

## 第3次作業-作業-HW3

學號：112111209

姓名：張香裕

作業撰寫時間：150 (mins, 包含程式撰寫時間)

最後撰寫文件日期：2024/11/28

本份文件包含以下主題：(至少需下面兩項，若是有多者可以自行新增)

- ☒ 說明內容
- ☒ 個人認為完成作業須具備觀念

### 說明程式與內容

開始寫說明，該說明需說明想法，並於之後再對上述想法的每一部分將程式進一步進行展現，若需引用程式區則使用下面方法，若為.cs檔內程式除了於敘述中需註明檔案名稱外，還需使用語法```語言種類 程式碼```，其中語言種類若是要用python則使用py，java則使用java，C/C++則使用cpp，下段程式碼為語言種類選擇csharp使用後結果：

```
public void mt_getResult(){  
    ...  
}
```

若要於內文中標示部分網頁檔，則使用以下標籤```html 程式碼```，下段程式碼則為使用後結果：

```
<%@ Page Language="C#" AutoEventWireup="true" ...>  
  
<!DOCTYPE html>  
  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head runat="server">  
<meta http-equiv="Content-Type" ...>  
    <title></title>  
</head>  
<body>  
    <form id="form1" runat="server">  
        <div>  
            </div>  
    </form>  
</body>  
</html>
```

更多markdown方法可參閱<https://ithelp.ithome.com.tw/articles/10203758>

請在撰寫"說明程式與內容"該塊內容，請把原該塊內上述敘述刪除，該塊上述內容只是用來指引該怎麼撰寫內容。

1. 請回答下面問題。

Ans:

```
def isFull(Stack,top,N):
    if top==N-1: #如果top指標指向堆疊頂端,傳回True
        return True
    else: #否則傳回False
        return False

def isEmpty(stack,top):
    if top==-1: #如果top指標為-1,傳回True
        return True
    else: #否則傳回False
        return False
```

2. 請回答下面問題。

Ans:

```
def can_knight_cover_board(N, startX, startY):
    # 定義騎士的 8 種可能移動方式
    moves = [
        (-2, -1), (-2, 1), (-1, 2), (1, 2),
        (2, 1), (2, -1), (1, -2), (-1, -2)
    ]

    # 初始化棋盤, 0 表示未訪問過
    board = [[0 for _ in range(N)] for _ in range(N)]

    # 初始化堆疊
    stack = [(startX, startY, 1)] # (x, y, step_count)
    board[startX][startY] = 1 # 標記起始點已訪問

    while stack:
        x, y, step = stack.pop()

        # 如果所有格子都已訪問, 返回 True
        if step == N * N:
            return True

        # 遍歷 8 種可能的移動方式
        for dx, dy in moves:
            nx, ny = x + dx, y + dy
            if 0 <= nx < N and 0 <= ny < N and board[nx][ny] == 0: # 確保在棋盤內
                # 且未訪問過
                board[nx][ny] = step + 1 # 標記下一步
                stack.append((nx, ny, step + 1))

        # 如果此路徑無法完成任務, 回溯
        board[x][y] = 0
```

```
    return False

# 輸入
N = int(input("請輸入棋盤大小 N (4 <= N <= 10): "))
startX = int(input("請輸入起始點 X 座標 (0<=X<N): "))
startY = int(input("請輸入起始點 Y 座標 (0<=Y<N): "))

# 輸出
if can_knight_cover_board(N, startX, startY):
    print("True")
else:
    print("False")
```

3. 請回答下面問題：

Ans:

```
def josephus(n, k):
    # 使用遞迴解決約瑟夫問題
    if n == 1:
        return 1
    else:
        return (josephus(n - 1, k) + k - 1) % n + 1

# 輸入
n = int(input("請輸入朋友數量 n: "))
k = int(input("請輸入報數間隔 k: "))

# 輸出
result = josephus(n, k)
print("最後留下來的的人的編號是:", result)
```

## 個人認為完成作業須具備觀念

開始寫說明，需要說明本次練習需學會那些觀念 (需寫成文章，需最少50字，並且文內不得有你、我、他三種文字)且必須提供完整與練習相關過程的notion筆記連結

- 對於堆疊(Stack)的結構應用及操作。
- 了解迴圈及遞迴的發法，解決約瑟夫問題。
- 堆疊結合條件判斷式。