

## 第2次作業-作業-HW2

學號：112111206

姓名：李儲亮

作業撰寫時間：180 (mins, 包含程式撰寫時間)

最後撰寫文件日期：2023/09/22

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:

```
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','<','>','?']]
    #因為部分按鍵具備雙重符號，部分按鍵有雙符號，因此分為下排和上排兩個陣列

    n=int(input())
    #n為要輸入幾組資料
    repeat=set()
    #此為了避免判定不適兩種符號按鍵時，輸出兩次結果
    for i in range(n):
        value,direction=input().split()
        #輸入按鍵，並判斷方向
        direction=int(direction)
        for j in range(len(alphabet1)):
            if value in alphabet1[j]:
                k= alphabet1[j].index(value)
                #判斷value在陣列的位子
                if direction==1:
                    ans=(alphabet1[j-1][k])
                elif direction==2:
                    ans=(alphabet1[j+1][k])
                elif direction==3:
                    ans=(alphabet1[j][k+1])
                elif direction==4:
                    ans=(alphabet1[j][k-1])
                #依照direction的數字來輸出對應方向的按鍵
                print(ans)
                repeat.add(ans)
                #將輸出的值加入集合，避免下一個判定重複輸出
        for j in range(len(alphabet2)):
            if value in alphabet2[j]:
                k= alphabet2[j].index(value)
                #第二次判定，判斷第二個陣列
                if direction==1:
                    ans=(alphabet1[j-1][k])
                elif direction==2:
                    ans=(alphabet1[j+1][k])
                elif direction==3:
                    ans=(alphabet1[j][k+1])
                elif direction==4:
                    ans=(alphabet1[j][k-1])
                if ans not in repeat:
                    print(ans)
                #照direction的數字來輸出對應方向的按鈕(最後的if判斷第一個陣列是否已經輸出過相同符號)
```

```
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:

```
#HW1_2py
num=input()
#依題目輸入,例如 : nums = [3, 0, 1 ]
nums=eval (num. split('=') [1].strip())
#只取數組的部分
n=len(nums)
#判斷數組長度
sum1=n* ( n+1 ) /2
#計算0加到n的總和
sum2=sum ( nums )
#計算數組內數字總和
miss=sum1-sum2
#相減來得知缺少的數字
print(int(miss))
```

3. 請回答下面問題：

a.

Ans:成立

因為BigO定理:  $f(n) = 2^{n+1}, g(n) = 2^n$

所以:  $f(n) \leq O(g(n))$

即:  $2^{n+1} \leq c \times g(n)$

因此可拆解:  $2^n \times 2^1 \leq c \times 2^n, 2^1 \leq c$

可以滿足BigO, 故此解成立

b.

Ans:不成立

因為BigO定理:  $f(n) = 2^2n, g(n) = 2^n$

所以:  $f(n) \not\leq O(g(n))$

即:  $2^2n \leq c \times g(n)$

因此可拆解:  $2^n \times 2^n \leq c \times 2^n, 2^n \leq c$

不能滿足BigO，故此解不成立。

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:           # n+1
        counter: int = number    # n
        while counter >= 1:      # (n+1+2)*n/2
            print(number, counter) # (n+1)*n/2
            counter = counter - 1 # (n+1)*n/2
        number = number - 1      # 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: # floor(log_2(n))+2
        print(number) # floor(log_2(n))+1
        number = number // 2 # floor(log_2(n))+1
```

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

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

c.

```
def calculateTimes (number: int, size: int) -> None:
    while number >= 1: # floor(log_2(n))+2
        while size >= 1: # (m+1)(floor(log_2(n))+1)
            print(number, size) # m(floor(log_2(n))+1)
            size = size - 1 # m(floor(log_2(n))+1)
        number = number // 2 # floor(log_2(n))+1
```

$$(1) T(n, m) = (3m + 3)\lfloor \log_2 n \rfloor + 3m + 4$$

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

d.

```
#if m=n (最大值)
def calculateTimes (number: int, size: int) -> None:
    while number >= 1: # floor(log_2(n))+2
        while size >= 1: # (n+1)(floor(log_2(n))+1)
            print(number, size) # n(floor(log_2(n))+1)
            size = size - 1 # n(floor(log_2(n))+1)
            number = number // 2 # floor(log_2(n))+1
```

```
#if m=n/2 (最小值)
def calculateTimes (number: int, size: int) -> None:
    while number >= 1: # floor(log_2(n))+2
        while size >= 1: # (n/2+1)(floor(log_2(n))+1)
            print(number, size) # n/2(floor(log_2(n))+1)
            size = size - 1 # n/2(floor(log_2(n))+1)
            number = number // 2 # floor(log_2(n))+1
```

$$(1) (3n + 3)\lfloor \log_2 n \rfloor + 3n + 4 \geq T(n) \geq \left(\frac{3n}{2} + 3\right)\lfloor \log_2 n \rfloor + \frac{3n}{2} + 4$$

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

## 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