

**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：**

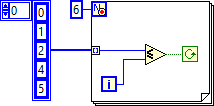
* **請勿擅自撕開任何彌封。若交還 NI 的測驗試題不完整或彌封破損，則將立即失去測驗資格。**
* 請遵守試卷上的要求。如果您填錯了准考證號碼，本次測驗將視作無效。
* 請在答案卡上填寫所有的答案。填寫在試卷上的答案將不納入計分標準。
* 請勿要求監考人員給予任何協助。若認為題目不甚清楚，則可標記該問題，並寫下您自己認為較合適的答案與理由，以供 NI 閱卷時參考之用。
* 您不可攜出此份模擬試題，亦不可用任何方式重現此份試題。在完成測驗之後，亦不可保留任何試題。

**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。**

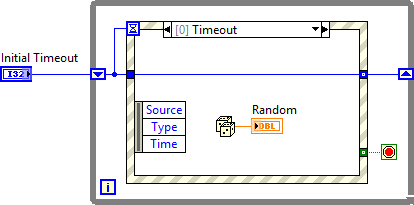
1. 關於 LabVIEW 程式執行程序，下列何者正確？(請選出所有正確的敘述)
2. 程式執行順序由控制元件的流程 (Control Flow) 所決定
3. 程式各元件須等到所有資料輸入節點後，才會執行並將資料輸出至端點
4. 程式執行順序由程式框圖的左邊依序執行到右邊
5. 通過程式各元件節點的資料流將決定程式執行順序
6. 下列 For Loop 將循環幾次？

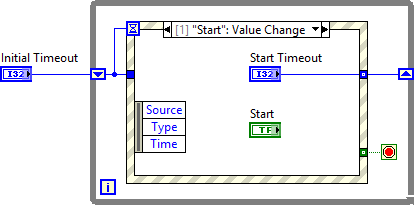


1. 3
2. 4
3. 5
4. 6
5. 使用者產生了共 100 個資料點，且不平均放置於一段時間之內。若維持此不平均的擷取點，則下列何項

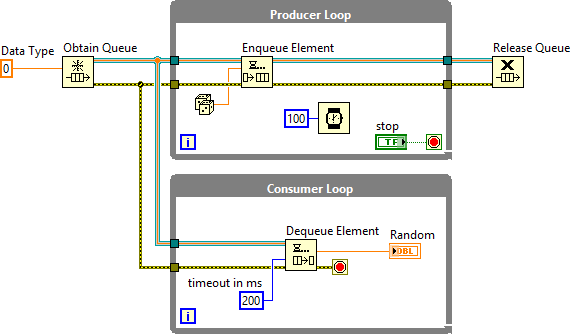
指示元最適於顯示資料？

1. Waveform Chart
2. Waveform Graph
3. XY Graph
4. Intensity Graph
5. 下列何項函式可以將浮點數的數值轉換為帶有小數點的字串？
6. 
7. 
8. 
9. 
10. 若下列程式一開始執行時 Random 不更新，當使用者按下 Start 按鍵， Random 才開始每 1 秒更新一筆亂數，則 Initial Time 跟 Start Time 應設定成何值，程式執行才會有上述功能？

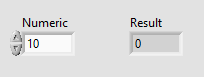


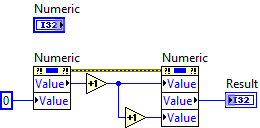


1. Initial Timeout = 1000, Start Timeout = -1
2. Initial Timeout = -1, Start Timeout = 1000
3. Initial Timeout = 1000, Start Timeout = 0
4. Initial Timeout = 0, Start Timeout = 1000
5. 針對下列 Producer/Consumer 程式架構，何者敘述正確？

