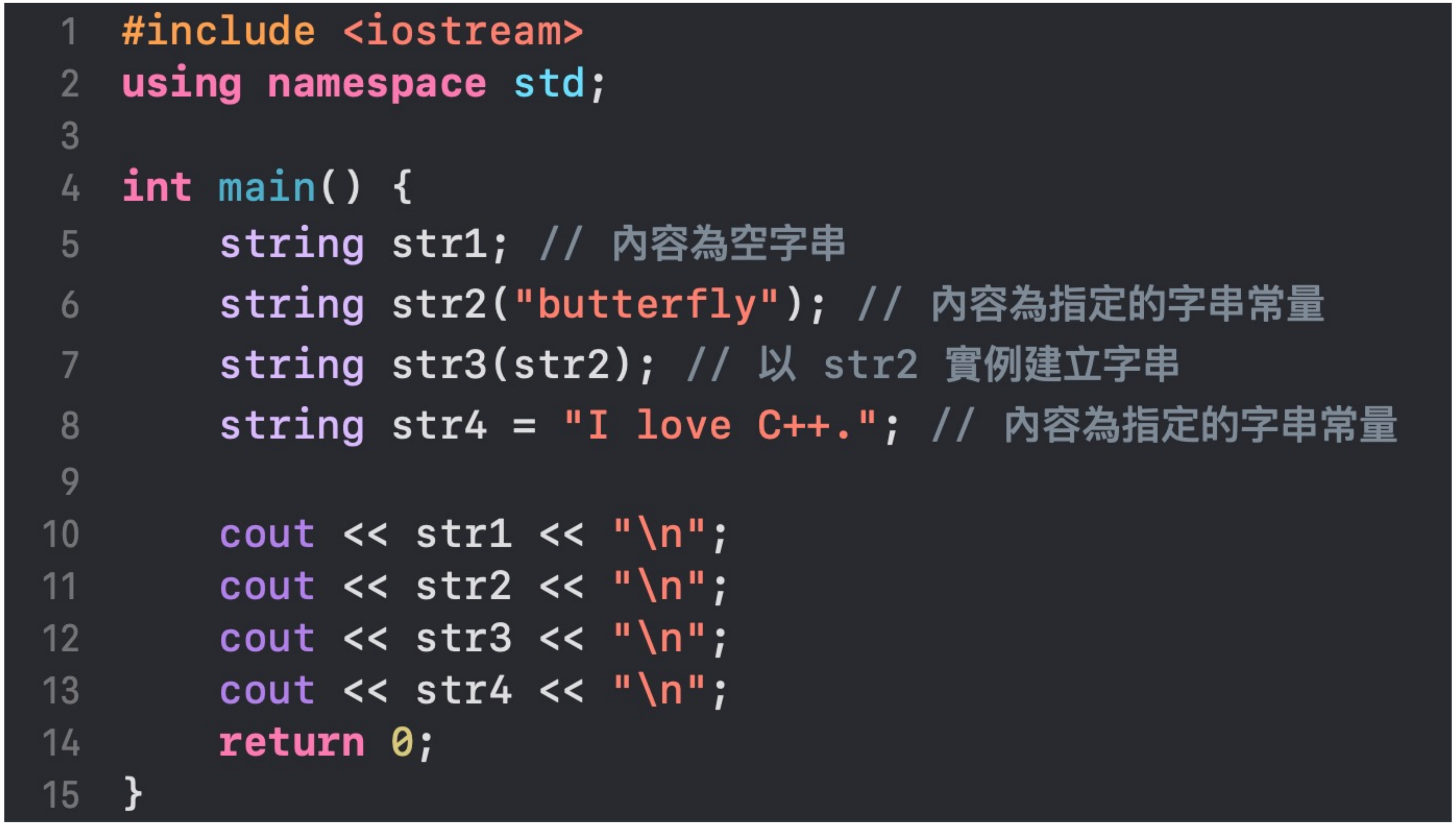
C++基礎語法 Unit-8

- C++字串(string)
- 常用字串函數

宣告字串 / 初始化

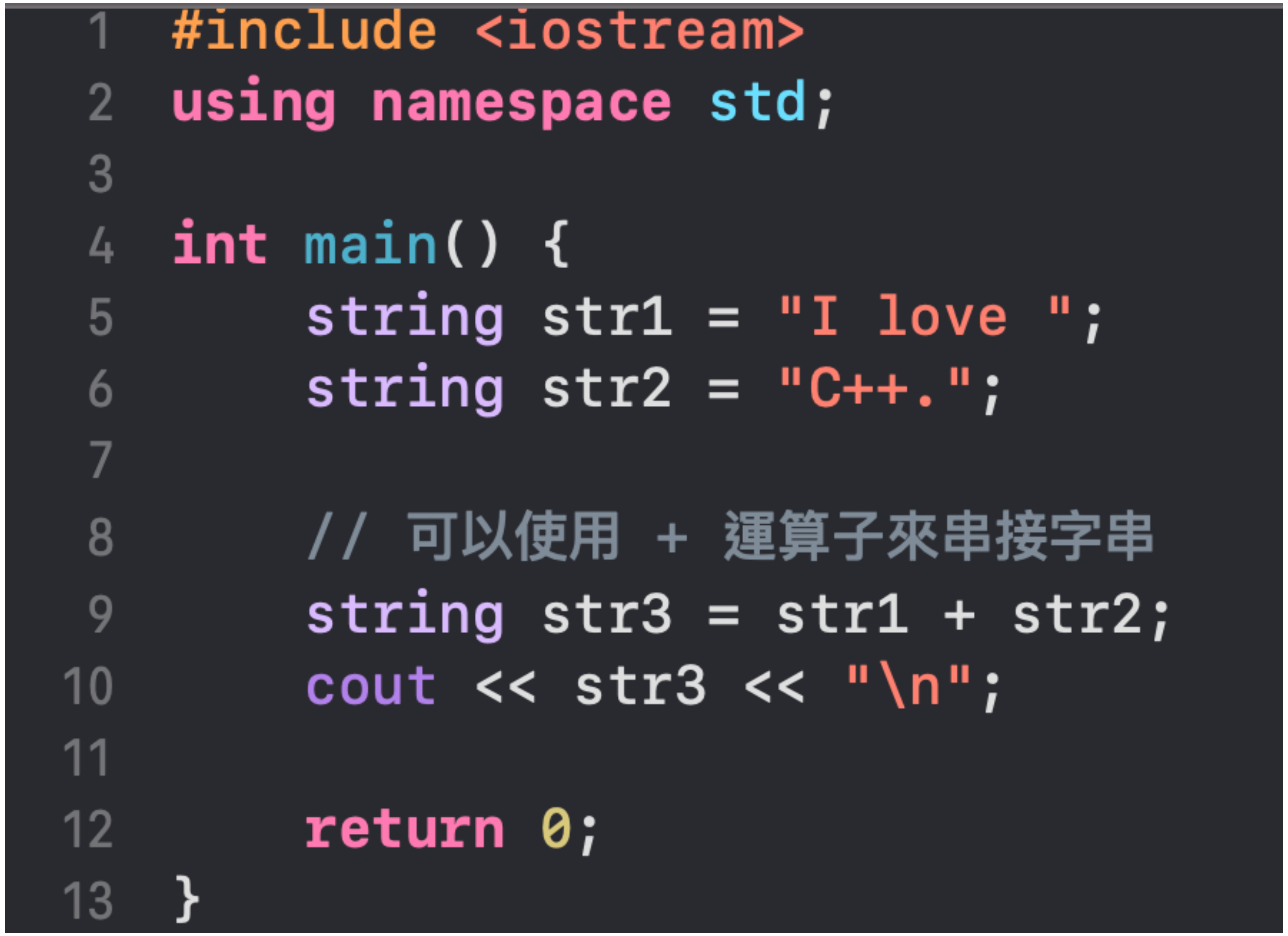
Example 8-1



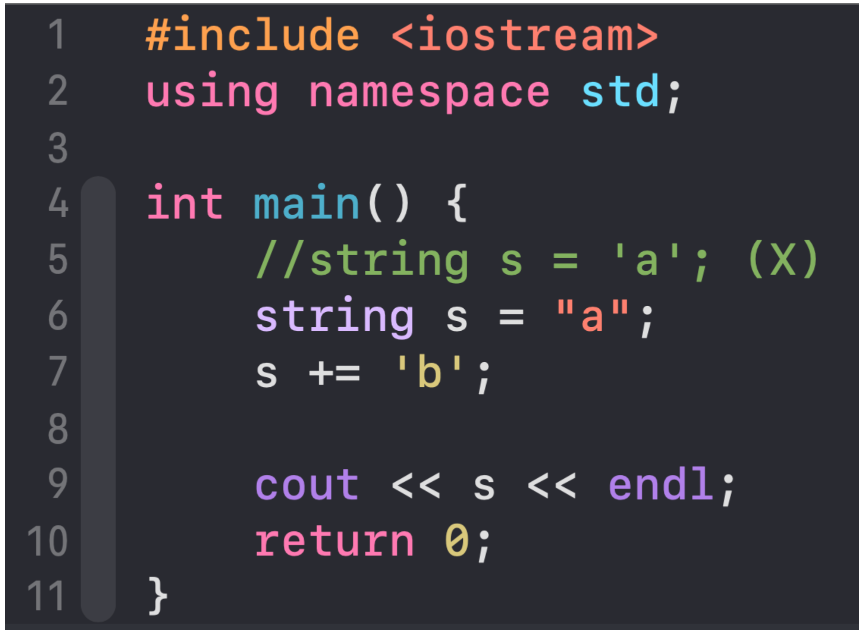
```cpp
#include <iostream>
using namespace std;

int main() {
    string str1; // 內容為空字串
    string str2("butterfly"); // 內容為指定的字串常量
    string str3(str2); // 以 str2 實例建立字串
    string str4 = "I love C++."; // 內容為指定的字串常量

    cout << str1 << "\n";
    cout << str2 << "\n";
    cout << str3 << "\n";
