在 strcpy.c 中，發生 bug 的地方是 while(str[len++])，len++是先去判斷後再加 1，即使是偵測到’\0’ ，雖然會判斷錯誤，但是 len 仍然會加 1。

以{src[]=”cs23!”}為例子，字串長度為 5。

int len=0;

src[0]=’c’ →len=1，

src[1] =’s’ →len=2，

src[2] =’2’ →len=3，

src[3] =’3’ →len=4，

src[4] =’!’ →len=5，

src[5] =’\0’ →len=6。

此時 len=6，雖然 src 的最後一個字元是\0，但是因為是先判斷後++，所以最後會被多加 1，導致 dst 會印出’ h’

在 fixed-strcpy.c 中，修正的地方是 while(str[++len])，++len 是先加再去判斷，以上面的例子來說，len 會改變如下。

int len=0;

len=1→ src[1]=’s’ ，

len=2→ src[2]=’2’ ，

len=3→ src[3]=’3’ ，

len=4→ src[4]=’!’ ，

len=5→ src[5]=’ \0’ →判斷錯誤後跳出迴圈。

此時 len=5 為結束字元的位置。

下為 i++及++j 的比較

```c
#include<stdio.h>
int main(){
    int i= 0,j= 0;
    int a=0,b=0;
    a=i++;
    b=++j;
    printf("i=%d\nj=%d\na=%d\nb=%d\n",i,j,a,b);
}
```

```
D:\課程、作業\c\strcpy.exe
i=1
j=1
a=0
b=1

--------------------------------
Process exited after 0.0406 seconds with return value 0
請按任意鍵繼續 . . .
```