

序

C++ 全方位學習一書入選 100 狀元閱讀書單科學類排名第 15，也是唯二入選的程式設計類書，因此筆者決定重新改版，以便造福更多想學習 C++ 程式設計的學生。**C++ 全方位學習第二版**保留**第一版**的編排特色，而刪除比較不實用的大型程式範例與標準範本程式庫，然後在各章習題中加入選擇題，使其更加適合用於技職院校及大學教學。

C++ 全方位學習第二版是針對專科與大學教學進度而編寫的，內容由淺入深、由簡而繁的方式帶領讀者學習 C++ 的基本程式設計、結構化程式設計、物件導向程式設計、與應用程式設計。同時，也適合作為初學 C++ 程式設計者的自修參考書。內容包括 C++ 語法、語法說明、簡單範例、與完整程式，C++ 語法方便讀者查詢敘述與函數的正確格式，語法說明列出敘述與函數的功能與參數的用法，簡單範例教導讀者如何使用敘述與函數，完整程式則實際將敘述與函數應用於程式中。

本書適用軟體：書中所有的語法、範例與程式都是使用 ANSI/ISO C++ 的標準，但也兼顧到 Visual C++ 的相容性，因此所有範例程式除了使用“Dev-C++ 5.0 beta 9.2 (4.9.9.2) (9.0 MB) with Mingw/GCC 3.4.2”編譯、連結與執行過之外，還在 Visual C++ 2012、Visual C++ 2013 環境下編譯、連結與執行過，讀者可以放心使用任何以 ANSI/ISO 為標準的 C++ 編譯器。

致謝：感謝各位讀者與各級學校師生數十年來給予筆者的肯定，您們數十年的支持與愛護是筆者繼續製作優良作品的原動力，希望您能再次給予**C++ 全方位學習第二版**一書支持與鼓勵。感謝碁峰資訊廖董事長暨全體同仁多年來的支持。

古頤榛 謹識

chapter

9

字元與字串

本章重點▶

- 9.1 C 型態字串函數
- 9.2 C 型態字元函數
- 9.3 字串與數值轉換函數
- 9.4 C++ 字串類別
- 9.5 習題

9.1 C 型態字串函數

C++ 提供二種字串型態：一是使用 `char` 定義的 C 型態字串，另一則是使用 `string` 定義的 C++ 型態字串。C 型態字串在 7.4 節已經介紹過，本節主要討論應用於 C 型態字串的函數，如取得字串長度 (`strlen`)、複製字串 (`strcpy`)、比較字串 (`strcmp`)、串接字串 (`strcat`) 等。至於 C++ 型態字串則將於 9.4 節討論。

9.1.1 取得字串長度 `strlen`



```
#include <string>
strlen (指定字串)
```


- **strlen** 函數是計算並傳回指定字串的位元組 (byte) 數。strlen 函數包含於 `string` 標題檔中，所以使用前須先插入 `string` 檔。

下面範例是取得字串長度。先宣告 `fixstr` 字串，然後利用 `strlen(fixstr)` 函數取得字串長度。

```
char fixstr[80] = "Ctype String";           //定義C型態字串
int fixlen = strlen(fixstr);               //取得字串長度
```

◎ 程式 9-01：取得輸入字串長度

```
1. //檔案名稱:d:\C++09\C0901.cpp
2. #include <iostream>
3. #include <cstring>                               //插入字串標題檔
4. using namespace std;
5.
6. int main(int argc, char *argv[])
7. {
8.     char instr[80];
9.     cout << "請輸入字串:";                       //顯示訊息字串
10.    cin.getline (instr, 80, '\n');                 //取得輸入字串列
11.    cout << "字串長度為:" << strlen(instr); //顯示字串長度
12.    cout << endl;
13.    system("PAUSE");
14.    return EXIT_SUCCESS;
15. }
```

 程式輸出

請輸入字串：**Ctype String** **Enter**
字串長度為：12

9.1.2 複製字串 strcpy



```
#include <string>
strcpy (目的字串, 來源字串)
```

- **strcpy** 函數是將來源字串複製到目的字串，其中目的字串必須是記憶體變數。且來源字串的長度必須小於目的字串所宣告的長度，否則複製時將會覆蓋到其他記憶體中的資料，嚴重的可能會造成當機。**strcpy** 函數包含於 **string** 標題檔中，所以使用前須先插入 **string** 檔。