    cout << str4 << "\n";
    return 0;
}
```

可以使用 + 運算子來串接字串

Example 8-2



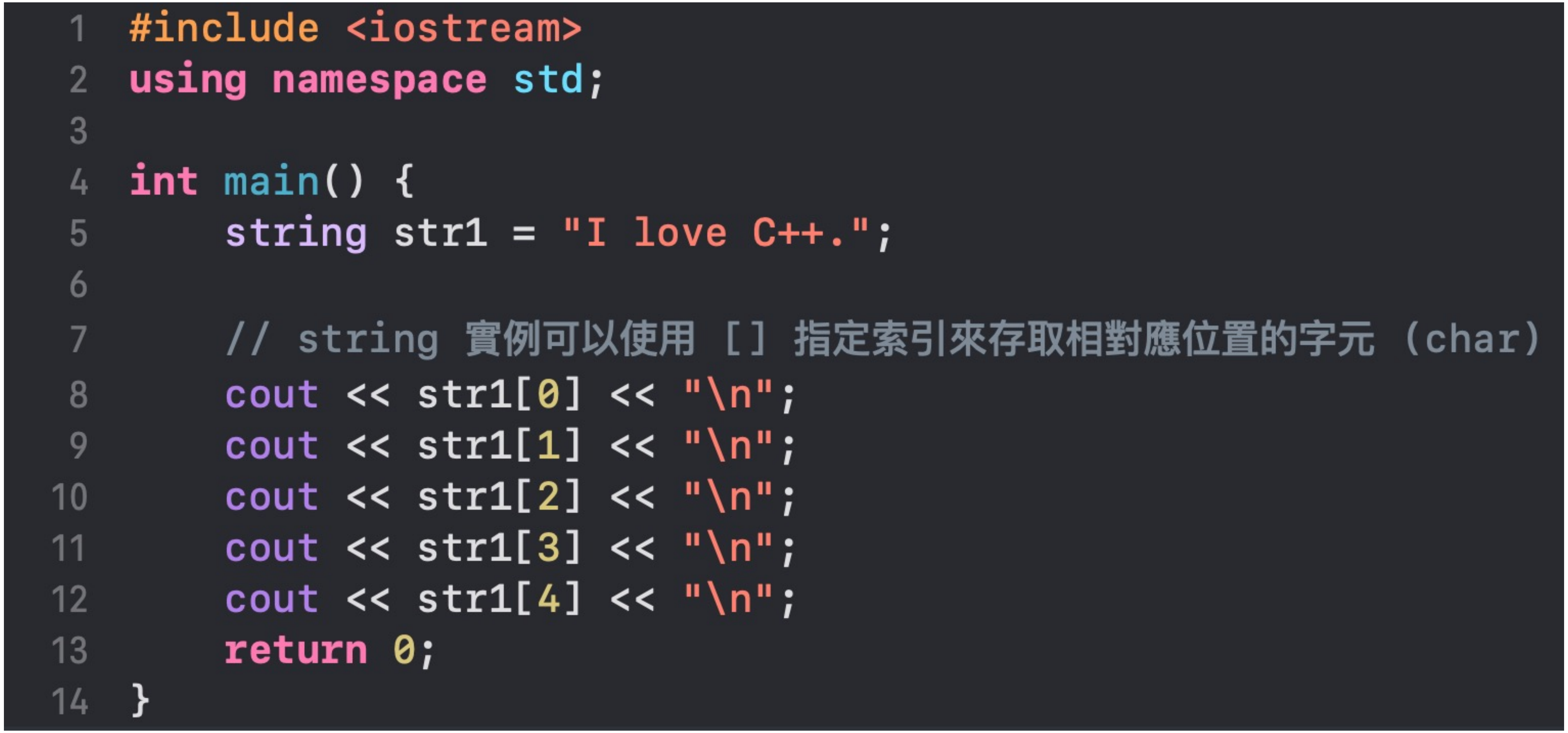
```cpp
#include <iostream>
using namespace std;

int main() {
    string str1 = "I love ";
    string str2 = "C++.";

    // 可以使用 + 運算子來串接字串
    string str3 = str1 + str2;
    cout << str3 << "\n";

    return 0;
}
```



```cpp
#include <iostream>
using namespace std;

int main() {
    //string s = 'a'; (X)
    string s = "a";
    s += 'b';

    cout << s << endl;
    return 0;
}
```