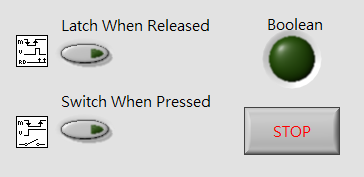


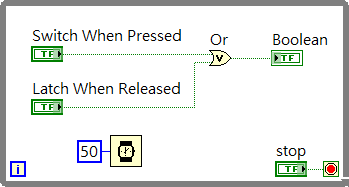
1. Consumer Loop的迴圈更新時間為Producer Loop的兩倍
2. 兩個迴圈之間以 Wait Time 函數的數值來同步化
3. 兩個迴圈之間以 Dequeue Element 的 Timeout 的數值來同步化
4. 兩個迴圈之間以 LabVIEW 內部時序來進行同步化
5. 將 Numeric 設為 10 並執行下列程式碼，則程式結束後 Numeric 和 Result 分別顯示何值？



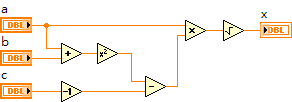


1. Numeric = 2 , Result = 2
2. Numeric = 2 , Result = 1
3. Numeric = 12 , Result = 12
4. Numeric = 12 , Result = 11
5. 執行程式 5 秒後按下 Latch When Release 按鍵，等待 3 秒放開Latch When Release 按鍵，接著經過 2 秒後按下 Switch When Pressed 按鍵，等待 5 秒放開 Switch When Pressed 按鍵，請問 Boolean 顯示器於執行程式後 7 秒和 12 秒時分別顯示何值？

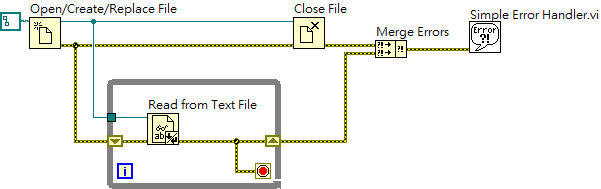




1. False, False
2. False, True
3. True, False
4. True, True
5. 下列何者變數的使用可避免因程式事件時序的不確定而導致的競賽狀態 (Race Condition)？
6. 區域變數 (Local Variable)
7. 全域變數 (Global Variable)
8. 共用變數 (Shared Variable)
9. 功能型全域變數 (Functional Global Variable)
10. 何項方程式等於下列程式碼？



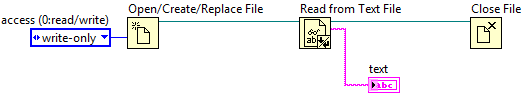
1. 假設啟動了自動錯誤處理 (Automatic error handling) 功能，則 LabVIEW 是否會於執行下列程式碼結束時顯示錯誤?

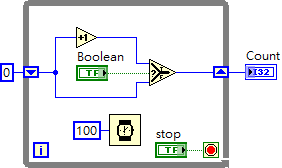
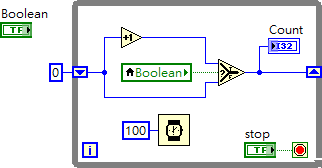
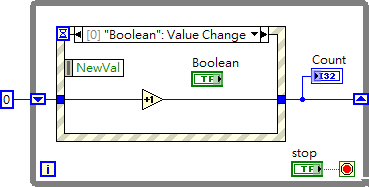
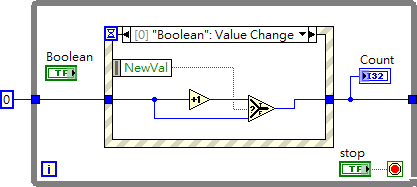


1. 否，此 VI 將不會停止
2. 否，因為 Simple Error Handler.vi 將會繼續記錄後續錯誤
3. 是，因為 Merge Errors 函式將只會顯示首次錯誤
4. 是，但因為 Merge Error 函式會收集全部錯誤，所以 LabVIEW 將顯示錯誤對話框達 2 次
5. 執行下列 VI 之後， **Data** 指示元將顯示何值？



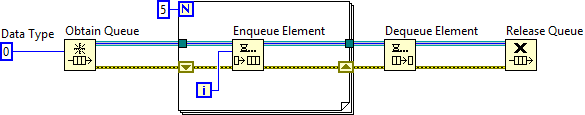
1. (Data x 5) + 2
2. (Data + 2) x 5
3. Data + 2
4. 無法確認
5. 「**Wait (ms)**」與「**Wait Until Next ms Multiple**」函式行為的差異是？
6. 首次呼叫「Wait Until Next ms Multiple」的時間延遲,可能短於接至函式輸入端點的特定 ms 秒數
7. 除非完成該程式碼區段的其他所有函式,「Wait (ms)」函式才會開始計時作業
8. 若某架構已包含「Wait Until Next ms Multiple Function」函式,則可於獨立執行緒中執行
9. 「Wait (ms)」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
10. 執行下列程式碼而發生錯誤的原因為何？



