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。

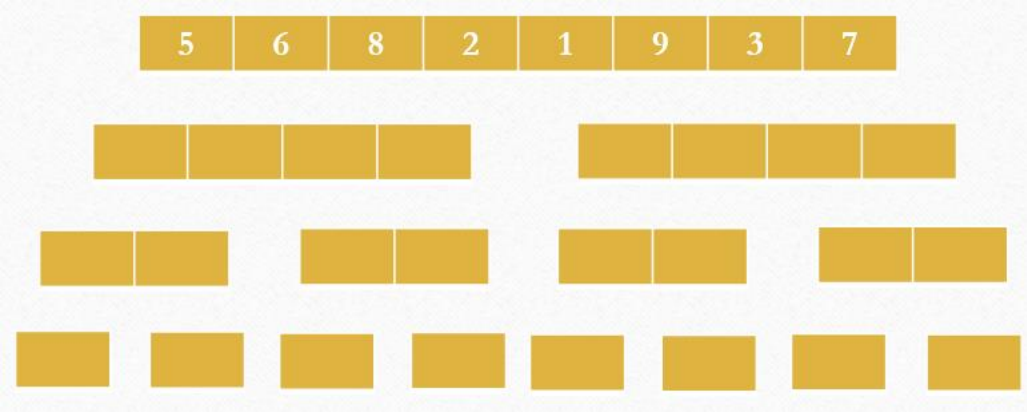
Table of Contents:

Introduction (required):

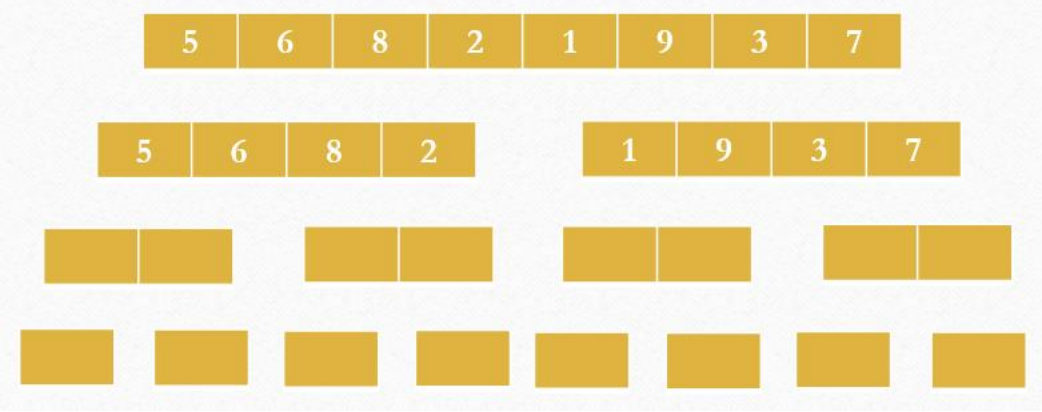
合併排序法，是排序演算法的一種，使用 **Divide and Conquer** 的演算法來實作。排序時需要額外的空間來處理，過程依照以下步驟進行：

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。

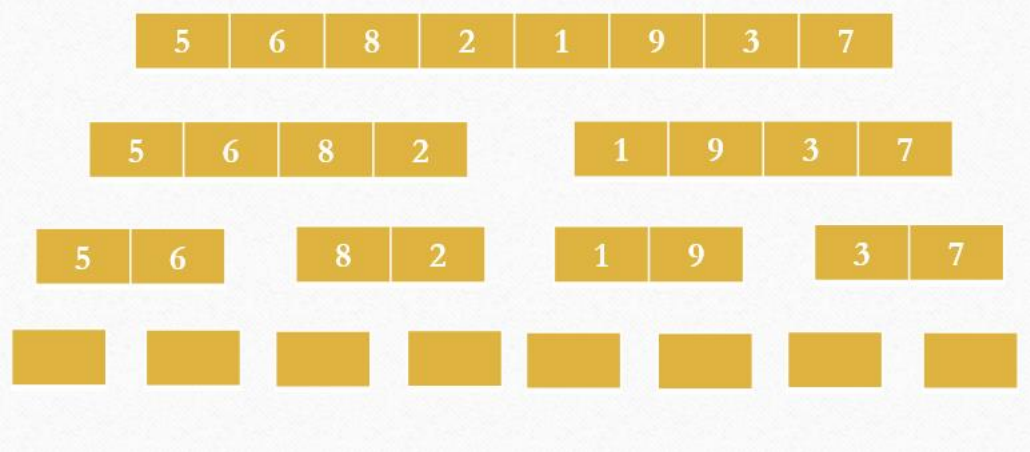
Step1:



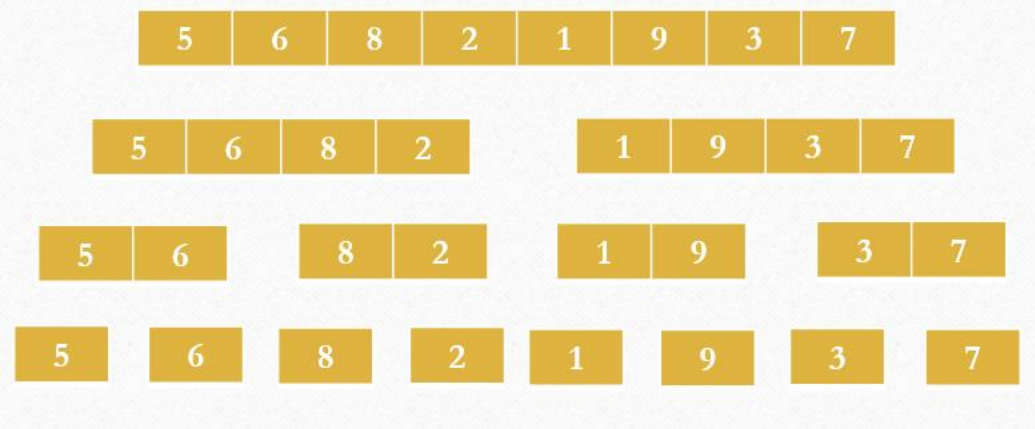
Step2:



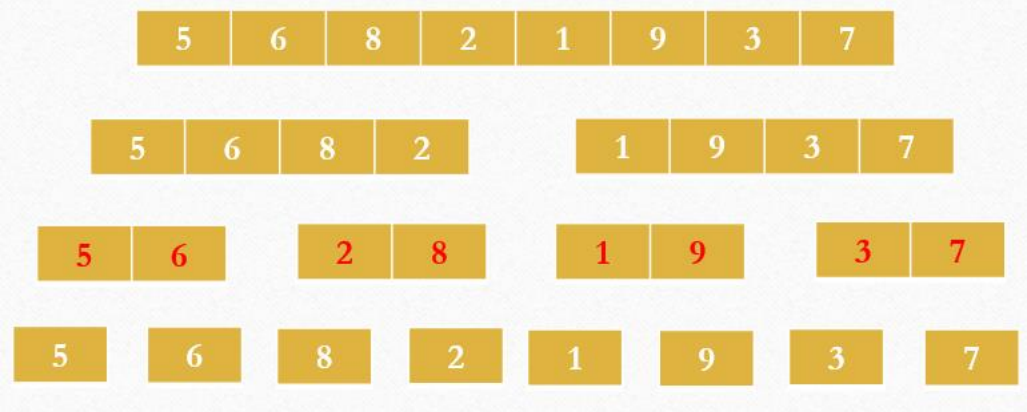
Step3:



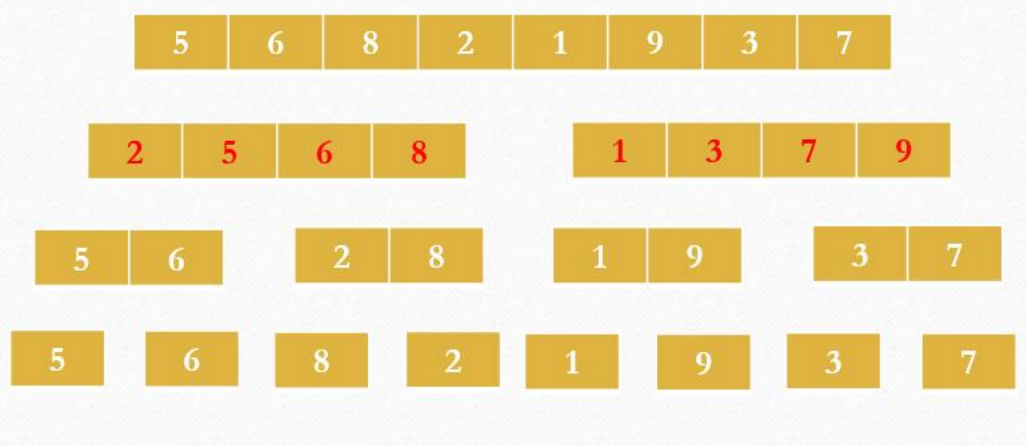
Step4:



