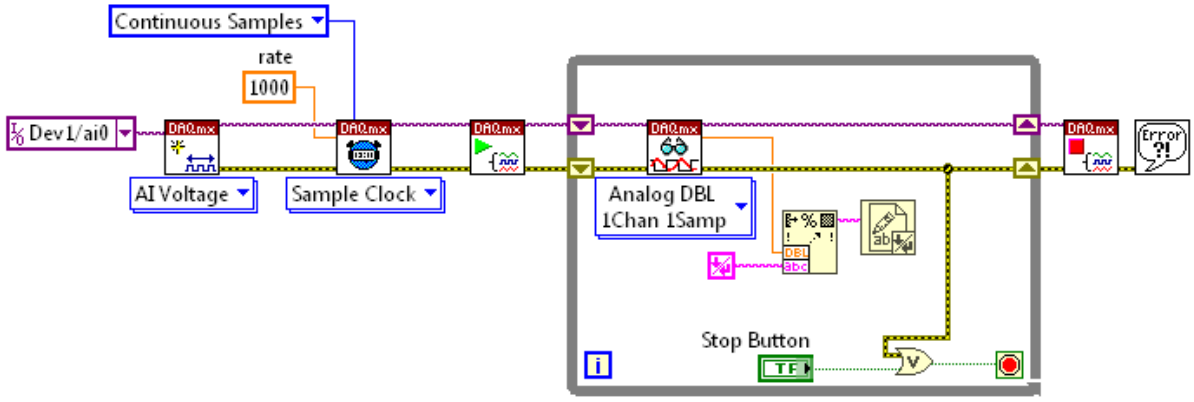


Q10: Which VI will continuously acquire and log data to a human readable text file?

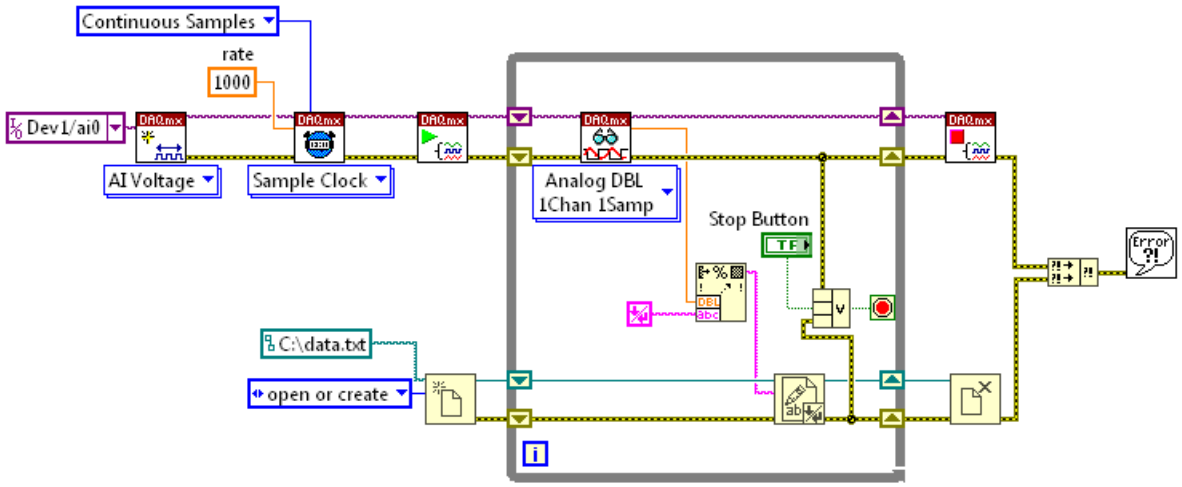
A:



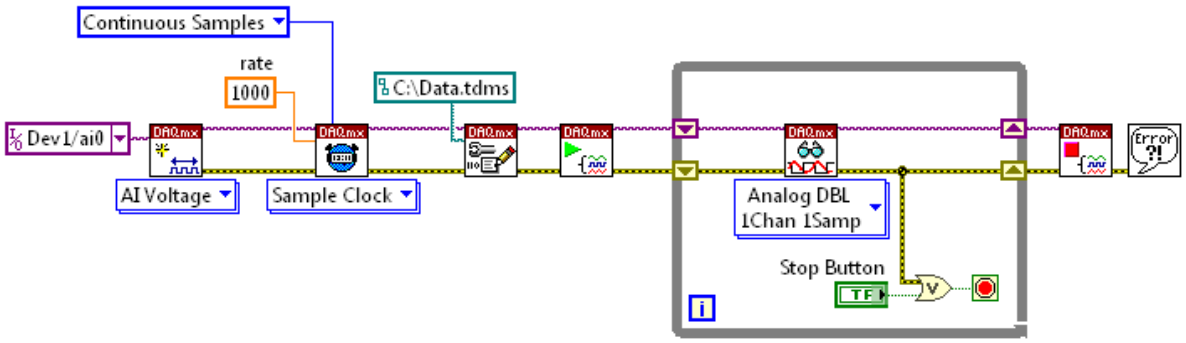
B:



C:



D:



Programming Best Practices

Sub-Topics

- SubVIs – Reusing Code
- Clusters – Grouping Data of Mixed Data Types
- Type Defs – Propagate Data Type Changes

Resources

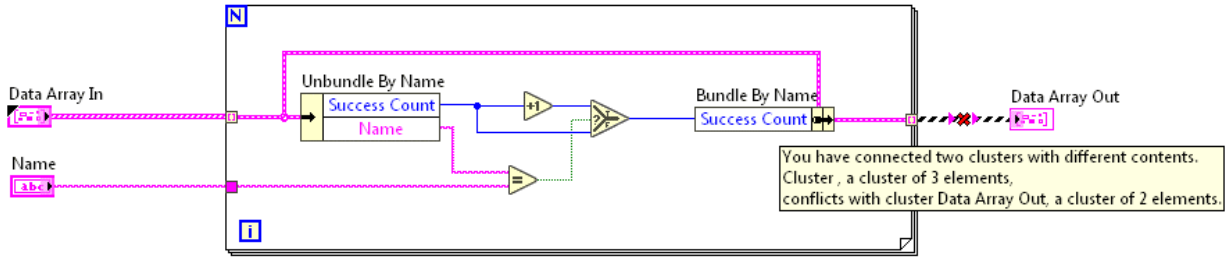
LabVIEW Help

- *Creating Custom Controls, Indicators, and Type Definitions*
- *Creating Modular Code*

Example Questions

Q1: When the cursor is over the broken arrow, the error message shown is displayed.

Data Array Out is the cluster with the correct content. How could this error be corrected?



A: Make Data Array Out into a Type Def, Create a copy of Data Array Out, change the copy to a control, and replace Data Array In with the new control.

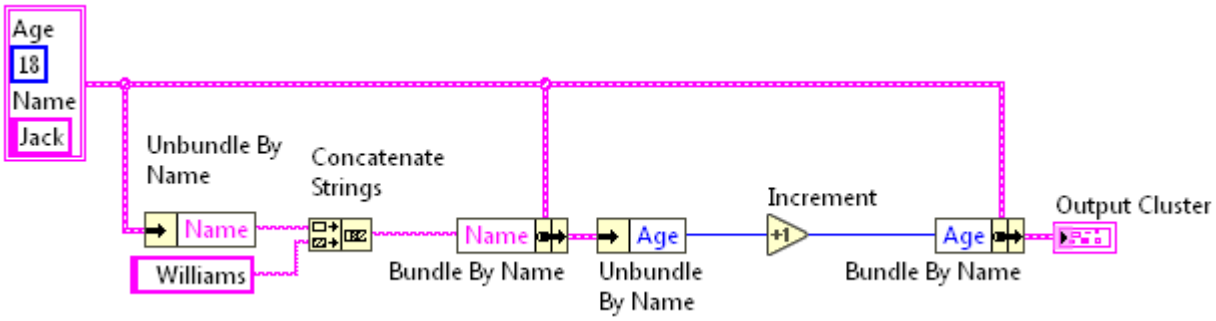
B: Delete the extra element from Data Array In, then save Data Array Out as a Type Def.

C: Right-click on Data Array In, choose "Open Type Def," then choose "Apply Changes" in the File menu of the Type Def.

D: Delete Data Array In, hit ctl-B to clear the broken arrow, then Right-click on the loop input terminal and select "Create -- Control".

Q2: What values will be displayed in the Output Cluster indicator after the VI completes execution?