可以用「下標」來存取指定位置的字元

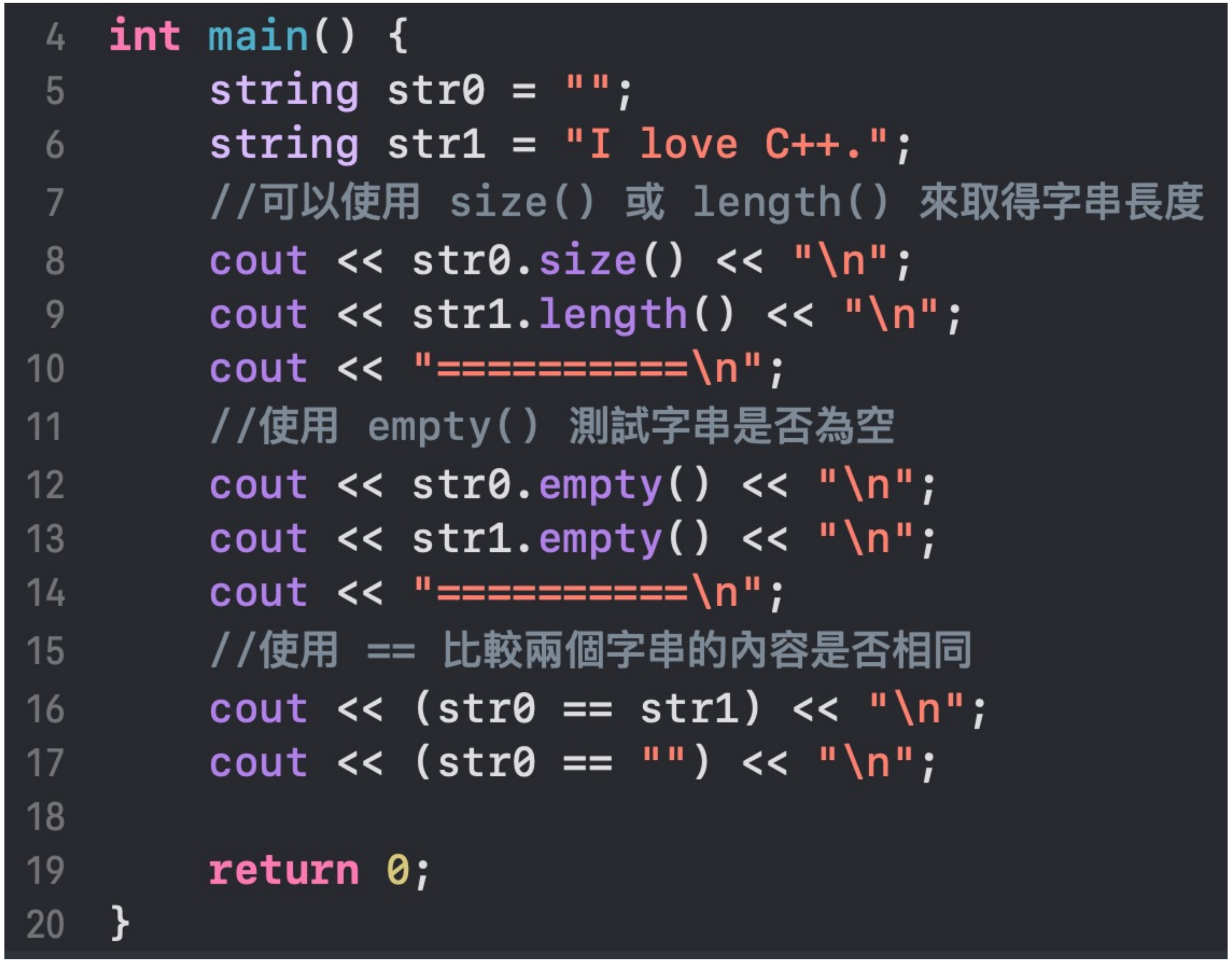
Example 8-3



```cpp
#include <iostream>
using namespace std;

int main() {
    string str1 = "I love C++.";

    // string 實例可以使用 [] 指定索引來存取相對應位置的字元 (char)
    cout << str1[0] << "\n";
    cout << str1[1] << "\n";
    cout << str1[2] << "\n";
    cout << str1[3] << "\n";
    cout << str1[4] << "\n";
    return 0;
}
```

size(), length(), empty(), ==

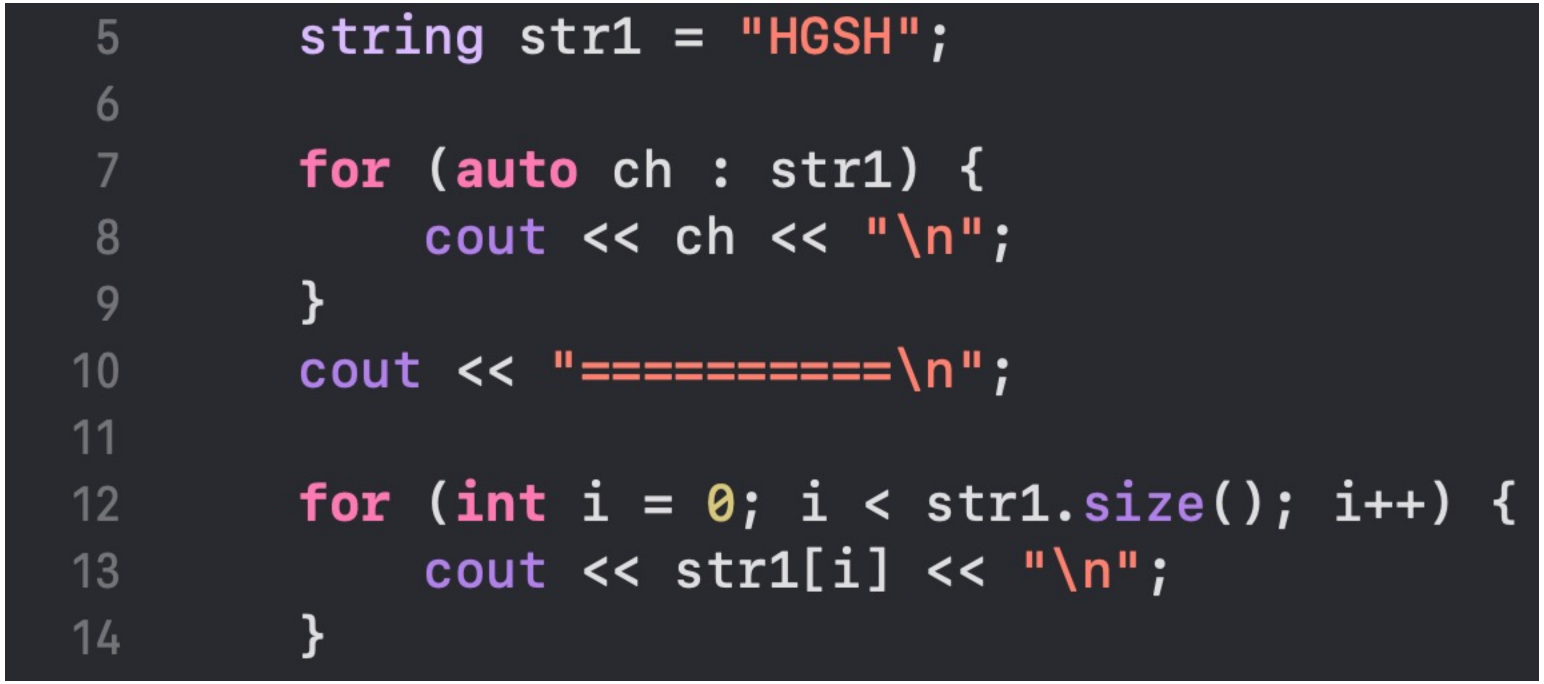
Example 8-4



```cpp
int main() {
    string str0 = "";
    string str1 = "I love C++.";
    //可以使用 size() 或 length() 來取得字串長度
    cout << str0.size() << "\n";
    cout << str1.length() << "\n";
    cout << "==========\n";
    //使用 empty() 測試字串是否為空
    cout << str0.empty() << "\n";
    cout << str1.empty() << "\n";
    cout << "==========\n";
    //使用 == 比較兩個字串的內容是否相同
    cout << (str0 == str1) << "\n";
    cout << (str0 == "") << "\n";

    return 0;
}
```