1. 檔案路徑並未接至 Open/Create/Replace File 函式的輸入
2. 未將函式間的錯誤叢集接線
3. 以錯誤存取而開啟檔案 File opened with the incorrect access
4. 以上皆是
5. Q15: 將Boolean 控制元設定為 Latching When 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？
6. 
7. 
8. 
9. 
10. 下列何項敘述錯誤？
11. subVI 的接頭面板 (Connector pane) 將定義輸入/出的接線處
12. subVI 接頭面板 (Connector pane) 端點的顏色,即對應其連接的資料型態
13. subVI 必具備 1 組圖示與接頭面板 (Connector pane)
14. 可透過 Functions Palette 而編輯 SubVI 圖示
15. 建置功能型全域變數 (Functional Global Variable) 時，下列何者為非必要？
16. 只執行一次的迴圈 (While Loop)
17. 為初始化的移位暫存器 (Shift Register)
18. 條件結構 (Case Structure)
19. 枚舉型數值控制元 (Enumerated Type Control)
20. 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）

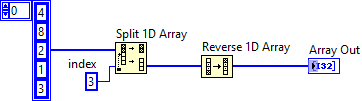
1. 輸入值必為純數 (Scalar)
2. 此輸入為建議使用,並非必要輸入
3. 此輸入為必要輸入
4. 若此輸入並未接線,則原 VI (Calling VI) 將中斷
5. 下列何者為 Dequeue Element 函式的輸出？



1. [0 1 2 3 4]
2. [4 3 2 1 0]
3. 0
4. 4
5. 若 START UP VI 回報錯誤,則狀態機的行為是？



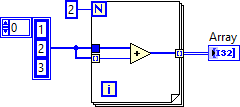
1. 轉為關機 (Shutdown) 狀態
2. 轉為待機 (Idle) 狀態
3. 停止並顯示錯誤對話框
4. 重新執行開機 (Startup) 狀態
5. 執行下列程式碼之後， Array Out 顯示為何？



1. 
2. 
3. 
4. 
5. 在頂層 VI 上有 1 組人機介面控制元，且必須透過 subVI 才能控制之。則必須將何項傳送至此

subVI？

1. 控制元的屬性 (Property)
2. 控制元的方法 (Method)
3. 控制元的參考 (Reference)
4. 控制元的資料型態 (Data Type)
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. 下列選項中，何時可使用 Probe 工具取代 Breakpoints？
11. 要減緩 VI 速度,並於接線中顯示數值
12. 呈現資料的流向
13. 不需暫停執行，亦可檢查接線上的資料
14. 於執行程序期間觀看 SubVI
15. 下列何項可令人機介面上的 LED 指示元顯示紅黃綠三種顏色？
16. 屬性節點 (Property Node)
17. 呼叫節點 (Invoke Node)
18. 區域變數 (Local Variable)
19. 全域變數 (Global Variable)
20. 執行下列 VI 之後，人機介面上的 Array 指示元顯示為何？



1. 
2. 
3. 
4. 
5. \_\_\_\_\_\_\_\_\_\_ 所顯示的完整波形將覆寫已儲存的資料;而 \_\_\_\_\_\_\_\_\_\_\_ 將定期更新資料,並保有先前

已儲存資料的歷史記錄。

1. Graphs; Charts
2. Charts; Plots
3. Plots; Graphs
4. Charts; Graphs
5. VI 是透過接頭面板 (Connector pane) 上的「Error In」與「Error Out」端點而呼叫 subVI。若已啟

