C Turnin作業1

國立屏東大學 資訊工程系 程式設計(一)

Turnin作業1

- Turnin Code: c.hw1
- Due Date: 2026/03/11 週三 晚上 23:59分截止 **Hard Deadline**

繳交方式說明

本次 Turnin 作業包含多個程式題，建議先為本次 turnin 要繳交的內容建立一個**外層資料夾**（例如「c.hw1」）切換到該資料夾後再為每一題建立一個**內層資料夾**（每一題的資料夾名稱已寫於題目前方，例如第一題的資料夾名稱為「p1」，第二題的為「p2」，以此類推），進入到內層資料夾才依照題目要求進行編撰。

同學們可參考如下命令列操作：

```
<ssh 登入系計中後>
[user@ws ~]$ mkdir c.hw1          # 在家目錄建立了一個名為「hw1」的資料夾
[user@ws ~]$ cd c.hw1             # 進入「hw1」資料夾
[user@ws c.hw0]$ mkdir p1         # 建立一個名為「p1」資料夾
[user@ws c.hw0]$ cd p1            # 切換到「p1」資料夾
[user@ws p1]$ joe SPNum.c         # 使用 JOE 編輯器對檔名為 SPNum.c 的檔案進行編輯
```

等到我們完成 p1 的撰寫後，請自行加以編譯與執行程式，確認**正確無誤**後回到**外層資料夾**使用 `turnin␣c.hw1␣p1↵` 指令完成**繳交第一題的整個資料夾**。

```
[user@ws p1]$ cd ..                # 回到上一層資料夾
[user@ws c.hw1]$ turnin c.hw1 p1 # 使用 turnin 指令提交 p1 的程式碼
Turning in:
 p1/SPNum.h -- ok
 p1/SPNum.c -- ok
All done.
[user@ws c.hw1]$
```

當然，你也可以等到本次作業要求的所有題目都在 hw1 資料夾裡完成後，一次將所有在**目前資料夾中的所有檔案**都加以上傳。

假設你已經在 hw1 資料夾裡完成所有題目，同時確認檔案的繳交格式正確，並且每個題目的程式檔案皆成功編譯並確認執行結果正確後，我們可以使用以下指令將多餘的（不需要繳交的）檔案加以刪除後，一次將所有檔案繳交：

```
[user@ws c.hw1]$ ls                    #檢視當前資料夾下有哪些內容
p1 p2 p3 p4 p5 p6
[user@ws c.hw1]$ rm -f */a.out         #移除所有子資料夾中的 a.out 檔案
[user@ws hw1]$ turnin c.hw1 .          #使用 turnin 指令繳交該資料夾下的所有內容
Turning in:
 ./p3/digitSum.c -- ok
 ./p3/digitSum.h -- ok
 ./p1/SPNum.h -- ok
 ./p1/SPNum.c -- ok
 ./p2/fibonacci.c -- ok
 ./p2/fibonacci.h -- ok
All done.
[user@ws c.hw1]$
```

如果繳交後想要查看已繳交的檔案及相關資訊，可以輸入 `turnin␣-ls␣c.hw1↵` 指令，例如：

```
[user@ws ~]$ turnin -ls c.hw1
.:
total 24
drwxrwx---. 2 turninman turnin 4096 Mar  4 16:32 p1
drwxrwx---. 2 turninman turnin 4096 Mar  4 16:32 p2
drwxrwx---. 2 turninman turnin 4096 Mar  4 16:32 p3

./p1:
total 8
-rw-rw----. 1 turninman turnin 139 Mar  4 16:32 SPNum.c
-rw-rw----. 1 turninman turnin  18 Mar  4 16:32 SPNum.h

./p2:
total 8
-rw-rw----. 1 turninman turnin 155 Mar  4 16:32 fibonacci.c
-rw-rw----. 1 turninman turnin  21 Mar  4 16:32 fibonacci.h

./p3:
total 8
-rw-rw----. 1 turninman turnin 151 Mar  4 16:32 digitSum.c
-rw-rw----. 1 turninman turnin  20 Mar  4 16:32 digitSum.h
[user@ws ~]$
```

