

Introduction to Computer Science Fall 2022 #17 Chi-Jen Wu



~~B1129004 男 陳韋澄~~ (已點表情回覆)

B1129016 男 李浩霆

B1129032 男 鄭昊宏

~~B1129036 男 謝天峻~~

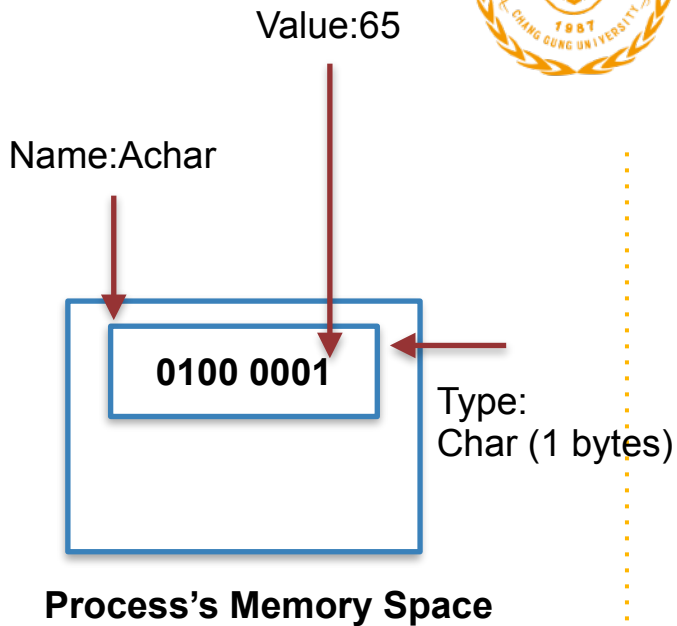
B1129054 男 劉宇翔

大一張哲維導師把以上五位同學的導
聚都排在1/05四 20:30 ~ 22:00，
請以上五位同學看到後不管時間可不
可以，都請回覆導師一下。

下午 2:07

Variables

- 值 (Value)
 - 在記憶體實際的二進位值
- 名字 (Name)
 - 在程式裡的代號
- 型別 (Type)
 - 在記憶體所佔的空間



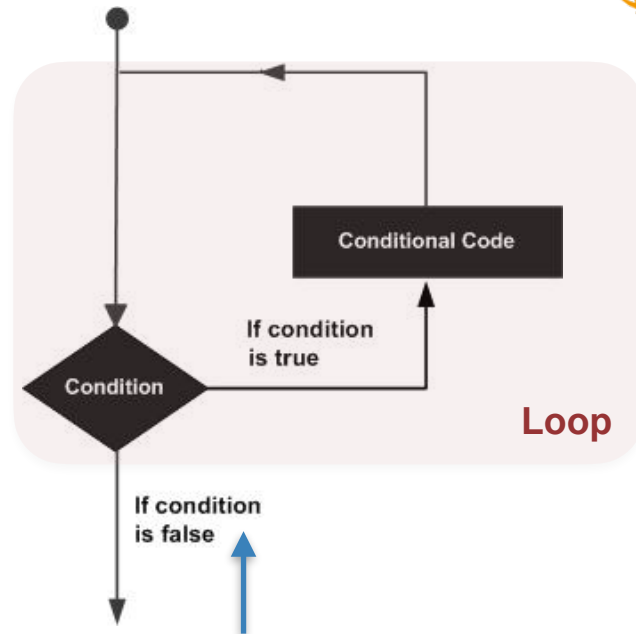
You should know that!

