



字符串类：求子串函数 substr

1. //返回 pos 开始的 len 个字符组成的字符串
2. `basic_string substr (size_type pos = 0, size_type len = npos) const;`

【参考程序】

```
1. #include<iostream>
2. #include<string>
3. using namespace std;
4. int main()
5. {
6.     string s = "Hello, Xiaozhi!";
7.     string s1 = s.substr(2);
8.     string s2 = s.substr(2, 3);
9.     cout << s1 << endl << s2 << endl;
10.    return 0;
11. }
```



字符串类：删除字符函数 erase

1. //删除 pos 开始的 n 个字符，返回修改后的字符串，源字符串也被修改
2. `basic_string& erase(size_type pos = 0, size_type len = npos);`

【参考程序】

```
1. #include<iostream>
2. #include<string>
3. using namespace std;
4. int main()
5. {
6.     string s = "Hello, Xiaozhi!";
7.     string s1 = s.erase(6);
8.     string s2 = s.erase(1, 2);
9.     cout << s << endl << s1 << endl << s2 << endl;
10.    return 0;
11. }
```

字符串类：插入字符函数 insert

```
1. #include<iostream>
2. #include<string>
3. using namespace std;
4. int main()
5. {
6.     string s = ",";
7.     s.insert(0, "heo"); //从第 0 个位置，插入字符串
8.     cout << s << endl ;
9.     s.insert(4, "world", 2); //从第 4 个位置，插入字符串的前 2 个字符
10.    cout << s << endl;
11.    s.insert(2, 2, 'l'); //从第 2 个位置，插入 2 次
12.    cout << s << endl;
13.    s.insert(s.end(), 'r'); //在末尾插入字符
14.    cout << s << endl;
15.    s += "ld!"; //在末尾插入字符串
16.    cout << s << endl;
17.    return 0;
18. }
```



字符串类：替换字符函数 replace

【参考程序】

```
1. #include<iostream>
2. #include<string>
3. using namespace std;
4. int main()
5. {
6.     string s = "Hello, world!";
7.     //删除第 7 个字符后的 6 个字符，以"jjm"代替
8.     s.replace(7, 6, "jjm");
9.     cout << s << endl;
10.    //删除第 7 个字符后 3 个字符，以"jianming"前 6 个字符代替
11.    s.replace(7, 3, "jianming", 6);
12.    cout << s << endl;
13.    //删除第 7 个字符后的 6 个字符，以 3 个'z'字符代替
14.    s.replace(7, 6, 3, 'z');
15.    cout << s << endl;
16.    // ... 还有很多用法
17.    return 0;
18. }
```



字符串类：查找系列函数 find

找字符串

- find: 从前往后找
- rfind: 从后往前找

找第一个出现在某个字符串中的字符

- find_first_of
- find_first_not_of
- find_last_of
- find_last_not_of

【参考程序】

```
1. #include<iostream>
2. #include<string>
3. using namespace std;
4. int main()
5. {
6.     string s = "Hello, world!";
7.     string filter = ",!";
8.     cout << s.find('l', 1) << endl ; //第 1 个开始查找字符'l'的首次出现
9.     cout << s.find("lo", 7) << " " << string::npos << endl; //字符串首次出现，不
    存在返回 string::npos
10.    cout << s.rfind("lo", 7) << endl; //往前能找到
11.    cout << s.find_first_of(filter, 2) << endl; //从第 2 个位置开始找在字符串
    filter 中的字符
12.    cout << s.find_last_of(filter) << endl; //同上，只是从后往前找
13.    cout << s.find_first_not_of(filter) << endl; //不在 filter 中的
14.    cout << s.find_last_not_of(filter, 100) << endl; //从后往前
15.    return 0;
16. }
```



例：字符串排序

【问题描述】

输入 10 个单词，按照字典顺序输出。

【参考程序】

```
1. #include<algorithm>
2. #include<iostream>
3. #include<string>
4. using namespace std;
5. int main()
6. {
7.     string cname[10];
8.     for (int i=0; i<10; i++)
9.     {
10.         getline(cin, cname[i]);
11.     }
12.     //利用 C++库函数排序
13.     sort(cname, cname+10);
14.     cout << "===output===" << endl;
15.     for (int i=0; i<10; i++)
16.     {
17.         cout << cname[i] << endl;
18.     }
19.     return 0;
20. }
```