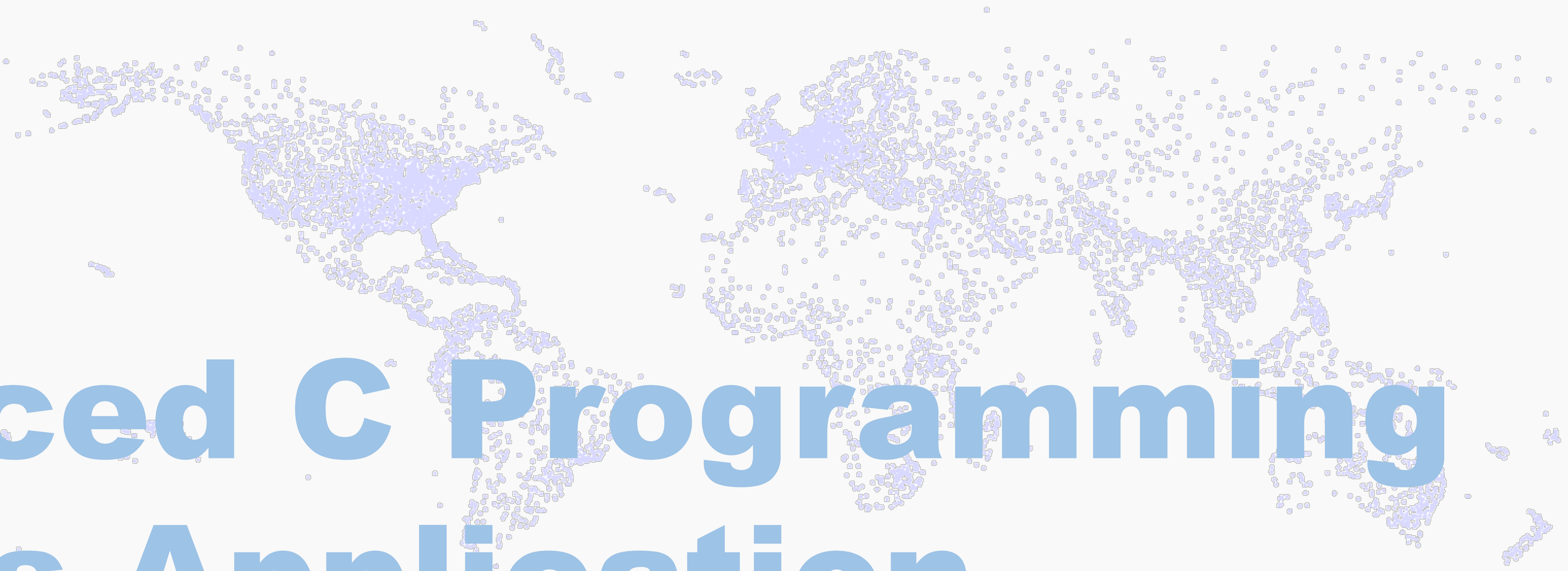


Advanced C Programming And It's Application

Pointer II: Pointer - Array & Function

Assistant Prof. Chan, Chun-Hsiang

Department of Artificial Intelligence, Tamkang University

Nov. 3, 2021

大綱

[1] Recall

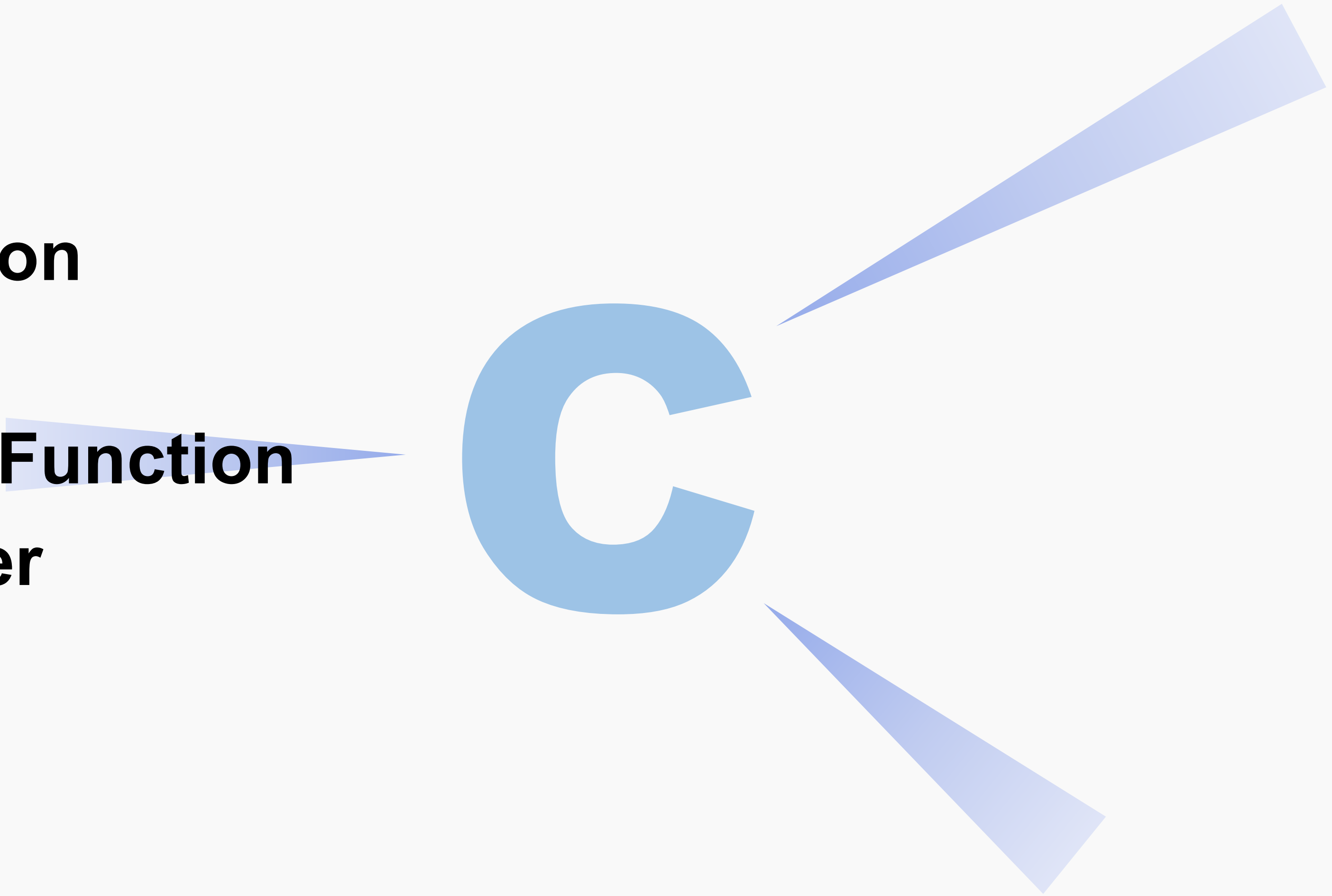
[2] Pointer & Function

[3] Array & Pointer

[4] Array Pointer in Function

[5] Pointer to Pointer

[6] Assignments



<Recall/>

複習Pointer I.

「*」在C/C++裡面有三個用處:
(1) 乘法 (Multiplication operator)
(2) 指標宣告 (Definition of a pointer)
(3) 取指標數值 (Dereferencing operator)

網站

宣告一個空指標 (null pointer):

```
/*Ex 6-1: Null Pointer */
printf("Ex 6-1: Null Pointer\n");
int *p = 0;
int *q = NULL;
```