- Integer
 - Overflow
 - MIN & MAX
 - Rounding
- floating-point
 - 浮點數的陷阱
 - Rounding
 - 定點數
 - 無法取%
 - 無法用邏輯運算子 (& | ~ ^)，(其實是可以，高級技巧)



Flow of Control

- if
- If else
- If else if else
- Switch case
- While
- For
- Do while loop
- For loop
- Do While loop
- Nested loops
- Break
- Continue



Condition 決定要不要轉彎



```
1 #include <stdio.h>
2
3 int main() {
4     double pi = 4.0;
5     int flag = 1;
6     long ipi = 0;
7     int x = 0;
8     for (int i = 3; i <= 1000000; i+=2) {
9         if (flag == 0) {
10             pi += (4.0/i);
11             flag++;
12         } else {
13             pi -= (4.0/i);
14             flag--;
15         }
16         ipi = pi * 100000; ←
17         if (ipi == 314159) {
18             x = i;
19             break;
20         }
21     }
22     printf("%d %.5f", x, pi);
23     // 272241 3.14159
24     return 0;
25 }
```

272241 3.14160

...Program finished with exit code 0
Press ENTER to exit console.

- 計算PI by Gregory-Leibniz Series
- $PI = (4/1) - (4/3) + (4/5) - (4/7) + (4/9) - \dots$
- 請利用 for loop 來計算 pi
- double PI = 4.0f;
- 找到 x 計算出來 = 3.14159 (精準五位數)



```
1 #include <stdio.h>
2
3 int main() {
4     for (int i=1; i<=9; i++) {
5         for (int j=1; j<=9; j++) {
6             printf("%d*%d=%d\t", i, j, i*j);
7         }
8         printf("\n");
9     }
10 }
```

● 九九乘法表

- 兩個loop
- 請改成一個loop

input								
1*1=1	1*2=2	1*3=3	1*4=4	1*5=5	1*6=6	1*7=7	1*8=8	1*9=9
2*1=2	2*2=4	2*3=6	2*4=8	2*5=10	2*6=12	2*7=14	2*8=16	2*9=18
3*1=3	3*2=6	3*3=9	3*4=12	3*5=15	3*6=18	3*7=21	3*8=24	3*9=27
4*1=4	4*2=8	4*3=12	4*4=16	4*5=20	4*6=24	4*7=28	4*8=32	4*9=36
5*1=5	5*2=10	5*3=15	5*4=20	5*5=25	5*6=30	5*7=35	5*8=40	5*9=45
6*1=6	6*2=12	6*3=18	6*4=24	6*5=30	6*6=36	6*7=42	6*8=48	6*9=54
7*1=7	7*2=14	7*3=21	7*4=28	7*5=35	7*6=42	7*7=49	7*8=56	7*9=63
8*1=8	8*2=16	8*3=24	8*4=32	8*5=40	8*6=48	8*7=56	8*8=64	8*9=72
9*1=9	9*2=18	9*3=27	9*4=36	9*5=45	9*6=54	9*7=63	9*8=72	9*9=81

...Program finished with exit code 0
Press ENTER to exit console.□



```
1 #include <stdio.h>
2
3 int main() {
4     for (int i=1, j=0; i <= 9; ) {
5         printf("%d*%d=%d\t", i, j, i*++j);
6         if (j == 9) {
7             i++;
8             j = 0;
9             printf("\n");
10        }
11    }
12 }
```

- 九九乘法表
 - 兩個loop
 - 請改成一一個loop

input

1*1=1	1*2=2	1*3=3	1*4=4	1*5=5	1*6=6	1*7=7	1*8=8	1*9=9
2*1=2	2*2=4	2*3=6	2*4=8	2*5=10	2*6=12	2*7=14	2*8=16	2*9=18
3*1=3	3*2=6	3*3=9	3*4=12	3*5=15	3*6=18	3*7=21	3*8=24	3*9=27
4*1=4	4*2=8	4*3=12	4*4=16	4*5=20	4*6=24	4*7=28	4*8=32	4*9=36
5*1=5	5*2=10	5*3=15	5*4=20	5*5=25	5*6=30	5*7=35	5*8=40	5*9=45
6*1=6	6*2=12	6*3=18	6*4=24	6*5=30	6*6=36	6*7=42	6*8=48	6*9=54
7*1=7	7*2=14	7*3=21	7*4=28	7*5=35	7*6=42	7*7=49	7*8=56	7*9=63
8*1=8	8*2=16	8*3=24	8*4=32	8*5=40	8*6=48	8*7=56	8*8=64	8*9=72
9*1=9	9*2=18	9*3=27	9*4=36	9*5=45	9*6=54	9*7=63	9*8=72	9*9=81

