

## CHAPTER 8 Strings and Vectors 字串和向量

Important English: string 字串, class 類別

8.1 An Array Type for Strings: also called C-strings; a C-string is an array

承襲自 C 語言的字串, 是 **array**, 叫做 **C-string**

8.2 The Standard string Class: also called strings; they are from C++ string class

C++語言的標準字串, 是 **class**, 叫做 **string**

8.3 Vectors:

an array cannot extend its size once it is declared;

a vector is used to overcome this problem

向量, 目的在彌補 array 一旦宣告後就不能擴大其 size 的缺點

After learning Chapter 8, you will be able to do the following. 學會 Chap 8 你能做什麼.

- ◆ Programming Project 1: 將「空白」和「大小寫」改正確  
Input: **the Answer to life, the Universe, and everything IS 42.**  
Output: **The answer to life, the universe, and everything is 42.**
- ◆ Programming Project 2: 輸出這行文字中有多少個 words, 以及每個 letter 出現幾次  
Input: **I say Hi.**  
Output: **3 words**  
**1 a**  
**1 h**  
**2 i**  
**1 s**  
**1 y**
- ◆ Programming Project 4: 將人名改成 Last Name, First Name Middle Initial  
Input: **Mary Average User**  
Output: **User, Mary A.**  
Input: **Mary A. User**  
Output: **User, Mary A.**  
Input: **Mary A User** 如果 middle initial 沒有加句點, 你要加上  
Output: **User, Mary A.**  
Input: **Mary User** 可以沒有 middle name or middle initial  
Output: **User, Mary**
- ◆ Programming Project 5: 將 4-letter words 都改成 “love”; 注意大小寫  
Input: **I hate you, you dodo!**  
Output: **I love you, you love!**  
Input: **John will run home.**  
Output: **Love love run love.**

- ◆ Programming Project 6: 將 digits 都改成 the character 'x'; John17 are not digits  
 Input: **My userID is john17 and my 4 digit pin is 1234 which is secret.**  
 Output: **My userID is john17 and my x digit pin is xxxx which is secret.**
- ◆ Programming Project 7: 將男女性的字眼都改成中性的字眼  
 Input: **See an advisor, talk to him, and listen to him.**  
 Output: **See an advisor, talk to her or him, and listen to her or him.**  
 Note: The **he** in **here** should not be replaced. 注意: here 中的 he 不該被改到

## 8.1 An Array Type for Strings C-strings C-字串

它們是由 C 語言而來。

一個 C-string 是用 an array of char 來儲存，並以 the null character '\0' 表示結束。

若宣告 `char s[10] = "Hi Mom!";` 則記憶體中儲存的會如下。

由於必須儲存 '\0'，所以 C-字串的最長長度會是 array 的最長長度減 1。

They are inherited 繼承 from the C language. A C-string is declared as **an array of char** and the **null character '\0'** is used to mark the end of a C-string.

For example, if we declare `char s[10] = "Hi Mom!";` then the array elements in the memory will be as follows. The maximum length of a C-string is the maximum length of it array - 1.

s[0]	s[1]	s[2]	s[3]	s[4]	s[5]	s[6]	s[7]	s[8]	s[9]
H	i		M	o	m	!	\0	?	?

**'\0'** is called the **null character**. `s` is a **C-string variable**.

The followings are some details about C-strings. C-strings 的細節 (不要死記)

- A C-string is declared as an array of characters. 宣告方式和 array 相同
- A C-string variable uses '\0' to mark its end. 用 '\0' 表結束
- You may initialize a C-string variable when you declare it. 宣告時, 可以給初值  
 For example, `char my_c_string[11] = "Hi there.";`
- You may omit the array size. 可以省略 array size  
 For example, `char short_string[ ] = "abc";`  
 is equivalent to 全等於 `char short_string[4] = "abc";`

會多給 1

If you omit the number inside [ ], the C-string variable will be given a size one character longer than the length of the C-string.