Lab 6-1:
寫一個程式來呈現「*」的三種用法。

</Recall>

Pointer & Function

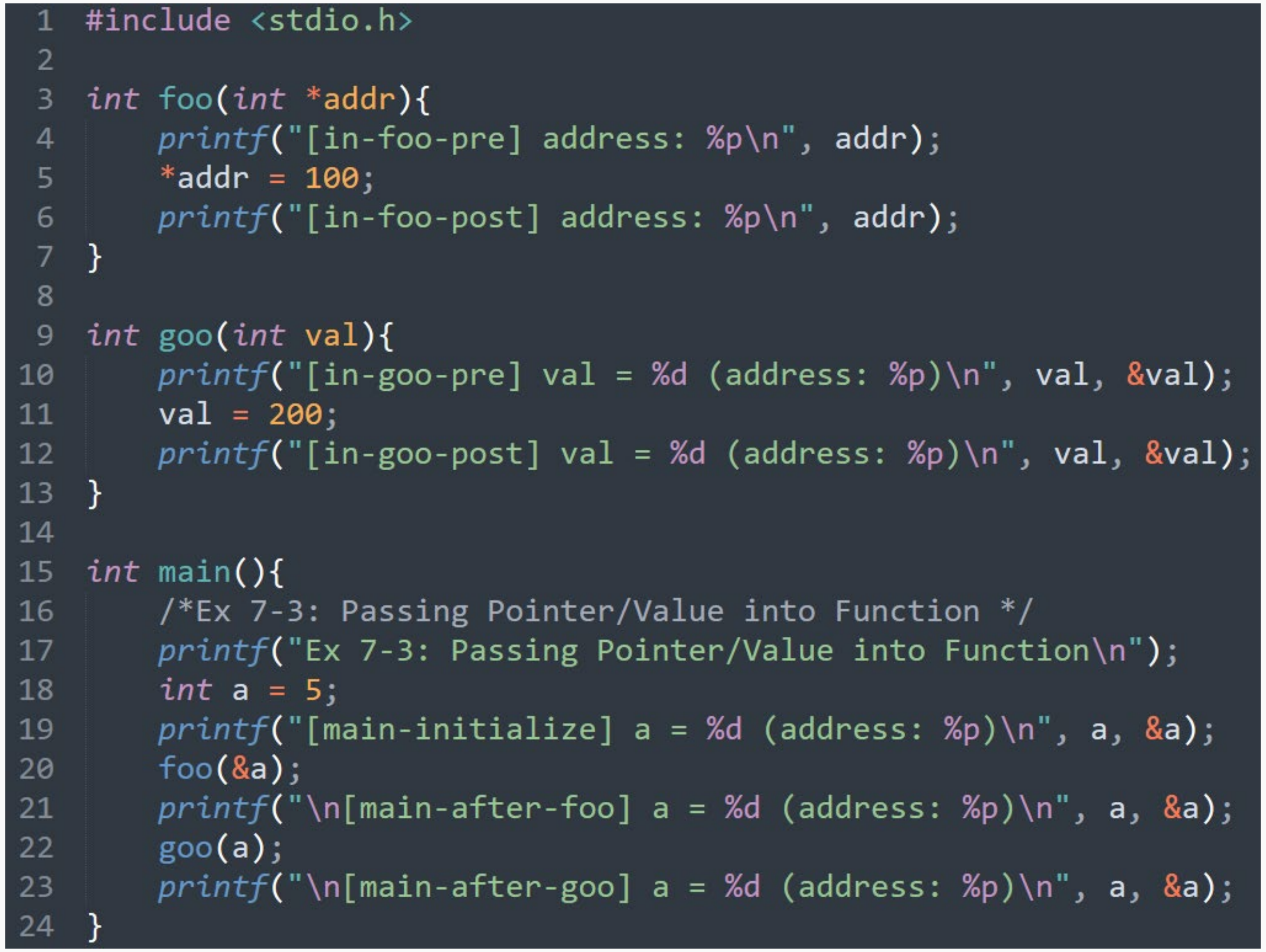
當我們熟悉指標的基本操作後，再來就是我們要教將Pointer塞到function裡面去。

```
#include <stdio.h>
int foo(int val, int *addr){
        printf("val = %d (address: %p)\n", a, addr);
}

int main(){
        /*Ex 6-2: Passing Pointer into Function */
        printf("Ex 6-2: Passing Pointer into Function\n");
        int a = 5;
        foo(a, &a);
        printf("a = %d (address: %p)\n", a, &a);
}
```

Pointer & Function

經過剛剛Ex 6-2之後，還記得我們在function那個單元有提到passing value into function，那這兩者哪裡不同?

宣告兩個function:
(1) int foo(指標)
(2) int goo(變數)
(3) main分別呼叫前面兩個函數並觀察結果。



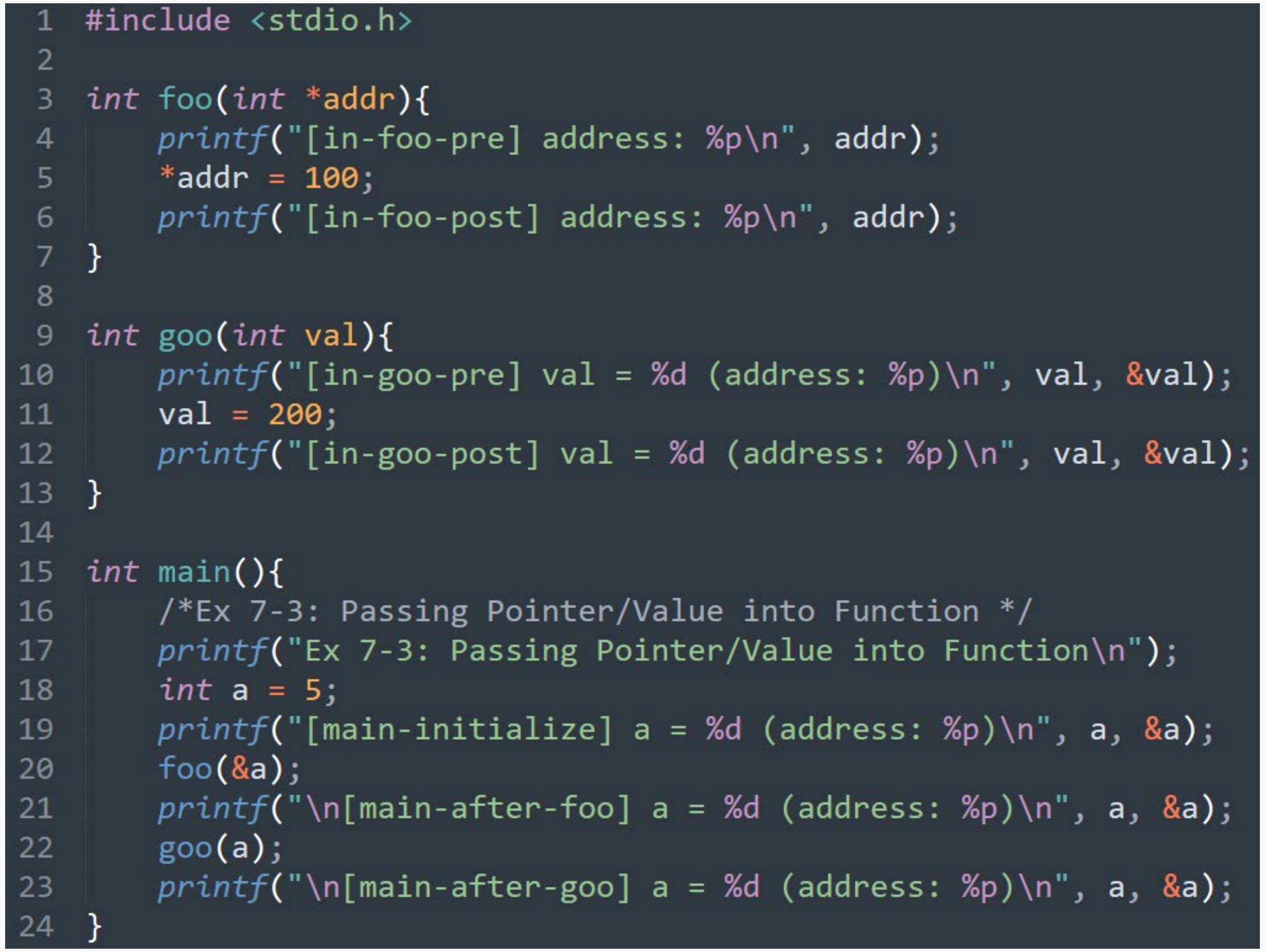
```c
#include <stdio.h>

int foo(int *addr){
    printf("[in-foo-pre] address: %p\n", addr);
    *addr = 100;
    printf("[in-foo-post] address: %p\n", addr);
}

int goo(int val){
    printf("[in-goo-pre] val = %d (address: %p)\n", val, &val);
    val = 200;
    printf("[in-goo-post] val = %d (address: %p)\n", val, &val);
}

int main(){
    /*Ex 7-3: Passing Pointer/Value into Function */
    printf("Ex 7-3: Passing Pointer/Value into Function\n");
    int a = 5;
    printf("[main-initialize] a = %d (address: %p)\n", a, &a);
    foo(&a);
    printf("\n[main-after-foo] a = %d (address: %p)\n", a, &a);
    goo(a);
    printf("\n[main-after-goo] a = %d (address: %p)\n", a, &a);
}
```

Pointer & Function

Lab 6-2:

(1) 先不要執行程式，請問這裡面每一個printf的數值是多少?

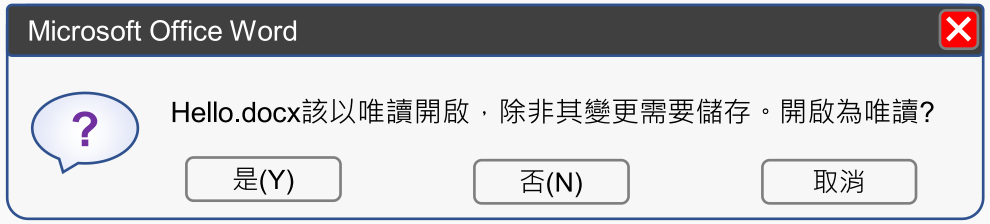
(2) 思考一下! 究竟函數中放指標與放變數差別在哪裡?



```c
#include <stdio.h>

int foo(int *addr){
    printf("[in-foo-pre] address: %p\n", addr);
    *addr = 100;
    printf("[in-foo-post] address: %p\n", addr);
}

int goo(int val){
    printf("[in-goo-pre] val = %d (address: %p)\n", val, &val);
    val = 200;
    printf("[in-goo-post] val = %d (address: %p)\n", val, &val);
}

int main(){
    /*Ex 7-3: Passing Pointer/Value into Function */
    printf("Ex 7-3: Passing Pointer/Value into Function\n");
    int a = 5;
    printf("[main-initialize] a = %d (address: %p)\n", a, &a);
    foo(&a);
    printf("\n[main-after-foo] a = %d (address: %p)\n", a, &a);
    goo(a);
    printf("\n[main-after-goo] a = %d (address: %p)\n", a, &a);
}
```

Pointer & Function

你們記不記得如果你已經開啟一個word檔，又在開啟一次，這時候是不是有一個熟悉的視窗呢?


Microsoft Office Word

Hello.docx該以唯讀開啟，除非其變更需要儲存。開啟為唯讀?

Lab 6-3:
想想看，有哪些情況你覺得「唯讀」是比「可讀可改」更適用的呢?

唯讀是甚麼? 就是Read Only (R)，只能看不能改!

