

在那段 strcpy 的 code 中，其實錯的也不是 string copy 的 function，是裡面用來計算字串長度的一行 while (見下圖)。

```
while(src[len++]); //wrong  
while(src[++len]); //right
```

那為甚麼上面是錯的下面是對的呢？

首先我們可以先把比較傳統的(?)的計算字串的 while 迴圈寫出來做比較(下圖)。

```
while(src[len] != '\0')len++;  
while(src[len] != NULL)len++;  
while(src[len])len++;
```

從上圖我們可以看出，若 while 的條件內沒有寫入運算子時，在此就是 != NULL 的意思。那以上圖的寫法來說，邏輯是這樣運行的，若 src[len] 內有字元時，len 會加 1，反之若裡面沒有字元了，則不會加 1。

若以 char src[] = "IU is a girl!" 且 len 從 0 開始，所以 src[0]~src[12] 之間會是有字元的，而字串長度為 13，那我們可以知道最後 len 應該要等於 13。

- 在 while(src[len])len++; 的這個寫法中邏輯會如下所述

len = 0 → src[0] 內有字元 → 判斷正確 → 進行 len++ (此時 len 為 1)

...

len = 12 → src[12] 內有字元 → 判斷正確 → 進行 len++ (此時 len 為 13)

len = 13 → src[12] 內為 '\0' → 判斷錯誤 → 跳出迴圈(此時 len 仍為 13)

得 len = 13

那讓我們回到最開始的那兩種寫法

```
while(src[len++]); //wrong  
while(src[++len]); //right
```

這種寫法與上一種寫法的差別在於，上一個是判斷完，正確才進行 len++，而這是在判斷的同時進行 len++或++len。

(i.e. 所以上一種寫法++len 或 len++不影響答案)

同樣以 char src[] = "IU is a girl!"，len 從 0 開始，len 最後應該要等於 13。

- 在 while(src[len++]); 中邏輯如下所述

len = 0 → src[0]內有字元 → 判斷正確 並 進行 len+1 (此時 len 為 1)

...

len = 12 → src[12]內有字元 → 判斷正確 並 進行 len+1 (此時 len 為 13)

len = 13 → src[13]內為 ' \0' → 判斷錯誤 並 進行 len+1 → 跳出迴圈

(此時 len 已為 14)

得 len = 14 會多 1

- 而在 while(src[++len]); 中邏輯則如下所述

len = 0 → len+1 (此時 len 為 1) 並 判斷正確 (src[1]內有字元)

...

len = 11 → len+1 (此時 len 為 12) 並 判斷正確 (src[12]內有字元)

len = 12 → len+1 (此時 len 為 13) 並 判斷錯誤 (src[13] 內為 ' \0') →

跳出迴圈

得 len = 13，是正確的