- Be sure you do not confuse the following initializations. They are not equivalent.

A C-string variable 和 an array of characters 不相同。所以下面兩個不相同

```
char short_string[ ] = "abc";           //the size is 4 and '\0' is stored
char short_string[ ] = {'a', 'b', 'c'}; //the size is only 3 and '\0' is not stored
```

- A C-string is an array, so you can use it just like any other array. For example, the following statements will change the C-string to a C-string of *the same length* consisting of all 'X' characters.

由於 C-string 是 array, 你可以對它做你對 array 的動作

下面的程式碼會將字串的內容都改成同長度的 XX...X

```
int i = 0;
while (my_string[i] != '\0')
{
    my_string[i] = 'X';
    i++;
}
```

- You should be very careful not to destroy the '\0' in a C-string. In the following example, '\0' is destroyed. 由於 C-string 是用 '\0' 表示結束, 所以千萬不要不小心毀掉 '\0'.  
下面就犯了這個錯

```
char happy_string[7] = "DoBeDo";
happy_string[6] = 'Z';           '\0' is destroyed
```

## ■ PITFALL (陷阱) Using = and == with C-string Variables

雖然在宣告 C-string variable 時, 可以用 = 來給初值, 但是在別的地方, 卻不能用 = 來 assign(給定)C-string variable 的值, 因為它的本質是陣列

You can use = when initialize a C-string variable, but you cannot use = to assign a value to a C-string variable.

```
So    char my_sting[10] = "Hello";    is legal 合法
but   char my_sting[10];
      my_string = "Hello";           is illegal 不合法
```

## ■ Some Predefined C-String Functions 內建函數

**strcpy(s1, s2)** Copy s2 to s1. 並不會檢查 s1 的長度是否足夠存入 s2

**strncpy(s1, s2, Limit)** The same as strcpy except that at most **Limit** characters are copied.

**strcat(s1, s2)** Append s2 to the end of s1. 並不會檢查 s1 的長度是否夠長

**strncat(s1, s2, Limit)** The same as strcat except that at most **Limit** characters are appended.

由於 C-strings 是來自於 C 語言，而 C 語言不會做太多的檢查，因此使用 predefined C-string functions 時，必須很小心 When you use the predefined C-string functions, **you have to know the cautions** listed below.

- **strcpy(s1,s2) and strcat(s1,s2) will not check if s1 is large enough to hold the result.**

- When using strcat, s1 must be a C-string variable. 用 strcat 時, s1 必須是 C-字串變數

Thus strcat(“I am”, “ your teacher.”); is not allowed. 不允許

You need to be careful to account for blanks when concatenating C-strings.

“空白”也要留意 For example,

```
char string_var[20] = “The rain”;
```

```
strcat(string_var, “in Spain”);
```

Then string\_var will become “**The rainin Spain**”, not “The rain in Spain”.

總之, C-strings 並不會為你做太多的檢查. 你必須小心

**strlen(s)**

Return the length of a. Note that **‘\0’ is not counted in the length.**

傳回 s 的長度. ‘\0’ 不算在長度內

**strcmp(s1, s2)**

Compare s1 and s2 according to lexicographic order.

以 lexicographic order (字母序) 來比較, 一個一個字元拿來比

**strncmp(s1, s2, Limit)**

The same as strcmp except that at most **Limit** characters are compared.

strcmp returns a **negative** value, a **positive** value, or **zero**, depending on whether s1 is **lexicographically** less, greater, or equal to s2.

- If s1 is less than s2, then strcmp returns a negative value, which means true.

若 s1 小, 則傳出 a **negative** value, 相當於 **true**.

- If s1 is greater than s2, then strcmp returns a positive value, which means true.

若 s1 大, 則傳出 a **positive** value, 相當於 **true**.

- If s1 is equal to s2, then strcmp returns zero, which means false.

