

**LabVIEW 基礎專業認證 (CLAD)**

**模擬測驗試題**

**注意：測驗過程中，禁止使用電腦或任何參考資料。**

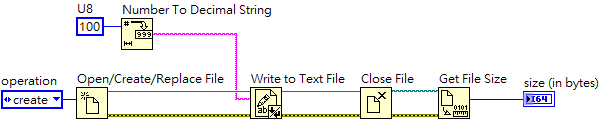
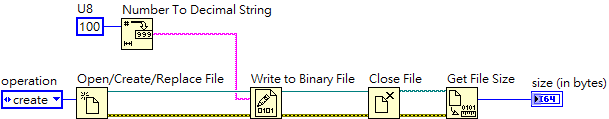
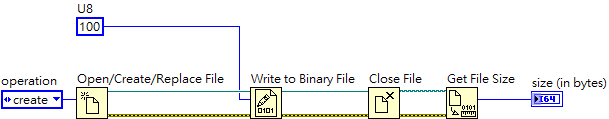
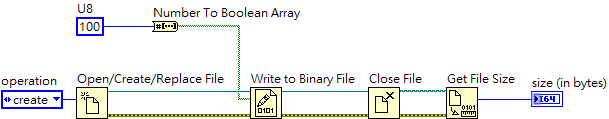
**說明：**

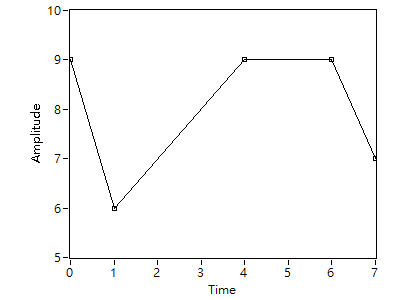
* **請勿擅自撕開任何彌封。若交還 NI 的測驗試題不完整或彌封破損，則將立即失去測驗資格。**
* 請遵守試卷上的要求。如果您填錯了准考證號碼，本次測驗將視作無效。
* 請在答案卡上填寫所有的答案。填寫在試卷上的答案將不納入計分標準。
* 請勿要求監考人員給予任何協助。若認為題目不甚清楚，則可標記該問題，並寫下您自己認為較合適的答案與理由，以供 NI 閱卷時參考之用。
* 您不可攜出此份模擬試題，亦不可用任何方式重現此份試題。在完成測驗之後，亦不可保留任何試題。

**測驗資訊：**

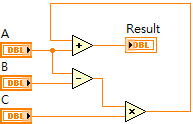
* 測驗時間：1 小時
* 測驗方式：多重選擇題
* 測驗題數：40 題
* 及格分數： 70%

**重要：當應試者完成測驗時，請將試卷放在所提供的信封內並密封之。將密封的信封交給 NI 監考人員。**

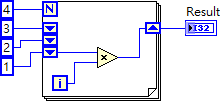
1. 下列何項端點將顯示 For Loop 或 While Loop 的迴圈循環計數 (Loop Iteration) ？
2. 
3. 
4. 
5. 
6. 若欲將數值 **100** 資料型態為 Unsigned Byte 儲存至文件檔，下列何種儲存方式所生成的文件檔案最小？
7. 
8. 
9. 
10. 
11. 若要顯示下圖數據，應使用哪一種圖表？



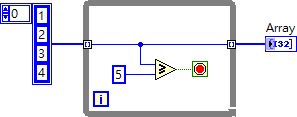
1. Waveform Chart
2. Waveform Graph
3. XY Graph
4. Dynamic Chart
5. 對 subVI 而言，若「**Context Help**」視窗中的輸入名稱是以粗體顯示，則下列何項條件正確？(請選出所有適用的敘述)
6. 輸入值必定為純數 (Scalar)
7. 此輸入為建議使用，而並非必要輸入
8. 此輸入為必要輸入
9. 若此輸入並未接線，則原 VI (Calling VI) 將中斷
10. 下列何項算數函式將優先執行？



