g s;  //生成一个空字符串s  
b)    string s(str) //拷贝构造函数 生成str的复制品  
c)    string s(str,stridx) //将字符串str内“始于位置stridx”的部分当作字符串的初值  
d)    string s(str,stridx,strlen) //将字符串str内“始于stridx且长度顶多strlen”的部分作为字符串的初值  
e)    string s(cstr) //将C字符串作为s的初值  
f)    string s(chars,chars\_len) //将C字符串前chars\_len个字符作为字符串s的初值。  
g)    string s(num,c) //生成一个字符串，包含num个c字符  
h)    string s(beg,end) //以区间beg;end(不包含end)内的字符作为字符串s的初值  
i)    s.~string() //销毁所有字符，释放内存  
都很简单，我就不解释了。  
2．字符串操作函数  
   这里是C++字符串的重点，我先把各种操作函数罗列出来，不喜欢把所有函数都看完的人可以在这里找自己喜欢的函数，再到后面看他的详细解释。  
a) =,assign()   //赋以新值  
b) swap()   //交换两个字符串的内容

c) +=,append(),push\_back() //在尾部添加字符

http://acm.nyist.net/JudgeOnline/problem.php?pid=308列题

**[cpp]** [view plain](http://blog.csdn.net/chl_3205/article/details/7522797) [copy](http://blog.csdn.net/chl_3205/article/details/7522797)

[print?](http://blog.csdn.net/chl_3205/article/details/7522797)

1. #include<stdio.h>
2. #include<string>
3. #include<iostream>
4. #include<algorithm>
5. **using** **namespace** std;
6. **int** main()
7. {
8. **int** x,n,max,i,j;
9. string s1,s2,s3;
10. scanf("%d",&x);
11. **while**(x--)
12. {   max=0;
13. cin>> s1;
14. s2=s1;
15. reverse(s2.begin(),s2.end());
16. n=s1.length();
17. **for**(i=0;i<n;i++)
18. {
19. **for**(j=1;j<=n-i;j++)
20. {
21. string::size\_type pos=s2.find(s1.substr(i,j));
22. **if**(pos!=string::npos)
23. {
24. **if**(max<j)
25. {
26. max=j;
27. s3=s1.substr(i,j);
28. }
29. }
30. }
31. }
32. cout<<s3<<endl;
33. }
34. **return** 0;
35. }

d) insert() //插入字符  
e) erase() //删除字符  
f) clear() //删除全部字符  
g) replace() //替换字符  EF BB BF 3C 3F 78 6D 6C  
h) + //串联字符串  
i) ==,!=,<,<=,>,>=,compare()  //比较字符串  
j) size(),length()  //返回字符数量  
k) max\_size() //返回字符的可能最大个数  
l) empty()  //判断字符串是否为空  
m) capacity() //返回重新分配之前的字符容量  
n) reserve() //保留一定量内存以容纳一定数量的字符  
o) [ ], at() //存取单一字符  
p) >>,getline() //从stream读取某值  
q) <<  //将谋值写入stream  
r) copy() //将某值赋值为一个C\_string  
s) c\_str() //将内容以C\_string返回  
t) data() //将内容以字符数组形式返回  
u) substr() //返回某个子字符串  
v)查找函数  
w)begin() end() //提供类似STL的迭代器支持  
x) rbegin() rend() //逆向迭代器  
y) get\_allocator() //返回配置器