

# 函數 ( 2 )

by 多拉A夢

我們就先假裝大家都會了

今天要講的函數是...

今天要講的函數是...

把剛剛學的指標應用上去

(如果對剛剛指標有什麼疑惑的點，請舉手跟我說一下哪裡不懂)

# 講一個大家比較有印象的

```
// elements 裡頭有 sizeofElement 這麼多個整數
int arrayGCD (int elements[], int sizeofElement);
```

在繳交的時候也只需要上傳 arrayGCD 的實作即可，請勿上傳完整程式碼，否則你可能會拿到 CE。

當你上傳程式碼片段以後，他會被放在以下位置：

```
#include <iostream>
#include <cmath>

int gcd(int a, int b) {
    int tmp;
    while(b){a%=b; tmp=a; a=b; b=tmp;}
    return a;
}

/* 你的程式碼會被放在這裡 */

int main() {
    int elements[1000], sizeofElement;
    while (std::cin >> sizeofElement) {
        for (int i = 0; i < sizeofElement; i++)
            std::cin >> elements[i];
        std::cout << "GCD of array = " << arrayGCD(elements, sizeofElement) << std::endl;
    }
    return 0;
}
```

# 講一個大家比較有印象的

```
// elements 裡有 1000 個 int 這麼多個整數  
int arrayGCD(int elements[], int sizeOfElement);
```

在繳交的時候也只需要上傳 arrayGCD 的實作即可，請勿上傳完整程式碼，否則你可能會拿到 CE。

當你上傳程式碼片段以後，它會被放在以下位置：

```
#include <iostream>
#include <cmath>

int gcd(int a, int b) {
    int tmp;
    while(b){a%=b; tmp=a; a=b; b=tmp;}
    return a;
}

/* 你的程式碼會被放在這裡 */

int main() {
    int elements[1000];
    while (true) {
        int sizeOfElement;
        for (int i = 0; i < sizeOfElement; i++)
            std::cin >> elements[i];
        std::cout << "GCD of array = " << arrayGCD(elements, sizeOfElement) << std::endl;
    }
    return 0;
}
```

別擔心... 只是給大家看一下

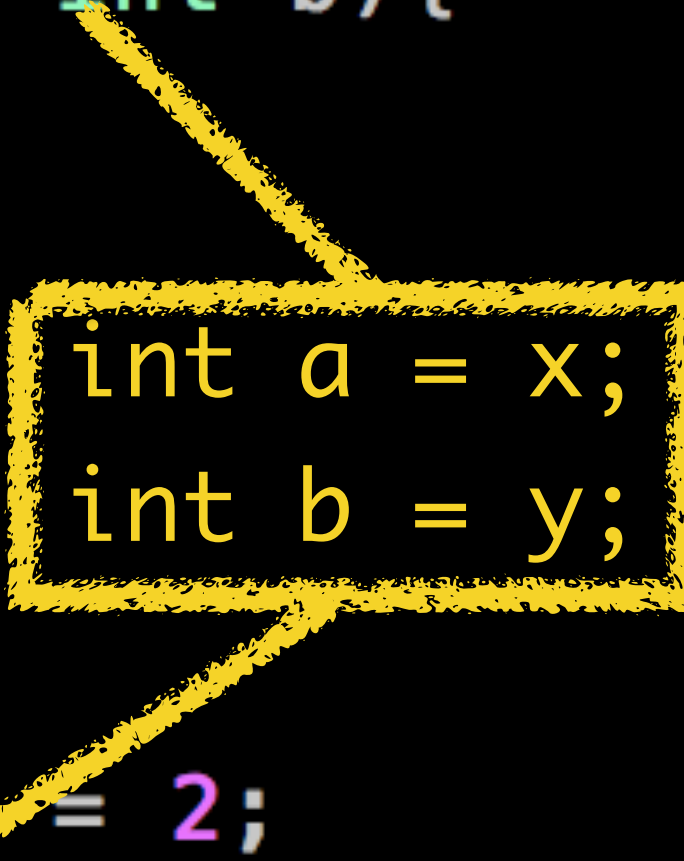
# 函數小複習

```
8 void swap(int x, int y){  
9     int tmp = x;  
10    x = y;  
11    y = tmp;  
12 }  
13  
14 int main(){  
15  
16     int x = 1, y = 2;  
17     swap(x, y);  
18     cout << x << ' ' << y << endl;  
19 }
```



# 函數小複習

```
8 void swap(int a, int b){  
9     int tmp = a;  
10    a = b;  
11    b = tmp;  
12 }  
13  
14 int main(){  
15  
16    int x = 1, y = 2;  
17    swap(x, y);  
18    cout << x << ' ' << y << endl;  
19  
20    return 0;  
21 }
```



結果：交換失敗（攤手

那該怎麼辦呢？

我們剛剛有學指標啊～

嘿嘿～

# 函數 + 指標

```
7  
8 void swap(int *a, int *b){  
9     int tmp = *a;  
10    *a = *b;  
11    *b = tmp;  
12 }  
13  
14 int main(){  
15  
16     int x = 1, y = 2;  
17     swap(&x, &y);  
18     cout << x << ' ' << y << endl;  
19  
20     return 0;  
21 }
```