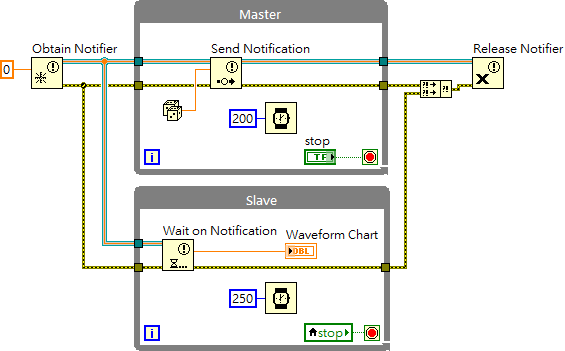
1. 加
2. 減
3. 乘
4. 無法確定
5. 執行下列 VI 之後，**Result** 指示元顯示為何？



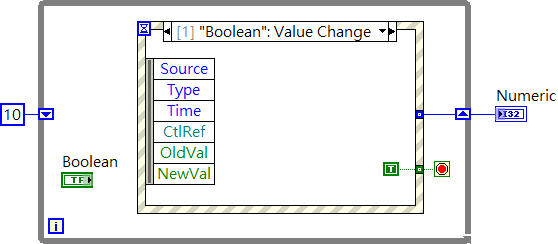
1. 0
2. 3
3. 6
4. 18
5. 何項變數可避免同步存取資料或程式碼時，會將競賽狀態 (**Race condition**) 降至最低？
6. 功能型全域變數 (Functional Global Variable)
7. 區域變數 (Local Variable)
8. 全域變數 (Global Variable)
9. 共用變數 (Shared Variable)
10. 下列關於變數 (**Variable**) 的功能描述何者有誤？
11. 區域變數 (Local Variable) 可於單一 VI 中傳遞資料
12. 共享變數 (Shared Variable) 可於單一專案中的多個 VI 之間傳遞資料
13. 全域變數 (Global Variable) 可於單一電腦中的多個 VI 之間傳遞資料
14. 功能型全域變數 (Functional Global Variable) 可跨多個電腦的多個 VI之間傳遞資料
15. 下列何項不是 LabVIEW 既有的除錯功能？
16. 停止值 (Stop Values)
17. 標示執行 (Highlight Execution)
18. 單步執行 (Single Stepping)
19. 中斷點 (Breakpoint)
20. 關於下列程式碼的描述，何項正確？

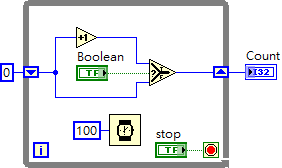
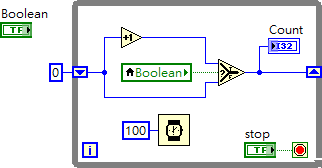
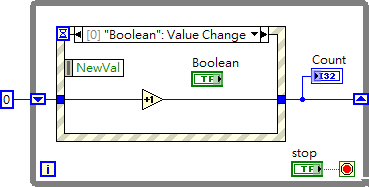
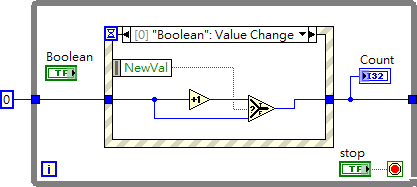


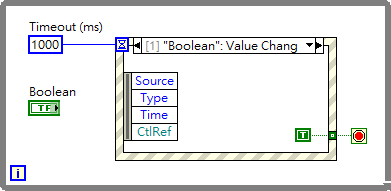
1. While Loop 將循環 1 次之後停止
2. While Loop 將循環 4 次之後停止
3. While Loop 將循環 5 次之後停止
4. While Loop 將無限循環
5. 若使用通知器 (Notifier) 同步兩組平行迴圈如下圖，將會產生什麼問題？



1. 因為通知器 (Notifier) 未具備 Timeout ，故會產生時序的問題
2. 因為通知器 (Notifier) 並不會緩衝資料，故會產生資料遺失的問題
3. 因為通知器 (Notifier) 只傳送單一數值，故會產生重複讀取的問題
4. 因為 Master 迴圈所傳送的資料， Slave 迴圈可能不會讀取，故會產生記憶體累加的問題
5. 執行下列程式碼，若按下 Boolean 按鍵後，則 **Numeric** 顯示器的數值為何？

