陳奕嘉 Alvin Tan

程式設計作業五
題目：**選擇排序法**

實現方法：

```
*********************************
Selection Sort in descending order
*********************************
Please enter the size of the array which undergoes selection sort (2-100): 8
Please enter the numbers one by one to do selection sort, in which a number is in the
    range of 0-99:
Number 1: 16
Number 2: 25
Number 3: 39
Number 4: 27
Number 5: 12
Number 6: 8
Number 7: 45
Number 8: 63
Origin: 16 25 39 27 12 8 45 63
Pass 1: 63 25 39 27 12 8 45 16
Pass 2: 63 45 39 27 12 8 25 16
Pass 3: 63 45 39 27 12 8 25 16
Pass 4: 63 45 39 27 12 8 25 16
Pass 5: 63 45 39 27 25 8 12 16
Pass 6: 63 45 39 27 25 16 12 8
Pass 7: 63 45 39 27 25 16 12 8
Pass 8: 63 45 39 27 25 16 12 8
Result: 63 45 39 27 25 16 12 8
```

應題目要求和參考助教提供的模板，此程式首先要求使用者輸入一個數字 n 以定義陣列能存放幾個待排序的數字。這部分有設置了防呆機制以便有效的數字能被讀取。n 的範圍在 2-100 是因爲對一個號碼進行排序沒有意義。

接著，在 i = 0; i < n 的 for 迴圈內，讓使用者依次輸入待排序數字，這裏也設置了同樣的防呆機制。每個數字的範圍設定為 0-99，0 也被考量進去是因爲它也有值。

到這裏，程式會列出未被排序的陣列，確定使用者輸入無誤。然後呼叫選擇排序的函數進行由大到小的排序。以上附了排序的每一項步驟，可以看到函數從 arr[0]開始，一直往下對比找出陣列中最大的數字，跟 arr[0]的數字做調換。過後就是 arr[1]往下對比除 arr[0]外最大的數字，也就是第二大的數字來進行調換，以此類推。最後顯示排序好的陣列。

心得：

此排序方法是個不穩定排序，因爲它涉及了相等元素的位置交換，導致它們的相對位置發生改變。但這種排序方法有著比穩定排序更好的效能，具有更佳的時間和空間效能，因爲元素交換的自由度沒有被限制。

我認爲排序可以讓我們更好地了解元素在電腦記憶體中是怎麼運作的。比如想要對一個陣列進行排序，必須通過改變每一個元素的索引，來達到調換位置的目的；有些方法如插入排序，必須把一部分元素都往后移才能空出位置。總之這部分的知識點是十分有用并且重要的。

參考資料：

https://youtu.be/W3dtyZr8rcY
https://moodle.ncku.edu.tw/course/view.php?id=28947