

魏宏凱 (112511142)：

在一開始的構思中，我們把數獨這個大題目分成「生成題目」與「印出陣列」兩個主要函式，我負責生成題目。雖然生成數獨題目看似不難，但我一直沒有好想法。原本最苦惱的是要生成整張數獨需要太多次 rand() 跟檢查，直到請教了一些資工系的學長以後，他們建議我「先生成部分數獨、再用解數獨的邏輯寫自己出的題目」，才讓我豁然開朗，參考他們的程式碼，把功能寫出來。

生成數獨時、因為久聞 c++ 的 rand() 函式其實很不夠 rand，我也自己額外請教上面提到資工系的朋友、他另外推薦我使用 mt19937 的函式庫。相較於 c++ 內建的 rand()，mt19937 使用的梅森旋轉算法可以產生更接近隨機的亂數。雖然在日常程式的練習上應該用哪個都沒差就是了、但我還是覺得很好玩所以玩了一下。

這次也嘗試使用 class 來寫：雖然以前對 class 不是很熟，但也感謝提供我思路的學長向我解釋，因此我不只完成了數獨，也提前預習了下學期的物件導向程式設計(?)

除了助教規定的功能與加分項以外，我也自己多發想並實現了幾項額外的功能：利用 stack 與 struct 實作的 undo, redo 功能、以及在一局數獨開始之前讓使用者選擇顏色的功能。因為以前都沒有把 struct 塞進 vector 或 stack 過，因此在實作的細節上、尤其在語法上請教了資工系的朋友，最終也成功多習得了一項看起來酷酷的技能。

此外，我在開發的過程中也認識 github 的使用方式，並且將自己的數獨程式碼放上 github、還寫了 readme。儘管不是為了分數，但我覺得為自己的學習留下紀錄也是相當重要的一件事。附上 repo link：<https://github.com/kevinwei1029/sudoku>

楊朋翰 (112511098)：

一開始宏凱跟我說要使用 Visual Studio 進行專案開發的時候，我還蠻好奇為什麼要特別去安裝檔案十幾 GB 的編譯器來開發；但他跟我介紹了許多功能，可以很方便的串接 github 與他人共同作業、有預覽錯誤功能可以不用直到按下編譯那刻才知道哪裡有錯、可以輕易看出各種 include 檔案來源、會自動顯示 object 與 class 下面有哪些函式與區域變數可以用。雖然宏凱說 Visual Studio 不太適合平常寫 LAB 跟 HW 使用、但我也很期待以後在開發大型專案的時候探索 Visual Studio 的更多功能。

我負責的是「印出陣列」的功能。一開始先寫了很多東西打算一次到位印出整個陣列，但印出來的都是排版歪七扭八的 |、--、跟數字。經過一陣子的苦惱以後，我決定從小的方塊開始寫，一步一步往上疊加，最終印出整個陣列。

後來，一方面是為了達成助教要求，也是為了排版美觀，我開始把邊框設定顏色，寫許多小函式實作其他的排版設計，以及在終端機上為數獨題目的各個部分加上不同的顯示顏色。看著原本黑白的終端機冒出五顏六色的色彩、真是好看！