本文使用「 」及「 」代表「空白字元」與Enter 換行字元」，並且將使用者輸入的部份使用灰階方式顯示。另外，題目的執行結果中，如果出現「(」、「)」、「:」、「;」、「.」與「,」等符號，皆為英文半形！

本學期作業繳交需要為每一題建立一個資料夾（資料夾名稱為該題題目前方之代號，第一題為p1第二題為p2餘以此類推），繳交方式可參考上述內容，任何未依

照正確繳交格式的檔案將以 0 分計。

p1 四角錐數

四角錐數（或稱爲金字塔數）是一個表示底面為正方形的金字塔能堆積的球數總和，其公式如下 $$S_n=1^2+2^2+3^2+...+n^2$$ 其中n是金字塔的層數。 請參考以下的main.c主程式（可在/home/stu/public/c2026s/c.hw1/p1/取得):

```
#include <stdio.h>
#include "SPNum.h"
int main()
{
   int n;
   printf("Please input an integer: ");
   scanf("%d", &n);
   printf("The quare pyramidal of %d is %d\n", n, SPNum(n));
}
```

你必須完成名為SPNum.c與SPNum.h的C語言程式，其中分別包含SPNum()函式的實作與其原形宣告SPNum()函式接生一個整數做為參數，並透過遞迴的方式將當前層的平方與前一項的和傳回作為運算結果。本題的相關程式將使用以下的Makefile進行編譯：

```
all: main.c SPNum.o
        cc main.c SPNum.o

SPNum.o: SPNum.c SPNum.h
        cc -c SPNum.c

clean:
        rm -f *.o *~ *.*~ a.out
```

此題的執行結果如下：

```
[3:23 user@ws p1] ./a.out↵
Please▲input▲an▲integer:▲0↵
The▲quare▲pyramidal▲of▲0▲is▲0↵
[3:23 user@ws p1] ./a.out↵
Please▲input▲an▲integer:▲1↵
The▲quare▲pyramidal▲of▲1▲is▲1↵
```

```
[3:23 user@ws p1] ./a.out↵
Please▲input▲an▲integer:▲8↵
The▲quare▲pyramidal▲of▲8▲is▲204↵
[3:23 user@ws p1] ./a.out↵
Please▲input▲an▲integer:▲15↵
The▲quare▲pyramidal▲of▲15▲is▲1240↵
[3:23 user@ws p1]
```

1. 請注意本題應繳交SPNum.c與SPNum.h兩個檔案，至於Main.c與Makefile則不需繳交。
2. 本題必須使用遞迴的方式，透過SPNum()函式的遞迴呼叫完成計算，若採用其他方式將不予計分。
3. 本題輸入介於0~100之間。

p2 費伯納西數列

費伯納西數列(fibonacci sequnce)的第n項的值等於其前兩項的和：
\[F_n=F_{n-1}+F_{n-2}\] 且其前兩項的值被定義為：
\[F_0=0 以及 F_1=1\]

請設計C語言程式fibonacci.c以及其標頭檔fibonacci.h以遞迴recursion方式設計一個名為fibonacci()的函式，並搭配下列的main.c程式（可在/home/stu/public/c2026s/c.hw1/p2/取得）完成第N項費伯納西數的輸出，其中N為大於等於0的正整數：

```
#include <stdio.h>
#include "fibonacci.h"

int main()
{
    int N;
    printf("N=? ");
    scanf("%d", &N);
    printf("F_%d=%d.\n", N, fibonacci(N) );
}
```