...Program finished with exit code 0
Press ENTER to exit console.



```
1 #include <stdio.h>
2
3 int main() {
4     for (int i=1, j=0; i<=9 ; ) {
5         printf("%d*%d=%d\t", i, j, i*++j);
6         (j==9) ? (i++, j=0, printf("\n")) : i ;
7     }
8     return 0;
9 }
```

● 九九乘法表

- 兩個loop
- 請改成一一個loop
- 三元運算子

input

1*1=1	1*2=2	1*3=3	1*4=4	1*5=5	1*6=6	1*7=7	1*8=8	1*9=9
2*1=2	2*2=4	2*3=6	2*4=8	2*5=10	2*6=12	2*7=14	2*8=16	2*9=18
3*1=3	3*2=6	3*3=9	3*4=12	3*5=15	3*6=18	3*7=21	3*8=24	3*9=27
4*1=4	4*2=8	4*3=12	4*4=16	4*5=20	4*6=24	4*7=28	4*8=32	4*9=36
5*1=5	5*2=10	5*3=15	5*4=20	5*5=25	5*6=30	5*7=35	5*8=40	5*9=45
6*1=6	6*2=12	6*3=18	6*4=24	6*5=30	6*6=36	6*7=42	6*8=48	6*9=54
7*1=7	7*2=14	7*3=21	7*4=28	7*5=35	7*6=42	7*7=49	7*8=56	7*9=63
8*1=8	8*2=16	8*3=24	8*4=32	8*5=40	8*6=48	8*7=56	8*8=64	8*9=72
9*1=9	9*2=18	9*3=27	9*4=36	9*5=45	9*6=54	9*7=63	9*8=72	9*9=81

...Program finished with exit code 0
Press ENTER to exit console.



```
4 int main() {  
5     int i = 12345;  
6     int t = 0;  
7     int d = 0;  
8  
9     if (i >= 1000) {  
10         t = i % 10000 / 1000;  
11     }  
12     d = i % 10;  
13     i -= t*1000;  
14     i -= d;  
15     i += d*1000;  
16     i += t;  
17     printf("i = %d\n", i);  
18     return 0;  
19 }  
20
```



- 把一個大於零的數的千位數字和個位數字調換
 - `int i = 12345;`
 - 12345 → 15342
 - 123 → 3120
 - 12 → 2010
 - 1 → 1000

Topics

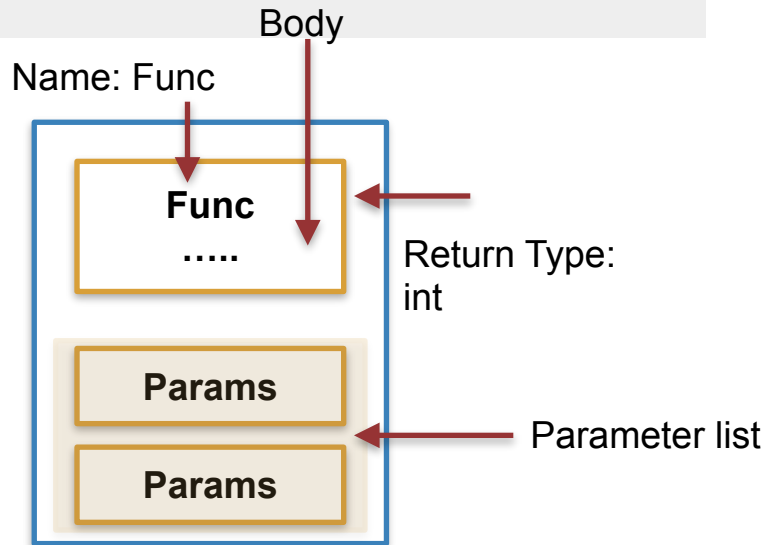
- Problem Solving with Programming Language
- C Programming
 - C Basics
 - Variables
 - Flow of Control
 - Function Basics ←
 - Programming with Arrays
 - Strings
 - Structures
 - Streams and File I/O
- Google Cloud Platform/Cloud Shell Editor (gcc/g++/Makefile)
- Google Cloud Platform/Cloud Source Repositories (git)



