



**《程序设计原理》实验报告**

**——问题 C: 时间计算器**

**学 号 3020205094**

**姓 名 刘宏伟**

**学 院 智能与计算学部**

**年 级 2021级**

**任课教师 杜朴风**

**2022年 12月 11日**

# 实验目的

练习字符串的处理，能够判断字符串组成图形的各种对称，同时注意越界等问题。

# 二．实验内容

1. **设计思路**

输入n，后续输入有n行。

接下来n行字符串，用一个char二维数组来存各行的字符串，在输入的同时维护这个图形的四个边界，以实现trim，即过滤掉四周的空格，得到需要判断对称的矩形范围，并且为没输入完全的位置补上空格。

然后对得到的矩形进行对称判断，可以利用对称的特点优化判断过程，节约时间。最终根据判断结果输出分数。

1. **代码实现**（写上必要的注释）

#include <bits/stdc++.h>

using namespace std;

// 定义char二维数组存储每一行字符

char s[100][100];

int main() {

// 定义n并输入n

int n;

cin >> n;

// 定义四个边界以将空白剔除，定义当前行k

int top\_pointer = 100, left\_pointer = 100, bottom\_pointer = -1, right\_pointer = -1, k = 0;

// 吃掉第一个回车

cin.get();

while (k < n) {

cin.getline(s[k], 100);

int sk\_length = strlen(s[k]);

// 表明此行是否是纯空白行，并且以下为维护四个边界

bool space\_flag = true;

for (int i = 0; i < sk\_length; i++) {

if (s[k][i] != ' ') {

if (space\_flag)

space\_flag = false;

if (i < left\_pointer)

left\_pointer = i;

if (i > right\_pointer) {

right\_pointer = i;

}

}

}

for (int i = sk\_length; i < 100; i++)

s[k][i] = ' ';

if (!space\_flag) {

if (k < top\_pointer)

top\_pointer = k;

if (k > bottom\_pointer)

bottom\_pointer = k;

}

k++;

}

// 储存好感度

int ans = 0;

// 通过计算中点,引入奇偶标识来减少判断次数

int mid\_x = (left\_pointer + right\_pointer) / 2;

int mid\_y = (top\_pointer + bottom\_pointer) / 2;

bool odd\_x = (left\_pointer + right\_pointer) % 2 == 0;

bool odd\_y = (top\_pointer + bottom\_pointer) % 2 == 0;

// 分别表明三种对称情况是否有可能成立

bool x = true,

y = true, xy = true;

// 在限定的范围内遍历需要判断的字符

for (int i = top\_pointer; i <= bottom\_pointer; i++) {

for (int j = left\_pointer; j <= right\_pointer; j++) {

if (i <= mid\_y && y) {

if (s[i][j] != s[bottom\_pointer - i + top\_pointer][j] || (odd\_y && 2 \* i + 1 == bottom\_pointer + top\_pointer && s[i][j] != ' '))

y = false;

}

if (j <= mid\_x && x) {

if (s[i][j] != s[i][right\_pointer - j + left\_pointer] || (odd\_x && 2 \* j + 1 == right\_pointer + left\_pointer && s[i][j] != ' '))

x = false;

}

if (j <= mid\_x && xy) {

if (s[i][j] != s[bottom\_pointer - i + top\_pointer][right\_pointer - j + left\_pointer])

xy = false;

}

}

}

// 计算并输出好感度

if (x)

ans += 1;

if (y)

ans += 2;

if (xy)

ans += 3;

cout << ans << endl;

return 0;

}

1. **结果展示**（Oj平台提交记录）



# 三．实验中遇到的问题和解决办法

我一开始使用了string数组来实现，然后发现判断对称时可能会存在越界问题，且难以通过初始化等方式来解决，于是改用预定大小的char数组来解决；还有一个可能的问题就是奇数行/列在判断对称时需要特别注意只有中间为空格才被视为对称。