

第2题——左旋字符串

字符串的左旋转操作是把字符串前面的若干个字符转移到字符串的尾部。请定义一个函数实现字符串左旋转操作的功能。比如，输入字符串"abcdefg"和数字2，该函数将返回左旋转两位得到的结果"cdefgab"。

示例 1:

输入: s = "abcdefg", k = 2

输出:"cdefgab"

示例 2:

输入: s = "lrloseumgh", k = 6

输出:"umghlrlose"

限制: $1 \leq k < s.length \leq 10000$

暴力解题法 (1)

```
char* reverseLeftWords(char* s, int n){  
  
    int i=0;  
    int len=strlen(s);  
    char *t=malloc(sizeof(char)*(len+1));  
    for(int i=0;i<(len-n);i++)  
    {  
        t[i]=s[n+i];  
    }  
    for(int i=0;i<n;i++)  
    {  
        t[len-n+i]=s[i];  
    }  
    t[len]='\0';  
    return t;  
}
```

评论区大佬的解法 (2)

```
char* reverseLeftWords(char* s, int n){
    int i=0;
    int len=strlen(s);
    char *t=malloc(sizeof(char)*(len+1));
    while(i<len)
    {
        t[i++]=s[(n+i)%len];
    }
    t[len]='\0';
    return t;
}
```

不得不说，大佬的这个%，用的太他妈的妙了

这个算法的精髓在于， $s[(n+i)\%len]$

如同表中展示，使用%完美解决

$(n+(i++))\%len$	2	3	4	5	6	0	1
i	0	1	2	3	4	5	6
n	2	2	2	2	2	2	2

另外，字符串的结束必须使用 '\0' 否则报错

reverse反转解法（3）

这个方法是在《代码随想录》的大佬那里看到的，非常牛逼

思路：

1.首先，我们C++中的reverse();可以把里面的元素逆置，只需要提供迭代器即可。

####2. 因此，要在N处左旋。我们可以分成三步：

原始	a	b	c	d	e	f	g
逆置a-b	b	a	c	d	e	f	g
逆置c-g	b	a	g	f	e	d	c
逆置首尾	c	d	e	f	g	a	b

```
class Solution {
public:
```

```
string reverseLeftWords(string s, int n) {  
    reverse(s.begin(),s.begin()+n);  
    reverse(s.begin()+n,s.end());  
    reverse(s.begin(),s.end());  
    return s;  
}  
};
```