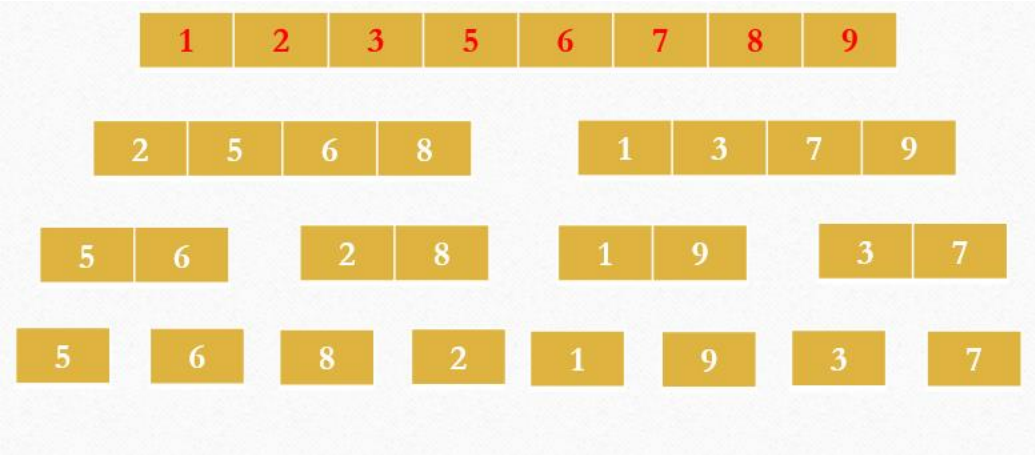
Step5:



Step6:



Step7:



:

Project plan or <u>Test Plan and TestCases</u> (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	2 3 5 9 14 23 67 5 23 47 90

	5 9 3 14 67 23 5 8 90 47 23 5 -1	
3	0	

Debug Log:
狀況**1**：沒有輸出

```
5
1
2
3
4
5
```

發現：**Macro** 沒寫好**(jmp M2)**，造成無限迴圈。

```
Carry MACRO Reg ;carry arr into buf
mov edx,[arr+4*Reg]
inc Reg
mov [buf+4*ecx],edx
jmp M2
ENDM
```

狀況2：印出來的東西不對

```
5
1
2
3
4
5
+0 +3 +0 +0 +0
```

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

```
MergeSort PROC
    push ebp
    mov ebp,esp
    mov eax,[ebp+12] ; eax=L+1
    inc eax
    mov ebx,[ebp+8] ; ebx=R
```

```
00C022A0  push        ebp
00C022A1  mov         ebp,esp
00C022A3  sub         esp,0FCh
00C022A9  push        ebx
00C022AA  push        esi
00C022AB  push        edi
00C022AC  lea         edi,[ebp-0FCh]
00C022B2  mov         ecx,3Fh
00C022B7  mov         eax,0CCCCCCCCh
00C022BC  rep stos    dword ptr es:[edi]
```

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)**

```
; if(L+1)<=R return;
cmp eax,ebx
jge MQuit
```

```
00C0250D  mov         dword ptr [ebp-24h],eax
00C02510  mov         eax,dword ptr [ebp-24h]
00C02513  cmp         eax,dword ptr [n]
00C02516  jge         main+0D5h (0C02545h)
```

3. 讀取傳入參數：版本**A**會使用$\boldsymbol{ebp + N}$。版本**B**則直接利用$\boldsymbol{dword\ ptr}$ [*變數名稱*]

```
    ; M=(L+R)>>1
    add ebx,[ebp+12]
    shr ebx,1
    push ebx
```

```
    if (R - L <= 1)return;
00C022BE  mov         eax,dword ptr [R]
00C022C1  sub         eax,dword ptr [L]
00C022C4  cmp         eax,1
00C022C7  jg          sol+2Eh (0C022CEh)
00C022C9  jmp         sol+15Fh (0C023FFh)
```

4. **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。

```
    if (R - L <= 1)return;
00C022BE  mov         eax,dword ptr [R]
00C022C1  sub         eax,dword ptr [L]
00C022C4  cmp         eax,1
00C022C7  jg          sol+2Eh (0C022CEh)
00C022C9  jmp         sol+15Fh (0C023FFh)
    int M = (R + L) / 2;
00C022CE  mov         eax,dword ptr [R]
00C022D1  add         eax,dword ptr [L]
00C022D4  cdq
00C022D5  sub         eax,edx
00C022D7  sar         eax,1
00C022D9  mov         dword ptr [M],eax
```

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

```
C:\Users\USER\masm\final_Group10>echo  %date% %a% %time%
 2019/01/09 週三 %a% 22:24:37.46

C:\Users\USER\masm\final_Group10>Group_FP < testdata.in > testdata.out

C:\Users\USER\masm\final_Group10>echo  %date% %a% %time%
 2019/01/09 週三 %a% 22:24:37.52

C:\Users\USER\masm\final_Group10>echo  %date% %a% %time%
 2019/01/09 週三 %a% 22:24:37.52

C:\Users\USER\masm\final_Group10>a < testdata.in > testdata.out

C:\Users\USER\masm\final_Group10>echo  %date% %a% %time%
 2019/01/09 週三 %a% 22:24:37.55
```

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