

C 語言常用函式

C.1 轉換函式

轉換函式可將字串轉成整數、長整數、浮點數；也可以將指定的數值轉成指定的數字系統並放入指定的字元陣列(字串)內。常用轉換函式如下：

函式	功能說明
atof	語法：double atof(const char *) 標頭檔：math.h 功能：將字串轉換成倍精確浮點數並傳回。轉換失敗時會傳回 0。
atoi	語法：int atoi(const char *) 標頭檔：stdlib.h 功能：將字串轉換成整數並傳回。轉換失敗時會傳回 0。
atol	語法：long atol(const char *) 標頭檔：stdlib.h 功能：將字串轉換成長整數並傳回。轉換失敗時會傳回 0。
itoa	語法：char *itoa(int n, char *str, int base) 標頭檔：stdlib.h 功能：長整數 n 轉成以數字系統 base 為底的字串，然後再指定給 str 字元陣列。 簡例：假設 char str[20]; ① itoa(70, str, 8) /* str 為 "106" 字串 */ ② itoa(70, str, 2) /* str 為 "1000110" 字串 */

函式	功能說明
ltoa	語法：char *ltoa(long l, char *str, int base) 標頭檔：stdlib.h 功能：將長整數 n 轉成以數字系統 base 為底的字串，然後再指定給 str 字元陣列。 簡例：假設 char str[20]; ①ltoa(70, str, 8) /* str 為 "106" 字串 */ ②ltoa(70, str, 2) /* str 為 "1000110" 字串 */
toascii	語法：int toascii(int c) 標頭檔：ctype.h 功能：將字元 c 轉成 ASCII 字元並傳回。
toupper	語法：int toupper(int c) 標頭檔：ctype.h 功能：將字元 c 轉成大寫英文字母並傳回。
tolower	語法：int tolower(int c) 標頭檔：ctype.h 功能：將字元 c 轉成小寫英文字母並傳回。

C.2 字串處理函式

字串處理函式可以對字串做字串合併、字串拷貝、字串搜尋、字串反轉、字串分割、字串大小寫轉換以及求字串長度...等功能，這些字串函式全部都宣告在 string.h 標頭檔內，所以要使用這些字串函式之前，必須在程式的最開頭撰寫 #include <string.h> 將字串函式含入到程式中。下表中字串函式以 str1 及 str2 兩個字串做舉例說明：

```
char str1[15]="碁峰";
char *str2="工作室";
```

函式	功能說明
strcat	<p>語法：char *strcat(char *str1, const char *str2)</p> <p>功能：將來源的 str2 字串連接至目地的 str1 陣列後面。但 str1 能容納的字元個數必須大於原來的 str1 加上 str2 的字元個數，否則執行時會發生錯誤。</p> <p>簡例：① strcat(str1, str2); /* 結果 str1 是 "碁峰工作室" 字串 */</p> <p>② char s1[8]="gotop"; strcat(s1, " is best"); /* 錯誤，s1 只能容納 8 個字元(包含 '\0')，*/ /* 因此 s1 連接 " is best" 字串後會超過 8 個字元個數，*/ /* 此如 s1 將無法放置 "gotop is best"，*/ /* 所以執行程式時會發生錯誤情形 */</p>
strncat	<p>語法：char *strncat(char * str1, const char *str2, size_t)</p> <p>功能：將來源的 str2 字串前面 size_t 個字元連接至目地的 str1 陣列後面。但 str1 能容納的字元個數必須大於原來的 str1 加上 str2 的字元個數，否則執行時會發生錯誤。</p> <p>簡例：① strncat(str1, str2, 4); /* 因為一個中文字代表 2 Bytes */ /* str1 為 "碁峰工作" 字串 */</p> <p>② char s1[20]="gotop"; strncat(s1, "△best", 3); /* 結果 s1 是 "gotop△be" 字串 */</p> <p>③ char s1[7]="gotop"; strncat(s1, "△best", 3); /* 錯誤，s1 無法放置 "gotop△be" 字串 */</p>
strcpy	<p>語法：char *strcpy(char *str1, const char *str2)</p> <p>功能：將 str2 字串複製到 str1 陣列，原來的 str1 內容會覆蓋。若 str2 字串的字元個數大於 str1 陣列所能容納的字元個數，則執行時會發生錯誤。</p> <p>簡例：① strcpy(str1, str2); /* 結果 str1 是 "工作室" 字串 */</p> <p>② char s1[5]="YMCA"; strcpy(s1, "Visual Basic"); /* 錯誤 */</p>