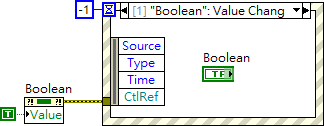
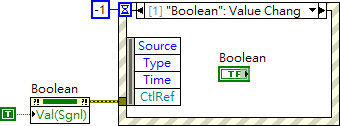
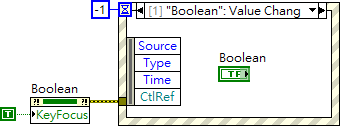
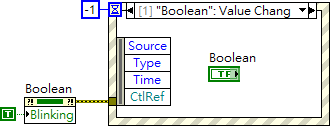


1. 0
2. 10
3. 無法確定，應參考其他 Event Case 的程式碼，才可決定該值
4. 此 VI 不會停止
5. 下列何項 Property Node 可在 subVI 中變更任意數字顯示器 (Digital Numeric) 的色彩設定？
6. 
7. 
8. 
9. 
10. 下列選項中，何時可使用 **Probe** 工具取代 Highlight Execution？
11. 減緩 VI 執行速度，並於接線上顯示數值
12. 呈現資料流向
13. 觀看接線上的即時數值
14. 於 VI 程序執行期間觀看 SubVI
15. 將Boolean 控制元設定為 **Latching Until Released**。若使用者按下 Boolean 控制元，則下列何項程式碼片段 (Snippet) 所回傳的Count 將顯示次數？
16. 
17. 
18. 
19. 
20. 若有設定 **Timeout** 的 Event Case ，則下列 Event 架構的 Timeout 事件將於何時執行？



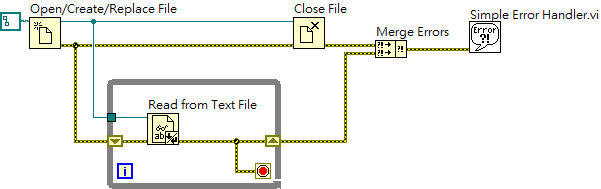
1. 若未發生其他事件，將於 1000 ms 之後執行
2. 不論是否發生其他事件，都將於 1000 ms 之後執行
3. 程式開始 1000 ms 之後，且使用者需按下Boolean 按鈕才會執行
4. 程式開始 1000 ms 之後，且發生其他事件才會執行
5. 此 Dequeue Element 函式要等待多久才可接收資料？



1. 1 millisecond (預設無接線)
2. 1 second (預設無接線)
3. 無限期等待
4. 不會等待且立刻回傳
5. 下列何者程式碼可以執行完程式碼，而不會無限期等待？
6. 
7. 
8. 
9. 
10. 應如何記載程式碼說明 (Document) VI，才能讓滑鼠游標移至 VI 圖示上方時，可於「Context Help」視

窗中顯示該 VI 的說明？

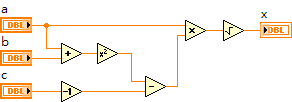
1. 在 VI Properties 對話框中設定 VI Description 欄位
2. 在「Show Context Help」視窗中鍵入訊息
3. 於人機介面上建立名為「VI Description」的自由標籤 (Free label)
4. 編輯 LabVIEW 的輔助說明檔案
5. 假設啟動了自動錯誤處理 (Automatic error handling) 功能，則 LabVIEW 是否會於執行下列程式碼結束時顯示錯誤?



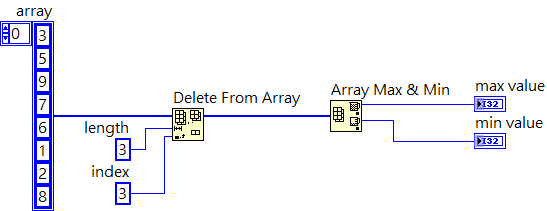
1. 否，此 VI 將不會停止
2. 否，因為 Simple Error Handler.vi 將會繼續記錄後續錯誤
3. 是，且 Merge Errors 函式將只會顯示首次錯誤
4. 是，但因為 Merge Error 函式會收集全部錯誤，所以 LabVIEW 將顯示錯誤對話框達 2 次
5. 執行下列 VI 之後， **Data** 指示元將顯示何值？



