

# C Note

## Ch9 Pointer

### #pointer as array(P.198)

```
Int a[arraysize];
```

```
Int *ptr=a;
```

```
*(ptr+i)=ptr[i]!=*ptr[i]
```

則ptr就是一個array了。

特殊用途：

```
ptr=&a[2]
```

則ptr[0]就是a[2],而不是從a[0]開始

### #multi-dimensional pointer array(P.202)

```
Int a[x][y][z];
```

```
Int *ptr=a[0][0]. Ptr是一格一格的陣列
```

```
Int (*arrayptr)=a[0]; arrayptr 是一行陣列 如果是要指到a一定要括號
```

```
Int (*matrixptr)=a; matrix 一次指到[3][4]的陣列
```

## Ch11 String

#字串為字元的陣列 也可以把這個陣列用一個指標來指。

Ex: char str[10];

```
Char *ptr="helloworld".
```

字串的最後一格要放'\0',記得留儲存空間。

### #常用函式 ( P .230)

---

1.strlen:讀字串長度

---

2.strcpy, strcat:皆為複製字串，但strcpy為clear後輸入, strcat為接在原本的後面。

注意strcpy, strcat不能用在用指標建立的記憶體上，因為指標不是字串本身，他只指到字串的記憶體上而已。所以要注意緩衝區覆蓋的問題。

要注意指標陣列不能用strcpy，因他的記憶體大小是一個指標

---

### 3.strncpy, strncat

能夠指定要複製幾個字元。

---

### Strcmp

比較兩個string 的大小。int strcmp(a,b) 如果a>b 回傳正數，a<b 回傳負數，a==b回傳0.

---

### Strchr, strstr

Strchr是在字串內尋找字元（由前面開始找） strchr（從後面開始找）

Strstr是在字串內尋找字串

三者都是回傳收尋目標的記憶體位置。

# 小技巧：當我們使用回傳位置的方法時，將原字串定義為矩陣就可以利用回傳值-cell得到是在字串裡的第幾個字。

Ex:char \*ptr=strchr(a[l], "c") a[l]=ace

ptr-a[l]=2

---

### Strspn, strcspn

strspn(a,b) 檢查a有幾個連續的字元出現在b中。

strcspn(a,b) 檢查a有幾個連續的字元沒出現在b中。

---

### Strtok

strtok(\*char string, char \*cutter);

If \*cutter=="abc" 則string會以 a,b,c三個分割子來切，然後把a,b,c的地方都換成"\0"

## Ch12 宣告修飾

### Static(P.287)

初始化只會在一開始時，後面都會抓前面的值

## Ch13位元運算（超級難！）

### 數字and. Or ...

a&b:比較a b的二進位 同理for or exclusive or...

### Shift(P.296)

<< 向左移 右邊補0

>>向右移 左邊補舊的MSB(最左邊的值)

## Ch16 檔案處理

Fopen fclose assert(file!=NULL) fgetc fputc sscanf...要記起來

Fscanf fprintf都跟原本的scanf printf一樣，只是輸出輸入在檔案裡而已，而不是螢幕

fprintf(stdout, format...)可以把檔案的內容print在螢幕上

fopen(filename,option(such as r,w,a))

R：讀已經建立的檔案

W：創立並讀新檔案

Fread fwrite 可以把檔案的內容讀出\入到其他地方，回傳值為成功讀/寫的數目

Fprintf 回傳成功輸出character(沒有很確定要注意"%d\n" 回傳的是2)

fscanf 回傳成功讀數目

Fseek 可以讀檔案的任意位元組

不要做用stdin讀東西fgets的蠢事

不要用fscanf +使用 rb wb，會出現無法判讀的情況 (format %s,%d ...)