



CS100

信息学算法入门

第四讲

课程资料下载地址

CS100信息学算法入门公布资料的固定网站
<http://pan.baidu.com/s/1jHDcMou>

请每次课前自行将资料下载到电脑

Hello World回顾

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

Hello, World!
Hello, World!



hello.cpp

字符串

字符串用" "括起来

"Hello, World! "

"David"

"I like apple and orange! "

字符用' '括起来

'A'

'c'

'+'

字符与字符串是两种不同的类型

声明字符串

```
#include <string> //包含头文件
```

```
string s1; //默认初始化，s1是一个空串  
string s2(s1); //s2是s1的副本  
string s2 = s1; //等价于s2(s1)  
string s3("value"); //s3是"value"的副本  
string s3 = "value"; //等价于s3("value")  
string s4 = (n, 'c'); //由n个连续字符'c'组成的串
```

```
string name; //声明空字符串  
name = "David"; //字符串赋值  
string city = "Shanghai"; //声明并初始化  
string a("hello"); //a="hello"  
string g(5, 'g'); //g="ggggg"
```

字符串输入输出

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  int main()
6  {
7      string s;
8      cin >> s;
9      cout << s << endl;
10     return 0;
11 }
```

cin读入遇空格、Tab、回车结束

读取整行

`getline(stream, str)` // `stream` 输入流, `str` 字符串变量名

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  int main()
6  {
7      string fullname;
8      cout << "Please input your full name." << endl;
9      getline(cin, fullname);
10     cout << "Your name is " << fullname << endl;
11     return 0;
12 }
```

Please input your full name.
David Yu
Your name is David Yu



readline.cpp

练习

打印个人信息。

声明name, email, mobile字符串，从键盘输入，并输出

```
Input your name David
Input your email address david@home.com
Input your mobile number 12345
Your name: David
Your email address: david@home.com
Your mobile number: 12345
```



information.cpp

下标访问

字符串包含的字符数为字符串的长度

`str.size()` // `str`为字符串，查看长度

访问字符串上的字符可以使用下标运算符`[]`

```
5  int main()
6  {
7      string l = "123456789";
8      cout << l << endl;
9      string title = "My string.";
10     cout << title << endl;
11     cout << title.size() << endl;
12     for (int i=0; i < title.size(); ++i) {
13         cout << title[i] << " ";
14     }
15     return 0;
16 }
```

```
123456789
My string.
10
M y   s t r i n g .
```



element.cpp

string类型举例

教科书例2.1-2.4

字符串处理函数

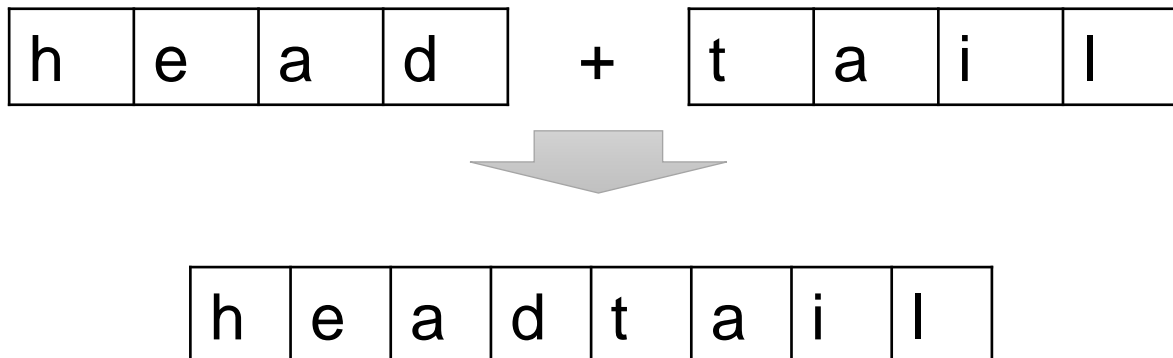
函数名	功能	举例
size()	返回字符串长度	<code>string s = "12345"; s.size(); //5</code>
clear()	清除字符串中所有元素	<code>string s = "hello"; s.clear(); //s=""</code>
empty()	判断字符串是否为空,字符串长度为0 返回true,否则false	<code>string s = "hello"; s.empty(); //false s.clear(); s.empty(); //true</code>

string操作

连接

+连接两个字符串，得到一个新的字符串

```
string s1 = "head";  
string s2 = "tail";  
string s=s1 + s2;
```



string操作

查找

查找find()