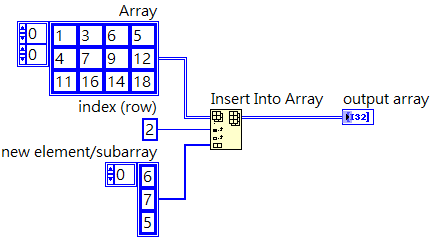
1. (Data x 5) + 2
2. (Data + 2) x 5
3. Data + 2
4. 無法確認
5. 何項方程式等於下列程式碼？



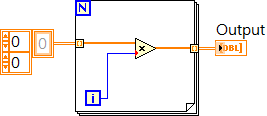
1. 在頂層 VI 上有 1 組必須透過 subVI 才能控制的人機介面控制元，則必須將何項傳送至此 SubVI ？
2. 控制元的屬性 (Property)
3. 控制元的方法 (Method)
4. 控制元的參考 (Reference)
5. 控制元的資料型態 (Data Type)
6. 對於 **Context Help** 顯示接頭面板端點,下列何項敘述正確？
7. 建議的輸入將以粗體字顯示
8. 必要的輸入將以粗體字顯示
9. 選用的端點將以斜體字顯示
10. 所有端點均以相同方式顯示
11. 下列何項程式碼可以創建包含兩條曲線的XY Grapg？
12. 創建包含兩組 X Array 以及 Y Array 的叢集 (Cluster)
13. 將一對 X Array 和 Y Array 綑綁成兩個叢集 (Cluster) ，再將兩組叢集創建成陣列 (Array)
14. 將一對 X Array 和 Y Array 綑綁成兩個叢集 (Cluster) ，再將創建包含此兩組叢集的叢集
15. 創建包含單個 X Array 和 Y Array 的叢集 (Cluster)
16. 下列程式碼執行過後， max value 和 min value 顯示的數值分別為何？



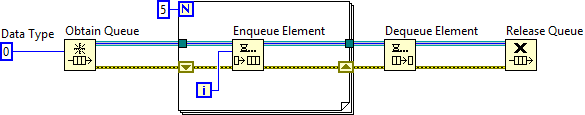
1. max value = 9, min value = 1
2. max value = 9, min value = 2
3. max value = 8, min value = 2
4. max value = 8, min value = 3
5. 執行下列程式碼之後， output array 顯示器的數值為何？



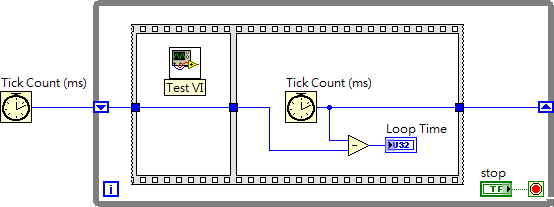
1. 
2. 
3. 
4. 
5. 針對屬性節點 (**Property Nodes**) 的預設行為，下列何項正確？(請選出所有適用的敘述)
6. 屬性節點 (Property Nodes) 可讓使用者透過程式設計的方式,進而設定人機介面物件的屬性
7. 屬性節點 (Property Nodes) 可更新人機介面物件中的數值
8. 單一人機介面的物件可使用超過 1 組的屬性節點 (Property Nodes)
9. 若屬性發生錯誤,則 LabVIEW 將處理其他屬性,並回報第一個發生的錯誤
10. Producer/Consumer 的設計模式會使用下列何者傳送資料以避免資料流失的問題？
11. 通知器 (Notifier)
12. 序列 (Queue)
13. 號誌 (Semaphore)
14. 區域變數 (Local Variable)
15. 此 VI 是否造成破裂的執行箭頭？