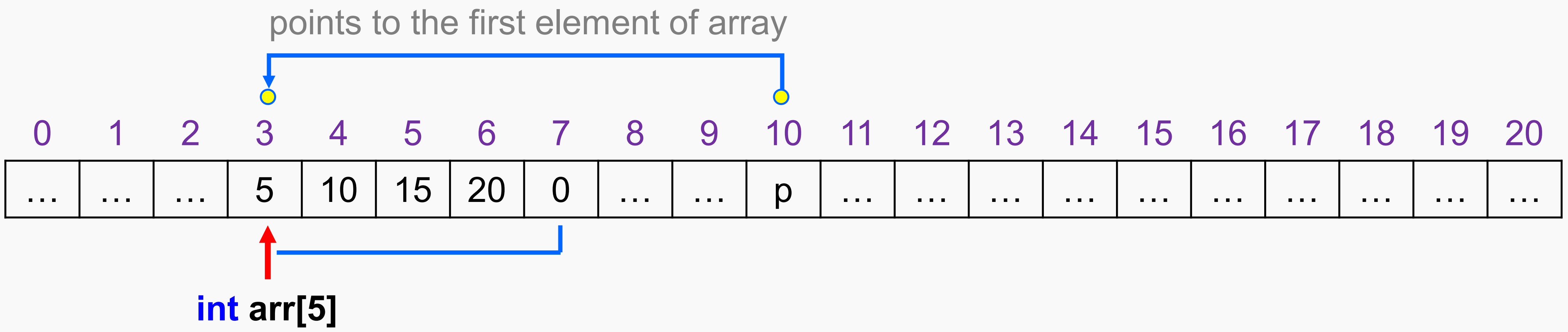
相對的就是 Read & Write (W)，可讀可改!

《愛蓮說》　　北宋。周敦頤

水陸草木之花，可愛者甚蕃；晉陶淵明獨愛菊，自李唐來，世人盛愛牡丹。予獨愛蓮之出淤泥而不染，濯清漣而不妖；中通外直，不蔓不枝；香遠益清，亭亭淨植，<u>可遠觀而不可褻玩焉</u>。予謂：菊，花之隱逸者也；牡丹，花之富貴者也；蓮，花之君子者也。噫！菊之愛，陶後鮮有聞。蓮之愛，同予者何人？牡丹之愛，宜乎眾矣！

Array & Pointer

矩陣其實就是一整串的連在一起數列，我們用下面這個例子來看一下。首先宣告一個整數陣列大小為五的int arr[5]，裡面分別儲存{5,10,15,20,0}。如果我們今天用一個指標指向矩陣第一個element (i.e., arr[0])，利用上週教過的「地址+1」取下一個element數值。



Array & Pointer

```
/*Ex 6-3: Pointer to Array Element */
printf("Ex 6-3: Pointer to Array Element\n");
int arr[5] = {5,10,15,20,0};
int *p = &arr[0];

for (int i=0; i<5; i++){
        printf("%d\t", *(p+i));
}
```

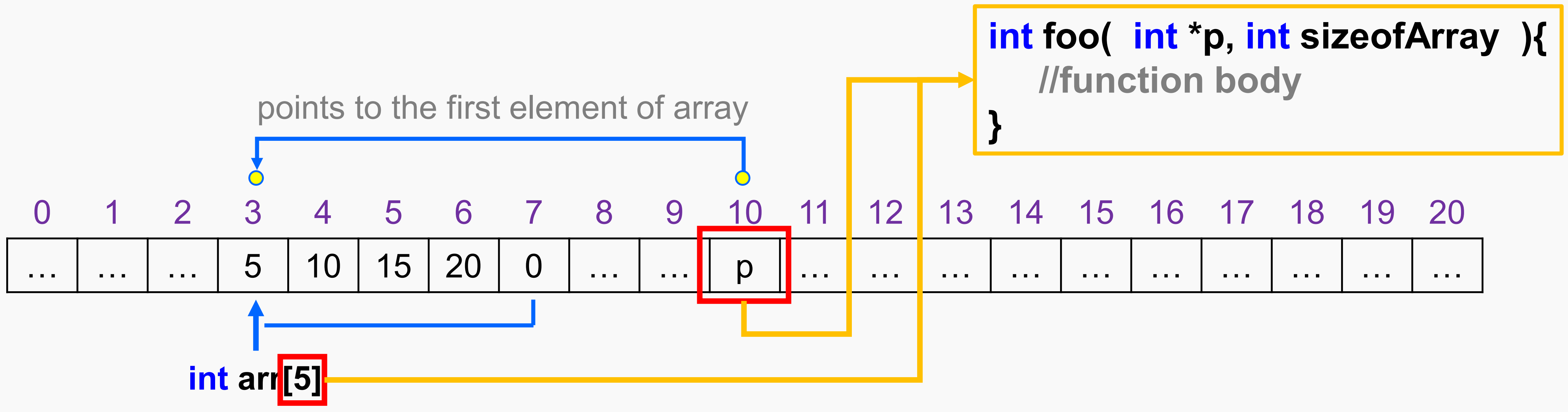
Lab 6-4:

宣告一個整數矩陣int arr[10] = {0,1,2,3,4,5,6,7,8,9}與一個指標指向arr[0]，假設搜尋到6會自動停止，並把掃描過程印出來，如下:

printf(矩陣索引值、矩陣中該元素數值、矩陣中該元素記憶體位置)

Array Pointer in Function

前面我們講過如何利用pointer掃描矩陣中每一個元素，這個應用很重要。因為這樣我們就不用將整個矩陣搬進函數裡面，我只要放指向第一個元素的指標與矩陣的大小，函數自然就可以知道整個矩陣的數值，也就可以接下去做運算。



```
int foo(  int *p, int sizeofArray  ){
     //function body
}
```

Array Pointer in Function

```
#include <stdio.h>
int foo(int *p, int sizeofArray){
	for (int i=0; i<sizeofArray; i++){
		printf("%d\t", p[i]);
	}
}

int main(){
	/*Ex 6-4: Array to Function */
	printf("Ex 6-4: Array to Function\n");
	int arr[10] = {0,1,2,3,4,5,6,7,8,9};
	foo(arr, 10);
}
```

Lab 6-5:

修改Ex 6-4 foo函數中的*p以下兩種表示方式，觀察程式執行結果並討論可能的原因。

(1) int array[];

(2) int array[10];

Array Pointer in Function

```
#include <stdio.h>
int foo(int *p, int sizeofArray){
	for (int i=0; i<sizeofArray; i++){
		printf("%d\t", p[i]);
	}
}

int main(){
	/*Ex 6-4: Array to Function */
	printf("Ex 6-4: Array to Function\n");
	int arr[10] = {0,1,2,3,4,5,6,7,8,9};
	foo(arr, 10);
}
```

Lab 6-6:

修改Ex 6-4的arr修改成以下三個，看程式執行的結果並找尋其原因。

(1) int arr1[];

(2) int *arr2 = 0;

(3) int arr3[100] = {0};

Array Pointer in Function

```
#include <stdio.h>
int foo1(int *p, int sizeofArray){
	for (int i=0; i<sizeofArray; i++){
		printf("%d\t", p[i]);
	}
}
int foo2(int *p, int sizeofArray){
	for (int i=0; i<sizeofArray; i++){
		printf("%d\t", *(p+i));
	}
}
int foo3(int p[], int sizeofArray){
	for (int i=0; i<sizeofArray; i++){
		printf("%d\t", p[i]);
	}
}
```

Is it an array or a pointer?

```
int main(){
	/*Ex 6-5: Array to Function */
	printf("Ex 6-5: Array to Function\n");
	int arr[10] = {0,1,2,3,4,5,6,7,8,9};

	foo1(arr, 10);
	printf("\n");
	foo2(arr, 10);
	printf("\n");
	foo3(arr, 10);
	printf("\n");
}
```

Lab 6-7:

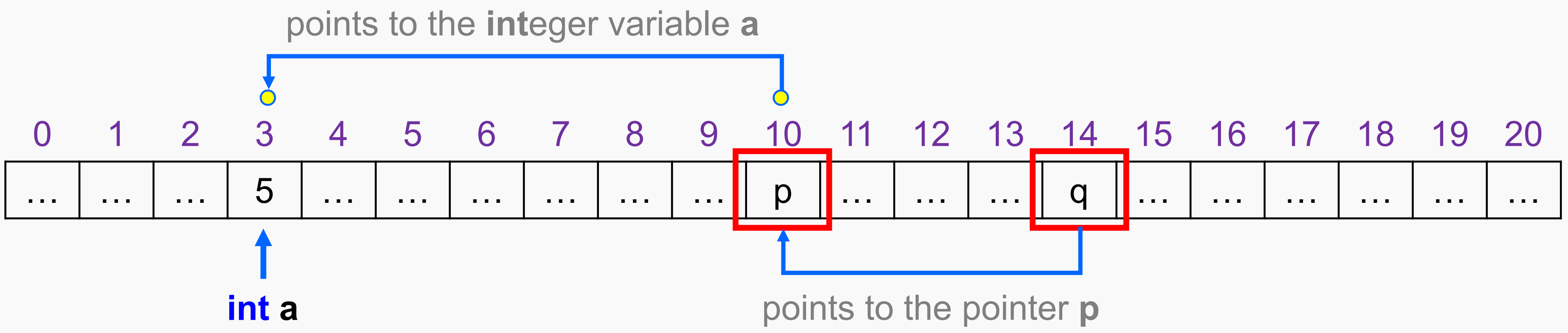
思考一下，這三個foo是用什麼樣的機制把結果印出來的?

Pointer to Pointer

那麼這邊我們要來介紹pointer的pointer，就是用來儲存pointer地址的pointer，也就是指向pointer的pointer!