Example 8-5

遍歷字串的每一個字元



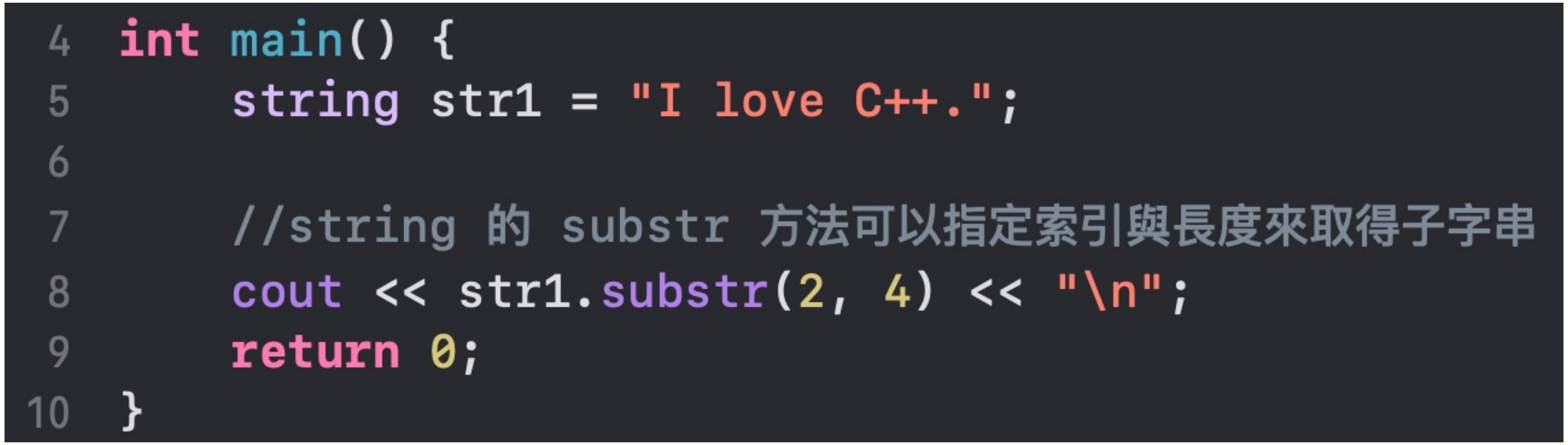
```
string str1 = "HGSH";

for (auto ch : str1) {
    cout << ch << "\n";
}
cout << "==========\n";

for (int i = 0; i < str1.size(); i++) {
    cout << str1[i] << "\n";
}
```

auto 自動變數類型
(C++ 11 起)

.substr(index, length)

Example 8-6

index	0	1	2	3	4	5	6	7	8	9	10
str1	I		l	o	v	e		C	+	+	.



```
int main() {
    string str1 = "I love C++.";

    //string 的 substr 方法可以指定索引與長度來取得子字串
    cout << str1.substr(2, 4) << "\n";
    return 0;
}
```

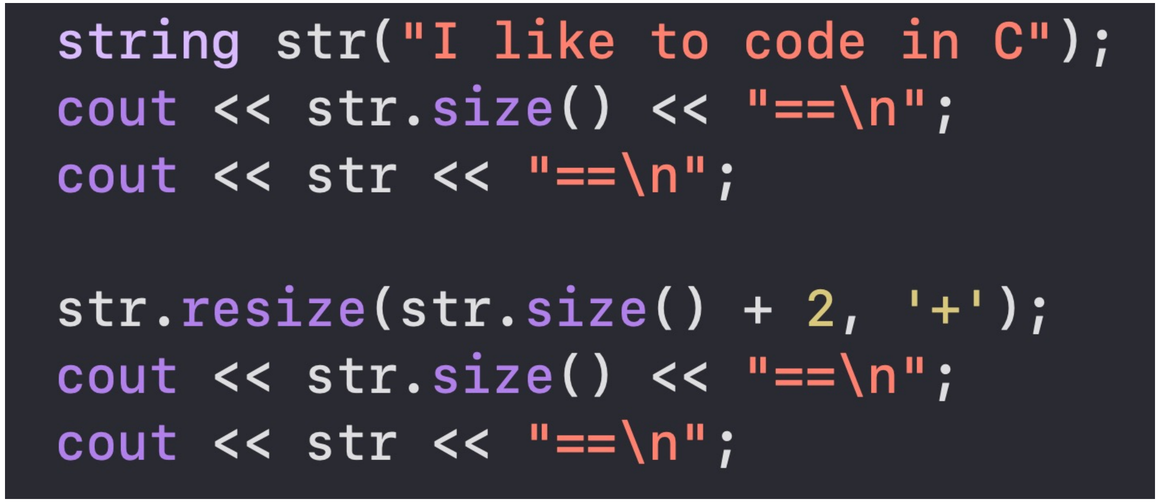
when length = -1, till the end of the string

.resize()

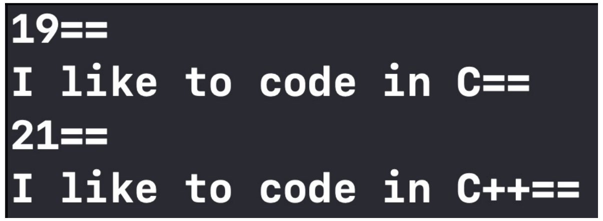
Example 8-7

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
str1	I		l	i	k	e		t	o		c	o	d	e		i	n		C

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
str1	I		l	i	k	e		t	o		c	o	d	e		i	n		C	+	+