Function Basics

```
return_type function_name( parameter list ) {  
    body of the function  
}
```

- Return Type
- Function Name
- Parameters
- Function Body



Process's Memory Space

Return Type

- 可以想像成：function本身就是一個變數

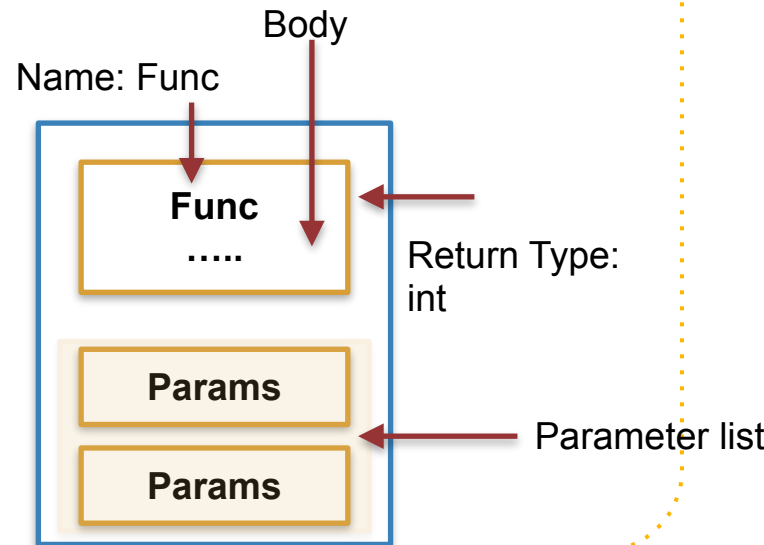
- int x = func_int();
- func_int() return **int**

- **Variables**

- Void

- **Function**

- return func_b();



Process's Memory Space

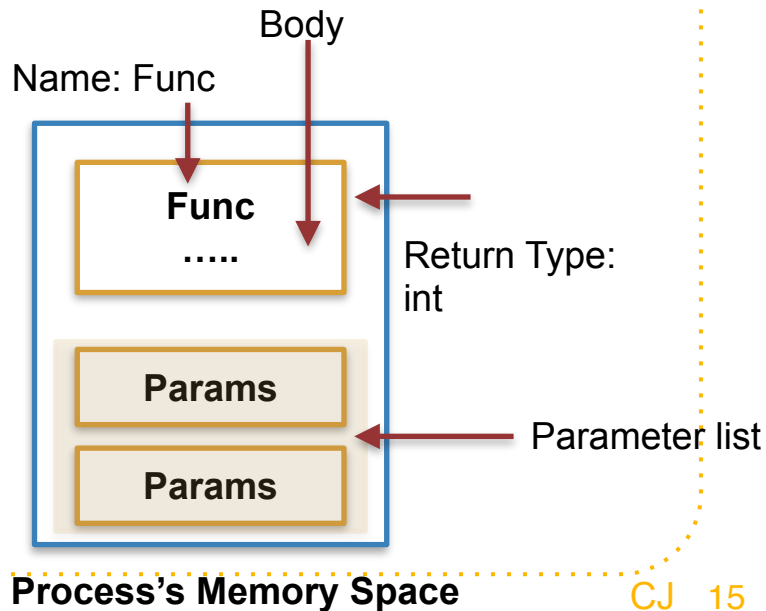


Function Name

- Coding style!
- 請使用以下方式命名
 - 動作_受詞()
 - `get_user_id()`, `set_student_name()`
 - 動作_名詞()
 - `動作()`, `abs()`, `printf()`, `save()`
- 不要叫什麼 `A()`, `B()`, `C()`

