作業思路

1. 大方向: 將整份作業拆解成main()以及insertionSort()，主要是為了要滿足insertionSort()可以使用template來寫的作業要求。
2. 小方向: 在main()內完成產生二維陣列並在陣列內儲存隨機a, b, c, d, e，並且可以計算出每列c的數目；而在insertionSort()內則會將c的數目透過insertion sort的方法由大排到小而列印出各自的index值，並且在遇到相同的c數目時會自動優先將index值由小排到大。

程式碼片段解析

1. 確保每次程式所產生的隨機值結果都會不同，方法是將現在的時間傳入隨機種子內。

一張含有 文字 的圖片

自動產生的描述

1. 讓使用者輸入n\*n array的n值後，即可在2D array內隨機產生a, b, c, d, e，隨機產生的方法是rand()。

一張含有 文字, 螢幕 的圖片

自動產生的描述

1. 開始計算每列c的數目，掃描整個二維陣列，只要存在c就會將counter加一，而後再針對insertionSort func.透過template實作出來。

一張含有 文字 的圖片

自動產生的描述

1. 在insertionSort() template內在產生第二個二維陣列rowCounterArr[][]，row放入index值，column放入numbers of ‘c’，如此一來就可以將index與numbers of ‘c’兩者關係互相綁定。

* 其他思考細節請見下方【思考過程 (錯誤嘗試)】

一張含有 文字 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述

1. 插入排序法(Insertion sort)的過程。

一張含有 文字, 螢幕 的圖片

自動產生的描述

1. 將insertion sort過後的index列印出來，邏輯是將排序過後的numbers of ‘c’去和第二個二維陣列rowCounterArr[][]做比較，只要numbers of ‘c’存在於此二維陣列中，就會去列印出對應的index值。

**rowCounterArr[][]存在的意義就是去綁定index和numbers of ‘c’**

一張含有 文字 的圖片

自動產生的描述

思考過程 (錯誤嘗試)

1. 使用std::map的key, value的關係，來綁定index, numbers of ‘c’兩者的關係，嘗試過後發現numbers of ‘c’會存在不unique的問題，就無法得出對應的index。

後來透過pointer指向numbers of ‘c’的方法嘗試將其變成unique，就可以將指標當成key去對照value，後來嘗試過後無法解決，個人推論是此處的pointer屬於連續的array，因此sorting過後透過for-loop印出時index值的順序也不會隨之改變。

一張含有 文字 的圖片

自動產生的描述

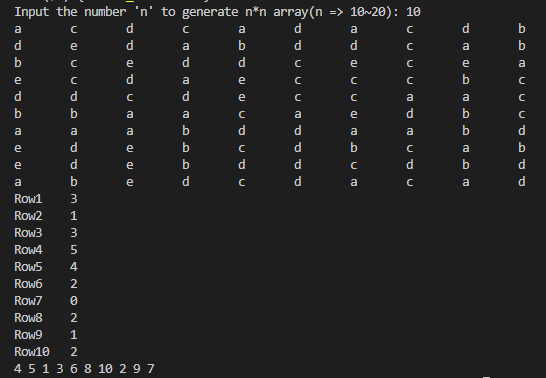
1. 使用vector的make\_pair()，來綁定index, numbers of ‘c’兩者之間的關係，後來嘗試過後發現使用的vector也是屬於連續的array，因此sorting過後透過for-loop印出時index值的順序也不會隨之改變。

一張含有 文字 的圖片

自動產生的描述

編譯結果

1. Input n=10



1. Input n=15

一張含有 文字 的圖片

自動產生的描述

1. Input n=20

一張含有 文字 的圖片

自動產生的描述