



Practice 3-1

Poker Card



Requirements

1. 你的程式必須可以產生52張撲克牌，並將他們"隨機"分配給4位玩家
 2. 輸入y開始遊戲，輸入n結束遊戲並離開程式
 3. 印出每個玩家手上的卡片，並顯示他們手上是否有對子
 4. 你的程式必須要可以重複執行，直到輸入n為止
-
- Note: 使用A~D代表花色，1~13代表數字
 - Note: 請記得在輸出結果後加上`Console.ReadKey(true)`，以便查看輸出結果



Hint

- 你可以以A_1~D_13來建立一個陣列，然後利用它來隨機發牌給玩家
- 你可以利用string.split, Array.indexOf, Array.resize等方法來撰寫你的程式

More Hint

- 如果你不知道怎麼隨機發牌的話

你可以用一個string array來存這52張卡(A_1~D_13)，並利用這個array來發牌給4位玩家(例如使用2D array `player[4][13]`來儲存)

在發牌時，你可以用`random.next(0, card.length)`來隨機選一張卡。發完卡之後，把陣列最後一項移動到該卡的位置。(ex. 你發出了B_1，則把最後的D_13複製到B_1的位置，此時陣列中會有兩個D_13，沒有B_1)

移動完之後，使用`Array.Resize`來調整card array的大小，同時移除最後一張卡片(D_13)，如此便能得到一個大小為`card.length-1`的array，同時也不會含有你已經給出的卡片。



More Hint (Cont.)

除了以上提到的作法，你也可以隨機挑選要發牌給哪個使用者。也就是說，隨機挑選哪個使用者會拿到A_1、A_2、...、D_13。使用這個方法的話你會使用到 `random.next(0, 4)` 和 `switch`

注意你不一定要使用上述的方法，只要你可以達到題目的要求即可

建議先自行想過一次後再考慮使用這些建議，別讓自己的思維被限制住了

Example

- Simulate code example:

```
file:///C:/Users/NightKirie/Desktop/1070925_Practice2/1070925_Practice2/bin/Debug/1070925_Practice2.EXE
輸入y來開始隨機發牌，輸入n結束遊戲:y
玩家1獲得的牌:C_13 C_12 C_11 C_10 D_10 C_9 C_8 C_7 D_6 C_6 C_5 D_3 C_4
玩家2獲得的牌:C_3 C_2 D_12 C_1 B_13 B_12 D_7 B_11 B_10 B_9 D_1 B_8 B_7
玩家3獲得的牌:D_9 B_6 B_5 B_4 D_2 B_3 B_2 B_1 D_5 A_13 A_12 A_11 D_8
玩家4獲得的牌:A_10 A_9 D_11 A_8 A_7 A_6 D_13 A_5 A_4 A_3 D_4 A_2 A_1
玩家1有對子
玩家2有對子
玩家3有對子
玩家4有對子
輸入y來開始隨機發牌，輸入n結束遊戲:
```

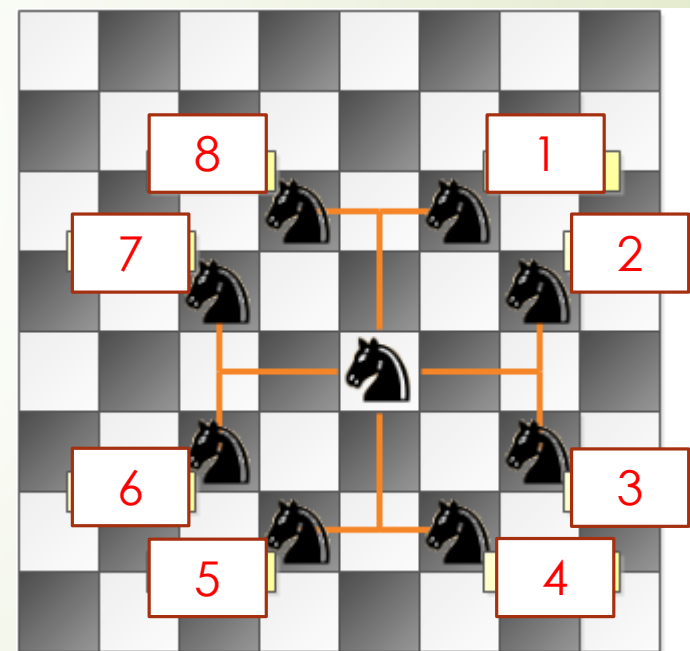
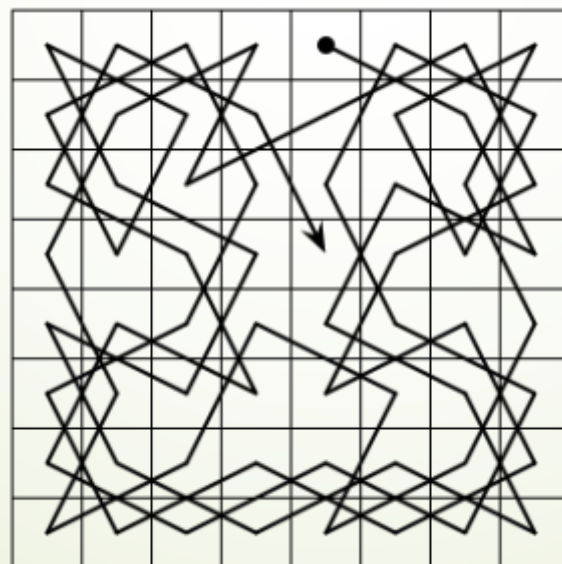