宣告方式: **ptr = NULL;

取值方式: **ptr



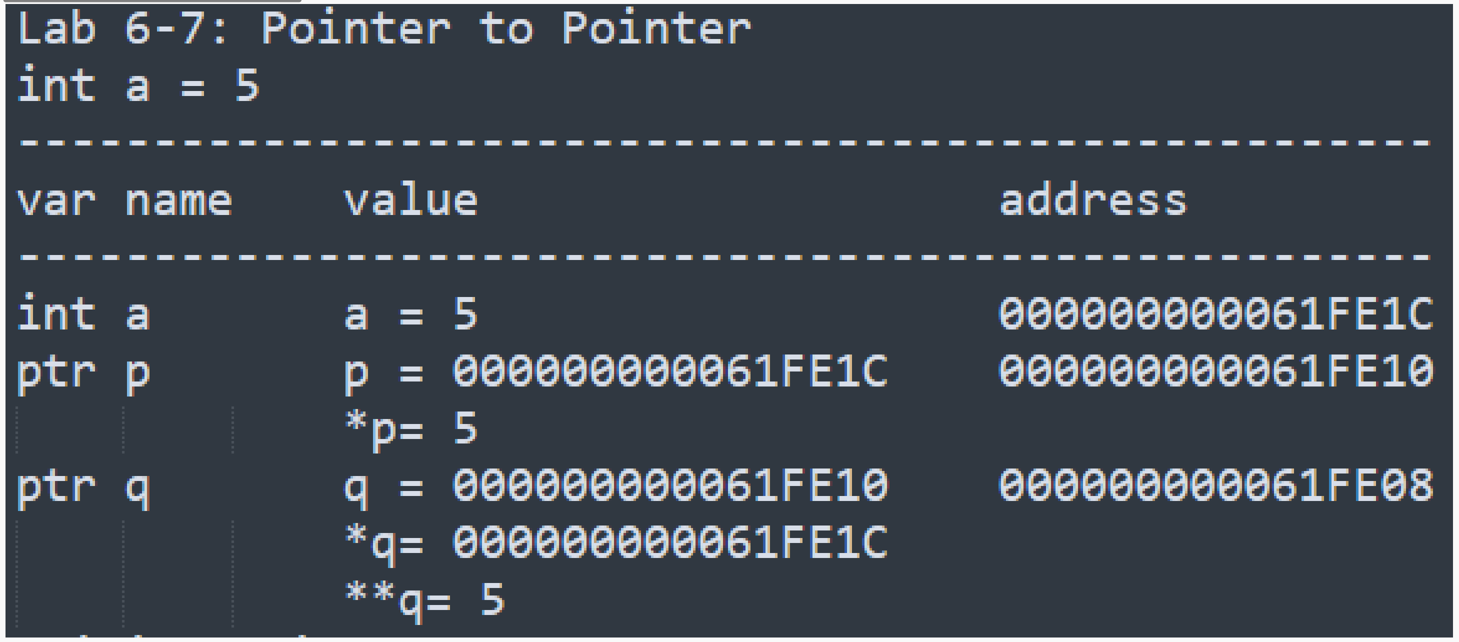
Pointer to Pointer

```
/*Ex 6-6: Pointer to Pointer */
printf("Ex 6-6: Pointer to Pointer\n");
int a = 5;
int *p = &a;
int **q= &p;
```

Lab 6-7:

練習將Ex 6-6的變數值以及地址印出來，如右圖所示。

Results:



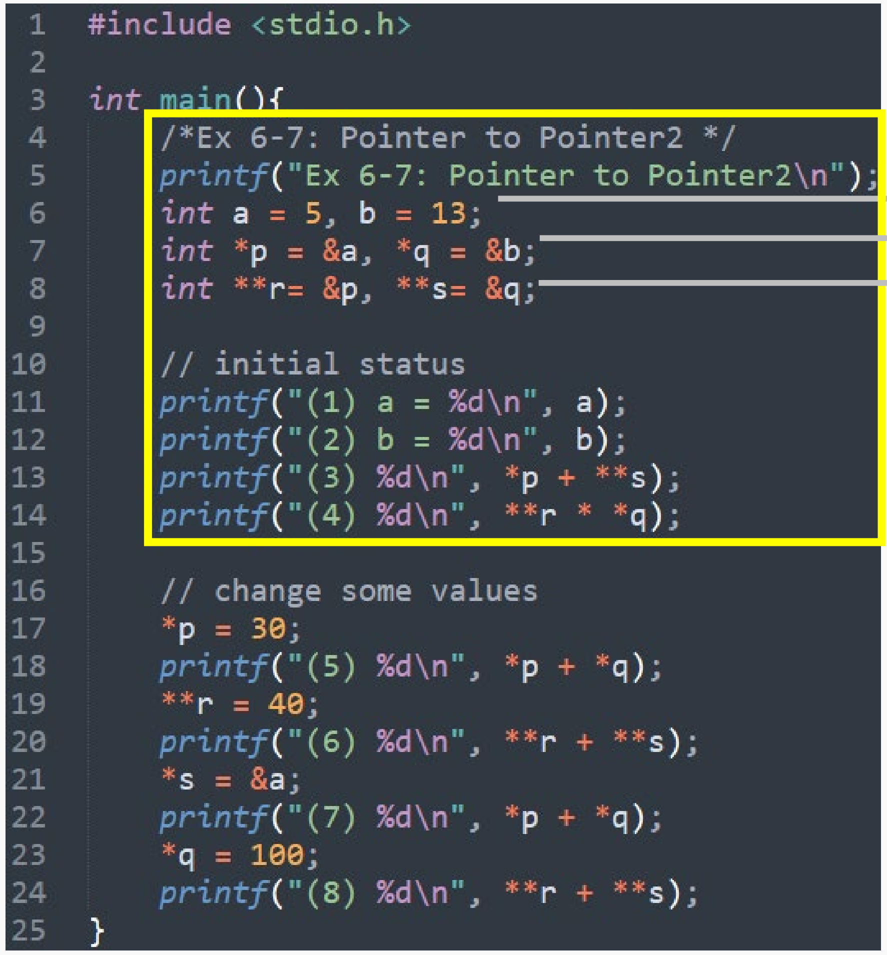
```
Lab 6-7: Pointer to Pointer
int a = 5
----------------------------------------------------
var name    value                   address
----------------------------------------------------
int a       a = 5                   000000000061FE1C
ptr p       p = 000000000061FE1C    000000000061FE10
            *p= 5
ptr q       q = 000000000061FE10    000000000061FE08
            *q= 000000000061FE1C
            **q= 5
```

Pointer to Pointer

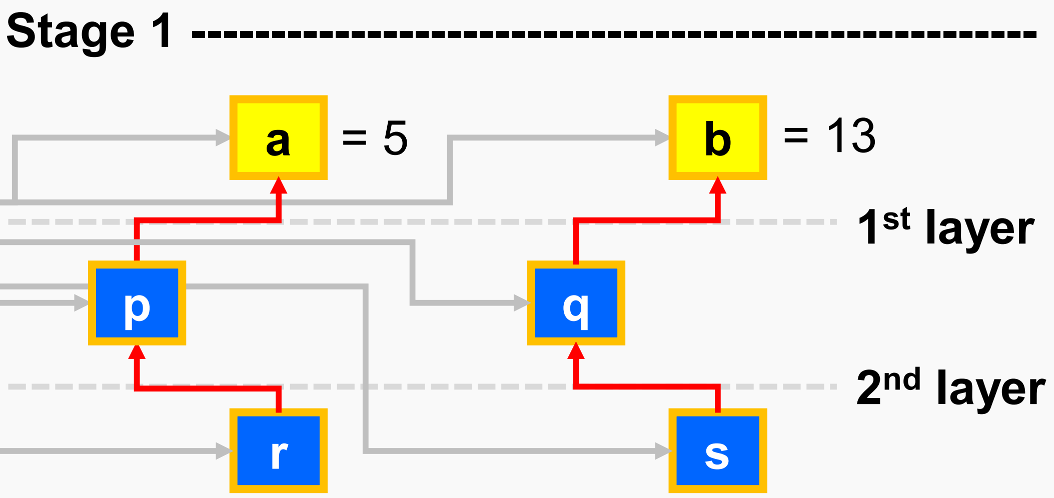
```
/*Ex 6-7: Pointer to Pointer2 */
printf("Ex 6-7: Pointer to Pointer2\n");
int a = 5, b = 13;
int *p = &a, *q = &b;
int **r= &p, **s= &q;

// initial status
printf("(1) a = %d\n", a);
printf("(2) b = %d\n", b);
printf("(3) %d\n", *p + **s);
printf("(4) %d\n", **r * *q);
```

```
// change some values
*p = 30;
printf("(5) %d\n", *p + *q);
**r = 40;
printf("(6) %d\n", **r + **s);
*s = &a;
printf("(7) %d\n", *p + *q);
*q = 100;
printf("(8) %d\n", **r + **s);
```