# 函數 + 指標

```
7
8 void swap(int *a, int *b){
9     int tmp = *a;
10    *a = *b;
11    *b = tmp;
12 }
13
14 int main(){
15
16     int x = 1, y = 2;
17     swap(&x, &y);
18     cout << x << " " << y << endl;
19
20     return 0;
21 }
```

int \*a = &x;  
int \*b = &y;

# 函數 + 指標

```
7
8 void swap(int *a, int *b){
9     int tmp = *a;
10    *a = *b;
11    *b = tmp;
12 }
13
14 int main(){
15
16     int x = 1, y = 2;
17     swap(&x, &y);
18     cout << x << " " << y << endl;
19
20     return 0;
21 }
```

`int *a = &x;`  
`int *b = &y;`

指標要熟悉喔～

# 函數 + 指標

```
7  
8 void swap(int *a, int *b){  
9     int tmp = *a;  
10    *a = *b;  
11    *b = tmp;  
12 }  
13  
14 int main(){  
15  
16     int x = 1, y = 2;  
17     swap(&x, &y);  
18     cout << x << ' ' << y << endl;  
19  
20     return 0;  
21 }
```

因為這裡的位置...

指標要熟悉喔～



# 指標：傳陣列的位置

```
7
8 int find_max(int s[], int size){
9     int MAX = s[0];
10    for(int i = 1; i < size; ++i)
11        if(MAX < s[i])
12            MAX = s[i];
13    return MAX;
14 }
15
16 int main(){
17
18     int s[5] = {5, 6, 100, 1, 10};
19     cout << "MAX = " << find_max(s, 5) << endl;
20
21     return 0;
22 }
```

# 指標：傳陣列的位置

```
7
8 int find_max(int s[], int size){
9     int MAX = s[0];
10    for(int i = 1; i < size; ++i)
11        if(MAX < s[i])
12            MAX = s[i];
13    return MAX;
14 }
15
16 int main(){
17
18     int s[5] = {5, 6, 100, 1, 10};
19     cout << "MAX = " << find_max(s, 5) << endl;
20
21     return 0;
22 }
```

這裡放 `*s` 也可以

這裡放 `&s[0]` 也可以

# 指標：直接回傳指標

```
7  
8 int* find_first_element(int s[], int size){  
9     return &s[0];  
10 }  
11  
12 int main(){  
13  
14     int s[5] = {5, 6, 100, 1, 10};  
15     int *ptr;  
16     ptr = find_first_element(s, 5);  
17     cout << "First element = " << *ptr << endl;  
18  
19     return 0;  
20 }
```

# 指標：直接回傳指標

```
7
8 int* find_first_element(int s[], int size){
9     return &s[0];
10 }
11
12 int main(){
13
14     int s[5] = {5, 6, 100, 1, 10};
15     int *ptr;
16     ptr = find_first_element(s, 5);
17     cout << "first element = " << *ptr << endl;
18
19     return 0;
20 }
```

丟指標陣列進去

&

回傳指標回來

```

/
8          // 傳指標陣列進去  _ _ _ _
9  int* get_first_element_pos(int *ptr[], int size){
10      return ptr[0];  //回傳ptr[0]裡面存的位置
11  }
12
13  int main(){
14      int x = 10, y = 100;
15      int *ptr[2];
16
17      ptr[0] = &x;      //指向 x
18      ptr[1] = &y;      //指向 y
19
20      int *ans = get_first_element_pos(ptr, 2);
21      // ^^^^^^  ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
22      // 宣告指標          這裡會回傳位置
23
24  /***** 檢查 *****/
25      cout << "x position = " << &x << endl;
26      cout << "ans position = " << ans << endl;
27
28      cout << "x value = " << x << endl;
29      cout << "value which ans point to = " << *ans << endl;
30
31      return 0;
32  }

```



```
8 // 傳指標陣列進去
9 int* get_first_element_pos(int *ptr[], int size){
10     return ptr[0]; //回傳ptr[0]裡面存的位置
11 }
```

```
12
13 int main(){
14     int x = 10, y = 100;
15     int *ptr[2];
16
17     ptr[0] = &x; //指向 x
18     ptr[1] = &y; //指向 y
```

回傳位置

```
19
20     int *ans = get_first_element_pos(ptr, 2);
21     // ^^^^^^ ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
22     // 宣告指標          這裡會回傳位置
```

```
23
24     /***** 檢查 *****/
25     cout << "x position = " << &x << endl;
26     cout << "ans position = " << ans << endl;
27
28     cout << "x value = " << x << endl;
29     cout << "value which ans point to = " << *ans << endl;
30
31     return 0;
32 }
```

小結：使用方法超多...

請善用網路自己學習噢



練習：TOJ 241

尋找最大值