**MergeSort**

**Programming Assignment #1**

依照PPT上的演算法，將Merge Sort以Java實作出來。  
Input: numbers from keyboard ;  
Output: a non-decreasing sorted sequence of N numbers.  
請繳交source code 和 一份說明文件，內需含執行畫面。  
無說明文件，以零分計算此作業。

**說明**

此 MergeSort 是採用 Recursive(遞迴)方式來實作，屬於 Divide and Conquer 演算法的實例，整個核心是把一個串列切成很多個子串列(divide)，切到不能再切剩餘一個數為止，之後並再逐一處理子問題將每個串列分別排序，最後再將子問題的結果合併(conquer)，以下分別來介紹 mergeSort 與 Merge 函式其功能與執行方式。

**mergeSort() 函式**

此函式主要是進行 Divide 的部分，是將一個串列切成很多個子串列。傳入參數分別為資料(list)、最左邊索引值(left)、最右邊索引值(right)，第二行是判斷目前串列是否只剩一個數，若成立則不再繼續切割，第四及第五行則是分別切開左右等份繼續執行 mergeSort() 函式，第六行是執行 Merge() 進行最後的排序合併。



**Merge() 函式**

此函式主要是進行 Conquer 的部分，首先先建立一個暫存的陣列名為 temp[]，長度為 right+1 加1的原因是因為陣列元素從零開始，接著建立變數 left\_end 來記錄左邊最後一個位置，以及建立變數 index 來記錄目前元素走訪的位置，最後再建立一個變數 origin\_left 將最左邊的變數儲存起來(最後搬移元素會用到)。

變數建立完後就可以左右兩串列比大小依序放入 temp 陣列中儲存(7~12行)，直到某一邊沒有元素比較時即結束回圈，接著下一步(14~22行)將其中一邊剩餘的元素依序地放入 temp 陣列中存放。

以上子串列都排序好後最後將排序好的 temp 陣列複製到 list 串列中，此時原先的 origin\_left 變數就派上用場，做為串列目前索引起始位置然後依序的放入 list 串列中即完成此一回合排序。  
  

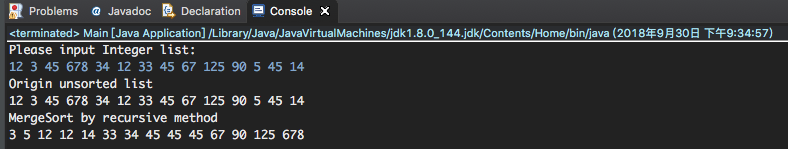

**main() 主函式**

主函式主要是要求使用者輸入一串整數形態的串列並執行排序演算法，我們的資料是以 ArrayList 來儲存，使用 ArrayList 優點為可以動態新增元素而不須一開始就決定資料長度，比起陣列相對來得方便，4~22行為判斷使用者的輸入是否合法，若輸入正整數以外的資料則會被例外函式抓取並告知使用者某數輸入不合法，執行此迴圈直到該輸入的數列完全為正整數為止。

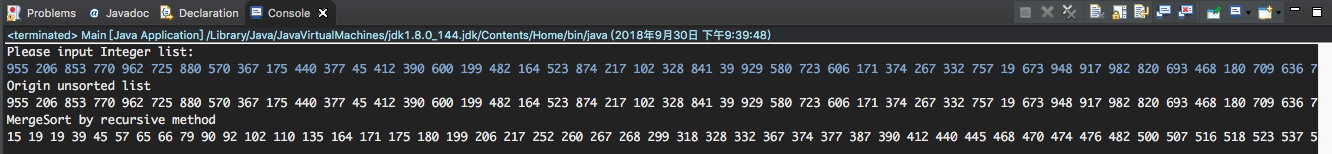
24~27行印出原始輸入的串列，接著第29行呼叫 mergeSort() 開始進入合併排序演算法，最後排序完成後31~33行為印出排序後的結果。  
  


**執行與測試**

* 測試一(輸入一筆正常測資)



* 測試二(一百筆資料測試)



* 測試三(不合法測試)  
  故意輸入不合法的資料，程式會告失使用者某筆資料有誤，並要求重新輸入數列到完全合法為止。

