

1A2B

Final Project Report

By

409261574 林家賢

X86 Assembly Language Fall 2020

Date Submitted: January 18, 2022

Abstract

我的書面報告內容涵蓋遊戲介紹、專案計畫及測試資料、程式碼討論、總結及工作分配。本報告將著重於釐清影片中介紹不清楚的部分,並以文字跟圖片詳細解析我設計的遊戲。

Introduction

與一般 1A2B 規則相同, 唯一的差別在於使用者可以選擇要遊玩的密碼長度(最低 1 位最高 9 位), 也就是說使用者選擇 8 時, 勝利條件為 8A0B、輸入 1 時勝利條件為 1A0B。

遊戲結束後會告訴使用者猜測了幾次, 此次數不將違規輸入計入猜測次數。

並在使用者輸入密碼長度與猜測數值時, 檢查是否有違規輸入, 包含: 選擇密碼長度時不為 1-9、輸入字母或符號、猜測數值與密碼長度不符合。

專案目標 : 1)處理輸入錯誤 2)讓使用者選擇密碼長度的 1A2B(1 至 9 位)

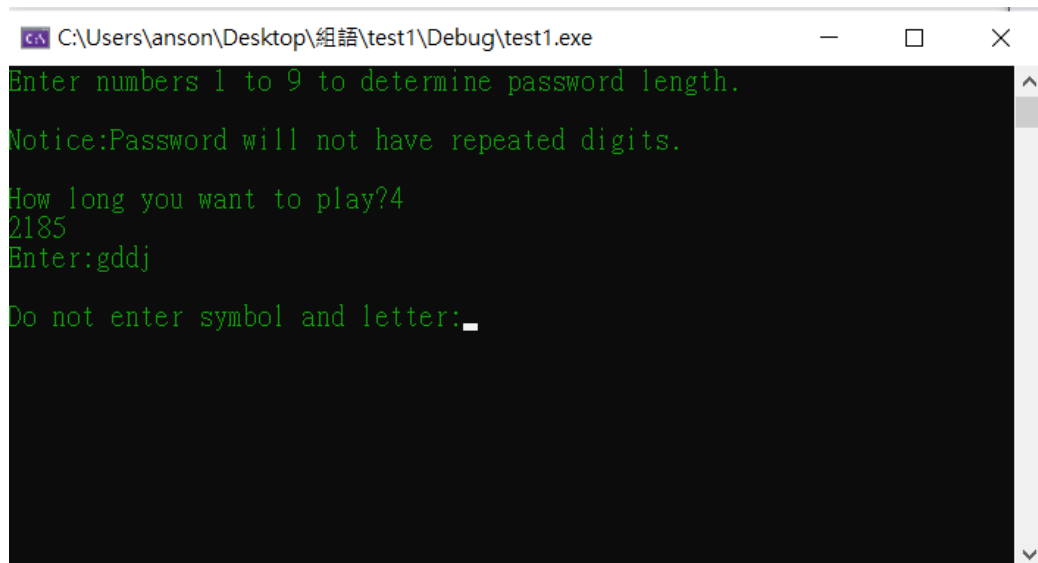
Test plan and Cases

一般遊戲畫面：

違規輸入的處理(使用者輸入字母符號或是輸入長度與設定不同)

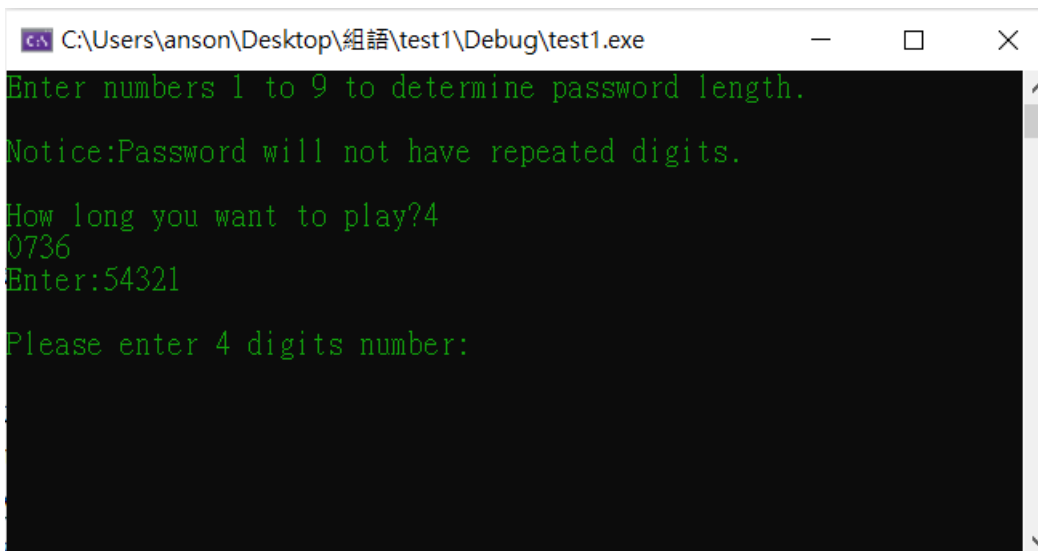
※為了講解方便, 所以直接顯示答案進行比對

1. 輸入字母符號



```
C:\Users\anson\Desktop\組語\test1\Debug\test1.exe
Enter numbers 1 to 9 to determine password length.
Notice:Password will not have repeated digits.
How long you want to play?4
2185
Enter:gddj
Do not enter symbol and letter:_
```