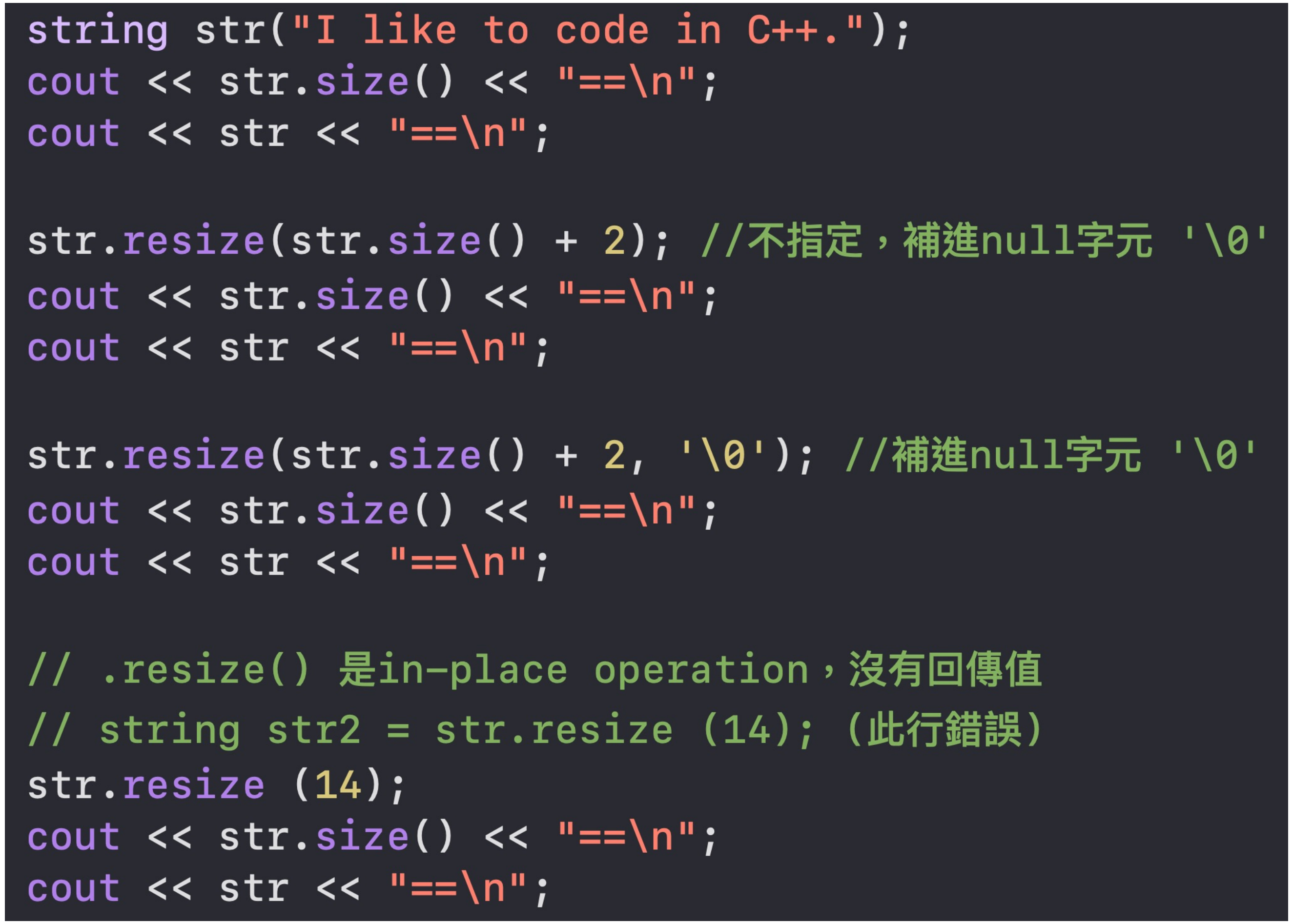
```
string str("I like to code in C");
cout << str.size() << "==\n";
cout << str << "==\n";

str.resize(str.size() + 2, '+');
cout << str.size() << "==\n";
cout << str << "==\n";
```



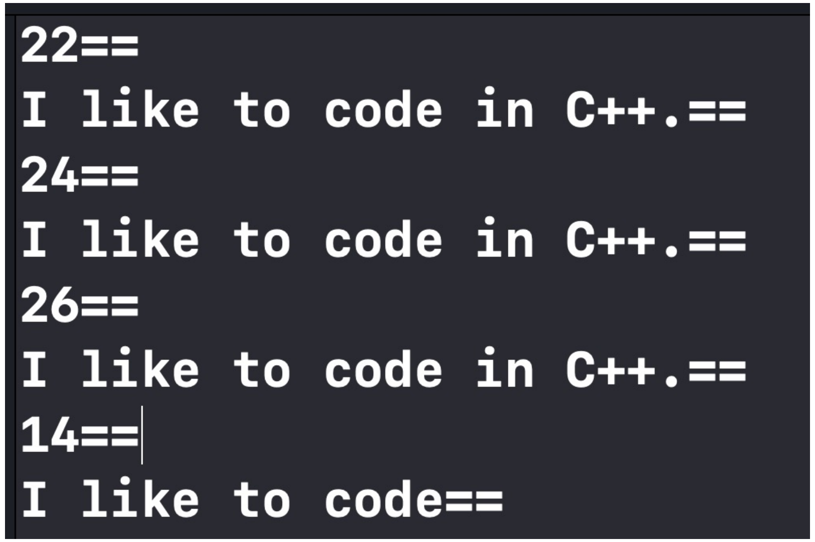
```
19==
I like to code in C==
21==
I like to code in C++==
```

.resize()

Example 8-7



```
string str("I like to code in C++.");
cout << str.size() << "==\n";
cout << str << "==\n";

str.resize(str.size() + 2); //不指定，補進null字元 '\0'
cout << str.size() << "==\n";
cout << str << "==\n";

str.resize(str.size() + 2, '\0'); //補進null字元 '\0'
cout << str.size() << "==\n";
cout << str << "==\n";

// .resize() 是in-place operation，沒有回傳值
// string str2 = str.resize (14); (此行錯誤)
str.resize (14);
cout << str.size() << "==\n";
cout << str << "==\n";
```