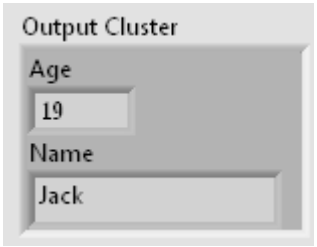
String constants are set to '/' Codes Display.



A:



B:



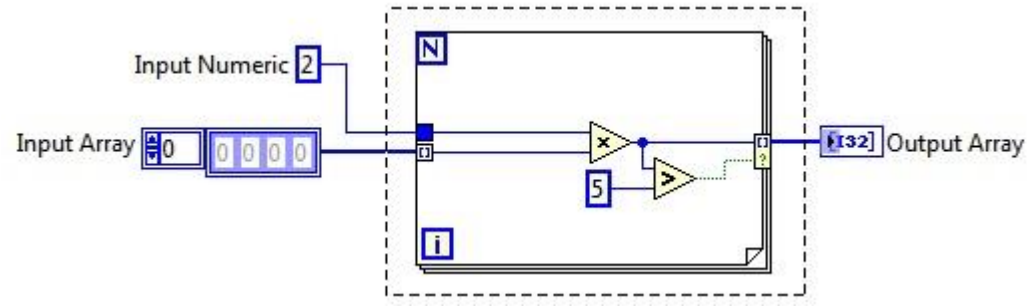
C:



D:



Q3: You select only the For Loop as shown by the dashed line in the image, and then select **Edit»Create SubVI** from the Edit menu. What will the code on the block diagram look like immediately after creating the SubVI?



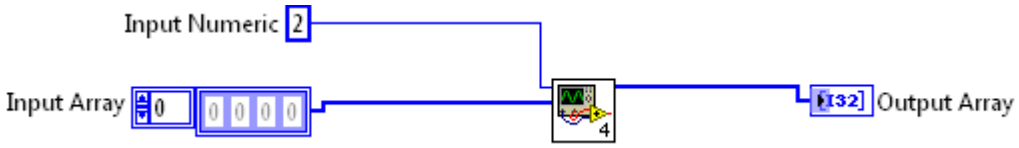
A:



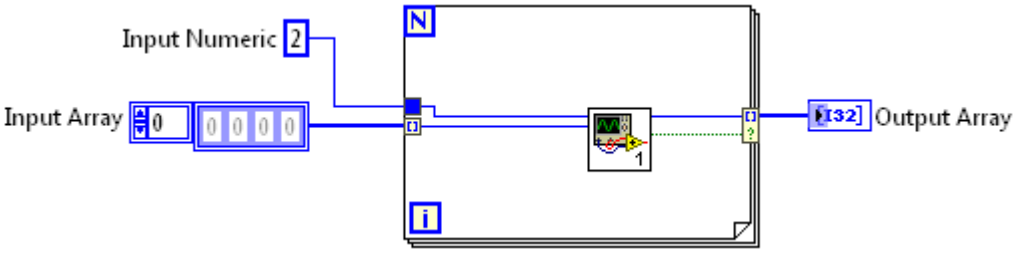
B:



C:



D:



Q4: Which of the following steps will **not** be done automatically when you create a SubVI from a section of Code using Edit>>Create SubVI?

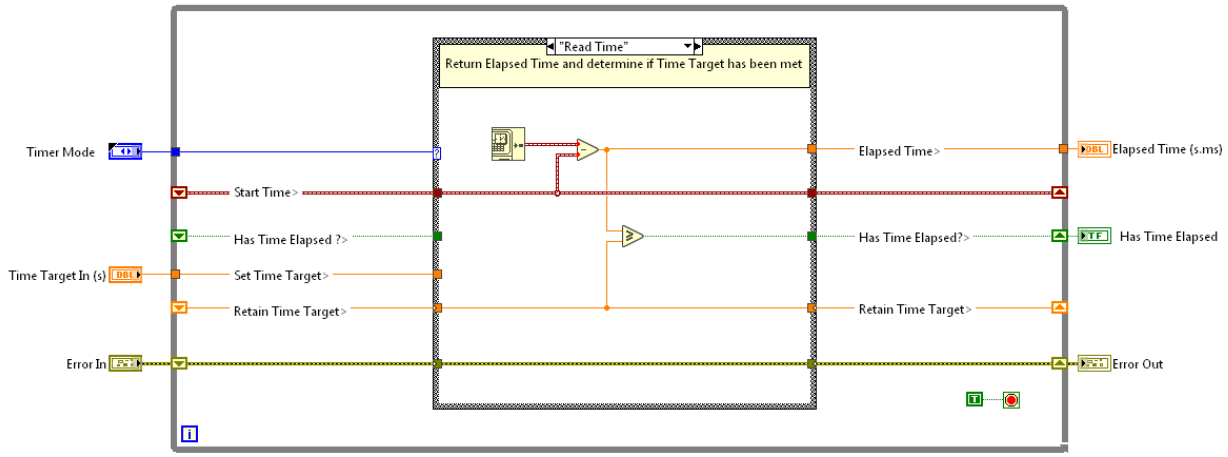
A: Assigning terminals to the appropriate indicators and controls.