Practice 3-2

騎士走棋盤

Explanation

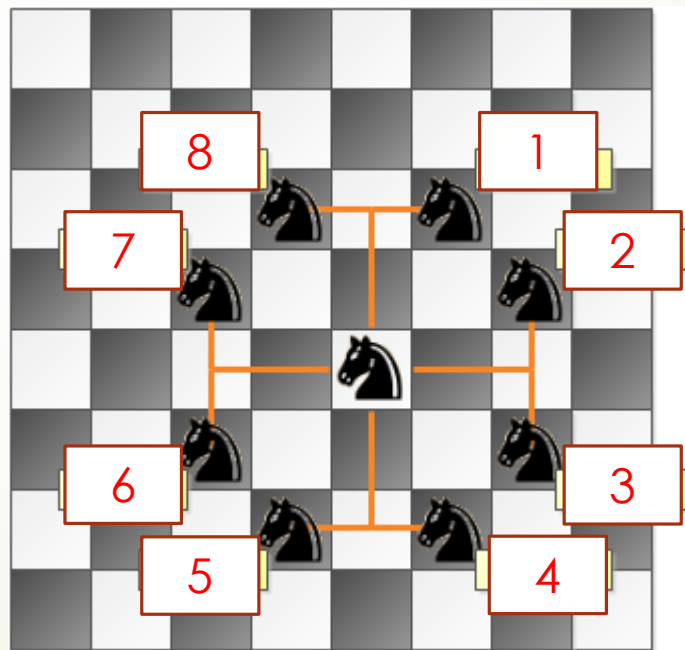
- 在一個棋盤上從隨意一個起點開始
- 每次動作依照"馬"的移動方式
- 走過的地方不重複走
- 最後棋盤上每個點都走過一次



Algorithm

➤ Warnsdorff規則

- 馬能走得必定是目前的位置的1~8種走法，1~8的走法中，每個當作又當作起點時，選擇接下來可以走的路徑最少的那個點，比如：當馬選擇走3的時候，他接下來能走的路剩下5條，所以3這個方向的值就是5，每次都先把最難走的路先走完，剩下的就容易了





Requirements

- 建立一個12*12的棋盤，先由左至右在由上至下排序1~144
 - 使用random來決定起始點，並印出起始點
 - 如果有解，則把路徑印出來
 - 如果無解，則印出"no answer"
-
- Note: 請記得在輸出結果後加上`Console.ReadKey(true)`，以便查看輸出結果

Example

```
file:///C:/Users/Administrator/Documents/Visual Studio 2015/Projects/20190921Csharp_prc3-2/20190921Csharp_prc3-2/bin/Debug/2019092...
起點 : 141
141 -> 131 -> 108 -> 83 -> 60 -> 35 -> 12 -> 22 -> 36 -> 11 -> 21 -> 7 ->
17 -> 3 -> 13 -> 38 -> 61 -> 86 -> 109 -> 134 -> 124 -> 138 -> 128 -> 142 ->
132 -> 118 -> 143 -> 120 -> 95 -> 72 -> 47 -> 24 -> 10 -> 20 -> 6 -> 31 ->
8 -> 33 -> 23 -> 48 -> 58 -> 81 -> 71 -> 96 -> 106 -> 129 -> 119 -> 144 ->
130 -> 140 -> 117 -> 107 -> 84 -> 94 -> 104 -> 127 -> 137 -> 114 -> 139 -> 116 ->
93 -> 70 -> 45 -> 59 -> 34 -> 9 -> 19 -> 5 -> 15 -> 1 -> 26 -> 49 ->
63 -> 73 -> 50 -> 25 -> 2 -> 16 -> 30 -> 40 -> 54 -> 44 -> 69 -> 46 ->
56 -> 79 -> 102 -> 92 -> 82 -> 105 -> 115 -> 125 -> 135 -> 121 -> 98 -> 123 ->
133 -> 110 -> 100 -> 77 -> 91 -> 68 -> 43 -> 57 -> 67 -> 90 -> 80 -> 103 ->
126 -> 136 -> 113 -> 88 -> 111 -> 101 -> 78 -> 55 -> 32 -> 42 -> 65 -> 75 ->
85 -> 99 -> 122 -> 112 -> 89 -> 66 -> 76 -> 53 -> 28 -> 18 -> 41 -> 51 ->
74 -> 97 -> 87 -> 64 -> 39 -> 29 -> 52 -> 62 -> 37 -> 14 -> 4 -> 27 ->

英
```