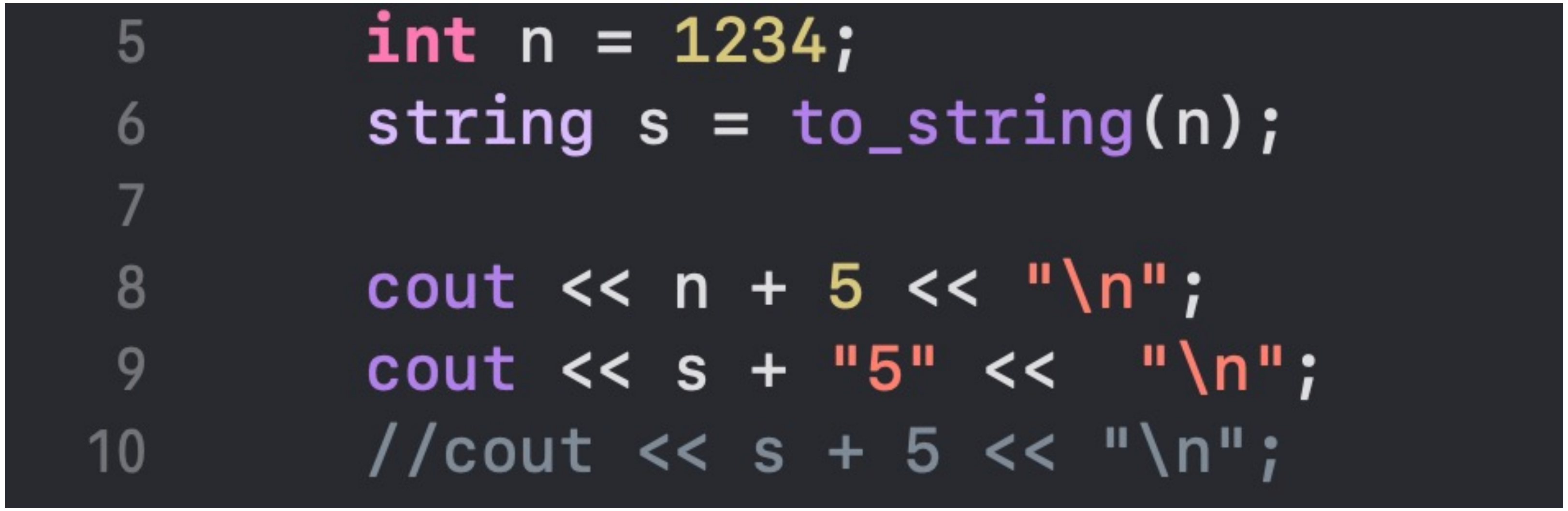


```
22==
I like to code in C++.==
24==
I like to code in C++.==
26==
I like to code in C++.==
14==
I like to code==
```

https://www.cplusplus.com/reference/string/string/resize/

to_string() 數字轉字串

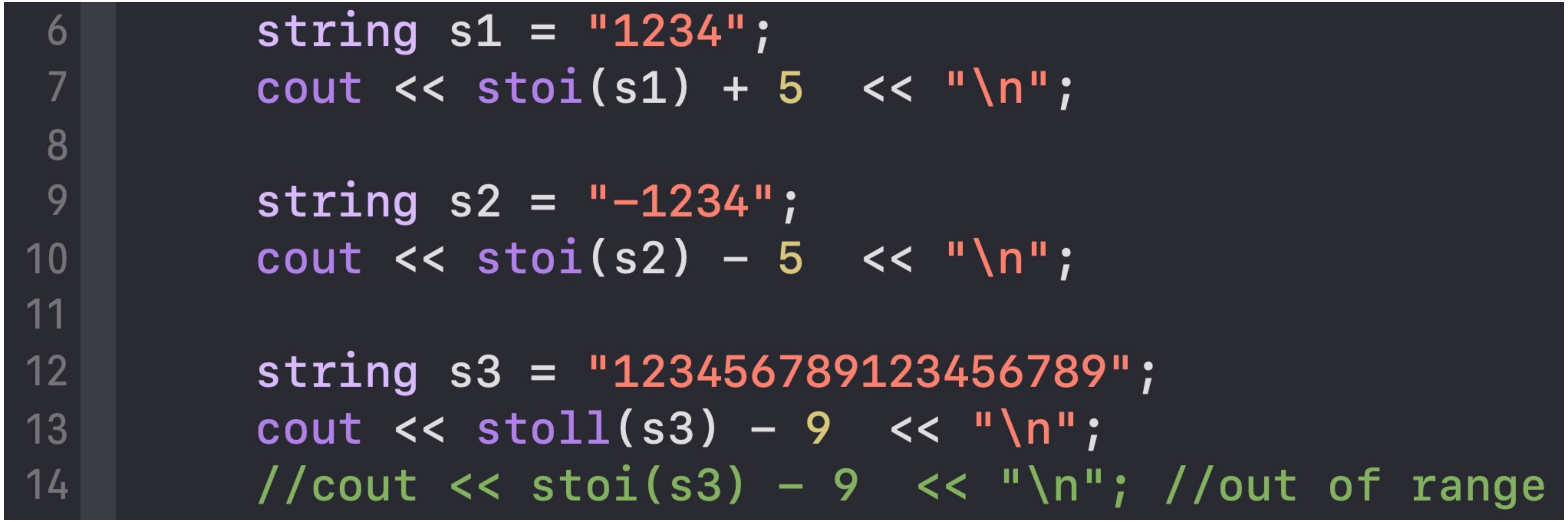
Example 8-8



```
    int n = 1234;
    string s = to_string(n);

    cout << n + 5 << "\n";
    cout << s + "5" <<  "\n";
    //cout << s + 5 << "\n";
```

stoi(), stoll() 字串轉數字

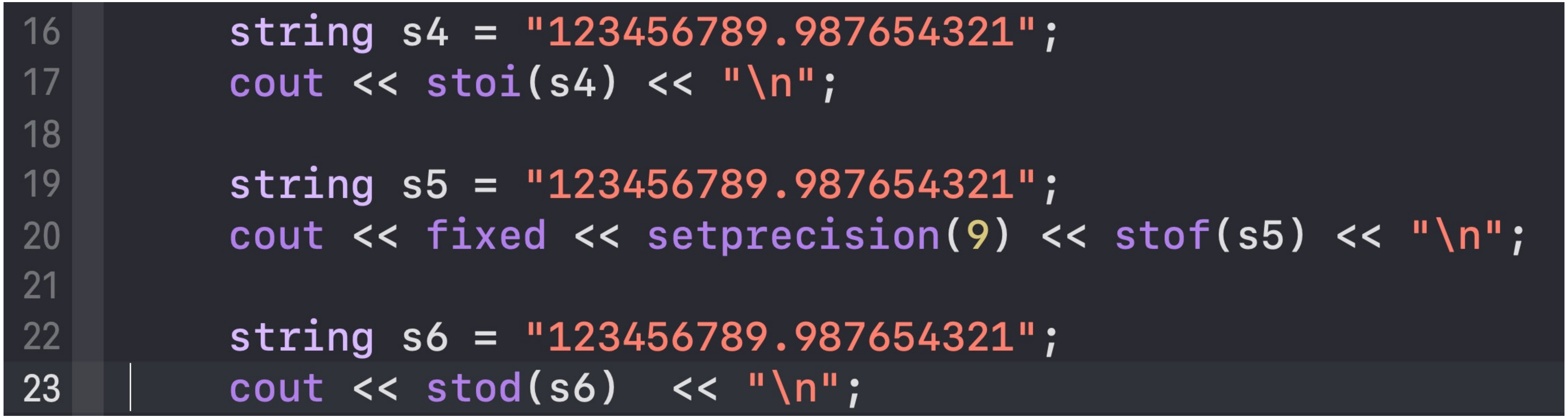
Example 8-9



```
    string s1 = "1234";
    cout << stoi(s1) + 5  << "\n";

    string s2 = "-1234";
    cout << stoi(s2) - 5  << "\n";

    string s3 = "123456789123456789";
    cout << stoll(s3) - 9  << "\n";
    //cout << stoi(s3) - 9  << "\n"; //out of range
```

