**X86 Assembly Language**

**Final Project Report Format**

**General Format:**

Your final project reports are to be typed, double spaced, with normal one-inch margins. Be careful to have correct spelling and proper grammar, as these will be taken into consideration when your report is graded. The sections of the report designated as **required** *MUST* be included in report. Those marked as **optional** are required if so indicated in the assignment document.

**Title Page (required):**

The title page should contain:

Title

Final Project Report

Author

X86 Assembly Language, Fall 2018

Date Submitted

The following page contain an example of a title page

Project Title

Final Project Report

By

406261523

康智詠

406261597

林子傑

X86 Assembly Language

Fall 2018

Date Submitted: December 25, 2018

**Abstract (required):**

**xchg**

輸入一個整數n，假如n <= 0 或 n>100 結束程式,之後輸入n筆數字，會輸出排序好的n筆數字。

**Table of Contents:**

**Introduction (required):**

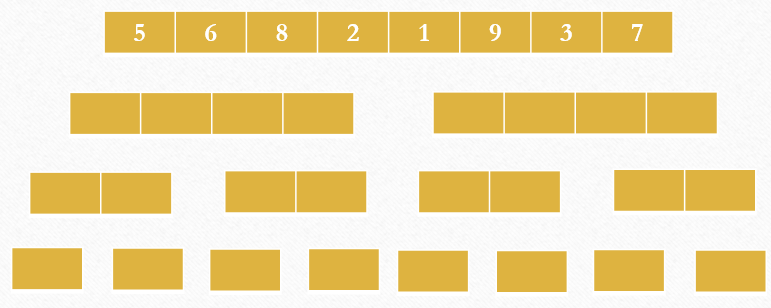
**合併排序法，是排序演算法的一種，使用Divide and Conquer的演算法來實作。排序時需要額外的空間來處理，過程依照以下步驟進行：**