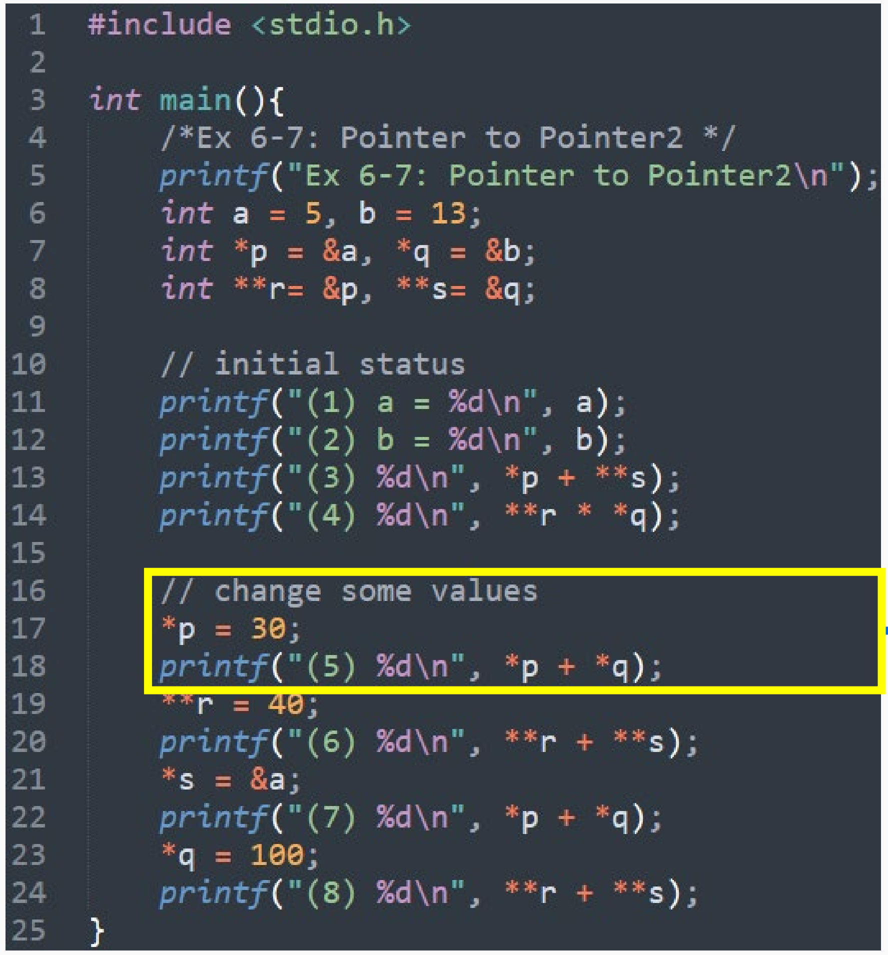
Pointer to Pointer



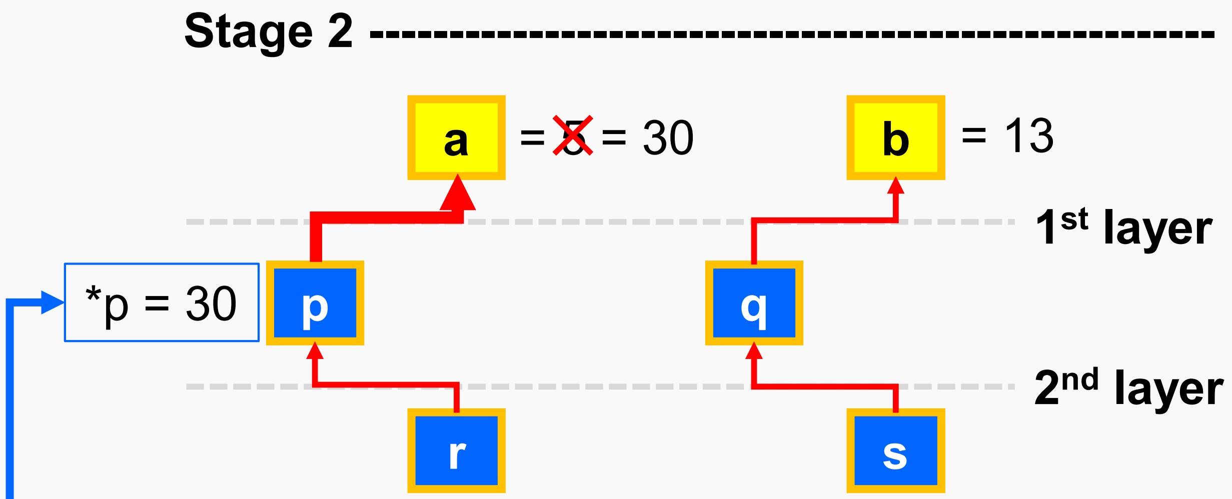
```c
#include <stdio.h>

int main(){
    /*Ex 6-7: Pointer to Pointer2 */
    printf("Ex 6-7: Pointer to Pointer2\n");
    int a = 5, b = 13;
    int *p = &a, *q = &b;
    int **r= &p, **s= &q;

    // initial status
    printf("(1) a = %d\n", a);
    printf("(2) b = %d\n", b);
    printf("(3) %d\n", *p + **s);
    printf("(4) %d\n", **r * *q);

    // change some values
    *p = 30;
    printf("(5) %d\n", *p + *q);
    **r = 40;
    printf("(6) %d\n", **r + **s);
    *s = &a;
    printf("(7) %d\n", *p + *q);
    *q = 100;
    printf("(8) %d\n", **r + **s);
}
```



Pointer to Pointer

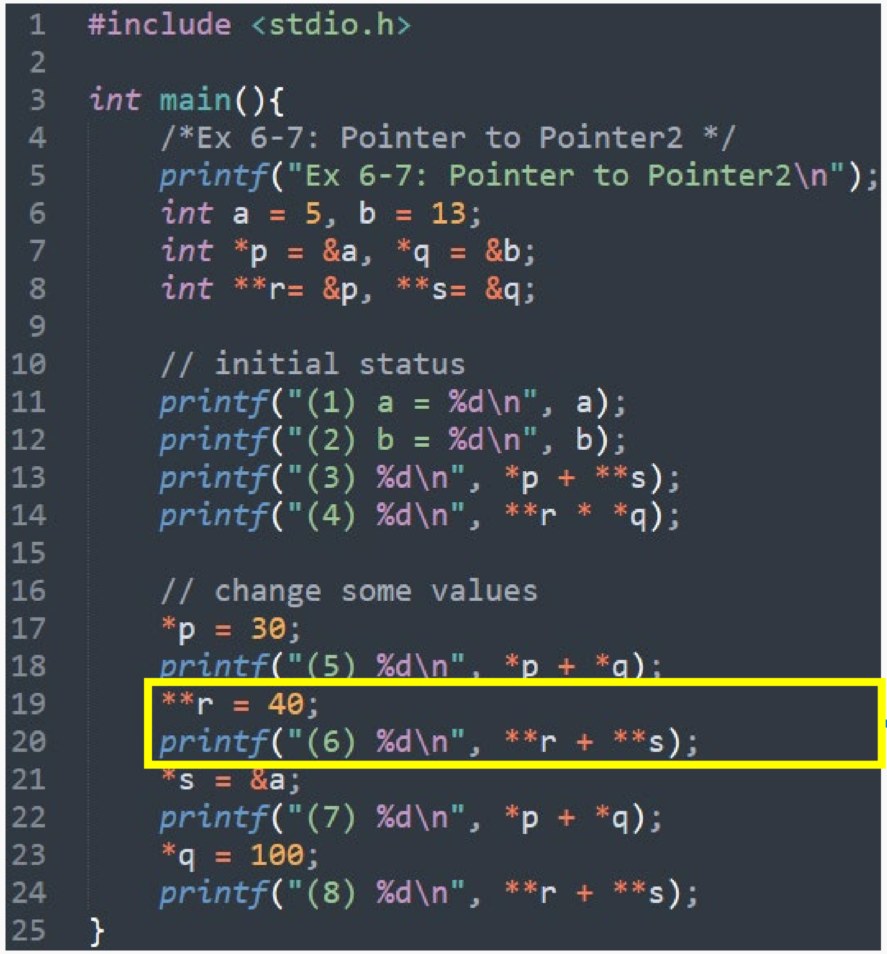


```c
#include <stdio.h>

int main(){
    /*Ex 6-7: Pointer to Pointer2 */
    printf("Ex 6-7: Pointer to Pointer2\n");
    int a = 5, b = 13;
    int *p = &a, *q = &b;
    int **r= &p, **s= &q;

    // initial status
    printf("(1) a = %d\n", a);
    printf("(2) b = %d\n", b);
    printf("(3) %d\n", *p + **s);
    printf("(4) %d\n", **r * *q);

    // change some values
    *p = 30;
    printf("(5) %d\n", *p + *q);
    **r = 40;
    printf("(6) %d\n", **r + **s);
    *s = &a;
    printf("(7) %d\n", *p + *q);
    *q = 100;
    printf("(8) %d\n", **r + **s);
}
```