stof(), stod() 字串轉數字

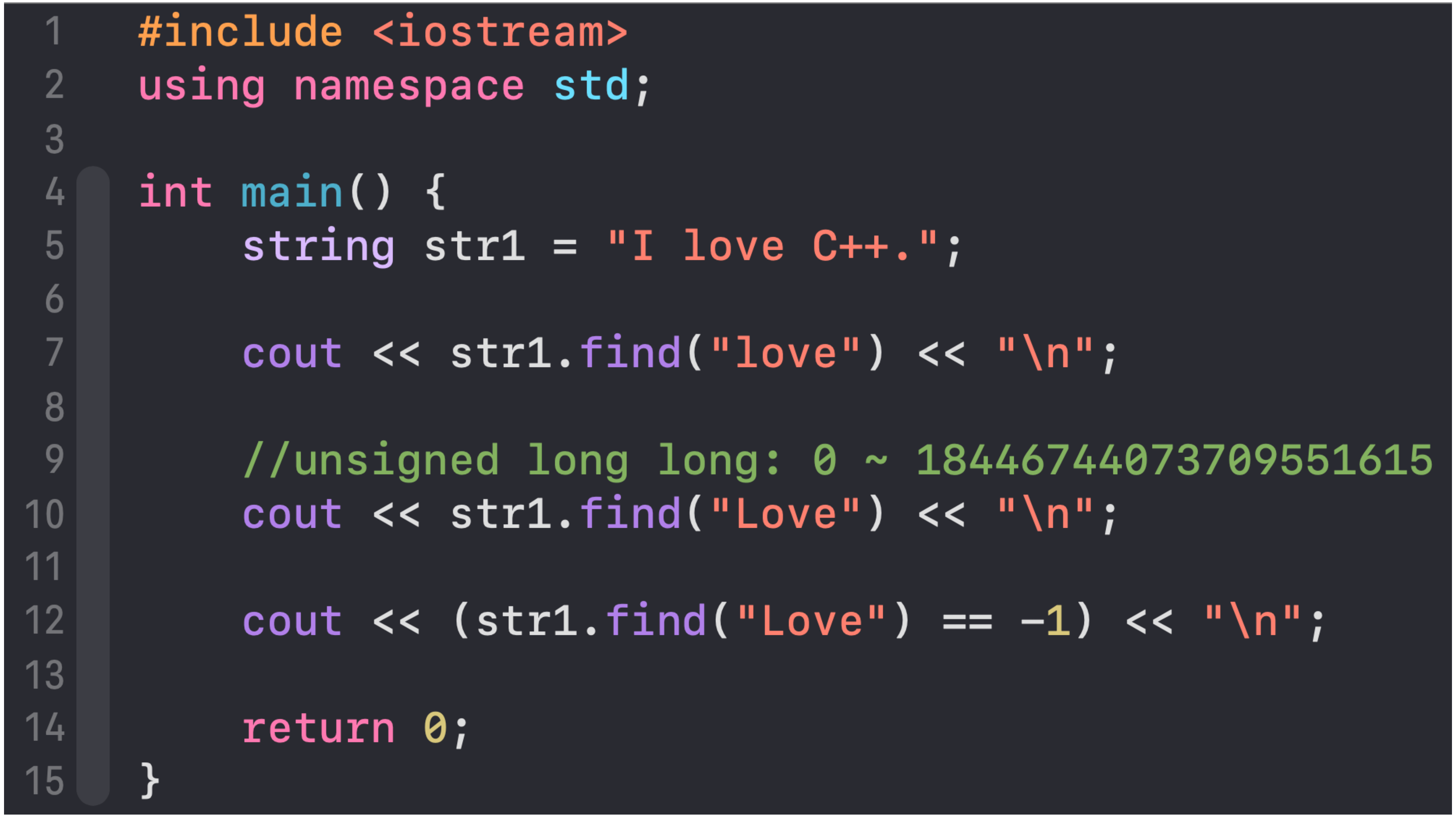
Example 8-9



```cpp
    string s4 = "123456789.987654321";
    cout << stoi(s4) << "\n";

    string s5 = "123456789.987654321";
    cout << fixed << setprecision(9) << stof(s5) << "\n";

    string s6 = "123456789.987654321";
    cout << stod(s6)  << "\n";
```

.find(string) 尋找子字串

Example 8-10

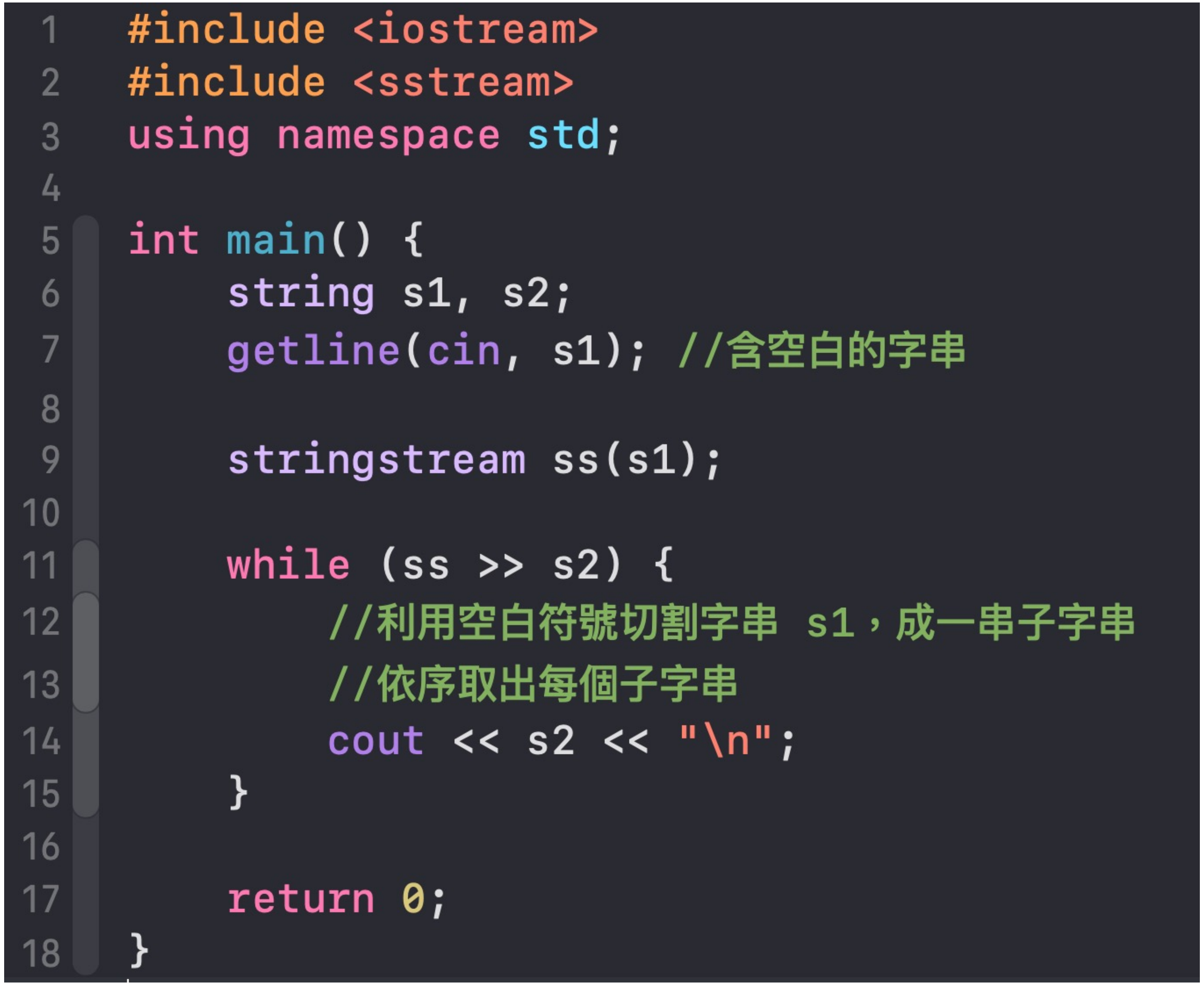


```cpp
#include <iostream>
using namespace std;

int main() {
    string str1 = "I love C++.";

    cout << str1.find("love") << "\n";

    //unsigned long long: 0 ~ 18446744073709551615
    cout << str1.find("Love") << "\n";

    cout << (str1.find("Love") == -1) << "\n";

    return 0;
}
```

