

## 1.編譯結果

```
jacob@jacob-VirtualBox:~/桌面/hw2$ gcc -o shogi shogi.c -lev
jacob@jacob-VirtualBox:~/桌面/hw2$ ls
README.md  shogi  shogi.c  shogi.dat  testshogi.dat
jacob@jacob-VirtualBox:~/桌面/hw2$ ./shogi
enter 'n' for new game, 'r' for shogi manual, 's' for saving the previous game, amd 'q' to quit:n
```

使用 gcc 此命令 compile，並以 ls 確認是否 compile

## 2.執行結果

```
jacob@jacob-VirtualBox:~/桌面/hw2$ gcc -o shogi shogi.c -lev
jacob@jacob-VirtualBox:~/桌面/hw2$ ls
README.md  shogi  shogi.c  shogi.dat  testshogi.dat
jacob@jacob-VirtualBox:~/桌面/hw2$ ./shogi
enter 'n' for new game, 'r' for shogi manual, 's' for saving the previous game, amd 'q' to quit:n
Game starts!
  9  8  7  6  5  4  3  2  1
  香 桂 銀 金 王 金 銀 桂 香 一
  步 飛 步 步 步 步 步 步 步 二
                           三
                           四
                           五
                           六
                           七
                           八
                           九
  步 步 步 步 步 步 步 步 步
  香 桂 銀 金 王 金 銀 桂 香
blue's turn
the total time of the game is 0 sec, and the total time the blue spend is 0 sec.
enter the origin of row(段, y-axis) or enter 0 to go back to last step:
  9  8  7  6  5  4  3  2  1
  香 桂 銀 金 王 金 銀 桂 香 一
  步 飛 步 步 步 步 步 步 步 二
                           三
                           四
                           五
                           六
                           七
                           八
                           九
  步 步 步 步 步 步 步 步 步
  香 桂 銀 金 王 金 銀 桂 香
blue's turn
the total time of the game is 5 sec, and the total time the blue spend is 5 sec.
enter the origin of row(段, y-axis) or enter 0 to go back to last step:
7
enter the origin of column(筋, x-axis):5
enter the destination of row(段, y-axis):6
enter the destination of column(筋, x-axis):6
wrong step!!
  9  8  7  6  5  4  3  2  1
  香 桂 銀 金 王 金 銀 桂 香 一
  步 飛 步 步 步 步 步 步 步 二
                           三
                           四
                           五
                           六
                           七
                           八
                           九
  步 步 步 步 步 步 步 步 步
  香 桂 銀 金 王 金 銀 桂 香
```

開新局，並偵測錯誤路線

```
blue's turn
the total time of the game is 15 sec, and the total time the blue spend is 15 sec.
enter the origin of row(段, y-axis) or enter 0 to go back to last step:
7
enter the origin of column(筋, x-axis):5
enter the destination of row(段, y-axis):6
enter the destination of column(筋, x-axis):5
action finishes! now it's red's turn
  9  8  7  6  5  4  3  2  1
  香 桂 銀 金 王 金 銀 桂 香 一
  步 步 步 步 步 步 步 步 步 二
                           三
                           四
                           五
                           六
                           七
                           八
                           九
  步 步 步 步 步 步 步 步 步
  香 桂 銀 金 王 金 銀 桂 香
the total time of the game is 16 sec, and the total time the blue spend is 16 sec.
  9  8  7  6  5  4  3  2  1
  香 桂 銀 金 王 金 銀 桂 香 一
  步 飛 步 步 步 步 步 步 步 二
                           三
                           四
                           五
                           六
                           七
                           八
                           九
  步 步 步 步 步 步 步 步 步
  香 桂 銀 金 王 金 銀 桂 香
red's turn
```

正確路線，改變棋盤並交換

```

the total time of the game is 133 sec, and the total time the blue spend is 109 sec.
9 8 7 6 5 4 3 2 1
香 桂 銀 金 王 金 銀 桂 香
步 飛 步 步 步 步 步 步
步 步 步 步 步 步 步 步
一二三四五六七八九

步 步 步 步 步 步 步 步
步 角 桂 銀 金 金 銀 桂 香
香 桂 銀 金 金 銀 桂 香
red's turn
the total time of the game is 135 sec, and the total time the red spend is 26 sec.
enter the origin of row(段, y-axis) or enter 0 to go back to last step:
6
enter the origin of column(筋, x-axis):5
enter the destination of row(段, y-axis):7
enter the destination of column(筋, x-axis):5
9 8 7 6 5 4 3 2 1
香 桂 銀 金 王 金 銀 桂 香
步 飛 步 步 步 步 步 步
步 步 步 步 步 步 步 步
一二三四五六七八九

步 步 步 步 步 步 步 步
步 角 桂 銀 金 金 銀 桂 香
香 桂 銀 金 金 銀 桂 香
red wins !!
end game