返回符合搜索条件的字符区间内的第一个字符的索引，没找到目标就返回npos。

所有的函数的参数说明如下：

第一个参数是被搜寻的对象。第二个参数（可有可无）指出string内的搜寻起点索引，第三个参数（可有可无）指出搜寻的字符个数。

string操作

查找

```
string str = "There are two needles in this haystack with needles.";
string str2("needle");
int found = str.find(str2);
if (found != string::npos)
    cout << "first needle found at: " << found << endl;
found = str.find("needles are small", found+1, 6);
if (found != string::npos)
    cout << "second needle found at: " << found << endl;
found = str.find("haystack");
if (found != string::npos)
    cout << "haystack? also found at:" << found << endl;
found = str.find('.');
if (found != string::npos)
    cout << "Period found at: " << found << endl;
```

```
first needle found at: 14
second needle found at: 44
haystack? also found at:30
Period found at: 51
```



find.cpp

string操作

取子串

取子串 `substr(位置, [字符串长度])`

`s.substr();` //返回s的全部内容

`s.substr(11);` //从索引11往后的子串

`s.substr(5,6);` //从索引5开始6个字符

string操作

插入

插入字符insert(位置, 字符串)

```
string s = "Tom";  
string s2 = "first ";  
s.insert(0, "my name is"); //my name Tom  
s.insert(3, s2); //my first name Tom
```


string操作

删除

删除字符erase(位置, [字符个数])

```
string s = "1234567890abcdef";  
s.erase(13); //从索引13开始往后全删除, s变为1234567890abc  
s.erase(7,5); //从索引7开始往后删5个, s变为1234567cdef
```

string操作

字符串比较

`==, !=, <, <=, >, >=, compare()`

字符串支持常见的比较操作符（`>`, `>=`, `<`, `<=`, `==`, `!=`）判断原则按照“当前字符特性”将字符按字典顺序进行逐一比较。字典排序靠前的字符小，比较的顺序是从前向后比较，遇到不相等的字符就按这个位置上的两个字符的比较结果确定两个字符串的大小。

另一个功能强大的比较函数是成员函数`compare()`。他支持多参数处理，支持用索引值和长度定位子串来进行比较。他返回一个整数来表示比较结果，返回值意义如下：`0`-相等 `>0`-大于 `<0`-小于。

```
string s("abcd");
```

```
s.compare("abcd"); //返回0
```

```
s.compare("dcba"); //返回一个小于0的值
```

```
s.compare("ab"); //返回大于0的值
```

```
s.compare(s); //相等
```

```
s.compare(0,2,s,2,2); //用"ab"和"cd"进行比较小于零
```

```
s.compare(1,2,"bcx",2); //用"bc"和"bc"比较。
```

字符串例题

教科书例题2.5-2.8

课外项目：统计字母的频率

输入一段话，统计各字母出现的次数

What I Have Lived For
by Bertrand Russell

Three passions, simple but overwhelmingly strong, have governed my life: the longing for love, the search for knowledge, and unbearable pity for the suffering of mankind. These passions, like great winds, have blown me hither and thither, in a wayward course, over a deep ocean of anguish, reaching to the verge of despair.

课外项目：替换加密

替换加密解密算法方案的起源可以追溯到凯撒（**Caesar**）时代，据说凯撒为了保障情报的可靠性而发明了凯撒密码。凯撒密码是一种简单的置换密码，在加密时，字母表中的每个字母都用其后的第三个字母表示，例如，**a**用**d**表示，**b**用**e**表示，其余以此类推。在解密时，只需要执行逆过程即可。

写**2**个程序，分别实现加密和解密功能。

加密程序，对用户输入的一个明文字符串，输出加密后的字符串。

解密程序，对用户输入的一个加密字符串，输出解密后的明文字符串。

作业

作业网站:

<http://120.132.20.20:8080/thrall-web/main#home>

题号: 130,131

附加题: 132