**程序员面试题精选100题(07)－翻转句子中单词的顺序[算法]**

2007-03-08 09:20:52|  分类： [字符串](http://zhedahht.blog.163.com/blog/#m=0&t=1&c=fks_081075092084086069081074084070080087080066083082) |  标签：[编程](http://zhedahht.blog.163.com/blog/" \l "m=0&t=3&c=编程)  [就业](http://zhedahht.blog.163.com/blog/#m=0&t=3&c=就业)  [找工作](http://zhedahht.blog.163.com/blog/#m=0&t=3&c=找工作)  [微软](http://zhedahht.blog.163.com/blog/#m=0&t=3&c=微软)  [面试题](http://zhedahht.blog.163.com/blog/#m=0&t=3&c=面试题)  |字号 订阅

题目：输入一个英文句子，翻转句子中单词的顺序，但单词内字符的顺序不变。句子中单词以空格符隔开。为简单起见，标点符号和普通字母一样处理。

例如输入“I am a student.”，则输出“student. a am I”。

分析：由于编写字符串相关代码能够反映程序员的编程能力和编程习惯，与字符串相关的问题一直是程序员笔试、面试题的热门题目。本题也曾多次受到包括微软在内的大量公司的青睐。

由于本题需要翻转句子，我们先颠倒句子中的所有字符。这时，不但翻转了句子中单词的顺序，而且单词内字符也被翻转了。我们再颠倒每个单词内的字符。由于单词内的字符被翻转两次，因此顺序仍然和输入时的顺序保持一致。

还是以上面的输入为例子。翻转“I am a student.”中所有字符得到“.tneduts a ma I”，再翻转每个单词中字符的顺序得到“students. a am I”，正是符合要求的输出。

参考代码：

///////////////////////////////////////////////////////////////////////  
// Reverse a string between two pointers  
// Input: pBegin - the begin pointer in a string  
//        pEnd   - the end pointer in a string  
///////////////////////////////////////////////////////////////////////  
void Reverse(char \*pBegin, char \*pEnd)  
{  
      if(pBegin == NULL || pEnd == NULL)  
            return;  
  
      while(pBegin < pEnd)  
      {  
            char temp = \*pBegin;  
            \*pBegin = \*pEnd;  
            \*pEnd = temp;  
  
            pBegin ++, pEnd --;  
      }  
}  
  
///////////////////////////////////////////////////////////////////////  
// Reverse the word order in a sentence, but maintain the character  
// order inside a word  
// Input: pData - the sentence to be reversed  
///////////////////////////////////////////////////////////////////////  
char\* ReverseSentence(char \*pData)  
{  
      if(pData == NULL)  
            return NULL;  
  
      char \*pBegin = pData;  
      char \*pEnd = pData;  
  
      while(\*pEnd != '\0')  
            pEnd ++;  
      pEnd--;  
  
      // Reverse the whole sentence  
      Reverse(pBegin, pEnd);  
  
      // Reverse every word in the sentence  
      pBegin = pEnd = pData;  
      while(\*pBegin != '\0')  
      {  
            if(\*pBegin == ' ')  
            {  
                  pBegin ++;  
                  pEnd ++;  
                  continue;  
            }  
            // A word is between with pBegin and pEnd, reverse it  
            else if(\*pEnd == ' ' || \*pEnd == '\0')  
            {  
                  Reverse(pBegin, --pEnd);  
                  pBegin = ++pEnd;  
            }  
            else  
            {  
                  pEnd ++;  
            }  
      }  
  
      return pData;  
}