

Lab01-热身灯光秀实验报告

姓名：王雪洋 学号：21130041034

2024.9.27

1 代码描述

(1) main() 函数如下所示：

```
int main(){
    int n;
    char str[256];
    scanf("%d",&n);
    gets(str); //读入字符串
    lightShow(str,n); //在这里对读入的字符串进行处理
    return 0;
}
```

(2) lightShow() 函数如下所示：

```
void lightShow(char *str,int n){
    //统计两种段的数量
    int X_cnt=0,O_cnt=0;
    char last = 'A'; //定义暂存字符，初值可以是除了“X”和“O”之外的任意
    for(int i=0;str[i]!='\0';i++){
        if(last!=str[i]){ //只要当前字符不同于暂存字符，就认为开始了新
            if(str[i]=='X'){
                X_cnt++;
            }else{
```

```

        O_cnt++;
    }
    last = str[i]; //更新暂存字符，以备下一个字符的对比
}
}
//选择段数少那种字符
char flag = 'O';
if(X_cnt<O_cnt){
    flag = 'X';
}
int cnt = 0,tmp_i=0;
//<2>遍历字符串根据所选字符的位置输出结果
for(int i=0;str[i]!='\0';i++){
    if(str[i]==flag){
        if(i==0||str[i-1]!=flag){
            tmp_i = i;
        }else{
            cnt++;
        }
    }else if(i>=1&&str[i-1]==flag){
        //<3>将指令“请 9 到 9 号同学切换开关”修改为“请 9
        if(cnt==0){
            printf("请%d号同学切换开关",tmp_i+1);
        }else{
            printf("请%d到%d号同学切换开关",tmp_i+1,tmp_i+1);
        }
        if(str[i+1]!='\0'){
            printf("\n");
        }
        cnt=0;
    }
}
}
}

```

该函数分为如下三个部分，我将逐个进行解释：

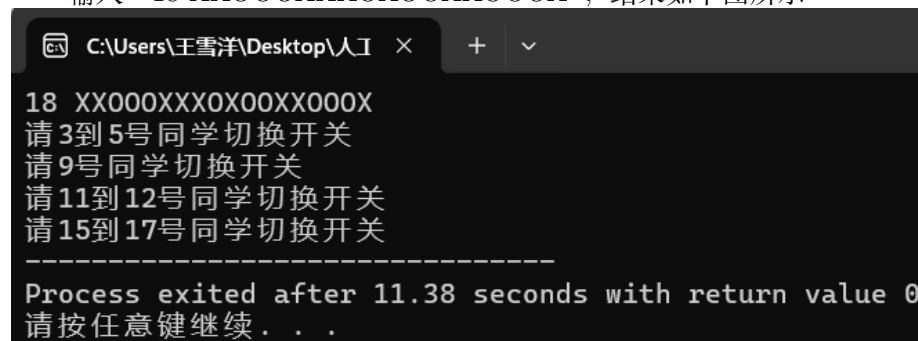
<1> 统计两种段的数量并选择较少的那一种。这里的两种段是指连续的“X”或者“O”形成的分段，例如“XXOOOXXXOXOOXXOOOX”中，连续的“X”有 5 段，连续的“O”有 4 段。通过一次遍历字符串就能得到这两种分段的数量，从而解决题目中“通过 5 次指令点亮了所有的灯。你也可以采用 4 次指令关闭所有的灯，但最佳方案无疑应该是后者。”的要求了。只要选择了段数量少的，就能满足最优化要求。

<2> 遍历字符串根据所选字符的位置输出结果。选择了段数较少的字符“O”之后，再次遍历字符串。遍历过程中，如果当前字符是“O”，那么根据当前位置选择（如果当前位置是字符串首字符或者上个字符不是“O”，说明从这里开始新的一段，记录当前位置为 tmpi，否则说明当前的“O”字符段仍在延续，那么记录跨度的 cnt 就要加一，知道当前字符不再是“O”，此时输出的两个数字分别是 tmpi 和 tmpi+cnt 即为所求结果）。

<3> 将指令“请 9 到 9 号同学切换开关”修改为“请 9 号同学切换开关”。这一步不难，只需要判断跨度 cnt 是否为 0 即可，根据判断结果输出不同格式的语句。

2 实验结果

输入“18 XXOOOXXXOXOOXXOOOX”，结果如下图所示：



```
C:\Users\王雪洋\Desktop\人工 x + v
18 XXOOOXXXOXOOXXOOOX
请3到5号同学切换开关
请9号同学切换开关
请11到12号同学切换开关
请15到17号同学切换开关
-----
Process exited after 11.38 seconds with return value 0
请按任意键继续...
```

3 问题解答

我使用的算法的空间复杂度为 $O(n)$ ，是用于存储字符串的空间。我相信还有更简洁的算法，比如采用双指针提高效率，但是要想将所有代码简化

到 12 行以下还是有点难度的。