函式	功能說明
strncpy	<p>語法：char *strncpy(char *str1, const char *str2, size_t)</p> <p>功能：將 str2 前面 size_t 個字元複製到 str1 陣列，原來的 str1 內容會被覆蓋。</p> <p>簡例：① strncpy(str1,str2,2); /* 一個中文字表示 2 Bytes */ /* 所以 str1 是 "工" 字串 */</p> <p>② char s1[5]="YMCA"; strncpy(s1, "best", 2); /* s1 是 "be" 字串 */</p> <p>③ char s2[5]= "YMCA"; strncpy(s2, "Visual Basic", 10); /* 錯誤 */</p>
strcmp	<p>語法：int strcmp(const char *str1, const char *str2)</p> <p>功能：str1 字串與 str2 字串由左至右以字元的 Ascii 碼進行比較，並傳回一個整數值。可透過傳返回值來判斷比較的結果，說明如下：</p> <ul style="list-style-type: none"> ① 若傳返回值等於 0，表示 str1 字串與 str2 字串相等 ② 若傳返回值小於 0，表示 str1 字串小於 str2 字串 ③ 若傳返回值大於 0，表示 str1 字串大於 str2 字串
strncmp	<p>語法：int strncmp(const char *str1, const char *str2, size_t)</p> <p>功能：對 str1 字串和 str2 字串最前面的 size_t 個字元進行比較，並傳回一個整數值，可透過傳返回值來判斷比較的結果：</p> <ul style="list-style-type: none"> ① 若傳返回值等於 0，表示 str1 字串與 str2 字串相等 ② 若傳返回值小於 0，表示 str1 字串小於 str2 字串 ③ 若傳返回值大於 0，表示 str1 字串大於 str2 字串
strlen	<p>語法：size_t strlen(const char *str1)</p> <p>功能：傳回字串的長度(字元個數)，但不包含 '\0' 字元(NULL)。</p> <p>簡例：① int n1=strlen(str1); /* 中文字佔用兩個字元 */ /* str1 為 "基峰" ，所以 n1=4 */</p> <p>② int n2=strlen("gotop"); /* n2=5 */</p>
strtok	<p>語法：char *strtok(char *str1, const char *str2)</p> <p>功能：使用 str2 來當做區隔字串，將 str1 字串中有 str2 字串之前的字串取出，然後再指定給 str1 字串。</p> <p>簡例：char s1[]="gotop△is△best"; char s2[]="△"; /* 使用△(空白)來當做區隔字串 */ strtok(s1, s2); /* 結果 s1 為 "gotop" */</p>

函式	功能說明
strchr	<p>語法：char *strchr(const char *str, int ch)</p> <p>功能：傳回搜尋 str 字串中第一次出現 ch 字元之位置的字串。若搜尋失敗則傳回 NULL。</p> <p>簡例：① char *str="gotop is best"; char *s2=strchr(str, 'o'); /* 結果 s2 為 "otop is bes" 字串 */ ② char *str="gotop is best"; char *s2=strchr(str, 'y'); /* 結果 s2 為 NULL */</p>
strrchr	<p>語法：char *strrchr(const char *str, int ch)</p> <p>功能：傳回搜尋 str 字串中最後出現 ch 字元之位置的字串。若搜尋失敗則傳回 NULL。</p> <p>簡例：① char *str="gotop is best"; char *s2=strrchr(str, 's'); /* 結果 s2 字串為 "st" */ ② char *str="gotop is best"; char *s2=strrchr(str, 'y'); /* 結果 s2 為 NULL */</p>
strrev	<p>語法：char *strrev(char *str1)</p> <p>功能：將 str1 字串進行前後順序反轉。</p> <p>簡例：char s1[]="gotop is best"; strrev(s1); /* 結果 s1 為 "tseb si potog" 字串 */</p>
strstr	<p>語法：char *strstr(const char *str1, const char *str2)</p> <p>功能：傳回 str1 字串中第一次出現 str2 位置的字串。若搜尋失敗會傳回 NULL。</p> <p>簡例：char *s1="gotop is best"; char *s2=strstr(s1, "is"); /* 結果 s2 為 "is best" 字串 */</p>
strlwr	<p>語法：char *strlwr(char *str1)</p> <p>功能：將 str1 字串的所有英文字母轉成小寫英文字母。</p> <p>簡例：char s1[20]="Gotop IS bEsT"; strlwr(s1); /* 結果 s1 為 "gotop is best" 字串 */</p>
strupr	<p>語法：char *strupr(char *str1)</p> <p>功能：將 str1 字串的所有英文字母轉成大寫英文字母。</p> <p>簡例：char s1[20]="Gotop IS bEsT"; strupr(s1); /* 結果 s1 為 "GOTOP IS BEST" 字串 */</p>

