陳奕嘉 E94115011

程式設計作業三
題目：1A2B

實現方法:
我的程式主要由 5 個部分組成的，分別是 switch-case 主幹、game mode 1、game mode 2、game mode 3、和 game mode 4.

Switch-Case:
首先是程式的主幹，題目要求要有 4 個游戲模式，所以就用 switch-case 來達到選擇 game mode 的作用。先設定一個整數變數 mode，來儲存使用者輸入的數字，帶進 switch 裏面尋找對應的 case，輸入 1 就去 case 1 的游戲模式，以此類推。若使用者輸入 1-4 以外的符號，就會執行 default 的情況，回到一開始 main menu 重新輸入數字。

Game Mode 1:
此模式是電腦產生一個 4 位數的 random number，使用者猜數字。這裏用陣列 R[ ]代表電腦選擇的 random number，因爲 10 個數字都不可以重複，所以 array 裏面 0-9 各只有一個。爲了把 array 數字打亂，設定一個 shuffle 函數。做法是讓電腦用 rand()%10 來隨機選擇 0-9 的兩個數字 x 與 y，作爲 R[ ]的 index，然後用 bubble sort 讓這兩個 index 對應在 array 裏的號碼交換位置，重複 1000 次。過後取 array 前四個號碼作爲 random number。

然後設定一個 do-while loop，結束條件是儅小 a 等於 4，等同于 4A0B。裏面先有一個 while loop 來讓使用者猜數字。因爲使用者可能很調皮輸入數字以外的符號，這裏使用 getchar( )來讀取 input。用 ctype.h 的 isdigit( )來接收數字，在 row 51 用跟 0 子元的差值轉換成 int 形態。Getchar 是一次讀取一個子元，只有接收到數字時，才會儲存進 G[ ]裏，順帶增加 count。所以儅輸入數字意外的符號或者小於 4 個數字，就符合加下來 count<4，而 count>4 很好理解，其中一個條件觸發就讓使用者重新輸入。Row 64 的條件是爲了防止重複輸入相同的號碼，若出發其中一個也是回去重新輸入。

接下來就到 for 回圈的部分，這邊是讓 random number 的每一個號碼去對比 guess number 的每一個號碼，儅對比相同但 R[ ]的 index 不等於 G[ ]的，增加一次 b；儅對比相同且 R[ ]的 index 等於 G[ ]的，增加一次 a。這裏有個 times 的計數器來計算猜了多少次。儅輸入的 guess number 得到 a=4，就跳出 while loop，結束游戲，回到 main menu。

Game Mode 2:
此模式是使用者輸入一個 4 位數的 random number，由電腦來猜。這裏 random number 的輸入模式和 Game Mode 1 的 guess number 輸入模式一樣，只是儲存在 array R[ ]裏。

接下來是電腦猜數字，電腦用 rand()%10 隨機選一個 0-9 的號碼，儲存在 array G[ ]裏，這裏也有防止重複相同號碼的機制，選到不重複號碼的 4 位數數字就跳出 while loop。

然後來到讓 random number 去對比電腦 guess number 的階段，電腦不看提示所以只需要設定 a 的計數器，同樣儅 a=4 就結束游戲，並告訴使用者電腦猜了幾次。

Game Mode 3:
此模式是電腦產生一個 5 位數的 random number，使用者猜數字。整個架構和 Game Mode 1 大致相似。需要改變的是給 array G[ ]增加多個儲存位置變成 G[5]；換成 count<5 ||

count>5；Row 181 的防止重複機制要增加多幾個 5 個號碼之間的組合；for loop 換成 i<5 和 j<5；a=5 時才結束猜數字階段；最後猜數字次數的回饋增加多一個 R[ ]。

Game Mode 4:

此模式是退出鍵。在 case 4 我設定了讓使用者確認是否要退出游戲的步驟，因爲大部分游戲都會有這個確認鍵。我先設定一個子元 choice，來儲存使用者的 y 或者 n。這邊我用了 ctype.h 裏面的 toupper 函數，確保不管是 Y 或 y 都會變成 Y。如果輸入 y 就結束整個程式運行，執行 return 0，輸入 n 就回到 main menu。此外的回答都重複提問 Y/N。

