

题目一 翻转单词顺序

面试题 58：翻转字符串

题目一：翻转单词顺序。

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。为简单起见，标点符号和普通字母一样处理。例如输入字符串"I am a student."，则输出"student. a am I"。

解

首先翻转句子中所有字符，然后翻转每个单词中的字符。

```
string ReverseSentence(string str)
{
    if(str.empty())
        return "";
    int length=str.size();
    int start=0,end=0;
    Reverse(str,0,length-1);
    int i=0;
    while(i<length)
    {
        while(i<length && str[i]==' ')
            i++;
        start=end=i;
        while(i<length && str[i]!=' ')
        {
            i++;
            end++;
        }
        Reverse(str,start,end-1);
    }
    return str;
}

void reverse(string &str,int start,int end) //引用形式
{
    while(start<end)
    {
        swap(str[start++],str[end--]);
    }
}
```

题目二 左旋转字符串

通过前面的分析，我们发现只需要调用 3 次前面的 Reverse 函数就可以实现字符串的左旋转功能。参考代码如下：

```
char* LeftRotateString(char* pStr, int n)
{
    if(pStr != nullptr)
    {
        int nLength = static_cast<int>(strlen(pStr));
        if(nLength > 0 && n > 0 && n < nLength)
        {
            char* pFirstStart = pStr;
            char* pFirstEnd = pStr + n - 1;
            char* pSecondStart = pStr + n;
            char* pSecondEnd = pStr + nLength - 1;

            // 翻转字符串的前面 n 个字符
            Reverse(pFirstStart, pFirstEnd);
            // 翻转字符串的后面部分
            Reverse(pSecondStart, pSecondEnd);
            // 翻转整个字符串
            Reverse(pFirstStart, pSecondEnd);
        }
    }

    return pStr;
}
```

题目三 翻转字符串里的单词

给定一个字符串，逐个翻转字符串中的每个单词。

示例 1:

输入: "the sky is blue"
输出: "blue is sky the"

示例 2:

输入: " hello world! "
输出: "world! hello"
解释: 输入字符串可以在前面或者后面包含多余的空格，但是反转后的字符不能包括。

示例 3:

输入: "a good example"
输出: "example good a"
解释: 如果两个单词间有多余的空格，将反转后单词间的空格减少到只含一个。

我们的做法是，先整个字符串整体翻转一次，然后再分别翻转每一个单词（或者先分别翻转每一个单词，然后再整个字符串整体翻转一次），此时就能得到我们需要的结果了。那么这里我们需要定义一些变量来辅助我们解题，storeIndex表示当前存储到的位置，n为字符串的长度。我们先给整个字符串反转一下，然后我们开始循环，遇到空格直接跳过，如果是非空格字符，我们此时看storeIndex是否为0，为0的话表示第一个单词，不用增加空格；如果不为0，说明不是第一个单词，需要在单词中间加一个空格，然后我们要找到下一个单词的结束位置我们用一个while循环来找下一个为空格的位置，在此过程中继续覆盖原字符串，找到结束位置了，下面就来翻转这个单词，然后更新i为结尾位置，最后遍历结束，我们剪裁原字符串到storeIndex位置，就可以得到我们需要的结果，代码如下：

```
class Solution {
public:
    string reverseWords(string s) {
        int n=s.size();
        int index=0;
        reverse(s.begin(),s.end());
        for(int i=0;i<n;i++)
        {
            if(s[i]!=' ')
            {
                int j=i;
                if(index!=0)//不是首单词
                {
                    s[index++]=' ';//每两个单词之间的空格
                }
                while(j<n && s[j]!=' ')
                {
                    s[index++]=s[j++];//覆盖
                }
                reverse(s.begin()+index-(j-i),s.begin()+index);
                i=j;
            }
        }
        s.resize(index);
        return s;
    }
};
```