

qsort 排序:

called by address

```
int cmpfunc (const void* a, const void* b)
{
    return (*(int*)b - *(int*)a);
}
```

通過調整此二數, 排序出來的 < 大 → 小
小 → 大

qsort (Array 型的變數名, 元素數, 元素大小, cmpfunc)

一定要有 cmpfunc, 才能使用 qsort

n = 7;

char array[n][200];

qsort (array, sizeof array / sizeof (array[0]), sizeof (*array),
cmpfunc)

※ 比較函數編寫規則:

- ① 如果 *a 排在 *b 前, 則要傳回負整數
- ② 如果 *a 排在 *b 後, 則要傳回正整數
- ③ 如果 *a 排在 *b 都行, 則要傳回 0

strcmp 比較二字串是否相同:

char str1[10], str2[12];

int num;

num = strcmp (str1, str2);

※ num 數字的大小代表二字串的差異為何

- ① 如果 $num < 0$, 則 $str1$ 長度 **大於** $str2$
- ② 如果 $num > 0$, 則 $str1$ 長度 **小於** $str2$
- ③ 如果 $num = 0$, 則 $str1$ 長度 **與** $str2$ **相等**

以 `getline` 自行設定字符串的分隔符 (C++ 獨有):

須先 `include <string.h>` 這個 header file

```
string str;  
getline ( cin , str , ',' )
```

如此一來, user 在輸入字詞時, 遇到 ',' 才會停止給 `str` 賦值

以 `pow` 算 a^b

須先 `include <math.h>` 這個 header file

```
double a = 3.0, b = 4.0, ans;  
ans = pow ( a , b ); // ans 之值為 81
```

注意 `pow` function 中的引數 a, b 須為 `double` 型的資料