下面範例是複製字串。先宣告 **source** 與 **target** 字串，起始 **source** 字串後利用 **strcpy(target, source)** 函數將 **source** 字串複製到 **target** 字串中。

```
char source[80], target[80];
source[80] = "Ctype String";           //起始 source 字串
strcpy(target, source);                //複製字串
```



程式 9-02：複製輸入字串

```
1. //檔案名稱：d:\C++09\C0902.cpp
2. #include <iostream>
3. #include <cstring>           //插入字串標題檔
4. using namespace std;
5.
6. int main(int argc, char *argv[])
7. {
8.     char source[80], target[80];
9.     cout << "請輸入來源字串：";           //顯示訊息字串
10.    cin.getline (source, 80, '\n');       //取得來源字串
11.    strcpy(target, source);              //複製字串
12.    cout << "複製後目的字串：" << target; //顯示目的字串
13.    cout << endl;
14.    system("PAUSE");
15.    return EXIT_SUCCESS;
16. }
```

程式輸出

請輸入來源字串：Ctype String

複製後目的字串：Ctype String

9.1.3 比較字串 strcmp



```
#include <string>
strcmp (字串 1, 字串 2)
```

- **strcmp** 函數將比較字串 1 與字串 2 的內容，並傳回代碼說明如下。
 - 傳回正數：表示字串 1 的 ASCII 碼大於字串 2 的 ASCII 碼。
 - 傳回零：表示二字串相等。
 - 傳回負數：表示字串 1 的 ASCII 碼小於字串 2 的 ASCII 碼。
- **strcmp** 函數包含於 **string** 標題檔中，所以使用前須先插入 **string** 檔。

下面範例是利用比較字串檢查輸入密碼。先宣告並起始 **password** 字串，再宣告 **instring** 字串，然後讀取鍵盤整列輸入（包含空白），比較輸入字串與 **password** 字串是否相等，若相等則 **flag==0**。

```
char password[80] = "2012";           //定義並啟始密碼
char instring[80];
cin.getline (instring, 80, '\n');    //輸入字串
int flag = strcmp(password, instring); //flag=比較字串結果
```

程式 9-03：檢查輸入密碼

```
1. //檔案名稱：d:\C++09\C0903.cpp
2. #include <iostream>
3. #include <cstring>           //插入字串標題檔
4. using namespace std;
5.
6. int main(int argc, char *argv[])
7. {
8.     char password[80] = "2012"; //定義並啟始密碼
9.     char instring[80];
10.    cout << "您有 3 次機會，";
11.    for (int i = 1; i <=3; i++) //輸入密碼迴圈
12.    {
13.        cout << "請輸入密碼：";
```

```

14.     cin.getline (instring, 80, '\n');    //取得來源字串
15.     int flag = strcmp(password, instring); //比較字串
16.     if (flag == 0)
17.     {
18.         cout << "恭喜您！密碼正確。";    //顯示目的字串
19.         break;                            //中斷迴圈
20.     }
21.     else
22.     {
23.         if (i != 3)                        //以計數值決定，
24.             cout << "還有 " << 3-i << " 次機會，"; //顯示的字串
25.         else
26.             cout << "對不起！沒機會了。";
27.     }
28. }
29. cout << endl;
30. system("PAUSE");
31. return EXIT_SUCCESS;
32. }

```

📌 程式輸出

您有 3 次機會，請輸入密碼：1999
 還有 2 次機會，請輸入密碼：2000
 還有 1 次機會，請輸入密碼：2001
 對不起！沒機會了。

📌 程式輸出

您有 3 次機會，請輸入密碼：2010
 還有 2 次機會，請輸入密碼：2011
 還有 1 次機會，請輸入密碼：2012
 恭喜您！密碼正確。

9.1.4 串接字串 strcat



```

#include <string>
strcat(字串 1, 字串 2)

```

- **strcat** 函數是將字串 2 串接到字串 1 之後，其中字串 1 必須是記憶體變數。

- `strcat` 函數包含於 `string` 標題檔中，所以使用前須先插入 `string` 檔。

下面範例是利用串接字串將 `first name` 與 `last name` 串接在一起。先宣告並起始 `first`、`last` 與 `full` 字串，利用 `strcat` 函數將 `first`、空白、與 `last` 字串串接到 `full` 字串中。

```
char first[80] = "Sharon";
char last[80] = "Stone";
char full[160] = "";
strcat(full, first);           //full="Sharon"
strcat(full, " ");           //full="Sharon "
strcat(full, last);           //full="Sharon Stone"
```