若 s1, s2 相等, 則傳出 **zero**, 相當於 **false**.

**Thus if s1 and s2 are the same, the Boolean value of strcmp(s1,s2) is false; if s1 and s2 are not the same, the Boolean value of strcmp(s1,s2) is true.**

因此只要 s1 和 s2 相同, 就得 false; 不相同, 就得 true. 千萬不要記反了喔!

## ■ **SELF-TEST EXERCISES** 1-12 全部都要做, 每題書上都有答案

- You can use **strcpy** to assign a value to a C-string variable. For example,

```
strcpy(my_string, “Hello”);
```

- You cannot use == to test if two C-strings are the same; you can use strcmp to do this. 你不能用 == 來檢查兩個 C-strings 是否相等; 但是可以用 strcmp 函數.

For example,

```
if strcmp(c_string1, c_string2)
    cout << "They are NOT the same.";
else
    cout << "They are the same.";
```

- To use strcpy or strcmp, you need to **#include <cstring>** or **#include <string.h>** or **#include <cstring.h>** 要寫那一個 depends on your machine

## ■ C-string Output

你可以用 **cout <<** 來 output C-strings. You may use **cout <<** to output C-strings.

上機 1

```
#include <iostream>
using namespace std;
const int N=50;

int main()
{
    char news[N] = "I wish you happy 2014!";
    cout << news << endl;

    system("PAUSE");
    return 0;
}
```

## ■ C-string Input

**\*\*\*IMPORTANT:** 你可以用 **cin >>** 來 input C-strings. 但是: **whitespace (blanks 空白, tabs, and line breaks 跳行) 全都會被跳掉.** 而且, 用 **cin >>** 來做輸入時, C-string variable 在讀到下一個 **whitespace** 時就不會再讀. You may use **cin >>** to input C-strings. However, **all whitespace (blanks, tabs, and line breaks) are skipped** when C-strings are read this way. Moreover, **each reading of input stops at the next space or line break.**

上機 2

```
#include <iostream>
using namespace std;
const int N=50;

int main()
{
    char news[N];
    cout << "Enter some input:\n";
    cin >> news;
    cout << news << endl;
```

Enter: I wish you happy 2014!

```

    system("PAUSE");
    return 0;
}

```

### 上機 3

```

#include <iostream>
using namespace std;
const int N=50;

```

```

int main( )
{

```

```

    char a[N], b[N];
    cout << "Enter some input:\n";
    cin >> a >> b;
    cout << a << b << "END OF OUTPUT \n";

```

Enter: Do be do to you!

```

    system("PAUSE");
    return 0;
}

```

You get:  
DobeEND OF OUTPUT

**\*\*\*IMPORTANT:** 如果你連 whitespace 也不想跳掉, 那可以改用 **getline**. If you do not want to skip whitespace, you can use **getline**.

### 上機 4

```

#include <iostream>
using namespace std;
const int N=50;

```

```

int main( )
{

```

```

    char a[N];
    cout << "Enter some input:\n";
    cin.getline(a, 20);
    cout << a << "END OF OUTPUT \n";

```

Enter: Do be do to you!

```

    system("PAUSE");
    return 0;
}

```

You get:  
Do be do to you!END OF OUTPUT

### \*\*\*IMPORTANT:

- getline 是讀一整行. 當讀到(1)一行結束, 或(2)第二個參數規定的字數 - 1 就停. When **getline** is executed, the entire line is read. **The reading stops when (1) the line ends or (2) the number of characters given by the second argument has been filled.**
- 注意:要保留一個位置給 null character '\0'. Notice that **19, not 20**, characters are read into the C-string variable. This is because the null character '\0' fills one array position. Try: 20 → 19 → 18 → 17 → 16

- 你當然也可以由檔案中輸入 C-strings, 或輸出 C-strings 到檔案中. `cin` can be replaced by any *file input* and `cout` can be replaced by any *file output*.