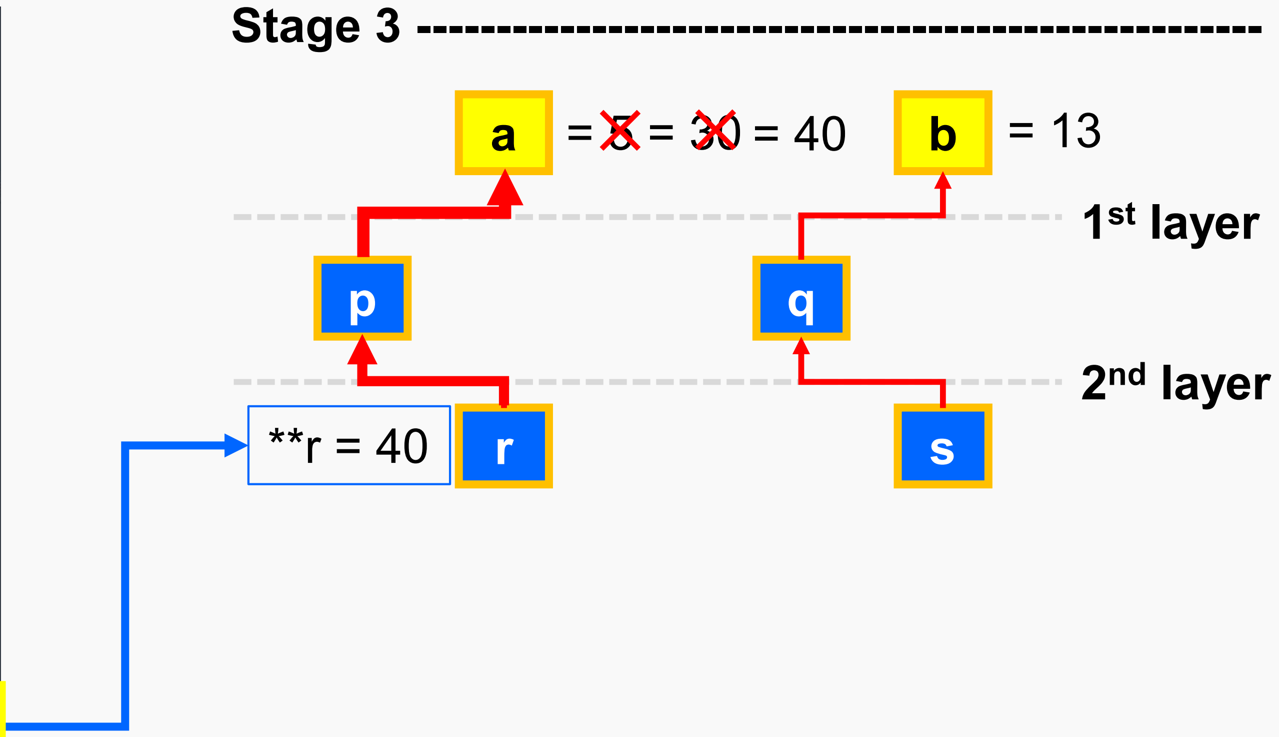


2021/11/03

Pointer to Pointer

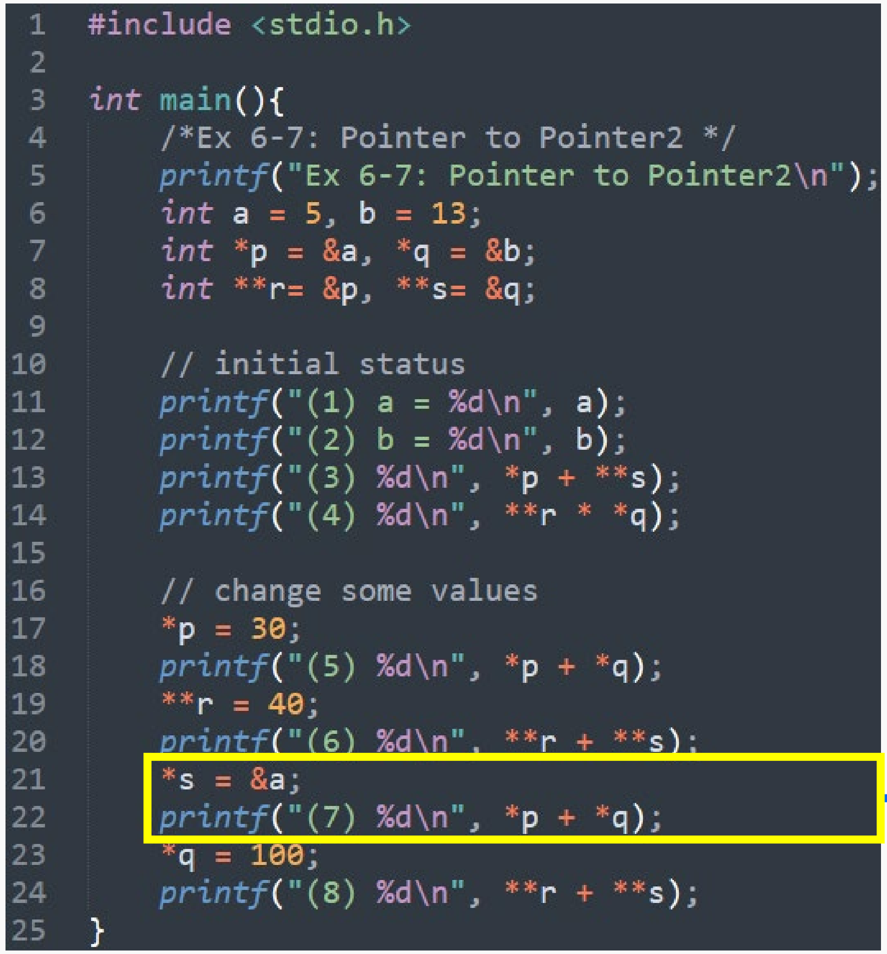


```
#include <stdio.h>

int main(){
    /*Ex 6-7: Pointer to Pointer2 */
    printf("Ex 6-7: Pointer to Pointer2\n");
    int a = 5, b = 13;
    int *p = &a, *q = &b;
    int **r= &p, **s= &q;

    // initial status
    printf("(1) a = %d\n", a);
    printf("(2) b = %d\n", b);
    printf("(3) %d\n", *p + **s);
    printf("(4) %d\n", **r * *q);

    // change some values
    *p = 30;
    printf("(5) %d\n", *p + *q);
    **r = 40;
    printf("(6) %d\n", **r + **s);
    *s = &a;
    printf("(7) %d\n", *p + *q);
    *q = 100;
    printf("(8) %d\n", **r + **s);
}
```

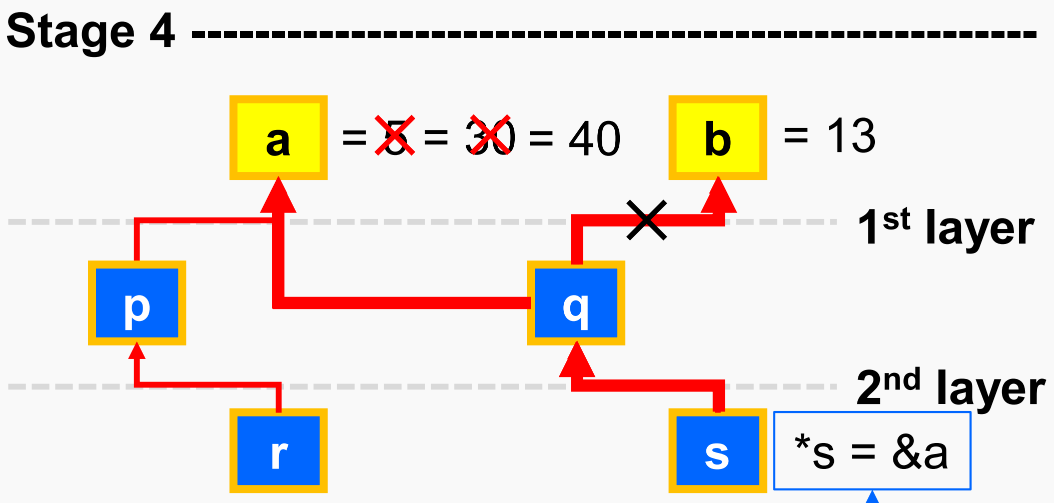


Pointer to Pointer

Lab 6-8:
如果 *s = &a 改成 **s = a，那(7)與(8)會是一樣的嗎?

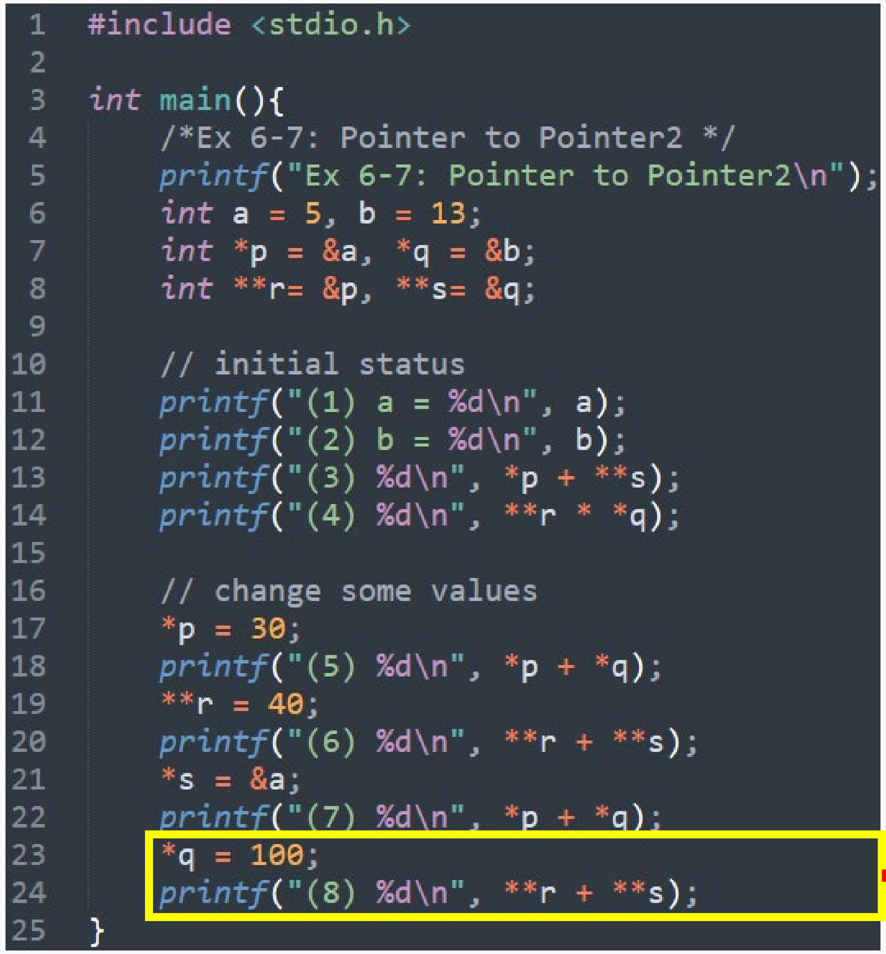


```c
#include <stdio.h>

int main(){
    /*Ex 6-7: Pointer to Pointer2 */
    printf("Ex 6-7: Pointer to Pointer2\n");
    int a = 5, b = 13;
    int *p = &a, *q = &b;
    int **r= &p, **s= &q;

    // initial status
    printf("(1) a = %d\n", a);
    printf("(2) b = %d\n", b);
    printf("(3) %d\n", *p + **s);
    printf("(4) %d\n", **r * *q);

    // change some values
    *p = 30;
    printf("(5) %d\n", *p + *q);
    **r = 40;
    printf("(6) %d\n", **r + **s);
    *s = &a;
    printf("(7) %d\n", *p + *q);
    *q = 100;
    printf("(8) %d\n", **r + **s);
}
```