Parameters

- int max(int num1, int num2) { }
- 兩個變數！
- 不要相信傳進來的是正確的
- 要假設他是錯的
- 要小心的再檢查一次！



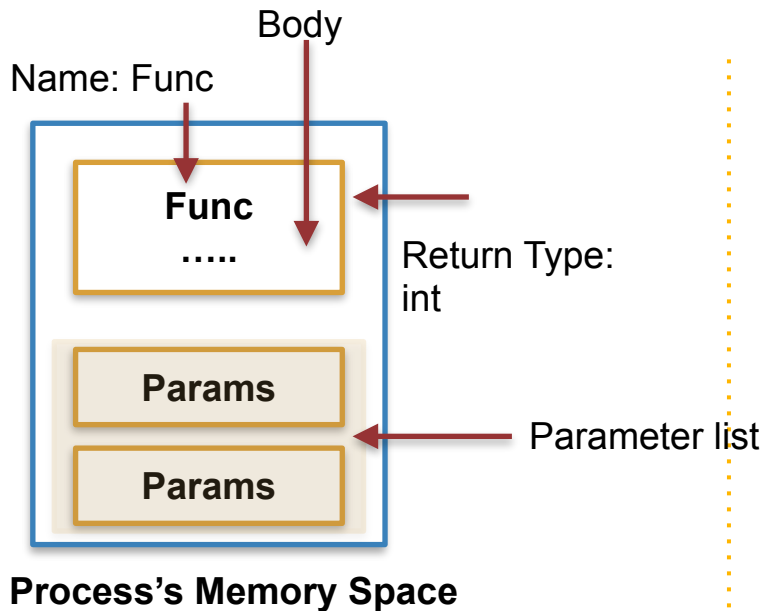
Parameters Call Type

- Call by value

- C語言都是 call by value
- 沒有其他方式

- Call by reference

- 透過傳遞記憶體位址的方式
- 可以直接對變數操作
 - Java
 - C語言也可以透過 pointer 的方式做到 (之後介紹)



Function Body

- Coding style!
- 不要超過50行！不要超過50行！不要超過50行！
- 不要超過50行！不要超過50行！不要超過50行！
- 不要超過50行！不要超過50行！不要超過50行！
- 不要超過50行！不要超過50行！不要超過50行！
- 不要超過50行！不要超過50行！不要超過50行！
- 不要超過50行！不要超過50行！不要超過50行！



An example

- Return the max number
- 小功能，小步驟

```
1 #include <stdio.h>
2
3 int max(int i, int j) {
4     return (i>j)? i :j;
5 }
6
7 int main() {
8
9     int n ;
10    n = max(10, 11);
11    printf("max = %d\n",n);
12
13    return 0;
14 }
```

max = 11

...Program finished with exit code 0
Press ENTER to exit console.



```
1 #include <stdio.h>
2
3 int absolute(int n) {
4     return (n<0)?-n:n;
5 }
6 int main() {
7
8     int n ;
9     n = absolute(-100);
10    printf("abs = %d\n",n);
11
12    return 0;
13 }
```

轉換絕對值

abs = 100

...Program finished with exit code 0
Press ENTER to exit console.



```
1 #include <stdio.h>
2
3 int absolute(int n) {
4     return (n<0)?-n:n;
5 }
6
7 int abs_max(int i, int j) {
8     int n = absolute(i);
9     int m = absolute(j);
10    return (n>m) ? n:m;
11 }
12
13 int main() {
14     int n ;
15     n = abs_max(-100, 8);
16     printf("abs_max = %d\n",n);
17
18     return 0;
19 }
```

不要去改變傳進來參數的值

結合兩個func

abs_max = 100

...Program finished with exit code 0
Press ENTER to exit console.



