# 7.7 函数和string对象

虽然C-风格字符串和string对象的用途几乎相同，但与数组相比，string对象与结构的更相似。例如，可以将一个结构赋给另一个结构，也可以将一个对象赋给另一个对象。可以将结构作为完整的实体传递给函数，也可以将对象作为完整的实体进行传递。如果需要多个字符串，可以声明一个string对象数组，而不是二维char数组。

程序清单7.14提供了一个小型示例，它声明了一个string对象数组，并将该数组传递给一个函数以显示其内容。

**程序清单7.14 topfive.cpp**

// topfive.cpp -- 处理一个字符串对象数组

#include <iostream>

#include <string>

using namespace std;

const int SIZE = 5;

void display(const string sa[], int n);

int main()

{

 string list[SIZE]; // 包含5个字符串对象的数组

 cout << "Enter your " << SIZE << " favorite astronomical sights:\n";

 for (int i = 0; i < SIZE; i++)

 {

 cout << i + 1 << ": ";

 getline(cin,list[i]);

 }

 cout << "Your list:\n";

 display(list, SIZE);

 cin.get();

 return 0;

}

void display(const string sa[], int n)

{

 for (int i = 0; i < n; i++)

 cout << i + 1 << ": " << sa[i] << endl;

}

下面是该程序的运行情况：

Enter your 5 favorite astronomical sights:

1:Orion Nebula

2:M13

3:Saturn

4:Jupiter

5:Moon

Your list:

1:Orion Nebula

2:M13

3:Saturn

4:Jupiter

5:Moon

对于该示例，需要指出的一点是，除函数getline()外，该程序像对待内置类型（如int）一样对待string对象。如果需要string数组，只需使用通常的数组声明格式即可：

string list[SIZE];

这样，数组list的每个元素都是一个string对象，可以像下面这样使用它：

getline(cin,list[i]);

同样，形参sa是—个指向string对象的指针，因此sa[i]是一个string对象，可以像下面这样使用它：

cout<< i + 1 << “: ” << sa[i] << endl;