[**字符串面试题（一）字符串逆序**](http://www.cnblogs.com/graphics/archive/2011/03/09/1977717.html)

**几点说明**

1. 所有题目全部来自网络，书籍，或者我自己的面试经历，本人只是负责搜集整理。在此对原作者表示感谢！

2. 我已经尽力确保文字及程序的正确性，但我毕竟是凡人，如果您发现了文章中的错误，或者有更好的解法，请一定留言相告，以免误导大家！

3. 所有代码都采用C/C++编写

很早就准备写一个字符串系列的面试题，本来已经写好了，大概有十几道题，但是写完才发现，文章好长，连我自己都没有耐心读下去了，索性就将其拆分成几个系列，一来分开后篇幅变小，看起来比较方便。二来也更有针对性，便于精雕细作。比如这篇，在原来的文章中只占很小的篇幅，但是独立出来才发现，东西也不少。既然是第一篇，就来个最最简单的字符串逆序吧。

字符串逆序可以说是最经常考的题目。这是一道入门级的题目，相信80%的程序员经历过这道题。给定一个字符串s，将s中的字符顺序颠倒过来，比如s="abcd"，逆序后变成s="dcba"。

**普通逆序**

基本上没有这么考的，放在这里主要是为了和后面的原地逆序做个对比。很简单，直接分配一个与原字符串等长的字符数组，然后反向拷贝一下即可。

[复制代码](javascript:void(0);)

char\* Reverse(char\* s)

{

//将q指向字符串最后一个字符

char\* q = s ;

while( \*q++ ) ;

q -= 2 ;

//分配空间，存储逆序后的字符串。

char\* p = newchar[sizeof(char) \* (q - s + 2)] ;

char\* r = p ;

// 逆序存储

while(q >= s)

\*p++ = \*q-- ;

\*p = '\0' ;

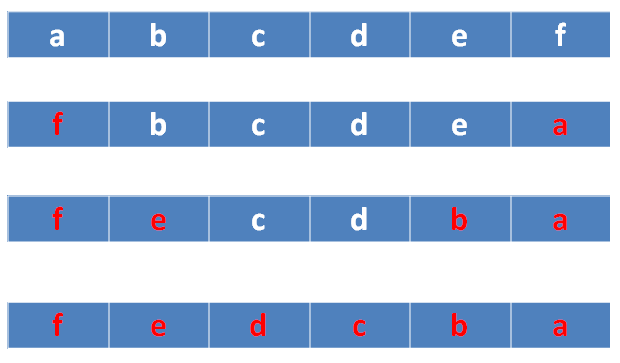
return r ;

}

[复制代码](javascript:void(0);)

原地逆序

英文叫做in-place reverse。这是最常考的，原地逆序意味着不允额外分配空间，主要有以下几种方法，思想都差不多，就是将字符串两边的字符逐个交换，如下图。给定字符串"abcdef"，逆序的过程分别是交换字符a和f，交换字符b和e，交换字符c和d。



一 设置两个指针，分别指向字符串的头部和尾部，然后交换两个指针所指的字符，并向中间移动指针直到交叉。

[复制代码](javascript:void(0);)

char\* Reverse(char\* s)

{

// p指向字符串头部

char\* p = s ;

// q指向字符串尾部

char\* q = s ;

while( \*q )

++q ;

q -- ;

// 交换并移动指针，直到p和q交叉

while(q > p)

{

char t = \*p ;

\*p++ = \*q ;

\*q-- = t ;

}

return s ;

}

[复制代码](javascript:void(0);)

二 用递归的方式，需要给定逆序的区间，调用方法:Reverse(s, 0, strlen(s)) ;

[复制代码](javascript:void(0);)

// 对字符串s在区间left和right之间进行逆序，递归法

void Reverse( char\* s, int left, int right )

{

if(left >= right)

return;

char t = s[left] ;

s[left] = s[right] ;

s[right] = t ;

Reverse(s, left + 1, right - 1) ;

}

[复制代码](javascript:void(0);)

三 非递归法，同样指定逆序区间，和方法一没有本质区别，一个使用指针，一个使用下标。

[复制代码](javascript:void(0);)

// 对字符串str在区间left和right之间进行逆序

char\* Reverse( char\* s, int left, int right )

{

while( left < right )

{

char t = s[left] ;

s[left++] = s[right] ;

s[right--] = t ;