## SELF-TEST EXERCISES 13, 14. 每題書上都有答案

### ■ **C-string-to-Number Conversions** C-字串轉換數字 **and Robust Input** 完備的輸入

- **atoi** (ascii to integer)  
The C-string "1234" and the number 1234 are not the same. "1234" is a string and 1234 is a number. You can use the function **atoi** to convert a C-string into an *integer*. For example, **int x = atoi("1234");** will make x to be the integer **1234**.  
However, if the argument does not correspond to an int value, then atoi returns 0. Thus **int x = atoi("#37");** will make x to be the integer **0**.  
atoi("1234")會得到 1234, atof("12.37")會得到 12.37. 注意: atoi 不接受 non-digit characters, 遇到這種 characters, 它會傳回 0. 例如: atoi("#37")會得到 0.
- **atol** (ascii to long integer)  
The function **atol** will convert a C-string into a *long integer*.
- **atof** (ascii to double)  
The function **atof** will convert a C-string into a *double*.  
For example, **double y = atof("12.37");** will make x to be the double **12.37**.
- **#include <cstdlib>**  
To use atoi, atol, or atof, your program must have: `#include <cstdlib>`

In the following, we will talk about **robust input**.

完備的輸入包容 user 在輸入時的各種習慣.

例如: user 在輸入 **1000** 元時, 有下列的可能, 然而全都是合理的.

**1000    \$1000    \$1,000**

- |  |                               |
|--|-------------------------------|
| <b>(1) Read the input as a string of characters.</b>                       | <u>將 input 當作 string 全部讀入</u> |
| <b>(2) Edit the string, remove non-digit characters like '\$' and ','.</b> | <u>去掉不要的</u>                  |
| <b>(3) Convert the string to a number.</b>                                 | <u>將 string 改成數字</u>          |

用(1)~(3)就可以包容 user 在輸入時的各種習慣.

要去掉 '\$' 和 ',' 等 non-digit characters, isdigit 函數可以派上用場.

用到 isdigit, 必須#include <cctype>

所有 non-digit characters 除掉後, 可用 **atoi, atol, atof** 將 string 轉換成 number.

You can use **isdigit** to remove non-digit characters.

To use **isdigit**, your program must have: **#include <cctype>**

After removing all non-digit characters, you can use atoi, atol, and atof to convert the strings into their corresponding numbers.

## C-strings to Integers

If input is	\$ 100	then output will be:	100
	100		100
	99%		99
	23% &&5 *12		23512

The following program contains a function **read\_and\_clean** that reads a line of input and discards all characters other than the digits '0' through '9'. 讀一行輸入，去掉所有 non-digit characters.

```
#include <iostream>
using namespace std;
#include <cstdlib>
#include <cctype>
```

```
void read_and_clean(int& n) //Read a line of input, discard all non-digit characters,
{ //and converts the string to an integer called n.
    const int SIZE = 6;
    char digit_string[SIZE];
    char next;
    cin.get(next);
    int i = 0;
    while (next != '\n')
    {
        if ( isdigit(next) && (i < SIZE - 1) )
        {
            digit_string[i] = next;
            i++;
        }
        cin.get(next);
    }
    digit_string[i] = '\0';
    n = atoi(digit_string);
}
```

```
void new_line() //Discard all the input remaining on the input line, 包括最後的 '\n'.
{
    char c;
    do
        cin.get(c);
    while (c != '\n');
}
```

```
int main()
{
    int n;
    char ans;
    do
    {
        cout << "Enter an integer and press Enter: ";
        read_and_clean(n);
        cout << "The integer is " << n << endl;
        cout << "Again? (yes/no): ";
        cin >> ans;
        new_line();
    } while ((ans != 'n') && (ans != 'N'));
    system("PAUSE");
    return 0;
}
```

## Check list and testing data for program05 測試完後繳回此單

student id: \_\_\_\_\_

TA: 羅、涂、姚、紀 [務必要圈出來]

