

第2次作業-作業-HW2

學號：1234567

姓名：葉崇聖

作業撰寫時間：180 (mins · 包含程式撰寫時間)

最後撰寫文件日期：2023/10/26

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. 問題如下圖所述，並回答下面問題。

Ans:

```
from typing import List

def getResult():
    alphabet1: List[List[chr]] = [['1','2','3','4','5','6','7','8','9','0'],
                                    ['Q','W','E','R','T','Y','U','I','O','P'],
                                    ['A','S','D','F','G','H','J','K','L',';'],
                                    ['Z','X','C','V','B','N','M',',','.','/']],

    alphabet2: List[List[chr]] = [['!','@','#','$','%','^','&','*','(',')'],
                                    ['Q','W','E','R','T','Y','U','I','O','P'],
                                    ['A','S','D','F','G','H','J','K','L',':'],
                                    ['Z','X','C','V','B','N','M','<','>','?']],

    #陣列排列 alphabet1 = 下排陣列 alphabet2 = 上排陣列

    #測試該筆數
    n = int(input("請輸入測試筆數 N :"))

    # 輸入符號跟方向
    for _ in range(n):

        value, direction = input("請輸入 符號 與 方向 (1:上 · 2:下 · 3:右 · 4:左):").split()
        direction = int(direction)
        found = False

        # alphabet1 尋找 value位置
        for i in range(len(alphabet1)):
            if value in alphabet1[i]:
                j = alphabet1[i].index(value)
                found = True

                #依據 direction 移動並在 alphabet1 中找出新位置的字符
                if direction == 1: #上
                    new_value = alphabet1[i-1][j]
                elif direction == 2: #下
                    new_value = alphabet1[i+1][j]
                elif direction == 3: #右
                    new_value = alphabet1[i][j+1]
                elif direction == 4: #左
                    new_value = alphabet1[i][j-1]
                else:
                    new_value = value
                print(new_value)
                break

    # 如果在 alphabet1 沒找到,再找 alphabet2 中查找
```

```

        if not found:
            for i in range(len(alphabet2)):
                if value in alphabet2[i]:
                    j = alphabet2[i].index(value)

                    #根據 direction 移動並找出新位置的字符
                    if direction == 1: # 上
                        new_value = alphabet2[i-1][j]
                    elif direction == 2: #下
                        new_value = alphabet2[i+1][j]
                    elif direction == 3: #右
                        new_value = alphabet2[i][j+1]
                    elif direction == 4: #左
                        new_value = alphabet2[i][j-1]
                    else:
                        new_value = value

                    print(new_value)
                    break

# 執行函式
getResult()

```

2. 給定一個包含 n 個不同數字的數組，這些數字的範圍是從 0 到 n 。找出數組中缺失的那一個數字。

Ans:

```

def missingNumber(nums):
    n = len(nums)
    total_sum = n * (n + 1) // 2 # 計算 0 到 n 的總和
    actual_sum = sum(nums) # 計算數組的總和
    return total_sum - actual_sum # 找出缺失的數字

# 測試範例
print(missingNumber([3, 0, 1])) # 輸出: 2
print(missingNumber([9, 6, 4, 2, 3, 5, 7, 0, 1])) # 輸出:

```

3. 請回答下面問題：

Ans:

```

a.
f(n)=2^n+1
g(n)=2^n
f(n)<=0(g(n))
2^n+1<=c*2^n    "2^n相消除"
2^1<=c           "成立"
a. Ans: 等於(成立)

```

b.
 $f(n) = 2^{2^n}$
 $g(n) = 2^n$
 $f(n) \leq O(g(n))$
 $2^{2n} \leq c * g(n)$
 $2^{2n+n} \leq c * 2^n$ "2^n相消除"
 $2^{3n} \leq c$ "不成立"
 b. Ans: 不等於(不成立)

4. 請問以下各函式，在進行呼叫後，請計算(1)執行次數 $T(n)$ ，並(2)透過執行次數判斷時間複雜度為何(請用Big-Oh進行表示)？

Ans:

a.

```
def calculateTimes (number: int) -> None:
    while number >= 1:
        counter:int = number
        while counter >= 1:
            print(number, counter)
            counter = counter - 1
        number = number - 1
```

#n+1
 #n
 #(n+1+2)*n/2
 #(n+1)*n/2
 #(n+1)*n/2
 #n

$$(1) T(n) = \frac{3}{2}n^2 + \frac{11}{2}n + 1$$

$$(2) T(n) = O(n^2)$$

b.

```
def calculateTimes (number: int) -> None:
    while number >= 1:
        print(number)
        number = number // 2
```

#floor(log₂n)+2
 #floor(log₂n)+1
 #floor(log₂n)+1

$$(1) T(n) = 3(\log_2 n) + 4$$

$$(2) T(n) = O(\log_2 n)$$

c.

```
def calculateTimes (number: int, size: int) -> None:
    while number >= 1:
        while size >= 1:
            print(number, size)
            size = size - 1
        number = number // 2
```

#floor(log₂n)+2
 #(m+1)
 #m(floor(log₂n)+1)

```
#m(floor(log_{2}n)+1)
    number = number // 2
```

```
#floor(log_{2}n)+1
```

$$(1) T(n, m) = (3m + 3)(\log 2n) + 3m + 4$$

$$(2) T(n, m) = O(m \log 2n)$$

```
d.
    #if m=n(最大值)
    def calculateTimes (number: int, size: int) -> None:
        while number >= 1:
            while size >= 1:
                print(number, size)
                size = size - 1
            number = number // 2

    #if m=n/2(最小值)
    def calculateTimes (number: int, size: int) -> None:
        while number >= 1:
            while size >= 1:
                print(number, size)
                size = size - 1
            number = number // 2
```

$$(1) (3n + 3)\lfloor (\log 2n) \rfloor + 3n + 4 \geq T(n) \geq \left(\frac{3n}{2} + 3\right)\lfloor (\log 2n) \rfloor + \frac{3n}{2} + 4$$

$$(2) T(n) = O(n \log 2n)$$

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