延申問題:

Q. 爲什麽不能用 random = rand() % 9000+1000 來隨機生成 random number?

A.我嘗試過這個方法，然後用

```
n1 = guess/1000;
n2 = (guess/100)%10;
n3 = (guess/10)%10;
n4 = guess%10;
```

來設定防止重複號碼的機制，但有個嚴重問題就是它不能生成一個以 0 為首的四位數號碼，range 只有 1000-9999，所以和助教給的例子'0234'不符。使用者輸入 0234 也會變成百位數 234，原因是 int 只能儲存整數。所以我換了一種方式，讓 random 和 guess number 以 array 的形式儲存起來，這種方式用來對比兩個數字也比較方便。

Q. 爲什麽 row 30 要加一個 getchar()?

A. Getchar( )的作用是清空輸入緩衝區，儅使用者輸入 1-4 其中一個號碼時，會使用鍵盤的 enter 鍵，增加了一個\n 的符號，所以會接著在 game mode 先接收到\n 才接收到 1-4，導致輸出不美觀。Getchar( )可以清空\n 讓使用者的 guess input 直接被讀取。

還有另一個作用是比如説儅輸入子元 a，程式會一直執行 default 的 case，導致 while loop 一直執行，Getchar( ) 可以清空緩衝區讓 while loop 停留在再次讓使用者輸入 1-4 的部分。

Q. 在 Game Mode 2 中電腦猜數字的概率是多少?

A. 電腦 1-3 就可以選中一個不重複號碼的 4 位數數字。選中之後需要平均 3862 次，區間為 0-20000 次，才可以在對比之下猜中和 random number 一樣的號碼。電腦要選中不重複號碼的數字且猜中 random number 平均需要 13896 次，區間為 0-31000 次，這個概率是合理的因爲是前面兩個概率的乘積。

根據定義 nPr = n! / (n - r)!，Game Mode 1 電腦生成數字有 10P4=5040 種排列組合，因爲會扣除號碼重複兩次以上的數字。同樣的 Game Mode 3 五位數模式有 10P5=25200 種排列組合。

討論:

通過這次 1A2B 小游戲的作業，讓我屬實受益匪淺。雖然只是一個小游戲，但仍然有很多細緻的地方要考慮，同時也讓基本功有很大的提升。

這也是第一次用 C 語言寫一個 switch-case 的程式，許多游戲都有用到讓玩家做選擇題的呈現方式，比如原神這游戲跟 npc 對話有用到 2 個 case 的 switch-case，雖然簡單明瞭但可以和玩家有很好的互動，作用是非常强大的。

還有這個作業有大量的防呆機制要考量，要經得起使用者的各種輸入手法，以防止程式當機。這裏也涉及到子元與數字的轉換，讓程式碼可以更加的靈活。

若是把題目改成號碼可以重複，Game Mode 1 四位數就有 10C4=10000 種方法組合起來，使用者輸入的限制條件只需要考量是子元或是數字和 count<4 || count>4 就可以了。Game Mode 2 電腦只需要 1 次就可以產生一個四位數號碼，對比出和使用者設定的 random number 所需要的 times 會大大降低。

從玩家的角度來説，因爲沒有號碼不重複這個有利條件，所以猜數字會更加的困難，不知道一個號碼會重複 2、3 或 4 次，也有可能 2 個號碼各出現兩次。Game Mode 3 是加倍困難。

從電腦的角度，Game Mode 2 反而對它來説更簡單，這是因爲電腦是照著 randomness 來猜數字，并不會思考和考慮原本的不重複號碼這個條件，導致一開始的所需次數較大。這裏可以反映出人類和電腦猜東西模式的差別，或許把這個游戲給 AI 玩，會出現和人類相似的答題模式。

參考資料:
https://youtu.be/pXyLkiMY2Lc