

B1029026 張晉鉞

正確:

```
#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);

}
```

錯誤:

```
#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;

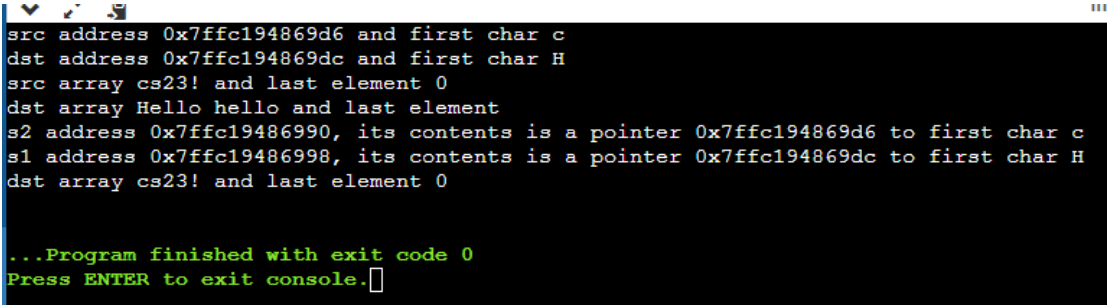
    // print the pointer variables address and their contents, and first char
    printf("s2 address %p, its contents is a pointer %p to\n", (void *)&s2, (void *)s2, *s2);
    printf("s1 address %p, its contents is a pointer %p to first\n", (void *)&s1, (void *)s1, *s1);

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

    return(s1);
}

```

正確的執行結果:



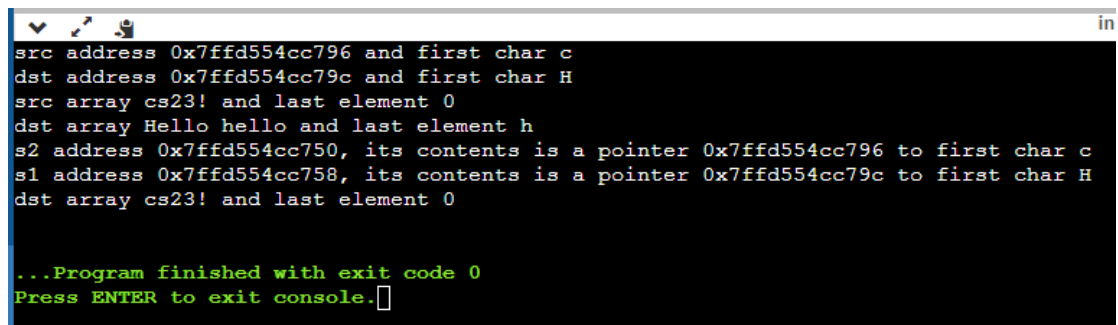
```

src address 0x7ffc194869d6 and first char c
dst address 0x7ffc194869dc and first char H
src array cs23! and last element 0
dst array Hello hello and last element
s2 address 0x7ffc19486990, its contents is a pointer 0x7ffc194869d6 to first char c
s1 address 0x7ffc19486998, its contents is a pointer 0x7ffc194869dc to first char H
dst array cs23! and last element 0

...Program finished with exit code 0
Press ENTER to exit console.

```

錯誤的執行結果:

A screenshot of a terminal window with a black background and white text. The text shows the execution of a program with several memory addresses and character pointers. At the bottom, it states the program finished with exit code 0 and prompts the user to press ENTER to exit the console. The window has a title bar with standard OS icons and the text 'in' on the right.

```
src address 0x7ffd554cc796 and first char c
dst address 0x7ffd554cc79c and first char H
src array cs23! and last element 0
dst array Hello hello and last element h
s2 address 0x7ffd554cc750, its contents is a pointer 0x7ffd554cc796 to first char c
s1 address 0x7ffd554cc758, its contents is a pointer 0x7ffd554cc79c to first char H
dst array cs23! and last element 0

...Program finished with exit code 0
Press ENTER to exit console.
```

Bug 是在 `while(src[len++]);`，應該改成 `while(src[++len]);`。 `len++`表示先做再加，這樣導致當字串跑到 `'0'` 值時，會再多加 1，會多跑一次才跳出迴圈。修正為 `++len` 時，就是先加再做了，此時便不會產生跑到 `'0'` 還會多跑一次才跳出迴圈。