```

## 當王被吃掉，遊戲結束

```

the total time of the game is 135 sec, and the total time the red spend is 26 sec.
enter 'n' for new game, 'r' for shogi manual, 's' for saving the previous game, and 'q' to quit:s
name for saving file(binary file as '.dat'):shogi.dat
successfully restored!
enter 'n' for new game, 'r' for shogi manual, 's' for saving the previous game, and 'q' to quit:r
enter 'o' to open saved files(it will recover the previous game, suggesting saving the game first.) or 'p' to review previous game:o
name for saving file(binary file as '.dat'):shogi.dat
*****1*****
 9 8 7 6 5 4 3 2 1
香 桂 銀 金 王 金 銀 桂 香
歩 歩 歩 歩 歩 歩 歩 歩
      歩
歩 歩 歩 歩 歩 歩 歩 歩
香 桂 銀 金 王 金 銀 桂 香
please enter 'f' to forward, 'b' to backward, 'q' to quit:f
*****2*****
 9 8 7 6 5 4 3 2 1
香 桂 銀 金 王 金 銀 桂 香
歩 歩 歩 歩 歩 歩 歩 歩
      歩
歩 歩 歩 歩 歩 歩 歩 歩
香 桂 銀 金 王 金 銀 桂 香
please enter 'f' to forward, 'b' to backward, 'q' to quit:f
*****3*****
 9 8 7 6 5 4 3 2 1
香 桂 銀 金 王 金 銀 桂 香
歩 歩 歩 歩 歩 歩 歩 歩
      歩
歩 歩 歩 歩 歩 歩 歩 歩
香 桂 銀 金 王 金 銀 桂 香
please enter 'f' to forward, 'b' to backward, 'q' to quit:f
*****4*****
 9 8 7 6 5 4 3 2 1
香 桂 銀 金 王 金 銀 桂 香
歩 歩 歩 歩 歩 歩 歩 歩
      歩
歩 歩 歩 歩 歩 歩 歩 歩
香 桂 銀 金 王 金 銀 桂 香

```

結束遊戲，將棋譜寫檔，並重新開始覆盤

```

please enter 'f' to forward, 'b' to backward, 'q' to quit:f
*****
  9  8  7  6  5  4  3  2  1
  香  桂  銀  金  王  金  銀  桂  香
  步  步  步  步  步  步  步  步  步
          步
步  步  步  步  步  步  步  步  步
香  角  桂  銀  金  王  金  銀  桂  香
please enter 'f' to forward, 'b' to backward, 'q' to quit:b
*****3*****
  9  8  7  6  5  4  3  2  1
  香  桂  銀  金  王  金  銀  桂  香
  步  步  步  步  步  步  步  步  步
          步  步
步  步  步  步  步  步  步  步  步
香  角  桂  銀  金  王  金  銀  桂  香
please enter 'f' to forward, 'b' to backward, 'q' to quit:b
*****2*****
  9  8  7  6  5  4  3  2  1
  香  桂  銀  金  王  金  銀  桂  香
  步  飛  步  步  步  步  步  步  步
          步
步  步  步  步  步  步  步  步  步
香  角  桂  銀  金  王  金  銀  桂  香
please enter 'f' to forward, 'b' to backward, 'q' to quit:q
enter 'n' for new game, 'r' for shogi manual, 's' for saving the previous game, and 'q' to quit:q
jacob@jacob-VirtualBox:~/桌面/hw2$

```

倒退步數，並結束系統

### 3.分析

與作業二的資料儲存方式比較，作業二是使用陣列儲存 **structure** 內的東西，我們預設了最大步數做為陣列空間，而這次作業則是用 **linked list** 的方式去堆資料，我們也可把這種資料儲存視為動態空間。儲存下棋資料就用 **stack**，毀棋就用 **pop** 的概念去做。

### 4.問題探討

#### (1)使 **linked list** 具備 **stack** 跟 **pop** 的功能

儲存下棋的過程並不難，其實就是堆疊或者刪除節點，但我遇到的問題主要是在覆盤的時候如何讓資料流往回走，因為一般的鏈結串列是單向的，資料因此無法往回走，因此我就加上了往前找的指位器，讓其變成雙向的鏈結串列。此外，在尋找資料的過程中，我使用了 **stack** 跟 **queue** 的概念，使用兩個指標去把頭跟尾指出來，不讓資訊流斷掉，並用 **temp** 去做移動。

#### (2)寫檔異常

目前在寫檔部分遇到的問題是資料流可能有異常，如果在關閉程式之前是可以讀取檔案的，但關閉程式後重新打開就會顯示錯誤。目前也找不到從哪邊 **DEBUG**。