程式 9-04：串接輸入字串

```
1. //檔案名稱：d:\C++09\C0904.cpp
2. #include <iostream>
3. #include <cstring>           //插入字串標題檔
4. using namespace std;
5.
6. int main(int argc, char *argv[])
7. {
8.     char first[80], last[80], full[160] = "";
9.     cout << "請輸入英文名字 (first name) : "; //顯示訊息字串
10.    cin.getline (first, 80, '\n');           //取得名字字串
11.    cout << "請輸入英文姓氏 (last name) : "; //顯示訊息字串
12.    cin.getline (last, 80, '\n');           //取得姓氏字串
13.    strcat(full, first);                     //附加名字字串
14.    strcat(full, " ");                       //附加空白字串
15.    strcat(full, last);                      //附加姓氏字串
16.    cout << "您的全名為：" << full << endl; //顯示全名字串
17.    system("PAUSE");
18.    return EXIT_SUCCESS;
19. }
```



程式輸出：粗體字表示鍵盤輸入

```
請輸入英文名字 (first name) : Sharon Enter
請輸入英文姓氏 (last name) : Stone Enter
您的全名為：Sharon Stone
```

9.2 C 型態字元函數

cctype 標題檔包含 C 型態的字元函數，如大寫轉成小寫（`tolower`）、與小寫轉成大寫（`toupper`）等函數。

9.2.1 大寫轉換小寫 `tolower`



```
#include <cctype>
tolower (字元)
```

- `tolower` 函數將測試參數字元是否為大寫字母（A~Z），若是則將該字元轉換成小寫字母（a~z）後傳回。`tolower` 函數只將大寫轉換成小寫，而不會影響數字、符號或其他字元。
- `tolower` 函數包含於 `cctype` 標題檔中，所以使用前須先插入 `cctype` 檔。

下面範例是將字串中的大寫字元轉換成小寫字元。以 `strlen` 函數取得 `flexStr` 字串長度，並以此長度作為迴圈重複的次數，在迴圈中每次讀取字串中的一個字元，並利用 `tolower` 函數將該字元轉換成小寫字元。

```
char flexStr[] = "Success is never ending. Failure is never final.";
int len = strlen(flexStr); //取得字串長度
for (int i = 0; i <= len; i++) //轉成小寫迴圈
    flexStr[i] = tolower(flexStr[i]); //轉成小寫字元
```

9.2.2 小寫轉換大寫 `toupper`



```
#include <cctype>
toupper (字元)
```

- `toupper` 函數將測試參數字元是否為小寫字母（a~z），若是則將該字元轉換成大寫字母（A~Z）後傳回。`toupper` 函數只將小寫轉換成大寫，而不會影響數字、符號或其他字元。
- `toupper` 函數包含於 `cctype` 標題檔中，所以使用前須先插入 `cctype` 檔。

下面範例是將字串中的小寫字元轉換成大寫字元。以 `strlen` 函數取得 `flexStr` 字串長度，並以此長度作為迴圈重複的次數，在迴圈中每次讀取字串中的一個字元，並利用 `toupper` 函數將該字元轉換成大寫字元。

```
char flexStr[] = "Success is never ending. Failure is never final.";
int len = strlen(flexStr); //取得字串長度
for (int i = 0; i <= len; i++) //轉成小寫迴圈
    flexStr[i] = toupper(flexStr[i]); //轉成大寫字元
```

程式 9-05：字串大小寫轉換

```
1. //檔案名稱:d:\C++09\C0905.cpp
2. #include <iostream>
3. #include <cstring> //插入字串標題檔
4. #include <cctype> //插入轉換字元檔
5. using namespace std;
6.
7. int main(int argc, char *argv[])
8. {
9.     char flexStr[] = "Success is never ending. Failure is never
    final.";
10.     cout << "字串轉換前:" << flexStr << endl; //顯示轉換前字串
11.     int len = strlen(flexStr); //取得字串長度
12.
13.     for (int i = 0; i <= len; i++) //轉成小寫迴圈
14.         flexStr[i] = tolower(flexStr[i]); //轉成小寫字元
15.     cout << "轉換小寫後:" << flexStr << endl; //顯示轉換後字串
16.
17.     for (int i = 0; i <= len; i++) //轉成大寫迴圈
18.         flexStr[i] = toupper(flexStr[i]); //轉成大寫字元
19.     cout << "轉換大寫後:" << flexStr << endl; //顯示轉換後字串
20.     system("PAUSE");
21.     return EXIT_SUCCESS;
22. }
```