**1. 將陣列分割直到只有一個元素。**

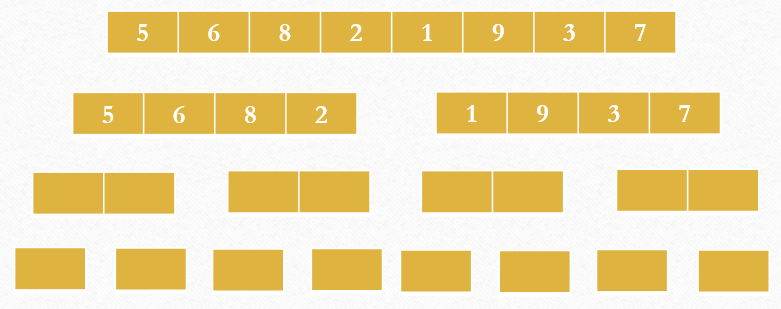
**2. 開始兩兩合併，每次合併同時進行排序，合併出排序過的陣列。**

**3. 重複2的動作直接全部合併完成。**

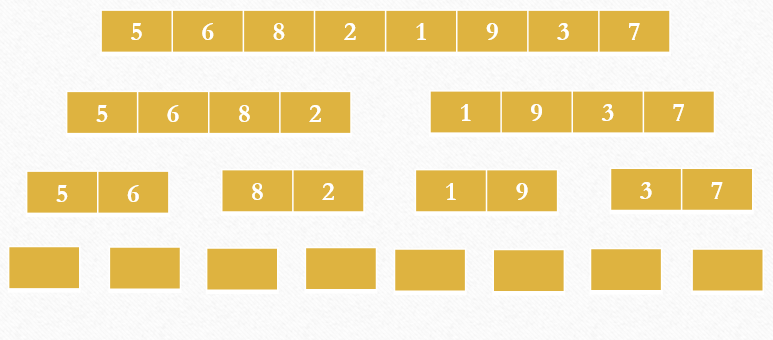
**Step1:**

****

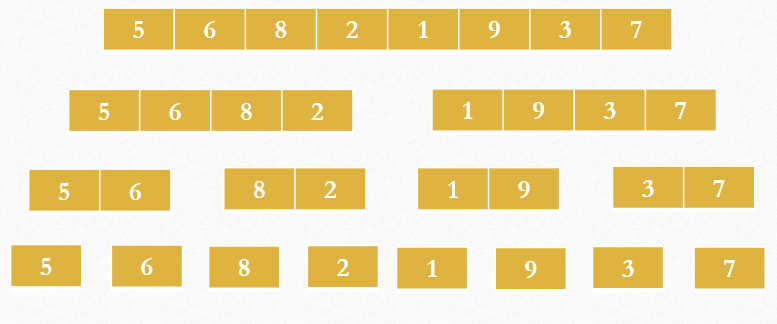
**Step2:**

****

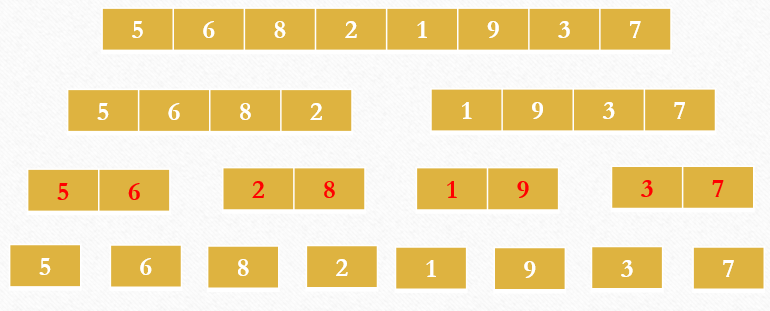
**Step3:**

****

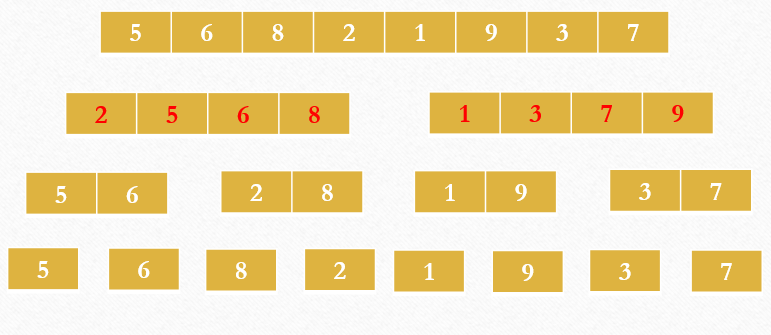
**Step4:**

****

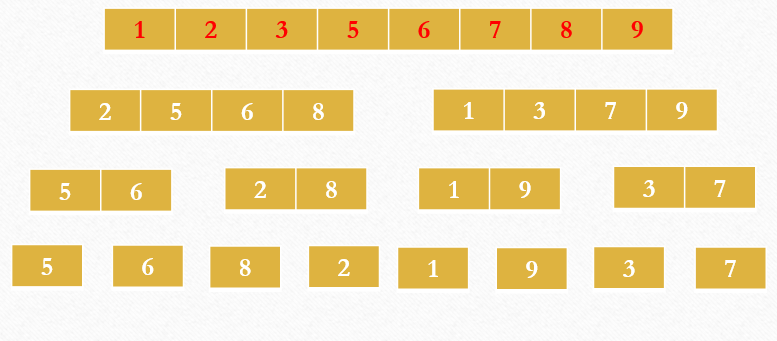
**Step5:**

****

**Step6:**

****

**Step7:**

**:**

**Project plan or Test Plan and TestCases (required):**

第一筆測資為正常輸入輸出

第二筆測資為多筆輸入輸出與不正常輸入

第三筆測資檢測輸入為0是否直接結束

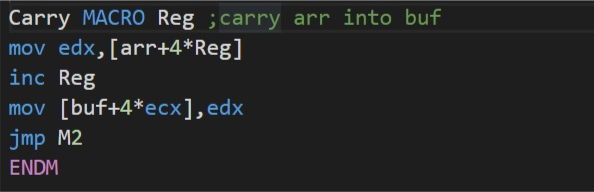
|  |  |  |
| --- | --- | --- |
| **Test Case Number** | **Input Values** | **Expected output** |
| 1 | 5  4  9  5  2  7  0 | 2 4 5 7 9 |
| 2 | 7  2  5  9  3  14  67  23  5  8  90  47  23  5  -1 | 2 3 5 9 14 23 67  5 23 47 90 |
| 3 | 0 |  |

**Debug Log:**

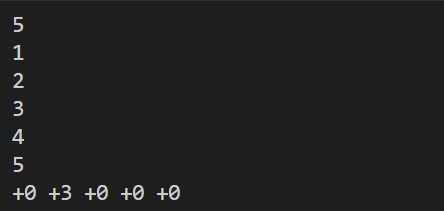
**狀況1：沒有輸出**

****

**發現：Macro 沒寫好(jmp M2)，造成無限迴圈。**



**狀況2：印出來的東西不對**

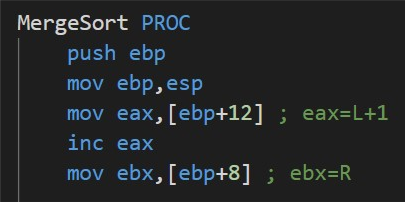
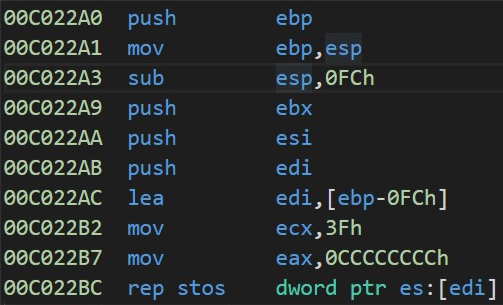
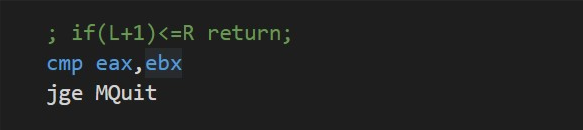
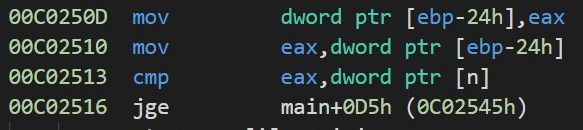
****