B: Connecting the controls and indicators on the front panel to terminals on the connector pane.

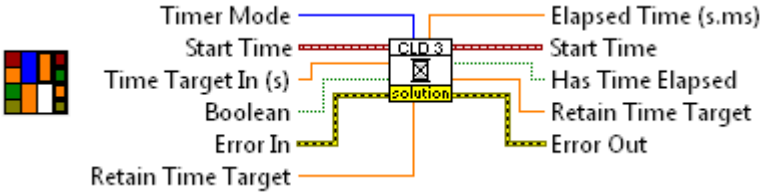
C: Configuring the Connector Pane to have enough terminals for all inputs and outputs.

D: Setting a text-based icon for the subVI.

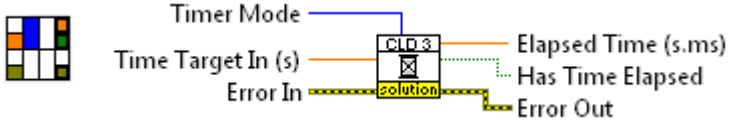
Q5: For this Functional Global Variable used as a timer, which connector pane configuration would be most in line with programming best practices and the default LabVIEW editor configuration?



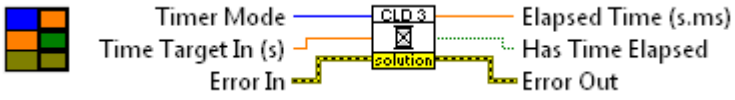
A:



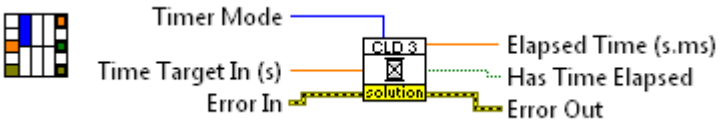
B:



C:



D:



APPENDIX I: SIMULATED DAQ FOR CLAD PREP

Use NI MAX to create the following simulated DAQ hardware. The configuration of this simulated cDAQ device matches the configuration of the physical cDAQ Demo Kit used in some classroom training. Using this configuration, you can create and run many of the VIs used in the HW-related questions in the CLAD exam and prep materials.

Create Simulated Hardware

1. Open Measurement & Automation Explorer (MAX).
2. Right-click **Devices and Interfaces** and select **Create New**.
3. In the Create New dialog box, select **Simulated NI-DAQmx Device or Modular Instrument**, and click the **Finish** button.
4. Expand **CompactDAQ Chassis,** select **NI cDAQ-9178**, and click **OK**.
    a. The device will appear in the **Devices and Interfaces** list, as **NI cDAQ-9173 "cDAQ*x*"**, where "*x*" is 1 unless you have other cDAQ-9178 devices (real or simulated).
5. Right-click the device name (NI cDAQ-9178 "CLAD") in the Devices and Interfaces list and select **Configure Simulated cDAQ Chassis**.
    a. Create simulated instruments for each of the slots as described in Table 2-1. Note: You will not enter the information from the Description column. This information is for reference and describes the module type.