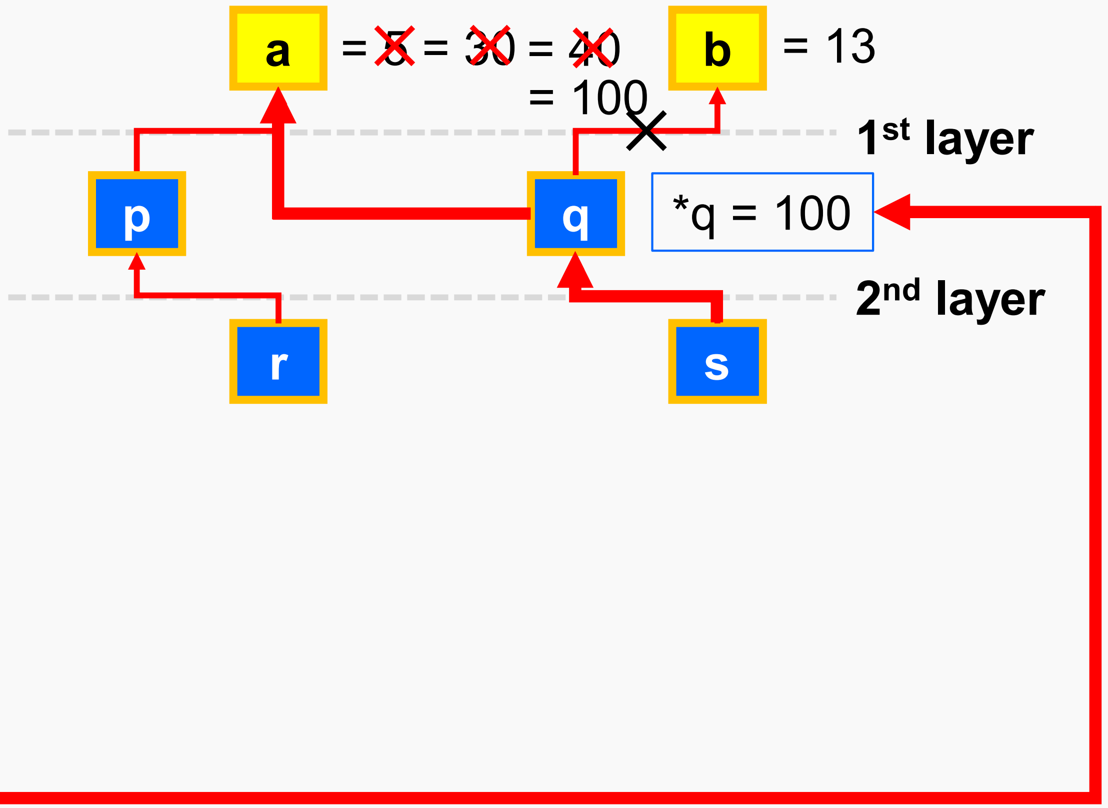
Pointer to Pointer

Lab 6-9:
將Lab 6-8的function拿來應用在這裡，分別印出stage 1-5的變數狀態。



```c
#include <stdio.h>

int main(){
    /*Ex 6-7: Pointer to Pointer2 */
    printf("Ex 6-7: Pointer to Pointer2\n");
    int a = 5, b = 13;
    int *p = &a, *q = &b;
    int **r= &p, **s= &q;

    // initial status
    printf("(1) a = %d\n", a);
    printf("(2) b = %d\n", b);
    printf("(3) %d\n", *p + **s);
    printf("(4) %d\n", **r * *q);

    // change some values
    *p = 30;
    printf("(5) %d\n", *p + *q);
    **r = 40;
    printf("(6) %d\n", **r + **s);
    *s = &a;
    printf("(7) %d\n", *p + *q);
    *q = 100;
    printf("(8) %d\n", **r + **s);
}
```

Stage 5 --



2021/11/03

作業一

大家有看過 Netflex 的《魷魚遊戲》嗎?
第三關 – 猜彈珠的遊戲

***** 規則 *****

1. 先拿出你的賭注(彈珠)，放在手掌心。
2. 說出對方的彈珠數是奇數或是偶數。
3. 猜錯的一方必須將彈珠交給猜對的一方。
4. 假設猜對的一方手上的彈珠數>猜錯的一方，則必須給足彈珠。
5. 沒有彈珠者，即Game over!

Photo from Wikipedia:
https://en.wikipedia.org/wiki/Squid_Game

作業一

第三關 – 猜彈珠的遊戲

*** 防呆機制 ***

1. 不能押比自己擁有更高的彈珠數
2. 不符合要求的情況，必須讓使用者一直輸入到正確為止。例如: 押的彈珠數、奇偶猜的方式。

Photo from Wikipedia:
https://en.wikipedia.org/wiki/Squid_Game

<Assignments/>

作業一

第三關 – 猜彈珠的遊戲

*** Status ***

需要把每一輪電腦以及使用者資訊印出來:

1. 彈珠現況
2. 猜奇or偶數
3. 實際結果

Photo from Wikipedia:
https://en.wikipedia.org/wiki/Squid_Game

</Assignments>

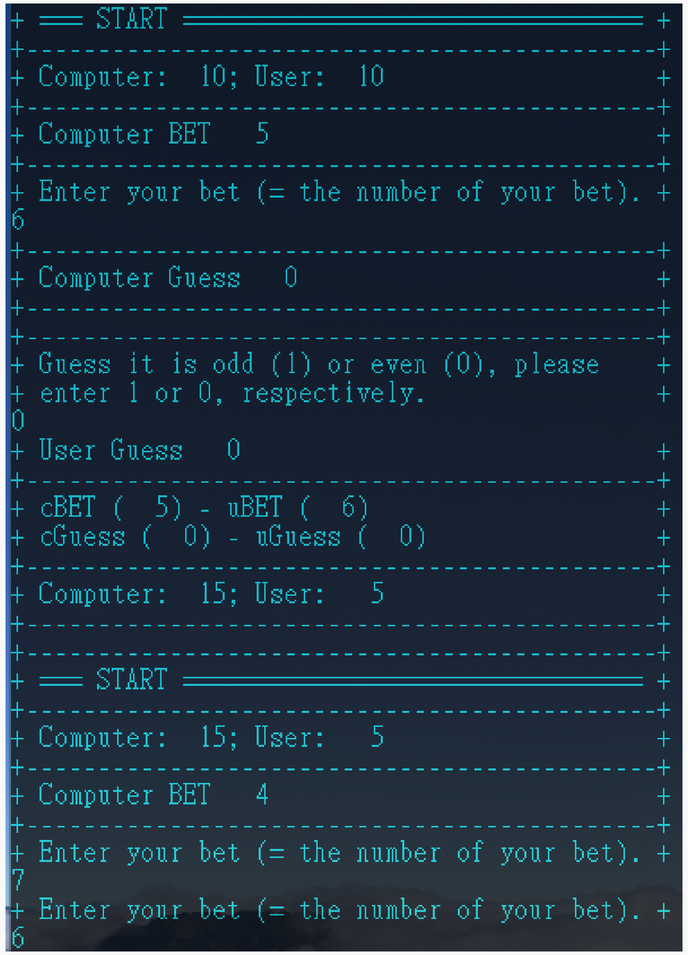
作業一

第三關 – 猜彈珠的遊戲 – random number generator

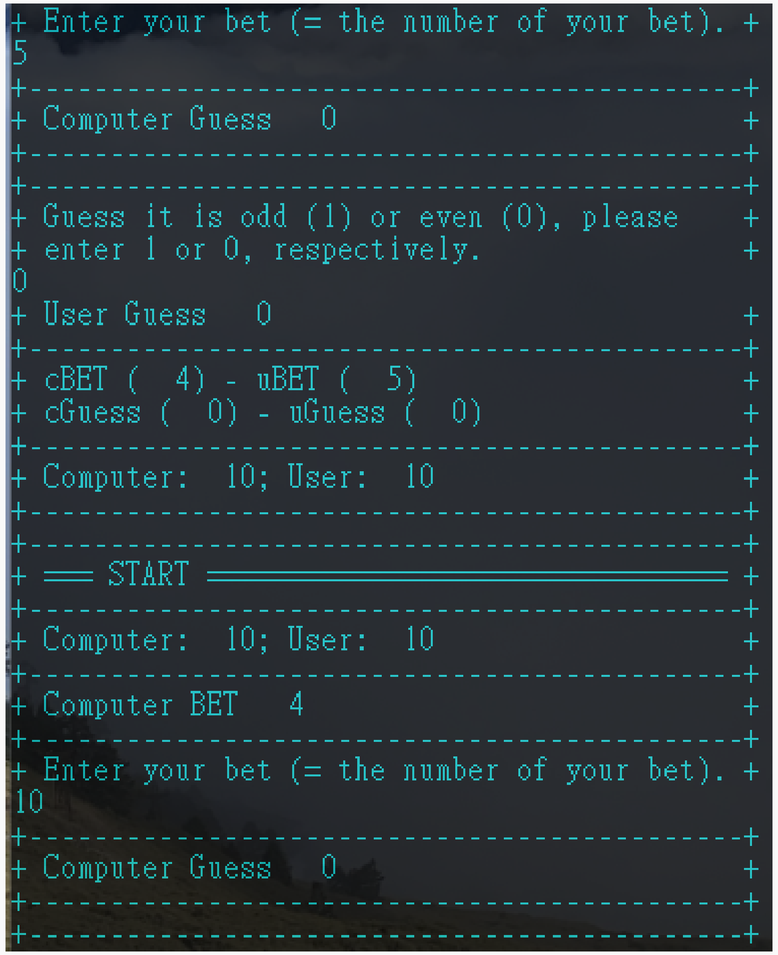
```
int get_rand(int range){
        time_t t;

        /* Initializes random number generator */
        srand((unsigned) time(&t));

        /* Print random numbers from 0 to range */
        int x = rand() % range;
        return x;
}
```

