总分：100，第一题10分，第二题20分，第三题20分，第四题30分，第五题20分

第一题：复杂度非最优扣5分

第二题：两个next数组各占10分，可以是优化版本也可以是非优化版本，next数组出错则得0分

第三题：功能未实现直接0分，复杂度非最优扣5分

第四题：三小问各占10分

第五题：功能未实现直接零分，每个复杂度对应5分，无法处理aacc的情况扣10分

第四章 字符串

1. 回文字符串

判断一个字符串s是否为回文字符串（即正读和倒读相同）。请简要说明实现算法，并指出时间复杂度和空间复杂度。

设置两个指针i和j，分别表示头指针和尾指针。每轮扫描时，头指针往后扫，尾指针往前扫，直到头尾指针重合。要求在每轮扫描的过程中，两个指针指向的字符都相等，则判断s为回文字符。

时间复杂度：O(n)；空间复杂度：O(1)

2. 求出下列模式串的next值

模式串 t=’abcdaabcddaba’；

模式串s=’XYYXYZXZXYYXZ’

1. t=’abcdaabcddaba’；

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| j=0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| t=a | b | c | d | a | a | b | c | d | d | a | b | a |
| nxt=-1 | 0 | 0 | 0 | 0 | 1 | 1 | 2 | 3 | 4 | 0 | 1 | 2 |

优化版本：

-1 0 0 0 -1 1 0 0 0 4 -1 0 2

1. s=’XYYXYZXZXYYXZ’

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| j=0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| t=X | Y | Y | X | Y | Z | X | Z | X | Y | Y | X | Z |
| nxt=-1 | 0 | 0 | 0 | 1 | 2 | 0 | 1 | 0 | 1 | 2 | 3 | 4 |

优化版本：

-1 0 0 -1 0 2 -1 1 -1 0 0 -1 4

3. 反转字符串

已知一个字符串A，由字母数字和“.”组成，存储在定长的数组中。我们需要将该字符串以“.”为分界位置，反转该字符串。例如：portal.pku.edu.cn，转化成cn.edu.pku.portal。设计一个算法，并分析算法的时间复杂度和空间复杂度。

做两次反转：1、将字符串全局反转。例：portal.pku.edu.cn转成：nc.ude.ukp.latorp；2、以“.”为分解点，部分反转。例：进一步cn.edu.pku.portal。

时间复杂度：O(n)，空间复杂度：O(1)

4. 找出下列程序的错误，并解释它为什么是错的。

1. void test1(){

char str[10];

char\* str1 = “0123456789”;

strcpy(str, str1);

std::cout<<str<<’\n’;

}

1. void test2(){

char str[10], str1[10];

for (int i=0; i<10; i++){

str1[i] = ‘a’;

}

strcpy(str, str1);

std::cout<<str<<’\n’;

}

1. void test3(char\* str1){

char str[10];

if (strlen(str1)<=10)

strcpy(str, str1);

std::cout<<str<<’\n’;

}

1) 对于函数test，这里str数组越界。因为字符串长度为10，还有一个结束符’\0’，所以str1字符长度为11，而str数组长度为10。这里越界了，但是并不会报错，无论是编译还是运行都可以正常通过。

2) 对于函数test2，最大的问题是str1没有结束符。因为strcpy的第二个参数是一个字符串常量，该函数是利用第二个参数的结束符来判断是否复制完毕。

3) 对于函数test3，这里也是越界问题，strlen函数得到的字符串除结束符外的长度，如果这里是等于10的话，就会导致str越界了。

5. 删除字符

删除字符串中的“b”和“ac”，需要满足如下的条件：

1) 字符串只能遍历一次

2) 不能够使用额外的空间，即O(1)的空间。

例如：acbac ==> ""；ababac ==> aa；bbbbd ==> d；aaccac ==> “”

请设计算法，得到输出的字符串。

注意aaccac这种情况，删除连在一起的a和c后，本来没有连在一起的a和c也应该删除。首先，在O(1)的空间复杂度的限制下，从字符串中删除某些字符，则需要把不需要删除的字符串移到前面，然后设置一个'\0'表示字符串的结尾。所以，我们设置如下两个指针i和j，i表示扫描字符串所在的位置，j表示当前有效的字符待拷贝的位置。

其次，删除的字符串中有“ac”两个字符，则删除时需要考虑前一个字符串。则分成两种情况，前一个字符为’a’，没有前一个字符或者前一个字符不为’a’。

算法用伪码表示如下：

j=0；

for i=0:(str长度-1)

{

if str[i]不等于b

{

if str[j-1]为a

{

if str[i]等于c

j--;

else

str[j++] = str[i];

}

else{

str[j++] = str[i];

}

}

}

str[j]=’\0’;

上面这种方法本质上是栈的变体，另一种做法是给a计数，碰到字符b对a的计数无影响，碰到c减少计数，碰到其他字符则把对应数目的a写入到结果中，并把当前a的计数重置为0