你必須完成名為fibonacci.c與fibonacci.h的C語言程式，其中分別包含fibonacci()函式的實作與其原形宣告fibonacci()函式接生一個整數做為參數，並透過遞迴的方式將其前兩項的和傳回做為運算結果。本題的相關程式將使用以下的Makefile進行編譯：

```
all: main.c fibonacci.o
        cc main.c fibonacci.o

fibonacci.o: fibonacci.c fibonacci.h
```

```
        cc -c fibonacci.c

clean:
        rm -f *.o *~ *.*~ a.out
```

此題的執行結果如下：

```
[3:23 user@ws p2] ./a.out↵
N=?▲0↵
F_0=0.↵
[3:23 user@ws p2] ./a.out↵
N=?▲1↵
F_1=1.↵
[3:23 user@ws p2] ./a.out↵
N=?▲3↵
F_3=2.↵
[3:23 user@ws p2] ./a.out↵
N=?▲10↵
F_10=55.↵
[3:23 user@ws p2]
```

1. 本題應繳交fibonacci.c與fibonacci.h兩個檔案，至於main.c與Makefile則不需繳交。
2. 本題必須使用遞迴方式，透過fibonacci()函式的遞迴呼叫完成計算，若採用其它方式將不予計分。
3. 由於遞迴程式耗用記憶體空間與處理器資源，且過大的項次將會發生整數型態溢位的問題。因此本題輸入之測試資料將不會超過40。

p3 整數位數加總

透過取餘數（%）與整除（/）將一個大數字拆解。
例如：123 的位數和是 $1+2+3=6$。
請設計一個C語言程式digitSum.c讓使用者輸入一個正整數。撰寫一遞迴函式計算該數字每一位數相加的總和。若輸入的值為負數，則顯示 “Error!” 請參考以下的main.c主程式（可在/home/stu/public/c2026s/c.hw1/p3/取得):

```
#include <stdio.h>
#include "digitSum.h"
int main()
{
   int n;
```

```
   printf("Enter an integer: ");
   if (scanf("%d", &n) != 1 || n < 0) {
      printf("Error!\n");
      return 0;
   }
   printf("The sum of digits is %d.\n", digitSum(n));
   return 0;
}
```

你必須完成名為digitSum.c與digitSum.h的C語言程式，其中分別包含digitSum()函式的實作與其原形宣告。digitSum()函式接生一個整數做為參數，並透過遞迴的方式計算該數字每一位數相加的總和傳回做為運算結果。本題的相關程式將使用以下的Makefile進行編譯：

```
all: main.c digitSum.o
        cc main.c digitSum.o

digitSum.o: digitSum.c digitSum.h
        cc -c digitSum.c

clean:
        rm -f *.o *~ *.*~ a.out
```

此題的執行結果如下：

```
[3:23 user@ws p3] ./a.out↵
Enter▲an▲integer:▲123↵
The▲sum▲of▲digits▲is▲6.↵
[3:23 user@ws p3] ./a.out↵
Enter▲an▲integer:▲732↵
The▲sum▲of▲digits▲is▲12.↵
[3:23 user@ws p3] ./a.out↵
Enter▲an▲integer:▲0↵
The▲sum▲of▲digits▲is▲0.↵
[3:23 user@ws p3] ./a.out↵
Enter▲an▲integer:▲-7↵
Error!↵
[3:23 user@ws p3]
```

1. 請注意本題應繳交digitSum.c與digitSum.h兩個檔案，至於main.c與Makefile則不需繳交。
2. 本題必須使用遞迴的方式，透過digitSum()函式的遞迴呼叫完成計算，若採用其他方式將不予計分。
3. 提示：利用 $n\%10$取得最後一位數，並利用 $n/10$取得剩餘位數。

From:
https://junwu.nptu.edu.tw/dokuwiki/ - **Jun Wu的教學網頁**
國立屏東大學資訊工程學系
CSIE, NPTU
Total: 260110

Permanent link:
https://junwu.nptu.edu.tw/dokuwiki/doku.php?id=c:2026spring:hw1

Last update: **2026/03/06 07:25**