程式輸出

```
字串轉換前:Success is never ending. Failure is never final.
轉換小寫後:success is never ending. failure is never final.
轉換大寫後:SUCCESS IS NEVER ENDING. FAILURE IS NEVER FINAL.
```

9.3 字串與數值轉換函數

字串形式的數值（“3.14159”）是不能當做算數運算的資料，所以 C++ 提供字串與數值間的轉換函數，例如字串轉成浮點數（`atof`）、字串轉成整數（`atoi`）、字串轉成長整數（`atol`）等函數，以及整數轉成字串（`itoa`）的函數。

9.3.1 轉成浮點數值 `atof`



```
#include <cstdlib>
atof (字串)
```

- `atof` 函數將字串參數轉換成浮點數值。`atof` 函數包含於 `cstdlib` 標題檔中，所以使用前須先插入 `cstdlib` 檔。

下面範例是將字串轉成浮點數值。利用 `atof` 函數將字串中的數值字元轉換成浮點數值。

```
s = "-1998.12E-25 "; //定義字串
x = atof( s ); //轉浮點數
x=-1.99812e-22
```

9.3.2 轉成整數值 `atoi`



```
#include <cstdlib>
atoi (字串)
```

- `atoi` 函數將字串參數轉換成整數數值。`atoi` 函數包含於 `cstdlib` 標題檔中，所以使用前須先插入 `cstdlib` 檔。

下面範例是將字串轉成整數數值。利用 `atoi` 函數將字串中的數值字元轉換成整數數值。

```
s = " 686 pigs "; //定義字串
i = atoi( s ); //轉換成短整數,i=686
```

9.3.3 轉成長整數值 `atol`



```
#include <cstdlib>
atol (字串)
```


- `atol` 函數將字串參數轉換成 `long` 型態的長整數數值。 `atol` 函數包含於 `cstdlib` 標題檔中，所以使用前須先插入 `cstdlib` 檔。

下面範例是將字串轉成長整數數值。以 `atol` 函數將字串中的數值字元轉換成 `long` 型態的長整數數值。

```
s = " 98686 dollars";           //定義字串
l = atol( s );                 //轉換成長整數,l=98686
```

程式 9-06：字串轉換數值

```
1. //檔案名稱:d:\C++09\C0906.cpp
2. #include <iostream>
3. #include <iomanip>
4. #include <cstdlib>           //轉換數值標題檔
5. using namespace std;
6.
7. int main(int argc, char *argv[])
8. {
9.     char *s; double x; int i; long l;
10.    setiosflags(ios::fixed|ios::right); //設定輸出格式
11.    cout << setw(7) << "字串\t" << setw(23) << "    數值" << endl;
12.
13.    s = "-1998.12E-25 ";       //定義字串
14.    x = atof( s );            //轉換成浮點數
15.    cout << setw(15) << s << "\t" << setw(15) << x << endl;
16.
17.    s = " 686 pigs ";         //定義字串
18.    i = atoi( s );           //轉換成短整數
19.    cout << setw(15) << s << "\t" << setw(15) << i << endl;
20.
21.    s = " 98686 dollars";     //定義字串
22.    l = atol( s );           //轉換成長整數
23.    cout << setw(15) << s << "\t" << setw(15) << l << endl;
24.    system("PAUSE");
25.    return EXIT_SUCCESS;
26. }
```

 程式輸出

字串	數值
-1998.12E-25	-1.99812e-22
686 pigs	686
98686 dollars	98686

9.3.4 整數轉成字串 itoa



```
#include <cstdlib>
itoa (整數數值, 字串變數, 數系基底)
```

- **itoa** 函數將整數參數轉換成字串後存入指定的字串變數中。
- **整數數值**是要被轉換的整數參數。
- **字串變數**是存放轉換後字串的字串變數。
- **數系基底**是整數數值的數字系統，8 代表八進位，10 表示十進位，16 表示十六進位。
- itoa 函數包含於 `cstdlib` 標題檔中，所以使用前須先插入 `cstdlib` 檔。

下面範例是將數值轉成轉成字串。分別以 `itoa` 函數將數值轉成八進位型態的字串，並存入 `intArray` 字串中。以 `itoa` 函數將數值轉成十進位型態的字串，並存入 `intArray` 字串中。以 `itoa` 函數將數值轉成十六進位型態的字串，並存入 `intArray` 字串中。

```
char intArray[10];
itoa(1234, intArray, 8);           //1234 轉成字串"2322"
itoa(1234, intArray, 10);         //1234 轉成字串"1234"
itoa(1234, intArray, 16);         //1234 轉成字串"4d2"
```