stringstream

stringstream

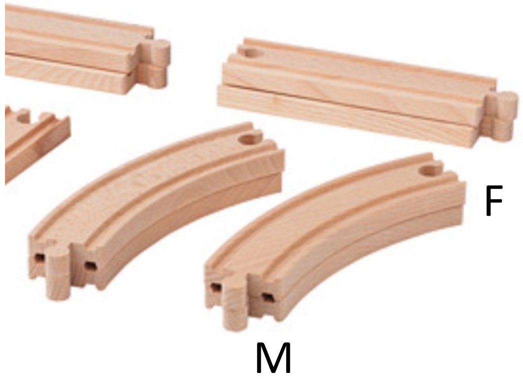
Example 8-11



```cpp
#include <iostream>
#include <sstream>
using namespace std;

int main() {
    string s1, s2;
    getline(cin, s1); //含空白的字串

    stringstream ss(s1);

    while (ss >> s2) {
        //利用空白符號切割字串 s1，成一串子字串
        //依序取出每個子字串
        cout << s2 << "\n";
    }

    return 0;
}
```

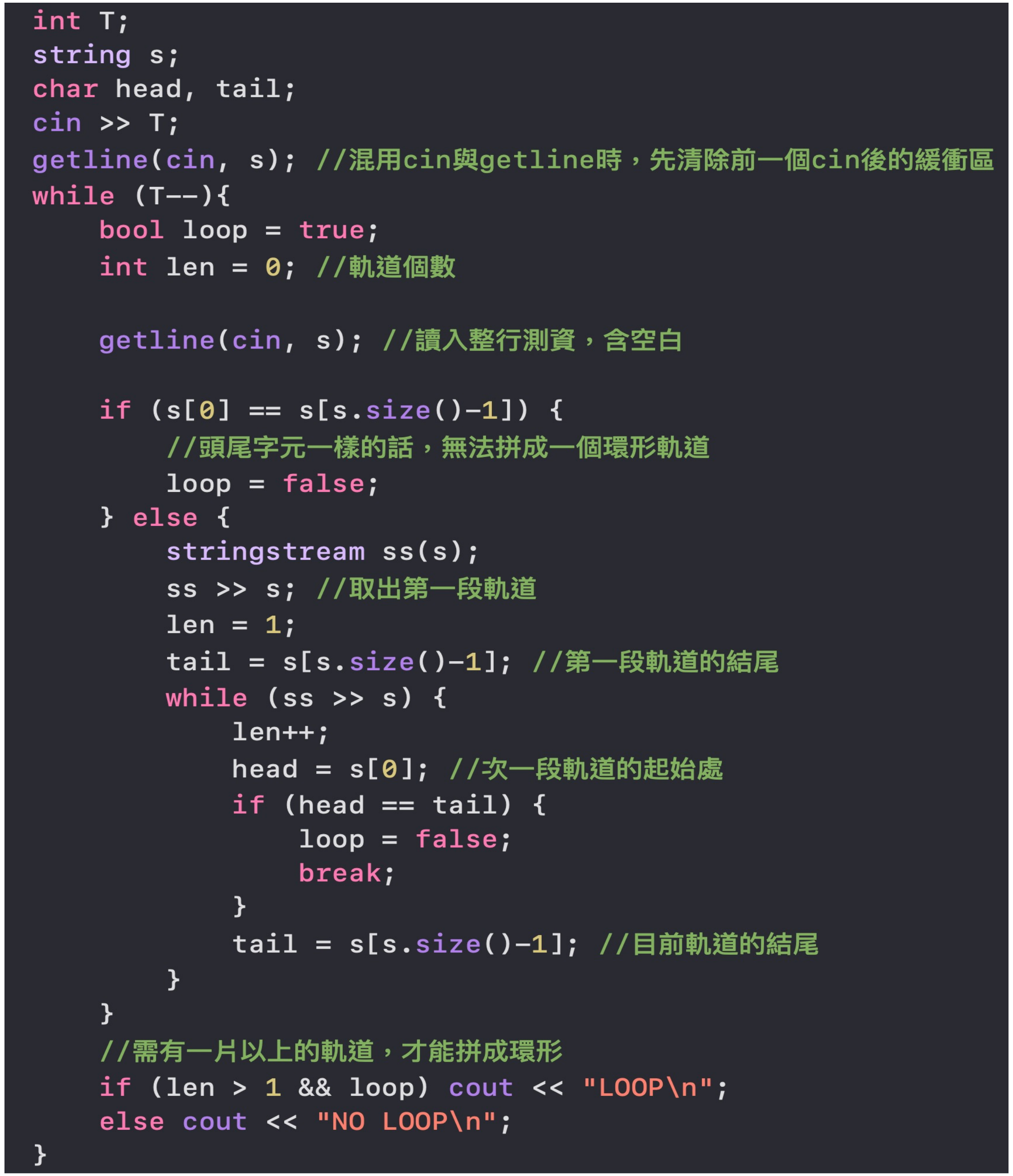
stringstream

【範例】ZeroJudge d275: 11586 – Train Tracks



【例1】 FM FF MF MM

【例2】 MF MF MF MF FF



```
int T;
string s;
char head, tail;
cin >> T;
getline(cin, s); //混用cin與getline時，先清除前一個cin後的緩衝區
while (T--){
    bool loop = true;
    int len = 0; //軌道個數

    getline(cin, s); //讀入整行測資，含空白

    if (s[0] == s[s.size()-1]) {
        //頭尾字元一樣的話，無法拼成一個環形軌道
        loop = false;
    } else {
        stringstream ss(s);
        ss >> s; //取出第一段軌道
        len = 1;
        tail = s[s.size()-1]; //第一段軌道的結尾
        while (ss >> s) {
            len++;
            head = s[0]; //次一段軌道的起始處
            if (head == tail) {
                loop = false;
                break;
            }
            tail = s[s.size()-1]; //目前軌道的結尾
        }
    }
    //需有一片以上的軌道，才能拼成環形
    if (len > 1 && loop) cout << "LOOP\n";
    else cout << "NO LOOP\n";
}
```