動自動錯誤處理 (Automatic error handling) 功能，且 subVI 回傳錯誤，則應如何避免 LabVIEW 顯示錯誤

對話框？

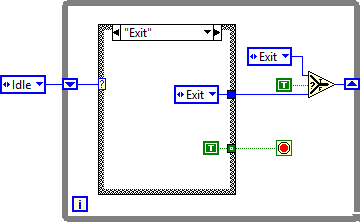
1. 點選 Tools » Options 功能表，並從 Block Diagram 分類中取消勾選Enable automatic error

handling in new VIs

1. 在 VI Properties 中找到 Category Execution，取消勾選Allow Debugging
2. 將 subVI 的「Error Out」端點,連至原 VI (Calling VI) 中的通道 (Tunnel)、端點 (Terminal),或指

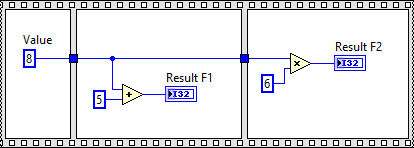
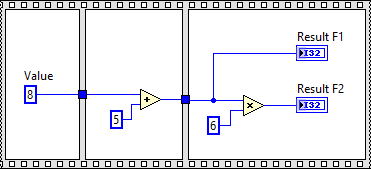
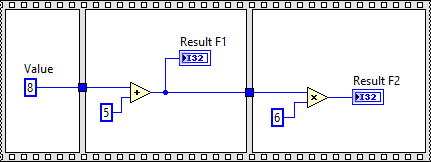
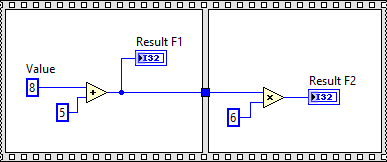
示元 (Indicator)

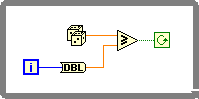
1. 對 subVI 按下滑鼠右鍵，再勾選Disable automatic error handling
2. 下列程式圖中的 While Loop 將循環幾次？



1. 0
2. 1
3. 2
4. 無限次
5. 下列何項不是 LabVIEW 既有的除錯功能?
6. 停止值 (Stop Values)
7. 執行標示 (Highlight Execution)
8. 單步執行 (Single stepping)
9. 中斷點 (Breakpoint)
10. 下列何項程式碼片段 (Snippet) 的功能等於此序列架構？



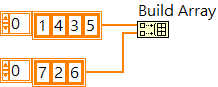
1. 
2. 
3. 
4. 
5. 下列程式碼的 While Loop 將執行幾次？



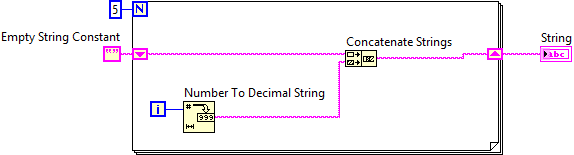
1. 0
2. 1
3. 2
4. 無限次
5. 應如何記載程式碼說明 (Document) VI，才能讓滑鼠游標移至 VI 圖示上方時，可於「Context Help」視

窗中顯示該 VI 的說明？

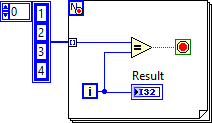
1. 在 VI Properties 對話框中設定 VI Description 欄位
2. 在「Show Context Help」視窗中鍵入訊息
3. 於人機介面上建立名為「VI Description」的自由標籤 (Free label)
4. 編輯 LabVIEW 的輔助說明檔案
5. 當選定「Concatenate Inputs」的設定，則 Build Array 輸出的結果為何？



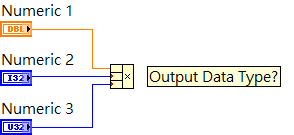
1. 在執行下列程式碼之後，Result 指示元將顯示何字串？



1. 12345
2. 01234
3. 123456
4. 012345
5. 在執行下列程式碼之後，Result 顯示元將顯示何值？



1. 0
2. 3
3. 4
4. 迴圈將不會停止
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. 執行下列 VI 之後，Result 指示元將顯示何值？



1. 5
2. 7
3. 12
4. 36
5. 何時透過設置全域變數 (Global Variable) 來傳送資料最為適當？
6. 不同電腦執行的 VI 之間
7. 主要 VI 與 subVI 之間
8. 平行的 While Loop 之間
9. 序列架構 (Sequence Structure) 的框架 (Frame) 之間
10. 若要定時監控使用者介面的更新，應於迴圈中使用下列何項時序函式？
11. 
12. 
13. 
14. 

