

資料結構 Homework#1 F74109016 葉惟欣
實作的程式碼

```
1 //F74109016 葉惟欣 finished in 2021/10/18
2 #include<stdio.h>
3 #include<stdlib.h>
4 #include<string.h>
5 int iterative(int n){ //傳入最後要印iterative第幾項
6     int f0 =0,f1 =1,f2,i;
7     if(n==0||n==1){
8         return n;
9     }
10    else{
11        for(i=1;i<n;i++){
12            f2 = f0+f1;
13            f0 = f1;
14            f1 = f2;
15        }
16        return f2;
17    }
18 }
19 int recursive(int i){ //傳入最後要印recursive第幾項
20     if(i==0){
21         return 0;
22     }
23     else if(i==1){
24         return 1;
25     }
26     else{
27         return recursive(i-1)+recursive(i-2);
28     }
29 }
30 int main(){
31     int len,num; //len represents the amount of questions.
32                 //num represents the amount of number
33     scanf("%d",&len);
34     int i;
35     char string [20];
36     int answer[len];
37     for(i=0;i<len;i++){
38         scanf("%s",string);
39         scanf("%d",&num);
40         if(strcmp(string,"recursive")==0){
41             answer[i] = recursive(num);
42         }
43         else if(strcmp(string,"iterative")==0){
44             answer[i] = iterative(num);
45         }
46     }
47     printf("\n");
48     for(i=0;i<len;i++){
49         printf("%d\n",answer[i]);
50     }
51     return 0;
```

遞迴的程式碼由 19 行開始，

當傳入的項數(i)為 1 則回傳 1，傳入的項數為 0 則回傳 0。

如果傳入的項數(i)非 0 與 1 則回傳 項數(i-1)的值加上項數(i-2)。

而項數(i-1)的值為何?則要再次呼叫 recursive 函數，來得到該值。
項數(i-2)的值為何?則要再次呼叫 recursive 函數，來得到該值。

舉例當要求 i=3 時 呼叫 recursive(i=3)

第一次呼叫 recursive(i=3)

呼叫 recursive(i=3)的函數，因為 i 不為 1 或 0 則回傳
recursive(i=2)+recursive(i=1)的值。

而因為這是回傳函數，所以會先呼叫函數得到其值才做回傳。

第二次呼叫 recursive(i=2)

呼叫 recursive(i=2)的函數 因為同樣 i 不為 1 或 0 則回傳
recursive(i=1)+recursive(i=0)。同樣會先呼叫函數得到其值才做回傳。

第三次呼叫 recursive(i=1)

呼叫 recursive(i=1)的函數 因為 i 為 1 則回傳 1。

回傳後就回到第二次呼叫 1+recursive(i=0)，再呼叫函數得到 recursive(i=0)其值才
做回傳。

第四次呼叫 recursive(i=0)

呼叫 recursive(i=0)的函數 因為 i 為 0 則回傳 0 給第二次呼叫。

回傳後就回到第二次呼叫 1+0，則再回傳 1 給第一次呼叫。

第五次呼叫 recursive(i=1)

當 recursive(i=2)在先前已經得到其值為 1 了，再呼叫 recursive(i=1)得到其值才做
回傳。因為 i=1 則直接回傳給第一次呼叫的 recursive(i=3)的函數裡。

因此我們可以在 recursive(i=3)裡面得到回傳值 1+1=2。則呼叫 recursive(i=3)得到
其值 2。

迭代函數的程式碼實作從第 5 行開始。

傳入的值是要求 iterative 的第幾項數。

一開始先設 f0 = 0 與 f1 =1;

如果要求的項數(n)為 1 或 0 則依照題意可直接回傳 n 的值。

如果項數(n)不為 1 或 0 時則做 for 迴圈。

舉例 $n=3$ 而言。

for 迴圈從 $i=1$ 開始 $i=1 < n=3$ 可以執行迴圈裡面的 body。

$f2 = f0(0) + f1(1) = 1 ==>$ 已經算出項數 2 了

項數 2 為現在要求的項數(計算邏輯：前兩項+前一項=現在要求的項數)

為了下一次項數的計算做的準備：

將 $f0$ 設為 $f1$ 則 $f0=1$ 將"現在要求的項數的前一項"設為"下一項的前兩項"

將 $f1$ 設為 $f2$ 則 $f1=1$ 將"現在要求的項數"設為"下一項的前一項"

$i++$

$i=2 < n=3$ 可以執行迴圈裡面的 body。

$f2 = f0(1) + f1(1) = 2 ==>$ 已經算出項數 3 了

項數 3 為現在要求的項數(計算邏輯：前兩項+前一項=現在要求的項數)

為了下一次項數的計算做的準備：

將 $f0$ 設為 $f1$ 則 $f0=1$ 將"現在要求的項數的前一項"設為"下一項的前兩項"

將 $f1$ 設為 $f2$ 則 $f1=2$ 將"現在要求的項數"設為"下一項的前一項"

$i++$ ， $i=3$ 不小於 $n=3$ 跳出迴圈

回傳目前的 $f2$ 其值為 2 也就是項數 3 的值。