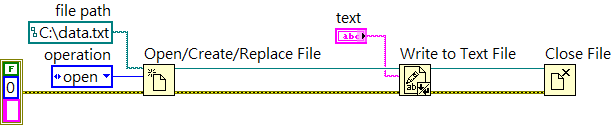
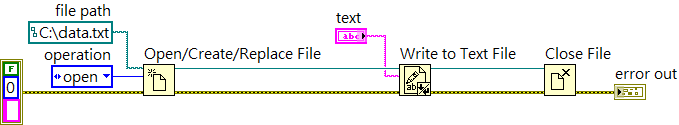
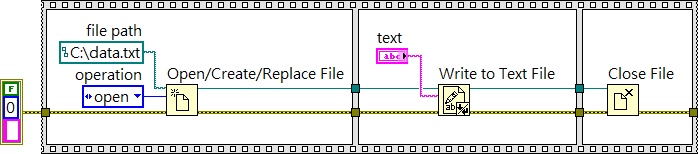
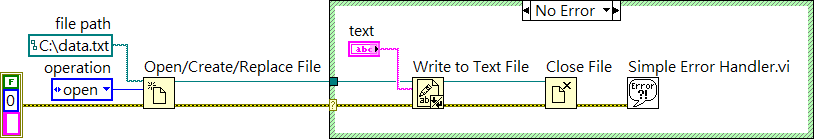


1. 是，因為迴圈次數 Count 未接線
2. 是， For 迴圈輸入為空陣列
3. 是，乘法器出現小紅點 Coercion Dot
4. 否
5. 下列何者為 Dequeue Element 函式的輸出？



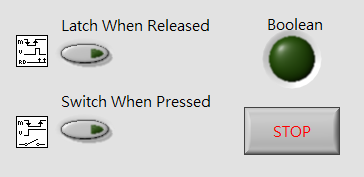
1. [0 1 2 3 4]
2. [4 3 2 1 0]
3. 0
4. 4
5. 下列程式碼用來計時 Test VI 在每次迴圈的運算時間須注意什麼樣的問題？

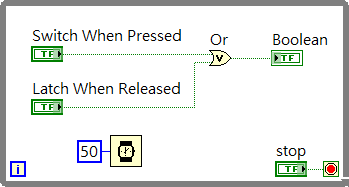


1. 兩個 Tick Count 所讀的時間來源不同
2. 使用 Tick Count 會有溢位重設 (Roll over) 的問題
3. Tick Count 輸出的資料型態為 U32 的正整數，資料相減會發生錯誤
4. 在主迴圈內使用的 Tick Count 函式和 Test VI 內使用的讀值不同，故無法進行計時
5. 若檔案「C:\data.txt」並不存在，則下列何項程式碼片段會跳出視窗回報錯誤？
6. 
7. 
8. 
9. 
10. 下列程式碼執行後， Round to Nearest 函式的輸出為何？

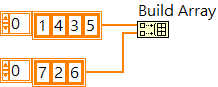


1. 具正負號的 32 位元整數，其值為 4
2. 具正負號的 32 位元整數，其值為 5
3. 雙倍精度的浮點數，其值為 4
4. 雙倍精度的浮點數，其值為 5
5. 執行程式 5 秒後按下 Latch When Released 按鍵，等待 3 秒放開Latch When Released 按鍵，接著經過 2 秒後按下 Switch When Pressed 按鍵，等待 5 秒放開 Switch When Pressed 按鍵，請問 Boolean 顯示器於執行程式後 7 秒和 12 秒時分別顯示何值？

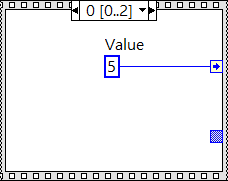
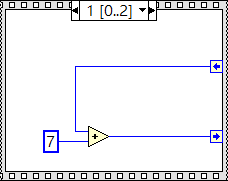
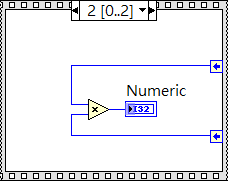




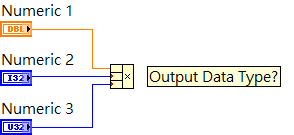
1. False, False
2. False, True
3. True, False
4. True, True
5. 若取消「**Concatenate Inputs**」的設定，則 Build Array 輸出的結果為何？



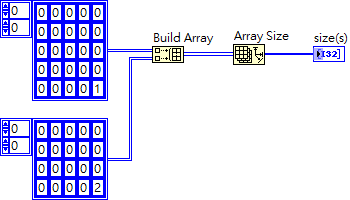
1. 下列 Stacked Sequence 架構的程式碼執行過後輸出的 Numeric 數值為多少？