計算圓面積

```
1 #include <stdio.h>
2
3 float circleArea(float r) {
4     return 3.14 * r * r;
5 }
6
7 int main() {
8     float radius=5.0f, area;
9
10    area = circleArea(radius);
11    printf("Area of circle = %.2f\n",area);
12
13    return 0;
14 }
```

input

Area of circle = 78.50

...Program finished with exit code 0
Press ENTER to exit console.



A Function for checking odd

```
1 #include <stdio.h>
2 int check_odd(int n) {
3     return n&0x1;
4 }
5
6 int main() {
7     printf("%d\n", check_odd(4));
8     return 0;
9 }
```



0

...Program finished with exit code 0
Press ENTER to exit console.



function 在程式設計的角色

- 你可以想像
- 進入一個 func
- 對CPU來說
- 就是開啟一個副本
- 裡面的東西都是新的，除了傳進來的參數，一切和之前的程式是沒有關係的
- 也因為這樣，我們可以讓一個函式做專心做一件事！

Print Pascal's Triangle

```
    1
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 1
```

```
...Program finished with exit code 0
Press ENTER to exit console. 
```




Print Pascal's Triangle

拆解步驟！ 用幾個小步驟取代大步驟

```
1  #include <stdio.h>
2  int main() {
3
4      // main loop for rows i=0 to 4
5      // print spaces |
6      // print numbers by ith row
7
8      return 0;
9  }
```

兩步驟

寫下你分好的小步驟

- 兩個小步驟
 - Print Spaces
 - Print numbers

```
1 #include <stdio.h>
2
3 void print_space(int rows, int r) {
4
5 }
6
7 void print_pascal_number(int rows, int r) {
8
9 }
10
11 int main() {
12     int rows = 5;
13
14     for (int r=0;r<rows; r++) {
15         print_space(rows, r);
16         print_pascal_number(rows, r);
17         printf("\n");
18     }
19
20     return 0;
21 }
```

Print Spaces

```
1 #include <stdio.h>
2
3 void print_space(int rows, int r) {
4     for (int space = 1; space <= rows - r; space++)
5         printf(" ");
6     }
7 }
8
```

```
11111
1111
111
11
1
```

...Program finished with exit code 0

實作第一個步驟

Print numbers

```
void print_pascal_number(int r) {  
    int coef;  
    for (int j = 0; j <= r; j++) {  
        if (j == 0 || r == 0) {  
            coef = 1;  
        } else {  
            coef = coef * (r-j + 1) / j;  
            // 1 6 15 20 15 6 1  
            // 1 (coef)  
            // 6 = 1 * (6/1)  
            // 15 = 6 * (5/2)  
            // 20 = 15 * (4/3)  
            // 15 = 20 * (3/4)  
            // 6 = 15 * (2/5)  
            // 1 = 6 * (1/6)  
        }  
        printf("%d ", coef);  
    }  
}
```



實作第二個步驟



每一個函式對應到一個步驟

每一個函式不要太多行

少行對於人說還可以看
行數越多，人越不可能看！

```
3- void print_space(int rows, int r) {
4-     for (int space = 1; space <= rows - r; space++) {
5-         printf(" ");
6-     }
7- }
8-
9- void print_pascal_number(int r) {
10-     int coef = 1;
11-     for (int j = 0; j <= r; j++) {
12-         if (j == 0) {
13-             coef = 1;
14-         } else {
15-             coef = coef * (r-j + 1) / j;
16-         }
17-         printf("%4d ", coef);
18-     }
19- }
20-
21- int main() {
22-     int rows = 7;
23-
24-     for (int r=0;r<rows; r++) {
25-         print_space(rows, r);
26-         print_pascal_number(r);
27-         printf("\n");
28-     }
29- }
```

```

      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
 1 5 10 10 5 1
1 6 15 20 15 6 1
```

