

(3 条消息) string 类的常用函数 (总结) _string 函数 _Laity (俗人) ! 的博客 -..

string 类常用成员函数

(1) 求长度

`int length(); int size();` // 求长度

(2) 比较 2 个字符串 `strcmp(s1, s2) == 0`

直接用 运算符 `< > == != <==> ==`

~~(3) 字符串复制 `strcpy(s1,s2); s2 -> s1`~~

可直接用运算符 `=`

~~(4) 字符串连接 `strcat(s1,s2); s1 + s2 -> s1`~~

直接用运算符 `+` 进行连接

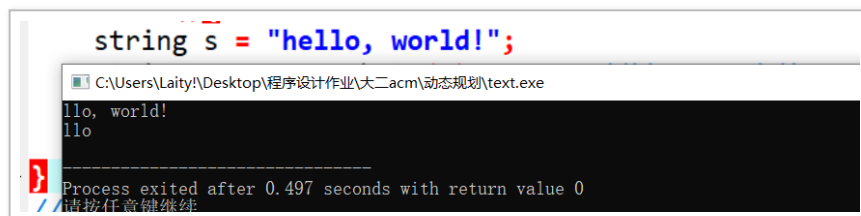
(5) 空串判断

`bool empty();` // 判断是否为空串

(6) 子串 `substr();`// 求子串 (重点)

`string substr(int pos = 0, int n = npos) const;` // 返回 pos 开始的 n 个字符组成的字符串

```
int main (){
    string s = "hello, world!";
    string ss1 = s.substr(2);          //llo, world!
    string ss2 = s.substr(2,3);        //llo
    cout << ss1 << endl << ss2 << endl;
}
```

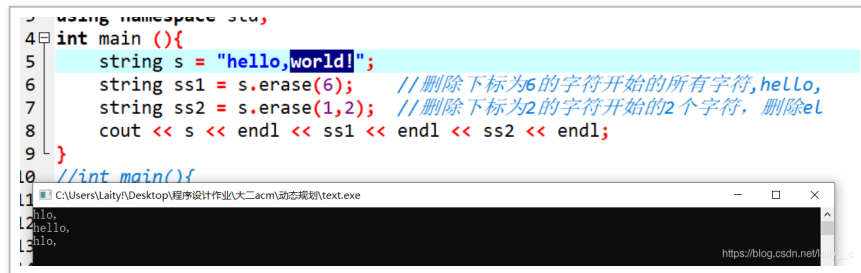


(7) `erase();` // 删除若干个字符 (重点)

`string &erase(int pos = 0, int n = npos);` // 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

注意：：： 原来的 s 也会改变；

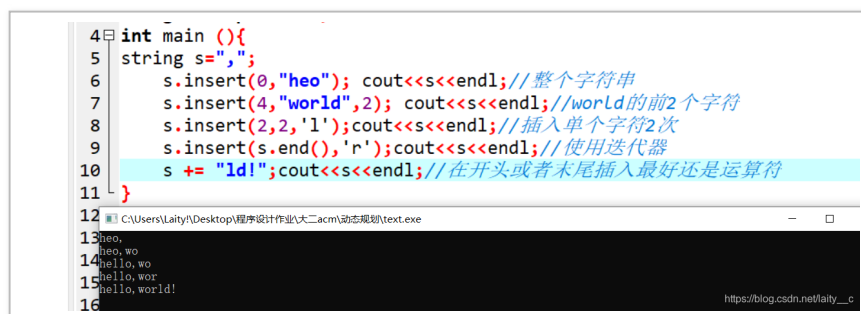
```
int main (){
    string s = "hello,world!";
    string ss1 = s.erase(6);    //删除下标为6的字符开始的所有字符,hello,
    string ss2 = s.erase(1,2); //删除下标为2的字符开始的2个字符,删除el
    cout << s << endl << ss1 << endl << ss2 << endl;
}
```



(8)insert();// 插入字符

```
string s=",";
s.insert(0,"heo"); cout<<s<<endl;//整个字符串
s.insert(4,"world",2); cout<<s<<endl;//world的前2个字符
s.insert(2,2,'l');cout<<s<<endl;//插入单个字符2次
s.insert(s.end(),'r');cout<<s<<endl;//使用迭代器
s += "ld!";cout<<s<<endl;//在开头或者末尾插入最好还是运算符
```

注意：：：是在下标为 i 前面加；

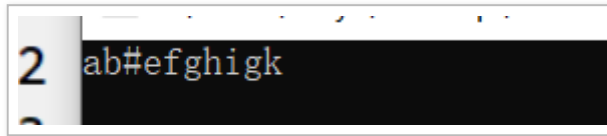


(9)replace();// 替换字符 (重点)

用法一：用 str 替换指定字符串从起始位置 pos 开始长度为 len 的字符

string& replace (size_t pos, size_t len, const string& str);

```
#include<iostream>
#include<string>
using namespace std;
int main()
{
    string str = "abcdefghigk";
    str=str.replace(str.find("c"),2,"#"); //从第一个a位置开始的两
    cout<<str<<endl;
    return 0;
}
```

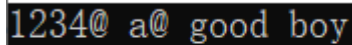


用法二：用 substr 的指定子串（给定起始位置和长度）替换从指定位置上的字符串

string& replace (size_t pos, size_t len, const string& str, size_t subpos, size_t sublen);

```
#include<iostream>
#include<string>
using namespace std;
int main()
{
    string str = "he is@ a@ good boy";
    string substr = "12345";
    str=str.replace(0,5,substr,substr.find("1"),4); //用substr的指
```