1. 12
2. 35
3. 60
4. 144
5. 下列三種資料型態不相同的數值相乘後的輸出結果資料型態為何？



1. Double Precision Floating Point, DBL
2. Long Integer, I32
3. Unsigned Long Integer, U32
4. Unsigned Long Long Integer, U64
5. 當Build Array 選定 Concatenate Inputs 之後，計算輸出陣列大小的 size(s) 值將為何？



1. [2 5 5]
2. [2 4 5]
3. [5 9]
4. [5 10]
5. 「**Wait (ms)**」與「**Wait Until Next ms Multiple**」函式行為的差異是？
6. 首次呼叫「Wait Until Next ms Multiple」的時間延遲,可能短於接至函式輸入端點的特定 ms 秒數
7. 除非完成該程式碼區段的其他所有函式,「Wait (ms)」函式才會開始計時作業
8. 若某架構已包含「Wait Until Next ms Multiple Function」函式,則可於獨立執行緒中執行
9. 「Wait (ms)」函式可讓迴圈不致佔用 100% 的 CPU 資源;「Wait Until Next ms Multiple」則無此項功能

**解答**

1. A  
   A : Loop Iteration ; B : Loop Condition (Stop if True) ; C : Loop Condition (Continue if True) ; D : Loop Count
2. C  
   A : 數字100轉成字串共3個字元，每個字元佔1 byte，寫入文字檔案共3 bytes  
   B : 字串和陣列寫入二進位檔預設會寫入4 bytes來表示字元數目或陣列大小以利讀檔，故共3+4=7 bytes  
   C : 資料型態為U8的數字只佔1 byte，故直接寫入二進位檔只佔1 byte  
   D : 由於LabVIEW中資料的最小單位為byte，即便是一個Boolean值也是用byte來儲存，而資料型態為U8的數字轉成Boolean Array的陣列大小為8，故寫入二進位檔共8+4=12 bytes
3. C  
   資料分布均勻而點距相同的資料用Waveform Chart或Waveform Graph顯示，而資料分布不均勻而點距皆不相同的資料則用XY Graph來顯示
4. C, D  
   粗體表示此輸入項為必要輸入(Require Input)，若未接線(Unwired)則此VI將無法執行
5. B  
   依資料流，“加”函式需要A和"乘”函式的輸出結果輸入才能執行；“乘”函式需要C和"減”函式的輸出結果輸入才能執行；“減”函式只需要A和B輸入即能執行，故最先執行的為”減”函式
6. A  
   1st : i(0) x 1 = 0 ; 2nd : i(1) x 2 = 2 ; 3rd : i(2) x 3 = 6 ; 4th : i(3) x 0(1st Result) = 0
7. A  
   功能型全域變數將記憶體定址至移位暫存器中，且VI設為Non-Reentrant不會有同步存取資料的問題
8. D  
   可跨電腦的多個VI之間傳遞資料的是透過網路傳輸的共享變數(Shared Variable)
9. A  
   停止值只用於停止VI運作而不能用於程式碼除錯
10. D  
    陣列中的數值在index大於3時未定義數值，Indexing功能輸入至While Loop為預設值0，故陣列中的所有數值將無法達成≧5的條件，While Loop將無限循環
11. B  
    通知器(Notifier)不會緩衝資料，故當資料寫入迴圈的時間小於資料讀取迴圈的時間時，將會產生資料遺失的問題
12. A  
    按下Boolean按鍵將會觸發Event Structure，Case中輸出至移位暫存器的數值為預設值0，且While Loop在此事件觸發後將結束迴圈，故迴圈結束輸出至Numeric顯示元的數值為0
13. B  
    A : strict只能變更特定數字顯示器色彩不能通用  
    B : 符合題目要求，可輸入任意數字顯示器的Reference到subVI中透過Property Node變更色彩  
    C : 只能和數字顯示器在同一個VI中變更，不能透過subVI  
    D : 變更的是單位標籤的色彩而非數字顯示器的色彩
14. C  
    A和B為打開Highlight功能除錯，Probe的功能為在VI正常執行時觀看接線上的即時數值
15. D  
    A : 迴圈結束後才會顯示計數值  