- Accomplish Note 1: **【看 e3 作業繳交區，確認有繳 2 個檔案，確認檔名正確】**  
P05\_0212204.cpp  
p05\_0212204.txt
- Accomplish Note 2: **【看 code】 six comments (at the beginning of the program)**  
//File Name: p05\_0212204.cpp  
//Author: Your Name  
//Email Address: your@yourmachine.bla.bla  
//Assignment Number: 5  
/\*Description: a program that computes the cost of a long-distance call.\*/  
//Last Changed: **November 12, 2013**
- Accomplish Note 3: **【看.txt file，確認有(1)(2)(3)】 a report (a .txt file) containing:**
  - (1) the purpose of this program,
  - (2) the difficulty you encountered when writing this program,
  - (3) anything special in your program.
- Output correct answers: **【全都要測】**

Test data				
# of test data	day	start time	call length	cost
1	Mo	09:00	30	240
2	Tu	18:30	50	250
3	wE	08:50	40	290
4	th	16:33	60	381
5	FR	23:55	25	85
6	Sa	08:45	60	180
7	Su	23:40	100	460

- 【看 code】** Make your program readable.  
**【1. blank lines 適當空行, 2. comments 適當加註解, 3. Indent 該凹就凹,該凸就凸】**
- 【助教 run】** Remind the user what kind of data should be inputted and what the output is.
- 【助教 run】** Allow the calculation to be repeated as often as possible.
- 【助教 run】** Provide additional functions.

## Check list and testing data for program07 測試完後繳回此單

student id: \_\_\_\_\_

TA: 羅、涂、姚、紀 [務必要圈出來]

- Accomplish Note 1: [【看 e3 作業繳交區，確認有繳 2 個檔案，確認檔名正確】](#)

P07\_0212204.cpp

p07\_0212204.txt

- Accomplish Note 2: [【看 code】](#) six comments (at the beginning of the program)

//File Name: p07\_0212204.cpp

//Author: Your Name

//Email Address: your@yourmachine.bla.bla

//Assignment Number: 7

/\*Description: Output a two-column list. \*/

//Last Changed: December 17, 2013

- Accomplish Note 3: [【看.txt file，確認有\(1\)\(2\)\(3\)】](#) a report (a .txt file) containing:

- (1) the purpose of this program,
- (2) the difficulty you encountered when writing this program,
- (3) anything special in your program.

- [【看 code】](#) Make your program readable. [【1.適當空行, 2.適當加註解, 3.該凹就凹,該凸就凸】](#)

- Output correct answers: [【全都要測】](#)

Test data										
Input 1	-12 3 -12 4 1 1 -12 1 -1 1 2 3 4 2 3 -12 end_value									
Input 2	5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 end_value									
Input 3	1 2 3 4 5 6 7 8 9 10 11 12 13 14 end_value									
Input 4	0 8 -5 8 -2 0 -5 9 -2 9 7 3 -3 3 7 -3 end_value									
Input 5	1000 -9999 -99 1000 1000000 -99 -9999 9999 -1000 1000000 end_value									
# of test data	1		2		3		4		5	
Output	N	Count	N	Count	N	Count	N	Count	N	Count
	4	2	5	15	14	1	9	2	1000000	2
	3	3			13	1	8	2	9999	1
	2	2			12	1	7	2	1000	2
	1	4			11	1	3	2	-99	2
	-1	1			10	1	0	2	-1000	1
	-12	4			9	1	-2	2	-9999	2
					8	1	-3	2		
					7	1	-5	2		
					6	1				
					5	1				
				4	1					
				3	1					
				2	1					
				1	1					

- [【助教 run】](#) Remind the user what kind of data should be inputted and what the output is.
- [【助教 run】](#) Allow the calculation to be repeated as often as possible.
- [【助教 run】](#) Provide additional functions.