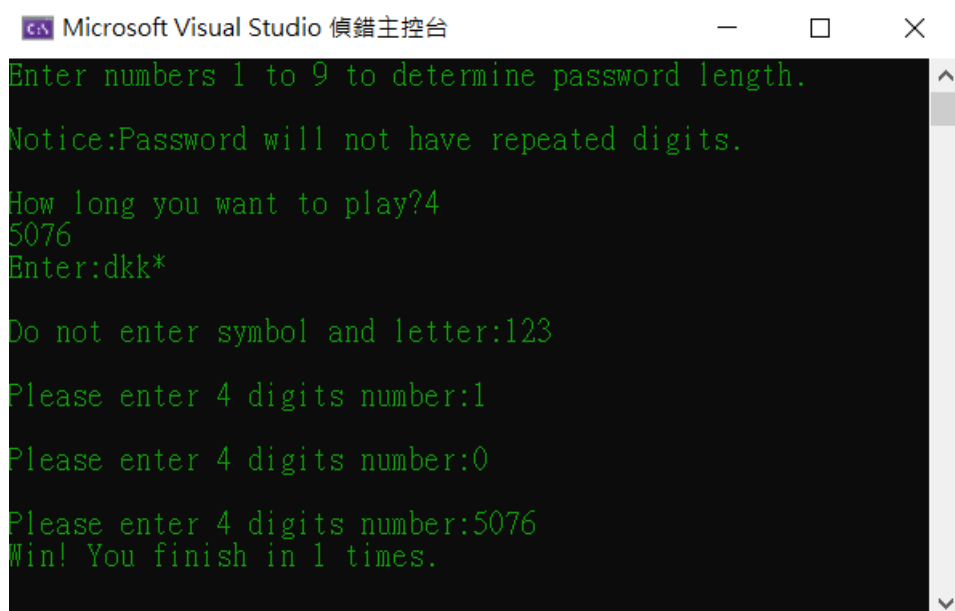
2. 輸入長度不符



```
C:\Users\anson\Desktop\組語\test1\Debug\test1.exe
Enter numbers 1 to 9 to determine password length.
Notice:Password will not have repeated digits.
How long you want to play?4
0736
Enter:54321
Please enter 4 digits number:
```

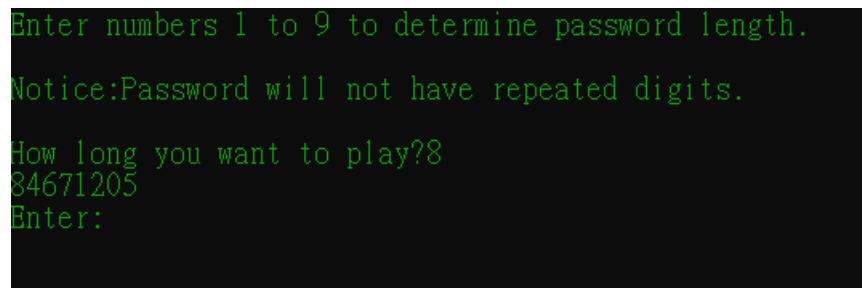
決定長度(password length)的違規輸入同理

違規輸入不計次數

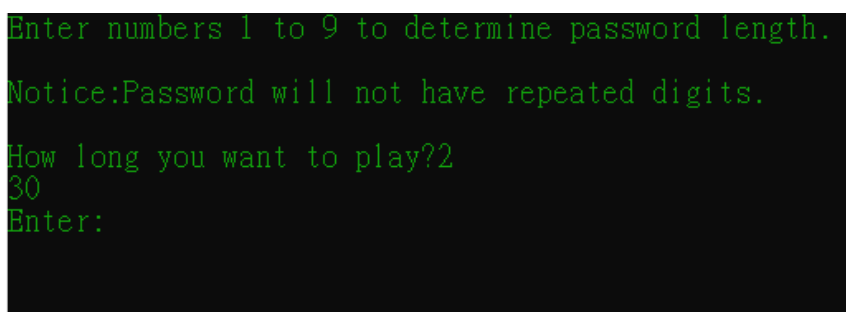


```
Microsoft Visual Studio 偵錯主控台
Enter numbers 1 to 9 to determine password length.
Notice:Password will not have repeated digits.
How long you want to play?4
5076
Enter:dkk*
Do not enter symbol and letter:123
Please enter 4 digits number:1
Please enter 4 digits number:0
Please enter 4 digits number:5076
Win! You finish in 1 times.
```