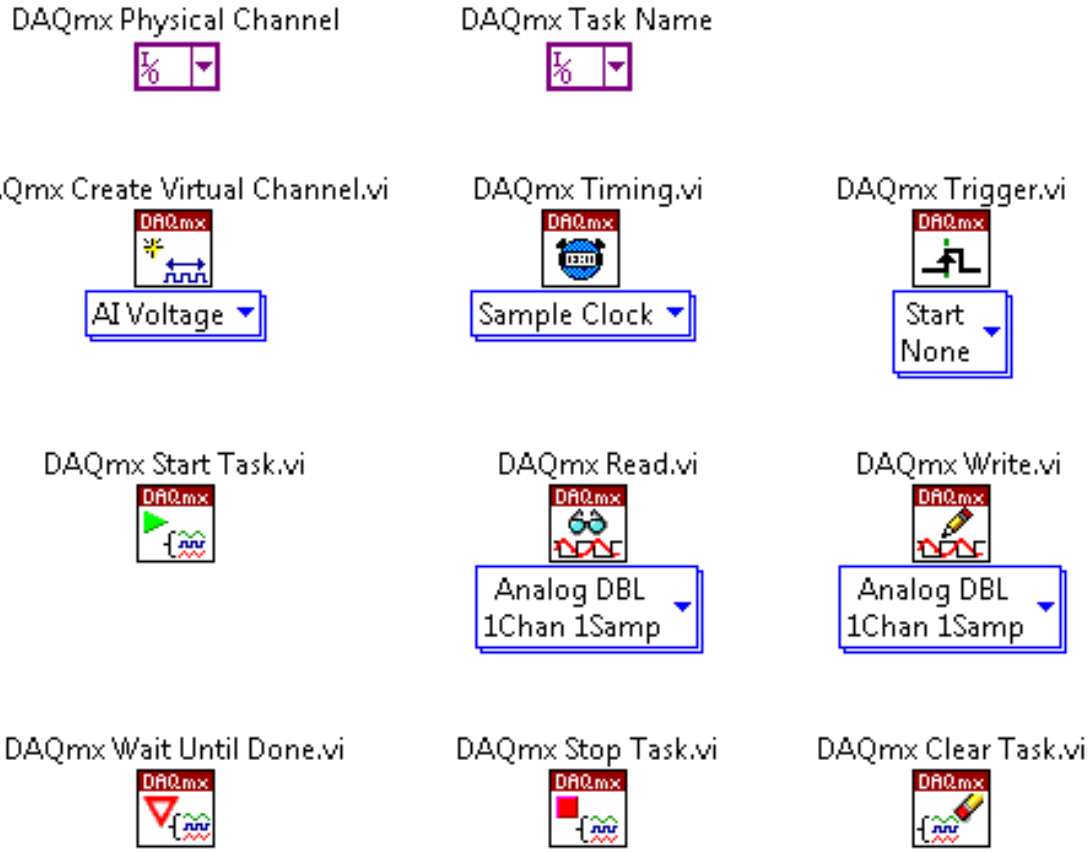
Table 2-1. Simulated CompactDAQ Device for LabVIEW Core 1/Core 2 and CLAD Exam

Slot	Module/Model	Description
1	NI 9236	Strain/Bridge Input
2	NI 9213	Thermocouple Input
3	NI 9472	Digital Output
4	NI 9263	Voltage Output
5	Empty	N/A
6	Empty	N/A
7	NI 9234	Sound and Vibration Input
8	NI 9215	Voltage Input

DAQmx VIs Included in CLAD Exam

The candidate must recognize and be familiar with the function of the following NI-DAQmx constants and VIs, found in the Functions » Measurement I/O » DAQmx palette. Any Acquire/ Analyze/Visualize/Log task will use similar VIs, regardless of the actual hardware used. The example questions in this Prep Guide should clarify the required degree of familiarity.

- Constants: DAQmx Physical Channel, DAQmx Task Name
- Configuration VIs: DAQmx Create Virtual Channel, DAQmx Timing, DAQmx Trigger
- Start & I/O VIs: DAQmx Start, DAQmx Read, DAQmx Write
- End VIs: DAQmx Wait Until Done, DAQmx Stop Task, DAQmx Clear Task



APPENDIX II: CORRECT RESPONSES FOR EXAMPLE QUESTIONS

Hardware

Q1: B

Q2: D

Q3: B

Q4: C

Q5: A

Q6: C

LabVIEW Programming Environment

Q1: C

Q2: D

Q3: C

Q4: B

Q5: D

Q6: C

LabVIEW Programming Fundamentals

Q1: C

Q2: C

Q3: B

Q4: D

Q5: A

Q6: B

Q7: C

Q8: C

Q9: D

Q10: C

Programming Best Practices

Q1: A

Q2: B

Q3: C

Q4: D

Q5: B