**解答**

1. B, D  
   A, C : 程式執行順序由資料流決定
2. B  
   此For Loop設置有Conditional Terminal，當執行到第四次時輸入迴圈的4 > i(3)，未滿足繼續迴圈的條件故即停止迴圈
3. C  
   均勻分布資料用Waveform Chart或Waveform Graph顯示；不均勻分布資料只能用XY Graph來顯示
4. D  
   Format Into String可將任意類型資料轉換為字串
5. B  
   Timeout設為 -1 時，不會發生Timeout事件；當觸發Start按鍵開關事件將Timeout設為1000時，每隔1000ms會觸發Timeout事件來更新亂數
6. D  
   由於Producer每100ms丟入一筆資料，故Consumer不會發生200ms的Timeout，而是依LabVIEW內部時序同步每100ms接收一筆資料
7. D  
   Property Node由最上層開始依序執行，第一個Property Node會先將Numeric預設的10讀出，然後將0寫進Numeric；第二個Property Node會先將前面讀出的10加1寫入Numeric，接著Result再讀出Numeric的數值為**11**，最後將11再加1寫入Numeric中，故程式執行後的Numeric讀值為**12**
8. 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**
9. D  
   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，故不會產生競賽狀態的問題
10. A
11. 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
12. 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
13. 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
14. C  
    A : Open/Create/Replace File函式的檔案路徑未接現在程式執行時會出現視窗選擇檔案而不會發生Error  
    B : 未將函式間的錯誤叢集接線，程式可執行且不會因而發生錯誤
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. D  
    SubVI圖示需進入Icon Editor編輯
17. D  
    除了Enum控制元之外，也可用String控制原來編輯FGV的執行狀態
18. 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
19. 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
20. C  
    回報的錯誤叢集的Status會滿足讓While Loop停止的條件，並且將錯誤叢集輸入Simple Error Handler而跳出錯誤視窗
21. D  
    Split 1D Array從index 3開始分離取出 [1 3] ，再經過Reverse 1D Array後輸出 [3 1]
22. 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
23. A, B, C, D
24. 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
25. A  
    LED指示元可透過屬性節點來改變顯示顏色
26. C  
    純數(Scale)與陣列(Array)相加會使得陣列中每個元素都加上純數，而For Loop執行兩次輸出1D Array結合為2D Array，其陣列大小為2x3
27. A
28. A
29. C  
    While Loop執行第一次會將下一次迴圈的State指向 ”Exit” ，第二次迴圈執行 ”Exit” 狀態即結束迴圈
30. A  
    停止值做為迴圈停止條件，而非LabVIEW除錯功能
31. A  
    Value(8) + 5 = Result F1 ; Value(8) x 6 = Result F2
32. B  
    Random輸出介於0~1之間的亂數，While Loop第一次迴圈亂數大於 i(0) ，故迴圈繼續；第二次迴圈小於 i(1) ，故迴圈中止
33. 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
34. C  
    選定Concatenate Inputs時，輸入Build Array的1D Array將會結合成1D Array而非2D Array，且第二個1D Array將接在第一個1D Array的後面
35. B  
    將For Loop執行5次的Iteration Count值分別為0~4轉換成字串再相接，故輸出為”01234”
36. B  
    輸入For Loop的Indexing Array中的元素有4個，故For Loop最多會執行四次，且其中陣列元素皆不會等於i值(0~3)，故不會滿足提前停止條件，For Loop執行4次後的i值為3
37. 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)
38. D  
    輸入6為Default Case，故輸出為62 = 36
39. B  
    A選項為Shared Variable的使用場合；C和D選項皆為同一個VI中，故為Local Variable的使用場合
40. B  
    除了B選項可延時設定迴圈更新時間之外，其餘選項皆是讀取目前時間