可由使用者決定密碼長度



```
Enter numbers 1 to 9 to determine password length.
Notice:Password will not have repeated digits.
How long you want to play?8
84671205
Enter:
```



```
Enter numbers 1 to 9 to determine password length.
Notice:Password will not have repeated digits.
How long you want to play?2
30
Enter:
```

Discussion

首先，我先判斷輸入的決定密碼長度有無違規輸入，如果沒有就往下執行，有的話，要求使用者重新輸入。之後產生一個該長度的亂數，並判斷此亂數是否有重複位數，重複的話，重新產生一個新亂數，直到不重複為止。再來，輸入猜測值並判斷有無違規輸入，若無違規則與密碼比較幾 A 幾 B，若不相同則輸出幾 A 幾 B 並跳回以重新輸入猜測值，相同時結束遊戲並顯示猜測次數。

※ 遊戲製作關鍵部分

處理亂數重複程式

```
;-----使亂數不重複-----  
;If there are duplicate digits, regenerate a set of passwords  
    mov ecx,key1  
    dec ecx  
    mov esi,offset Key  
P1:  
    mov eax,aa  
    inc eax  
    mov aa,eax  
  
    push ecx  
  
    mov ebx,key1  
    sub ebx,eax  
    mov ecx,ebx  
  
    mov eax,[esi]  
    mov edi,offset Key  
    push eax  
    mov eax,aa  
    mov ebx,4  
    mul ebx  
    add edi,eax  
    pop eax  
P2:  
    mov ebx,[edi]  
    cmp eax,ebx  
    jne LLL  
    pop ecx  
    jmp L9          ;Repeated digits are found, so regenerate the password  
LLL:  
    add edi,4  
    Loop P2  
    pop ecx  
    add esi,4  
Loop P1
```

設計內容：

將亂數以 2 個字串儲存 並以迴圈互相比較如有重複跳回產生新亂數
若無重複繼續執行程式

判斷 A、B:

1. 判斷 B

```
-----
;Determine whether each bit of the guessed value exists in the password,if it exists then B is incremented by one
mov ecx,key1
mov esi,offset InputtN

M1:
mov edi,offset Key
push ecx
mov ecx,key1
N0:
    mov eax,[esi]
    mov ebx,[edi]
    ;call dumpregs
    cmp eax,ebx
    jnz N1

    mov edx,B
    inc edx
    mov B,edx
N1:
    add edi,4
Loop N0
    add esi,4
    pop ecx
Loop M1
```

2. 判斷 A

```
-----
;Check whether each digit of the guessed value is the same as the password in the corresponding position,
;if it is the same, add one to A, and subtract one to B
mov ecx,key1
mov esi,offset InputtN
mov edi,offset Key
M2:
mov eax,[esi]
mov ebx,[edi]
cmp eax,ebx
jne N2
    mov edx,B
    dec edx
    mov B,edx
    mov edx,A
    inc edx
    mov A,edx
N2:
    add esi,4
    add edi,4
Loop M2
```

A、B 判斷方法:

我先判斷 B 之後再處理 A, 因為 B 判斷可知猜測值有幾個數字與密碼相同
再由 A 可知數字位置是否相同, 而 B 的值減掉 A 的值就為 B 真正的值。

Conclusion:

一開始，我原本想要挑戰做五子棋的小遊戲，不過在做互動介面時，因不熟悉又接近期末了，權衡之下我改做 1A2B。

程式設計最初目標就是一般的 1A2B, 但在做完 debug 後, 我認為可以增加遊戲彈性與互動性, 於是則產生了最終的完成版-可由使用者決定密碼長度的 1A2B。

過程中遇到的問題也不少:

首先，違規輸入的種類繁多, 尤其是密碼原本應為數字構成, 但為了考慮使用者輸入文字, 因此 ReadDec 全都得改為 ReadString, 並在判斷完所有可能的違規輸入後, 再轉為數字以進行接下來的程式作業。

其次，產生亂數的方法，為了配合遊戲規則不能重複數字，我使用最直觀的辦法，如有重複位數則再產生一次, 而這個辦法以原規則四位數 1A2B 來說確實沒什麼問題, 因為隨機產生的亂數為無位數重複的機率約為 50%, 但當密碼長度上升到 9 位數時, 產生符合規則的密碼機率只剩下 0.3%, 會導致遊戲的前置作業有了 1 至 2 秒的等待時間, 所以這個產生無位數重複亂數的部分是值得我這組繼續深入研究的議題