|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 测试方法 | 测试结果 | 测试结论 |  |
| 1  Main函数是否可以正常运行 | 随机输入字符串观察能否正常输出 |  | 通过 |  |
| 2  检测函数setrandomstring函数能否正常能够使用 | 随机输入字符串观察能否正常输出 |  | 还未检测 |  |
| 3 |  |  |  |  |

测试代码#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define M 10

#define N 35

void setRandomString(String \*s);

void sortString(String\*sArray);

typedef struct

{

char content[N];

int length;

}String;

int main()

{

String string1;//生成一个结构体

setRandomString(&string1 );//设置字符串内容

printf("未排序的字符串为：%s\n长度是：%d\n",string1.content,string1.length);

sortString(&string);//将字符串排好序

printf("已排序的字符串为： %s\n长度是：%d\n",string1.content,string1.length);

return 0;

}

void setRandomString(String \*s)//设置字符串内容

{

srand(time(NULL));

String.length=rand()%(N+1);//生成随机数确定长度

int i=0;

char arr[String.length]={0};

for(i=0;i<String.length;i++)

{

int m=rand()%(M-1)+1;//将生成的随机数放入字符串中

arr[i]='m';

}

}

void sortString(String\*sArray)//将乱序的字符串排列号顺序

{

}

测试结论

完成要求1 2