Photo from Wikipedia:
https://en.wikipedia.org/wiki/Squid_Game

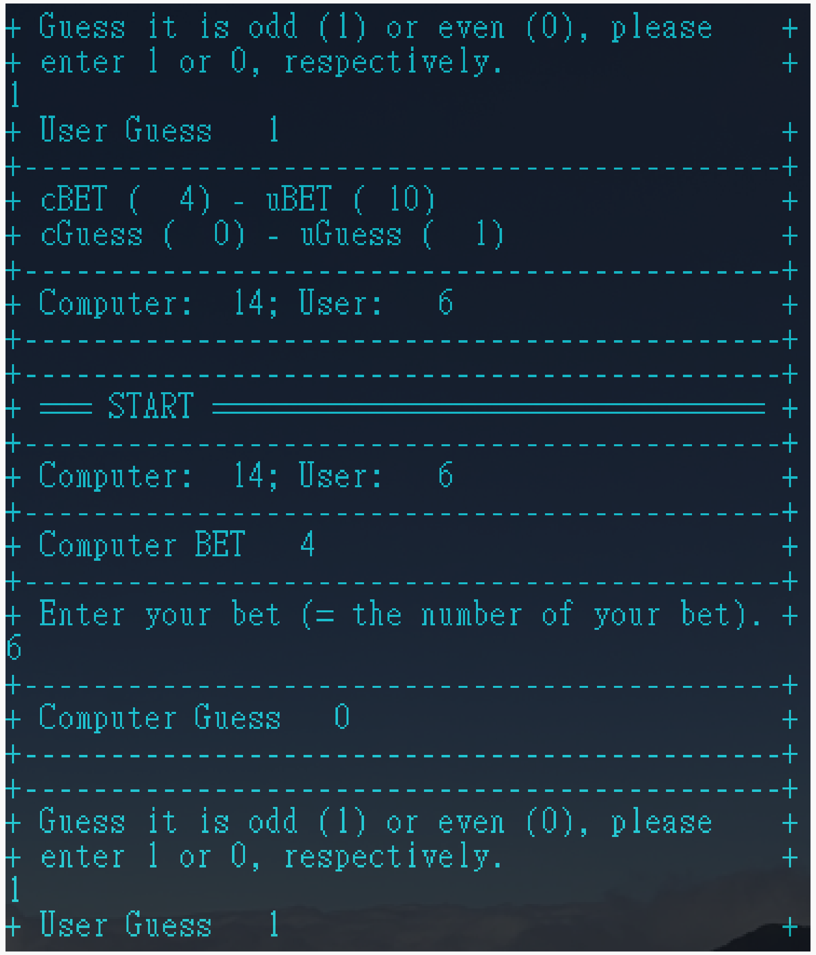
作業一



```
+ == START ================================= +
+------------------------------------------+
+ Computer:  10; User:  10                  +
+------------------------------------------+
+ Computer BET    5                         +
+------------------------------------------+
+ Enter your bet (= the number of your bet). +
6

+------------------------------------------+
+ Computer Guess    0                       +
+------------------------------------------+
+------------------------------------------+
+ Guess it is odd (1) or even (0), please   +
+ enter 1 or 0, respectively.               +
0
+ User Guess    0                           +
+------------------------------------------+
+ cBET (  5) - uBET (  6)                   +
+ cGuess (  0) - uGuess (  0)               +
+------------------------------------------+
+ Computer:  15; User:   5                  +
+------------------------------------------+
+------------------------------------------+
+ == START ================================= +
+------------------------------------------+
+ Computer:  15; User:   5                  +
+------------------------------------------+
+ Computer BET    4                         +
+------------------------------------------+
+ Enter your bet (= the number of your bet). +
7
+ Enter your bet (= the number of your bet). +
6
```

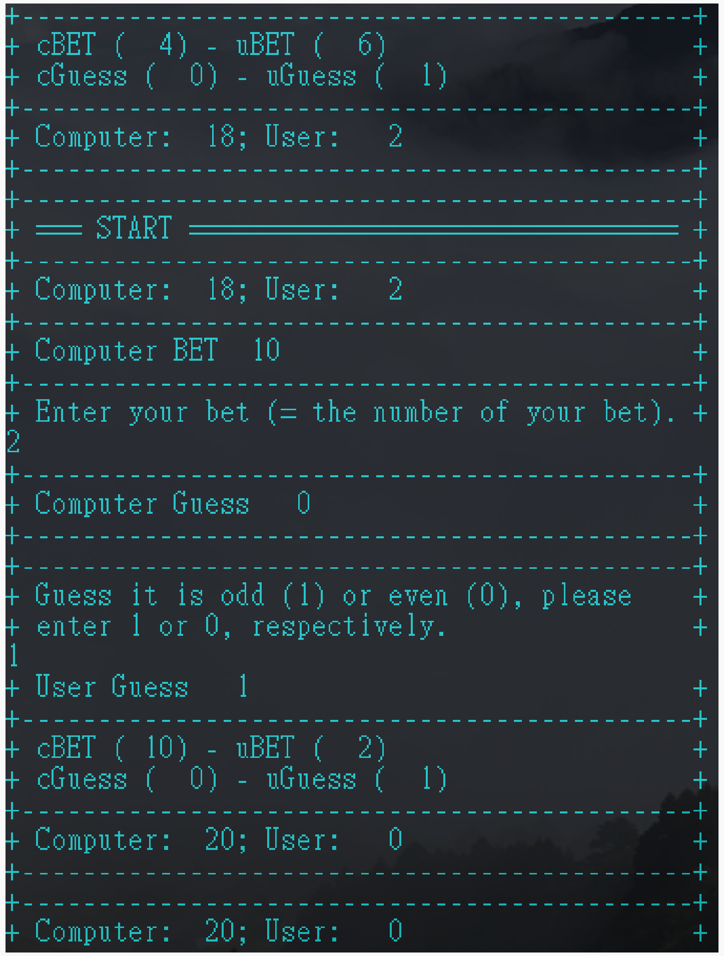


```
+ Enter your bet (= the number of your bet). +
5

+------------------------------------------+
+ Computer Guess    0                       +
+------------------------------------------+
+------------------------------------------+
+ Guess it is odd (1) or even (0), please   +
+ enter 1 or 0, respectively.               +
0
+ User Guess    0                           +
+------------------------------------------+
+ cBET (  4) - uBET (  5)                   +
+ cGuess (  0) - uGuess (  0)               +
+------------------------------------------+
+ Computer:  10; User:  10                  +
+------------------------------------------+
+------------------------------------------+
+ == START ================================= +
+------------------------------------------+
+ Computer:  10; User:  10                  +
+------------------------------------------+
+ Computer BET    4                         +
+------------------------------------------+
+ Enter your bet (= the number of your bet). +
10

+------------------------------------------+
+ Computer Guess    0                       +
+------------------------------------------+
+------------------------------------------+
```

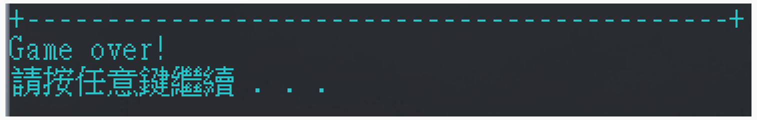


```
+ Guess it is odd (1) or even (0), please   +
+ enter 1 or 0, respectively.               +
1
+ User Guess    1                           +
+------------------------------------------+
+ cBET (  4) - uBET ( 10)                   +
+ cGuess (  0) - uGuess (  1)               +
+------------------------------------------+
+ Computer:  14; User:   6                  +
+------------------------------------------+
+------------------------------------------+
+ == START ================================= +
+------------------------------------------+
+ Computer:  14; User:   6                  +
+------------------------------------------+
+ Computer BET    4                         +
+------------------------------------------+
+ Enter your bet (= the number of your bet). +
6

+------------------------------------------+
+ Computer Guess    0                       +
+------------------------------------------+
+------------------------------------------+
+ Guess it is odd (1) or even (0), please   +
+ enter 1 or 0, respectively.               +
1
+ User Guess    1                           +
```

作業一



```
+--------------------------------------------+
+ cBET (  4) - uBET (  6)                     +
+ cGuess (  0) - uGuess (  1)                 +
+--------------------------------------------+
+ Computer:  18; User:   2                    +
+--------------------------------------------+
+--------------------------------------------+
+ ══ START ═══════════════════════════════    +
+--------------------------------------------+
+ Computer:  18; User:   2                    +
+--------------------------------------------+
+ Computer BET  10                            +
+--------------------------------------------+
+ Enter your bet (= the number of your bet). +
2
+--------------------------------------------+
+ Computer Guess   0                          +
+--------------------------------------------+
+--------------------------------------------+
+ Guess it is odd (1) or even (0), please     +
+ enter 1 or 0, respectively.                 +
1
+ User Guess   1                              +
+--------------------------------------------+
+ cBET ( 10) - uBET (  2)                     +
+ cGuess (  0) - uGuess (  1)                 +
+--------------------------------------------+
+ Computer:  20; User:   0                    +
+--------------------------------------------+
+--------------------------------------------+
+ Computer:  20; User:   0                    +
```



```
+----------------------------------------------+
Game over!
請按任意鍵繼續 . . .
```

References

- C語言: 超好懂的指標，初學者請進 ~
- 蔣宗哲教授 – 程式設計(一) 講義
- Netflex 魷魚遊戲