15 Pascal's Triangle

```

      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
 1 5 10 10 5 1
1 6 15 20 15 6 1
1 7 21 35 35 21 7 1
1 8 28 56 70 56 28 8 1
1 9 36 84 126 126 84 36 9 1
1 10 45 120 210 252 210 120 45 10 1
1 11 55 165 330 462 462 330 165 55 11 1
1 12 66 220 495 792 924 792 495 220 66 12 1
1 13 78 286 715 1287 1716 1716 1287 715 286 78 13 1
1 14 91 364 1001 2002 3003 3432 3003 2002 1001 364 91 14 1

```

...Program finished with exit code 0
Press ENTER to exit console.



有了 function 才有了程式設計

- Function —> 一切重頭開始了
 - Input —> process (function) —> output
 - 規劃程式結構的能力
- 在高階程式語言裡
 - 沒有學會 function
 - 等於是在寫程式，不是程式設計
 - 沒有學會 function
 - 等於沒有規劃程式結構的能力
 - 無法更進一步寫出code size更大的程式

**function是
程式語言的
賢者之石**

印出數字金字塔：
考慮函式後再寫一次

```
    1
   2 2
  3 3 3
 4 4 4 4

...Program finished with exit code 0
Press ENTER to exit console.█
```



```

1 #include <stdio.h>
2
3 void print_spec(int i, int n) {
4     for (int spec=0; spec<n-i; spec++) {
5         printf(" ");
6     }
7 }
8
9 void print_num(int n) {
10    for (int i=0; i < n; i++) {
11        printf("%d ", n);
12    }
13    printf("\n");
14 }
15
16 int main() {
17     int n = 7;
18     for (int i=1; i<=n; i++) {
19         print_spec(i, n);
20         print_num(i);
21     }
22
23     return 0;
24 }

```

6行

14行

對比
一下

```

1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
6 6 6 6 6 6
7 7 7 7 7 7 7

```

...Program finished with exit code 0
Press ENTER to exit console.

```

1 #include <stdio.h>
2
3 int main() {
4     int n = 4;
5     int spc = n;
6
7     for (int i=1; i<=n; i++) {
8         for (int k=spc; k>=1; k--) {
9             printf(" ");
10        }
11        for (int j=1; j<=i; j++) {
12            printf("%d ", i);
13        }
14        printf("\n");
15        spc--;
16    }
17 }

```

```

1
2 2
3 3 3
4 4 4 4

```

...Program finished with exit code 0
Press ENTER to exit console.

最後兩位加法

- Input 159
- Output: 14
- func1: 找出最後1位
- func2: 找出最後2位
- 把他們加起來
- 可以把func再加一個參數
 - 要回傳第幾位

```
1 #include <stdio.h>
2 int lastDigit(int n) { ←
3     return (n%10);
4 }
5
6 int secondLastDigit(int n) { ←
7     return ((n/10)%10);
8 }
9
10 int addTwoDigits(int i) {
11     return lastDigit(i) + secondLastDigit(i);
12 }
13
14 int main() {
15     int n=159;
16     int sum = addTwoDigits(n);
17     printf("sum = %d\n", sum);
18
19     return 0;
20 }
```

sum = 14

...Program finished with exit code 0
Press ENTER to exit console.



```
1 #include <stdio.h>
2
3 int main() {
4     printf("Main");
5     int fun() { ←
6         printf("fun");
7
8         int view() {
9             printf("view");
10        }
11        return 1;
12    }
13    view();
14 }
```

C 語言

沒有 nested
function

不要寫出這種東西

Conclusion

- C Basics
 - Function Basics

```
return_type function_name( parameter list ) {  
    body of the function  
}
```

- function是程式語言的賢者之石



明天有QZ#2 & HW#8



Thanks!

Open for any questions

CJ Wu

cjwu@mail.cgu.edu.tw