程式 9-07：數值轉成八、十、十六進位字串

```
1. //檔案名稱:d:\C++09\C0907.cpp
2. #include <iostream>
3. #include <iomanip>
4. #include <cstdlib>           //轉換數值標題檔
5. using namespace std;
6.
7. int main(int argc, char *argv[])
```

```

8.  {
9.    char intArray[10];
10.
11.    itoa(1234, intArray, 8);           //1234 轉成字串"2322"
12.    cout << "1234 轉成八進位型態字串:\\" << intArray << "\\n";
13.
14.    itoa(1234, intArray, 10);        //1234 轉成字串"1234"
15.    cout << "1234 轉成十進位型態字串:\\" << intArray << "\\n";
16.
17.    itoa(1234, intArray, 16);       //1234 轉成字串"4d2"
18.    cout << "1234 轉成十六進位型態字串:\\" << intArray << "\\n";
19.    system("PAUSE");
20.    return EXIT_SUCCESS;
21. }

```

📌 程式輸出

```

1234 轉成八進位型態字串: "2322"
1234 轉成十進位型態字串: "1234"
1234 轉成十六進位型態字串: "4d2"

```

9.4 C++字串類別

C++ 字串類別是一個抽象的資料型態，它不是 C++ 原本內建的資料型態，如 `int` 或 `char`。C++ 字串類別與字串類別函數是定義於 C++ 的新型標題檔中，而 C 型態的字串標題檔（`cstring`）並沒有定義這些函數。所以使用這些函數以前，必須插入 C++ 新型的標題檔（`string`）。

9.4.1 建立 C++字串



```
#include <string>
```

```

string 物件名稱;           //第一式
string 物件名稱("字串");  //第二式
string 物件名稱 = "字串";  //第三式
string 物件名稱("字元", 長度); //第四式
string 物件名稱(字串物件); //第五式
string 物件名稱(字串物件, 起始, 長度); //第六式

```

- **string** 是建立字串物件的關鍵字。在第一式中 **string** 僅建立物件並未給予物件初值，在第二式與第三式中，**string** 不僅建立物件名稱並給予該物件初值，第四式中物件的初值是重複的字元，第五式是複製另一個字串物件的資料作為新字串物件的初值，第六式擷取另一個字串物件的部分字串作為新字串物件的初值。
- **string** 函數定義於 `<string>` 標題檔中，所以使用前必須插入 `<string>` 標題檔與加入 `using namespace std;` 敘述。

下面範例是 C 型態字串。它們是使用 C 型態的指標或陣列定義字串，第一式是定義並起始字串指標變數，第二式則是定義並起始字串陣列變數。

```
char *name = "JOHN";
char name[20] = "JOHN";
```

下面範例是 C++ 型態字串。它們是使用 C++ 字串類別定義的字串，`string s1`；只宣告字串物件 `s1` 並未起始字串資料，`string s2("JOHN ARCHER")` 宣告並起始 `s2` 物件為“JOHN ARCHER”，`string s3 = "MARY ARCHER"` 宣告並起始 `s3` 物件為“JOHN ARCHER”，`string s4("A", 4)` 宣告並起始 `s4` 物件為“AAAA”，`string s5(s2)` 宣告 `s5` 並指定 `s5` 等於 `s2`，`string s6(s2, 0, 4)` 宣告 `s6` 物件並指定 `s6` 等於 `s2` 物件的前 4 個字元。

```
string s1; //宣告 s1
string s2("JOHN ARCHER"); //宣告 s2 = "JOHN
ARCHER"
string s3 = "MARY ARCHER"; //宣告 s3 = "MARY
ARCHER"
string s4("A", 4); //宣告 s4 = "AAAA"
string s5(s2); //宣告 s5 = "JOHN
ARCHER"
string s6(s2, 0, 4); //宣告 s6 = "JOHN"
```