}

return s ;

}

[复制代码](javascript:void(0);)

**不允许临时变量的原地逆序**

**使用异或操作**

[复制代码](javascript:void(0);)

// 使用异或操作对字符串s进行逆序

char\* Reverse(char\* s)

{

char\* r = s ;

//令p指向字符串最后一个字符

char\* p = s;

while (\*(p + 1) != '\0')

++p ;

// 使用异或操作进行交换

while (p > s)

{

\*p = \*p ^ \*s ;

\*s = \*p ^ \*s ;

\*p = \*p-- ^ \*s++ ;

}

return r ;

}

[复制代码](javascript:void(0);)

**按单词逆序**

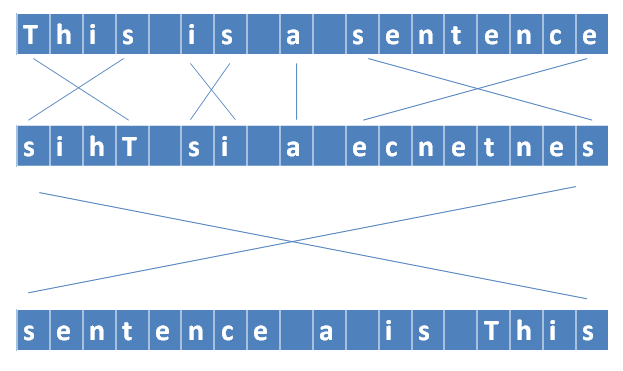
给定一个字符串，按单词将该字符串逆序，比如给定"This is a sentence"，则输出是"sentence a is This"，为了简化问题，字符串中不包含标点符号。

分两步

1 先按单词逆序得到"sihT si a ecnetnes"

2 再整个句子逆序得到"sentence a is This"

对于步骤一，关键是如何确定单词，这里以空格为单词的分界。当找到一个单词后，就可以使用上面讲过的方法将这个单词进行逆序，当所有的单词都逆序以后，将整个句子看做一个整体（即一个大的包含空格的单词）再逆序一次即可，如下图所示，第一行是原始字符换，第二行是按单词逆序后的字符串，最后一行是按整个句子逆序后的字符串。



代码

[复制代码](javascript:void(0);)

// 对指针p和q之间的所有字符逆序

void ReverseWord(char\* p, char\* q)

{

while(p < q)

{

char t = \*p ;

\*p++ = \*q ;

\*q-- = t ;

}

}

// 将句子按单词逆序

char\* ReverseSentence(char\* s)

{

// 这两个指针用来确定一个单词的首尾边界

char\* p = s ; // 指向单词的首字符

char\* q = s ; // 指向空格或者 '\0'

while(\*q != '\0')

{

if (\*q == '')

{

ReverseWord(p, q - 1) ;

q++ ; // 指向下一个单词首字符

p = q ;

}

else

q++ ;

}

ReverseWord(p, q - 1) ; // 对最后一个单词逆序

ReverseWord(s, q - 1) ; // 对整个句子逆序

return s ;

}

[复制代码](javascript:void(0);)

**逆序打印**

还有一类题目是要求逆序输出，而不要求真正的逆序存储。这题很简单，有下面几种方法，有的方法效率不高，这里仅是提供一个思路而已。

先求出字符串长度，然后反向遍历即可。

void ReversePrint(const char\* s)  
{  
    int len = strlen(s) ;  
    for (int i = len - 1; i >= 0; --i)  
        cout << s[i];  
}

如果不想求字符串的长度，可以先遍历到末尾，然后在遍历回来，这要借助字符串的结束符'\0

[复制代码](javascript:void(0);)

void ReversePrint(const char\* s)

{

const char\* p = s ;

while (\*p)

\*p++ ;

--p ; //while结束时，p指向'\0'，这里让p指向最后一个字符

while (p >= s)

{

cout <<\*p ;

--p ;

}

}

[复制代码](javascript:void(0);)

对于上面第二种方法，也可以使用递归的方式完成。

void ReversePrint(const char\* s)  
{  
    if(\*(s +1) != '\0')  
        ReversePrint(s + 1) ;  
    cout << \*s ;  
}

关于数组的面试题，请看[这里](http://www.cnblogs.com/graphics/archive/2010/08/24/1761620.html)

== THE END==

Happy coding!