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



```
1234@ a@ good boy
```

(10)find 系列函数;// 查找字串



```
C++ 中 string 的 find() 函数的用法  
(列题见 F - All in All UVA - 10340  
)
```

string 的 find() 函数用于找出字母在字符串中的位置。

s.find(str,position)

find() 的两个参数:

str: 是要找的元素

position: 字符串中的某个位置, 表示从从这个位置开始的字符串中找指定元素。

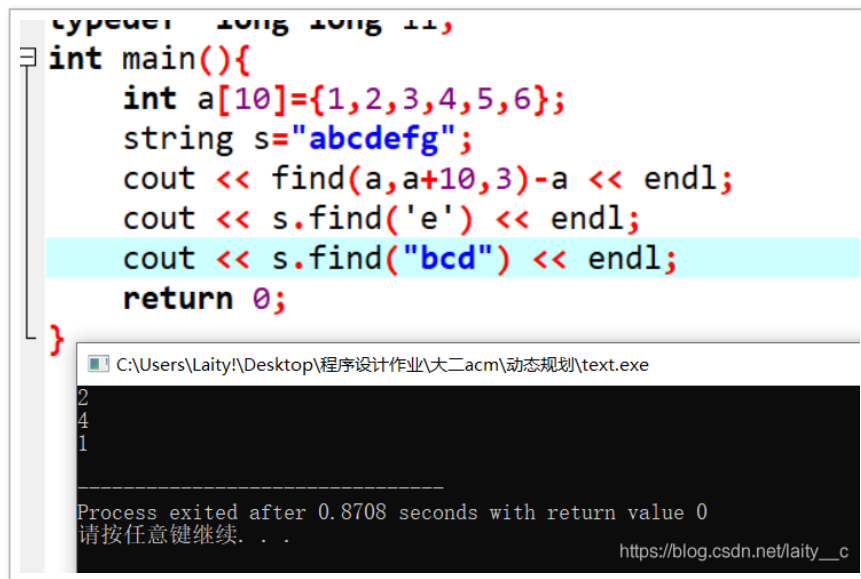
可以不填第二个参数, 默认从字符串的开头进行查找。

返回值为目标字符的位置, 当没有找到目标字符时返回 string::npos。 (或 -

1)

数组的 find(pos,pos+n,x) 用法 与 string 的不同

```
#include<iostream>
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
int main(){
    int a[10]={1,2,3,4,5,6};
    string s="abcdefg";
    cout << find(a,a+10,3)-a << endl;
    cout << s.find('e') << endl;
    cout << s.find("bcd") << endl;
    cout << s.find('e',4) << endl;//从下标为4开始搜索，输出-1;
    return 0;
}
```



```
typedef long long ll;
int main(){
    int a[10]={1,2,3,4,5,6};
    string s="abcdefg";
    cout << find(a,a+10,3)-a << endl;
    cout << s.find('e') << endl;
    cout << s.find("bcd") << endl;
    return 0;
}
```

C:\Users\Laity!\Desktop\程序设计作业\大二acm\动态规划\text.exe

2
4
1

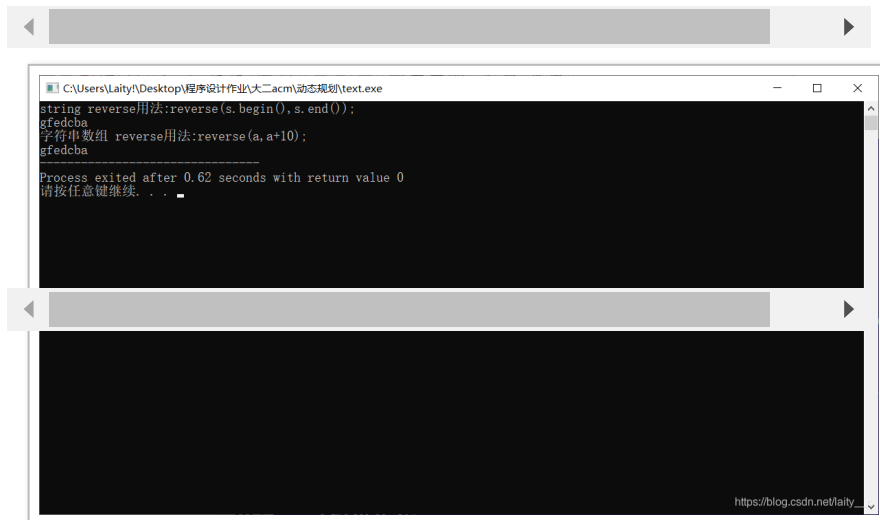
Process exited after 0.8708 seconds with return value 0
请按任意键继续. . .

https://blog.csdn.net/laity__c

reverse () 函数的使用

```
#include<iostream>
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
int main(){
    char a[8]="abcdefg";
```

```
string s="abcdefg";
cout << "string reverse用法:reverse(s.begin(),s.end());" << endl;
reverse(s.begin(),s.end());
cout << s << endl;
cout << "字符串数组 reverse用法:reverse(a,a+10);" << endl;
reverse(a,a+7);
for(int i=0;i<7;i++){
    cout << a[i];
}
return 0;
}
```



string 的 STL 中 sort 使用

sort (s.begin(),s.end()) ;

全文完

本文由 简悦 SimpRead 优化，用以提升阅读体验

使用了 全新的简悦词法分析引擎^{beta}，[点击查看详细说明](#)