    B : Boolean按鍵的開關狀態為Latch時無法使用Local Variable，故此程式碼將無法執行  
    C : 按下Boolean按鍵即計數一次，但放開按鍵將會再重複計數一次  
    D : 加上Switch可避免按鍵重複計數的問題，只在按下按鍵時計數一次
16. A  
    在未有任何事件發生達Timeout輸入值時才會觸發Timeout事件
17. C  
    Dequeue Element的Timeout預設值為 -1，即會無限期等待資料接收
18. B  
    Property Node中的Value(Signal)可以觸發Event Structure
19. A  
    VI Property對話框中的VI Description欄位的輸入說明將會顯示於此VI的Context Help中
20. C  
    Open File將因檔案路徑錯誤而輸出Error，Close File接收到Error不會執行並將Error輸出至Merge Errors；下面迴圈會因Open File產生的Error而停止迴圈，並將Error傳至Merge Errors。雖然輸入至Merge Errors的有兩個Errors，但Merge Errors只會將最先讀到的Error輸出到Simple Error Handle，故只會出現一次錯誤視窗
21. D  
    使用Local Variable將無法透過接線上的資料流來確定程式執行順序，故此段程式碼將無法確認執行後的數值
22. A
23. C  
    控制元的參考(Reference)可輸入至subVI再透過Property Node以及Invoke Node來調整該控制元
24. B  
    必要輸入(Require Input)在Context Help中將以粗體表示
25. B  
    包含兩組陣列分別代表X和Y座標的叢集Build Array輸入至XY Graph可顯示多條曲線
26. B  
    從原陣列中刪除 [7 6 1] ，剩餘陣列 [3 5 9 2 8] 中最大值為9，最小值為2
27. C  
    將 [6 7 5] 插入第三列，空白處將會補0，原先第三列將會移到下一列
28. A, B, C, D
29. B
30. D  
    此程式碼可以執行，但For Loop執行次數為0次
31. C  
    Queue的行為是先進先出，故最先排進Queue的元素為0，第一個被取出的元素也將是0
32. B  
    因Tick Count計數的資料型態為U32，當計數器發生溢位時，相減會出現負值
33. A  
    A選項的Close File的Error Out為接線，故發生錯誤將會跳出錯誤視窗；而B和C選項因Close File的Error Out有接線至顯示元(Indicator)或結構(Structure)上，故即使發生錯誤也不會跳出錯誤視窗；D選項在Open File的Error Out接線至Error Case，發生錯誤時將不會執行 ”No Error” Case中的程式碼，故不會跳出錯誤視窗
34. A  
    Round To Nearest會選擇最靠近輸入數值的”偶數”輸出，故4和6中最靠近4.5的為4
35. B  
    Latch When Released需等到按下按鍵後放開，其狀態才會轉為True，並在程式讀取後回復至False。在7秒時仍未放開Latch When Released按鍵，故Boolean值為**False**；而Switch When Pressed則在按下按鍵時即轉變狀態為True，且需在下一次重新按下時才會回復成False，故程式在12秒時的Boolean值為**True**
36. D  
    若Build Array輸入陣列大小不相等時，其輸出陣列大小會以最大的為主，而空的值會補0
37. C  
    5 x (5+7) = 60
38. A  
    因資料型態需相同才能運算，當出現Coercion Dot時，程式會強迫將其轉換為最通用的資料型態，而圖中最通用的資料型態為雙倍精度浮點數(DBL)
39. C  
    選定Concatenate Input後，輸出為二維陣列而非三維陣列，故其陣列的Column數目相同，而Row數目相加即為輸出陣列大小
40. A  
    A : 因Wait Until Next ms Multiple需上一次迴圈的結束時間來決定需延時多久以達到輸入值，故迴圈首次執行的時間可能會縮短  
    B : Wait函式為迴圈一開始即執行延時，而非迴圈內程式執行完成之後  
    C : 無此函式功能  
    D : 兩者皆可讓迴圈延時而不致占用100%的CPU資源