函式	功能說明
strxfrm	<p>語法：size_t strxfrm(const char *str1, const char *str2, size_t)</p> <p>功能：將 str2 前面 size_t 個字元替換成 str1 陣列前面的 size_t 個字元，並傳回 str2 字串的長度。若替換失敗會發生執行時期錯誤。</p> <p>簡例：char s1[20]="gotop is best"; strxfrm(s1, "how are you", 5); /* 傳回 "how are you" 字串長度 11，接著再將 s1 字串前面 */ /* 5 個字元換成 "how are you" 字串最前面 5 個字元， */ /* 結果 s1 為 "how a is best" 字串 */</p>

C.3 字元判斷函式

如下字元判斷函式可以用來判斷某一個字元是否為數字、英文字母、標點符號或控制字元...等。這些函式都宣告於 ctype.h 標頭檔。

函式	功能說明
isalnum	<p>語法：int isalnum(int c)</p> <p>功能：判斷字元 c 是否為英文字母或數字，若是傳回非 0 值，否則傳回 0。</p>
isalpha	<p>語法：int isalpha(int c)</p> <p>功能：判斷字元 c 是否為英文字母，若是傳回非 0 值，否則傳回 0。</p>
isascii	<p>語法：int isascii(int c)</p> <p>功能：判斷字元 c 是否為 0~127 之 ASCII 值，若是傳回非 0 值，否則傳回 0。</p>
isctrl	<p>語法：int isctrl(int c)</p> <p>功能：判斷字元 c 是否為控制字元，若是傳回非 0 值，否則傳回 0。</p>
isdigit	<p>語法：int isdigit(int c)</p> <p>功能：判斷字元 c 是否為十進位數字，若是傳回非 0 值，否則傳回 0。</p>

函式	功能說明
isxdigit	語法：int isxdigit(int c) 功能：判斷字元 c 是否為十六進位數字，若是傳回非 0 值，否則傳回 0。
isspace	語法：int isspace(int c) 功能：判斷字元 c 是否為空白字元？若是傳回非 0 值，否則傳回 0。
isgraph	語法：int isgraph(int c) 功能：判斷字元 c 是否為可列印字元，若是傳回非 0 值，否則傳回 0。此函式不會將空白字元當成可列印字元。
isprint	語法：int isprint(int c) 功能：判斷字元 c 是否為可列印字元，若是傳回非 0 值，否則傳回 0。
isupper	語法：int isupper(int c) 功能：判斷字元 c 是否為大寫英文字母，若是傳回非 0 值，否則傳回 0。
islower	語法：int islower(int c) 功能：判斷字元 c 是否為小寫英文字母，若是傳回非 0 值，否則傳回 0。
ispunct	語法：int ispunct(int c) 功能：判斷字元 c 是否為標點符號字元，若是傳回非 0 值，否則傳回 0。

