

一、【實驗目的】：

What was your design? What were the concepts you have used for your design?

Q1: 定義一個含有 10 個無號整數的陣列，並計算它們平方和，結果放在 R4。

Q2: 比較兩個日期（年 → 月 → 日），如果 $date1 < date2$ ，則 $R0 = 1$ ；否則 $R0 = -1$ 。

Q3: 反向點積 $result = a1 \cdot b_n + a2 \cdot b_{n-1} + \dots + a_n \cdot b_1$ ，將結果儲存在 R4。

二、【遭遇的問題】：

What problems you faced during design and implementation?

Q1: 計算平方和時，我一開始忘記陣列元素是 32-bit，導致位址偏移量寫錯。

Q2: 日期比較的時候，邏輯上需要依序判斷年、月、日，但我一開始分支條件寫得太複雜，容易出現判斷混亂。

Q3: 反向點積比較麻煩，因為 A 陣列要從前面往後，B 陣列要從後面往前，我在處理指標遞增與遞減時常常混淆，還有一開始不知道要先把 B 的指標移到最後一個元素。

三、【解決方法】：

How did you solve the problems?

Q1: 我重新確認了 ARM 組合語言 LDR 指令的位址計算方式，加入正確的位移 (LSL #2) 才成功存取到每個陣列元素。

Q2: 我將邏輯簡化成「先比年，再比月，最後比日」的三層結構，這樣程式就清楚很多，也不會漏判。

Q3: 我利用一個計數器控制迴圈次數，並在程式開始時先把 B 陣列的位址移動到最後一個元素，再用 $[Rn], \#-4$ 的方式往前取值。經過多次模擬與檢查，才確認乘積與加總的結果正確。

在找尋這些問題的解決方法與問題點時，我有使用 ChatGPT 協助我找尋與解決問題。包含實驗結報的內容修改與潤飾都有使用 ChatGPT 協助。

四、【未能解決的問題】：

Was there any problem that you were unable to solve? Why was it unsolvable?

這次三個題目基本上都能完成，沒有遇到嚴重無法解決的錯誤。

唯一的疑慮是例如在 Lab1.1 和 Lab1.3，如果陣列元素數值過大，可能會造成加總結果溢位，但並沒有特別要求處理溢位，因此我沒有額外作。