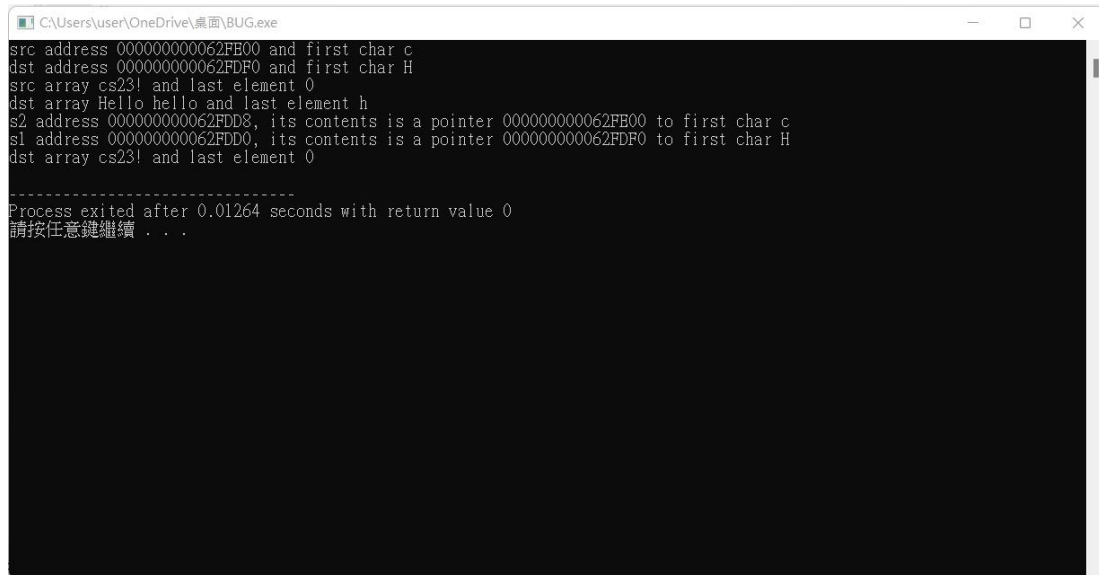


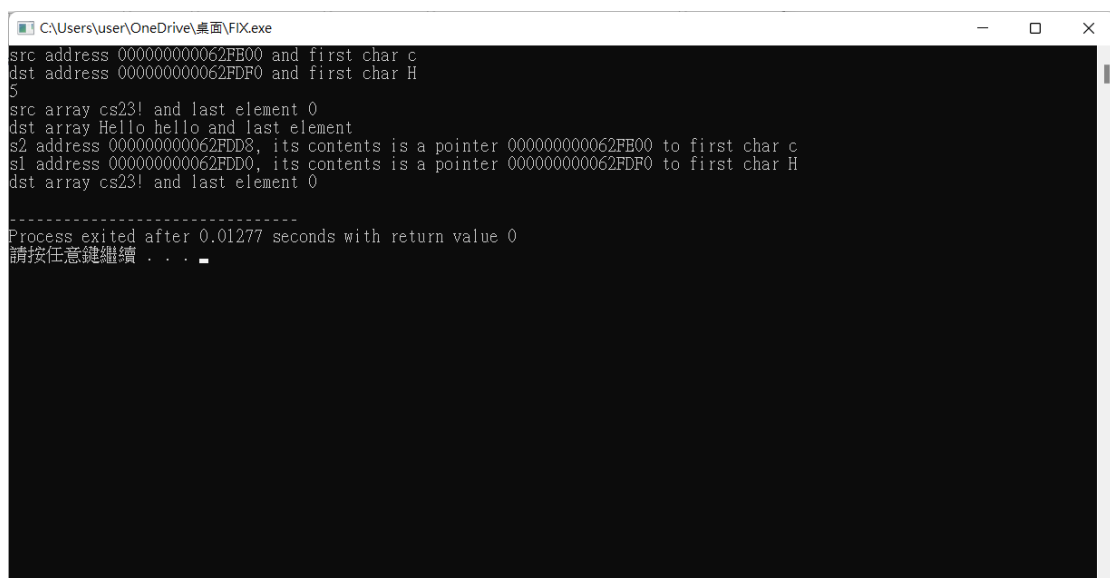
閱讀完那篇文章，我從中了解到，C 在做數值的交換時，要用 pointer 去操作，如果只是單純把值給複製是沒辦法做交換的。而在 strcpy.c 的程式碼中，有錯誤的部分也不是 string copy 的地方，是用來計算長度的程式碼。原先是 len++，後來修改成++len。



```
C:\Users\user\OneDrive\桌面\BUG.exe
src address 000000000062FEB0 and first char c
dst address 000000000062FDF0 and first char H
src array cs23! and last element 0
dst array Hello hello and last element h
s2 address 000000000062FDD8, its contents is a pointer 000000000062FEB0 to first char c
s1 address 000000000062FDD0, its contents is a pointer 000000000062FDF0 to first char H
dst array cs23! and last element 0

-----
Process exited after 0.01264 seconds with return value 0
請按任意鍵繼續 . . .
```

上圖為修改前的結果，判斷 cs123 字串的长度為 6



```
C:\Users\user\OneDrive\桌面\FIX.exe
src address 000000000062FEB0 and first char c
dst address 000000000062FDF0 and first char H
5
src array cs23! and last element 0
dst array Hello hello and last element
s2 address 000000000062FDD8, its contents is a pointer 000000000062FEB0 to first char c
s1 address 000000000062FDD0, its contents is a pointer 000000000062FDF0 to first char H
dst array cs23! and last element 0

-----
Process exited after 0.01277 seconds with return value 0
請按任意鍵繼續 . . .
```

上圖為修改後的結果，判斷 cs123 字串的长度為 5

使用 while(src[len++]) 程式是先判斷完再進行++，舉例來說“cs123”是一

個長度為 5 的字串，但使用 `while(src[len++])` 會在最後字串 `'\0'` 的部分判斷完再幫他加 1，所以會得到長度為 6 的字串。而使用 `while(src[++len])` 程式是先++再進行判斷，舉例來說“cs123”是一個長度為 5 的字串，使用 `while(src[++len])` 在最後字串 `'\0'` 的部分不會有再幫他+1 的問題。