C.4 數學函式

下表常用數學函式皆宣告於 math.h 標頭檔中：

函式	功能說明
sin	語法：double sin(double a) 功能：傳回 a 徑度的正弦值。
cos	語法：double cos(double a) 功能：傳回 a 徑度的餘弦值。
tan	語法：double tan(double a) 功能：傳回 a 徑度的正切值。
asin	語法：double asin(double a) 功能：傳回正弦值為 a 的徑度。
acos	語法：double acos(double a) 功能：傳回餘弦值為 a 的徑度。
atan	語法：double atan(double a) 功能：傳回正切值為 a 的徑度。
atan2	語法：double atan2(double a, double b) 功能：傳回正弦值 a/b 的徑度。
exp	語法：double exp(double x) 功能：傳回 e^x 值，e 為自然對數。
log10	語法：double log10(double x) 功能：傳回 $\text{Log}_{10}x$ 的值。
log	語法：double log(double x) 功能：傳回 $\text{Log}_e x$ 的值(自然對數值)，即數學上的 $\text{Ln}(x)$ 。
labs	語法：long labs(long n) 功能：傳回長整數 n 的絕對值。
fabs	語法：double fabs(double n) 功能：傳回浮點數 n 的絕對值。
floor	語法：double floor(double n) 功能：傳回小於或等於 n 的最大整數。
ceil	語法：double ceil(double n) 功能：傳回大於或等於 n 的最小整數。

函式	功能說明
pow	語法：double pow(double a, double b) 功能：傳回數值 a 的 b 次方。 簡例：printf("%d", pow(5, 3)); /* 求出 5 ³ , 即印出 125 */
sqrt	語法：double sqrt(double n) 功能：傳回數值 n 的平方根。 簡例：int n=sqrt(16); /* n=4 */ double d=sqrt(33); /* d=5.744563 */
fmod	語法：double fmod(double a, double b) 功能：傳回 a 除以 b 的餘數。 簡例：double d=fmod(1.8, 0.5); /* d=0.3 */

下表函式皆宣告於 stdlib.h 標頭檔中：

函式	功能說明
abs	語法：int abs(int n) 功能：傳回整數 n 的絕對值。 簡例：printf("%d", abs(-2)); /* 印出 2 */ Printf("%d", abs(-2.3)); /* 印出 2 */
max	語法：<type> max(<type> x, <type> y) 功能：傳回 x, y 兩數之中的最大值。
min	語法：<type> min(<type> x, <type> y) 功能：傳回 x, y 兩數之中的最小值。
rand	語法：int rand(void) 功能：傳回介於 0~32767 之間的亂數。
srand	語法：void srand(unsigned int x) 功能：以 x 當亂數產生器的種子。 簡例：通常都是以時間來當做亂數產生器的種子，寫法如下： srand((unsigned)time(NULL));

C.5 其他常用函式

函式	功能說明
exit	<p>語法：void exit(int)</p> <p>標頭檔：stdlib.h</p> <p>功能：結束程式之前將緩衝區內的資料寫回指定的檔案，最後再關閉檔案並結束程式。</p>
malloc	<p>語法：void malloc(size_t)</p> <p>標頭檔：stdlib.h</p> <p>功能：動態配置記憶體の起始位址指定給指標變數使用，如果記憶體空間不足，則呼叫 malloc() 函式會傳回 NULL。</p>
free	<p>語法：void free(void *)</p> <p>標頭檔：stdlib.h</p> <p>功能：使用 malloc() 函式動態配置記憶體之後，若不要再使用動態配置的記憶體空間時，可以使用 free() 函式指定要釋放的記憶體指標。</p>
getdate	<p>語法：void getdate(date *d)</p> <p>標頭檔：dos.h</p> <p>功能：此函式用來取得目前系統日期並傳回 date 結構，此結構有 da_year、da_mon、da_day 欄位，分別用來表示系統日期的年、月、日。</p> <p>簡例：假設目前系統日期是 2021/4/6，則：</p> <pre>struct date d; /* 宣告 date 結構，其結構變數名稱為 d */ getdate(&d); printf("%d", d.da_year); /* 印出 "2021"，da_year 為年 */ printf("%d", d.da_mon); /* 印出 "4"，da_mon 為月 */ printf("%d", d.da_day); /* 印出 "6"，da_day 為日 */</pre>

函式	功能說明
time	<p>語法：time_t time(time_t *t)</p> <p>標頭檔：time.h</p> <p>功能：取得格林威治時間 1970 年 1 月 1 日 00:00:00 到目前系統時所經過的秒數，然後再將該秒數放到 t 指標所指向的記憶體位址。</p>
clock	<p>語法：clock_t clock(void)</p> <p>標頭檔：time.h</p> <p>功能：傳回程式執行時所經過的時間。通常都是將 clock 函式的傳返回值除以 CLK_TCK 來取得程式執行時所經過的秒數。</p>