HW4

B0929060 張孟佳

使用 swap()function 時，變換的變數是使用另一塊記憶體，所以傳回主程式時原本的變數不會受到 swap 改變。但如果使用指標時，是以記憶體位置操作記憶體位置的值，回傳到主程式時就能夠成功轉換。

*字號有兩種用法：第一種是用來宣告指標變數，第二種是用來取指標所指向的值。

```
/*
   File: strcpy.c

   Description: Copies a source string to a destination. Keeps copying
   until it finds the NULL char in the source char string

   Input: char pointers for source (s2) and destination (s1)

   Output: returns the pointer to the destinatio (s1)
*/

#include<stdio.h>
#include<stdlib.h>

char *my_strcpy(char * , const char * );

int main()
{

   char src[] = "cs23!";
   char dst[]="Hello hello";
   char *curdst;
   int len=0;

   printf("src address %p and first char %c \n", (void *)&src, src[0]);
   printf("dst address %p and first char %c \n", (void *)&dst, dst[0]);

   // compute where NULL character is '\0' ASCII 0

   while(src[len++]);
```

```
    // print out the char arrays and various addresses.

    printf("src array %s and last element %d\n", src, atoi(&src[len]));
    printf("dst array %s and last element %c\n", dst, dst[len]);

    // do the copy

    curdst= my_strcpy(dst, src);

    // check to see if the NULL char is copied too.

    printf("dst array %s and last element %d\n", dst, atoi(&dst[len]));

    return 0;

}
char *my_strcpy(char *s1, const char *s2) {

    register char *d = s1;

    // print the pointer variables address and their contents, and first char

    printf("s2 address %p, its contents is a pointer %p to first char %c \n", (void *)&s2,
(void *)s2, *s2);
    printf("s1 address %p, its contents is a pointer %p to first char %c \n", (void *)&s1,
(void *)s1, *s1);

    while ((*d++ = *s2++));
    return(s1);

}
```

在 strcpy.c 和 fixedstrcpy.c 裡面，修正的地方是 while(src[len++])改成 while(src[++len])。使用前者的 len++值當作 index 會使印出的 last element 不是正確的'\0'，在 index 對應 dst 字串中的元素會為 dst[6]=h。而使用後者的++len，才能夠印出正確的 last element，而 dst 字串的元素會是 dst[5]= (空格)。