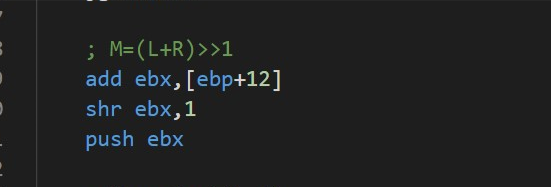
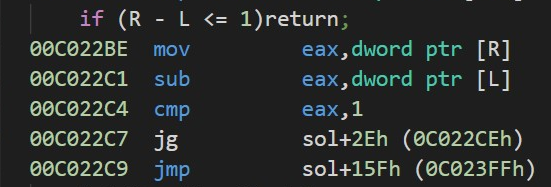
**發現：我們設的全域變數Ｍ會隨著遞迴而改變，在return時Ｍ沒有改回來，所以造成錯誤。**

**解決：將M放進Stack，return後再回覆。**

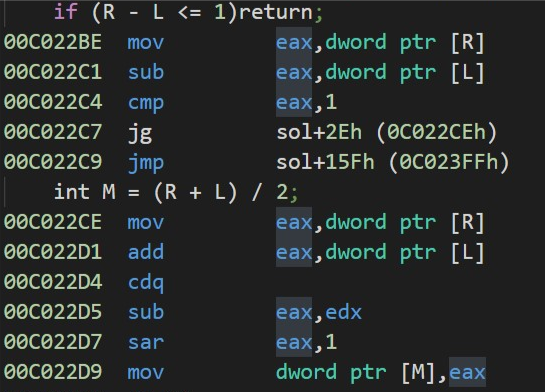
**Discussion or Analysis (required):**

**比較自己寫的(以下稱版本A)及Visual Studio反組譯C/C++產生的(以下稱版本B)程式碼**

1. **呼叫函式後的初始化：版本A(左圖)只有將ebp儲存在堆疊中，而版本B(右圖) 為了保護資料不被意外執行，多了一些清空記憶體的指令** ****
2. **執行位置跳轉：版本A會傳入自己設的標籤(例如上圖的MQuit)。版本B則是傳遞位置(例如下圖的main+0D5h)** **** 
3. **讀取傳入參數：版本A會使用。版本B則直接利用**

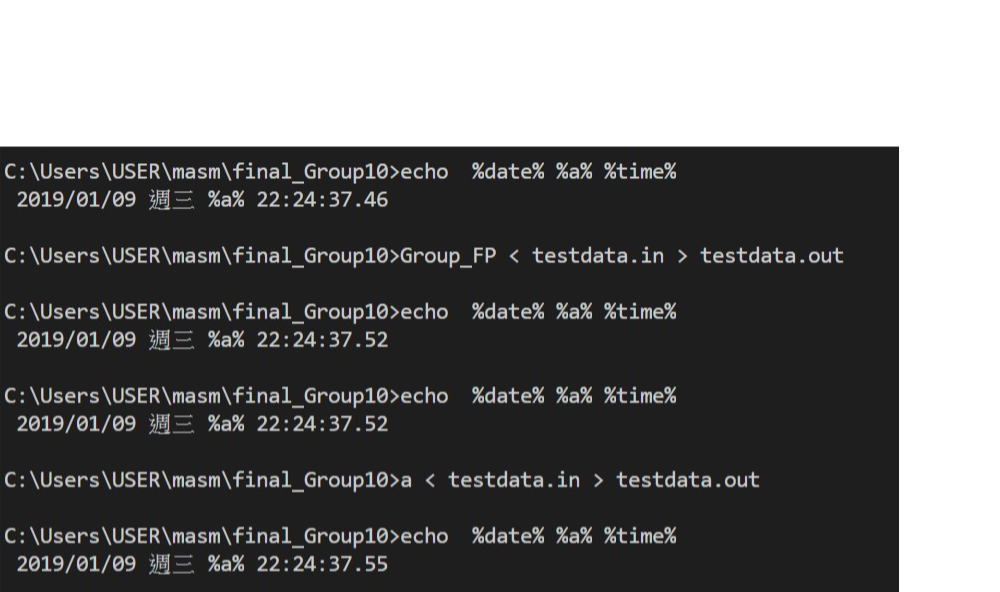
** **

1. **B版本的缺點：不一定能產生出最少行的code** **，例如下圖第6行可以刪除，把第7行的jmp改成jle結果是一樣的。**

****

**時間比較**

**我們用了陣列大小為100的測資輸入1000次做比較，發現組合語言(final\_Group10)(0.6sec)C/C++語言(a)(0.4sec)所花的時間差不多，C/C++語言比較快一點**

****

**Conclusion (required):**

**用Visual Studio反組譯的code，因為經過一些優化及系統化，比較方便，而自己寫的較彈性，能用macro來減少行數，增加可讀性。**

**Team Work Arrangement (If you are in a group):**

**林子傑: 提供merge sort 模板、debug、 寫程式 、對bug進行截圖，提供報告意見**

**康智詠: debug、 寫程式 、指導寫組合語言、打報告**