程式 9-08：建立並顯示字串

```
1. //檔案名稱：d:\C++09\C0908.cpp
2. #include <iostream>
3. #include <string> //C++型態 string 標題檔
4. using namespace std;
5.
6. int main(int argc, char *argv[])
7. {
8.     string s1("c++ string class"); //宣告並起始 s1
```

```

9.     string s2 = s1;
10.    string s3(s1);
11.    string s4(s1, 4, 12);
12.    string s5(s1, 0, 3);
13.    cout << "s1 = " << s1 << endl;           //輸出 s1
14.    cout << "s2 = " << s2 << endl;           //輸出 s2
15.    cout << "s3 = " << s3 << endl;           //輸出 s3
16.    cout << "s4 = " << s4 << endl;           //輸出 s4
17.    cout << "s5 = " << s5 << endl;           //輸出 s5
18.    system("PAUSE");
19.    return EXIT_SUCCESS;
20. }

```

📌 程式輸出

```

s1 = c++ string class
s2 = c++ string class
s3 = c++ string class
s4 = string class
s5 = c++

```

9.4.2 輸入 C++字串



```

#include <iostream>
using namespace std;

getline (cin, 字串物件)

```

- **getline** 是 C++ 型態的 `iostream` 新型標題檔的函數，它可以讀取包含空白的字串。
- **getline** 函數包含於 `iostream` 新型標題檔中，所以使用前必須先插入 `iostream` 檔。

下面範例是使用 `cin` 讀取鍵盤輸入字串，但 `cin` 視空白為分隔符號，所以字串中空白以後的部份將被切除，例如鍵盤輸入“Hello world!”而 `cin` 只能讀取“Hello”。

9.5 習題

選擇題

- _____是取得 C++ 型態字串長度的函數，假設字串名稱為 s1。
 - s1.strlen()
 - s1.length()
 - size(s1)
 - max_size(s1)
- _____是 C 型態比較二字串的函數，假設字串名稱為 s1 與 s2。
 - s1.strcmp(s2)
 - s1.compare(s2)
 - strcmp(s1, s2)
 - compare(s1, s2)
- _____是 C++ 型態比較二字串的函數，假設字串名稱為 s1 與 s2。
 - s1.strcmp(s2)
 - s1.compare(s2)
 - strcmp(s1, s2)
 - compare(s1, s2)
- _____是 C 型態將 s2 字串附加到 s1 字串之後的函數。
 - strcat(s1, s2)
 - concat(s1, s2)
 - append(s1, s2)
 - insert(s1, s2)
- _____是 C++ 型態將 s2 字串附加到 s1 字串之後的函數。
 - s1.strcat(s2)
 - s1.concat(s2)
 - s1.append(s2)
 - s1.insert(s2)
- 假設 C++ 字串 s1 = “word”，_____是將 “l” 插入 s1 的第 3 個字元的敘述。
 - s1.insert(“l”, 3);
 - s1.insert(‘l’, 3);
 - s1.insert(3, “l”);
 - s1.insert(3, ‘l’);

7. 假設 C++ 字串 `s1 = "word"`，_____是以 "a" 取代 `s1` 的第 1 個字元的敘述。
- a) `s1.replace(1, 1, "a");` b) `s1.replace("a", 1, 1);`
c) `replace(s1, 1, "a");` d) `replace(s1, "a", 1);`

實作題

- 寫一個 C++ 程式，將下列字串中的所有小寫英文字母轉換成大寫英文字母，其餘中文與符號不變。
"C++ 學習講堂的內容與程式適合任何 C++ 編譯器，如 Visual C++、Borland C++、GNU C++ 等等。"
- 寫一個 C++ 程式，將下列字串中的所有大寫英文字母轉換成小寫英文字母，其餘中文與符號不變。
"C++ 學習講堂的內容與程式適合任何 C++ 編譯器，如 Visual C++、Borland C++、GNU C++ 等等。"
- 寫一個 C++ 程式，計算字串中的英文單字個數。
 - 在 `main` 函數中，定義一個字串指標，由鍵盤輸入一字串並存入指標位址，然後呼叫並傳遞字串給 `wordCount` 函數，最後輸出英文單字的個數。
 - 定義一個 `wordCount` 函數，接收呼叫敘述傳遞的字串指標參數，然後計算並傳回字串中英文單字的個數。
- 寫一個 C++ 程式，更改字串中的所有英文句子的第一個字元為大寫。
 - 在 `main` 函數中，定義一個字串指標，由鍵盤輸入一字串並存入指標位址，然後呼叫並傳遞字串給 `capital` 函數，最後輸出更改後的字串。
 - 定義一個 `capital` 函數，接收呼叫敘述傳遞的字串指標參數，然後將字串中的所有英文句子的第一個字元改為大寫。