判断回文字符串实验报告

目录

[一：需求分析 2](#_Toc86410247)

[题目解读 2](#_Toc86410248)

[输入输出 2](#_Toc86410249)

[剖析程序 3](#_Toc86410250)

[二: 概要设计 3](#_Toc86410251)

[问题解决思路 **3**](#_Toc86410252)

[主函数伪代码 **3**](#_Toc86410253)

[class stack 定义 **4**](#_Toc86410254)

[class queue 定义 **4**](#_Toc86410255)

[函数调用关系 5](#_Toc86410256)

[输入处理 5](#_Toc86410257)

[三：调试分析报告 6](#_Toc86410258)

[四：设计实验的回顾讨论 6](#_Toc86410259)

[五：用户使用说明 6](#_Toc86410260)

[六：测试结果 7](#_Toc86410261)

[七：实验结果 7](#_Toc86410262)

题目：判断回文字符串

班级：2020211306 姓名：马天成 学号：2020211376 分工：撰写代码+实验报告

班级：2020211306 姓名：黄洪建 学号：2020211371 分工：撰写代码

班级：2020211306 姓名：马紫薇 学号：2020211392 分工：撰写实验报告

# 一：需求分析

## 题目解读

在键盘上输入一个01字符串，以“#”作为结束标志，判别它是否为回文字符串。

## 输入输出

输入一个以“#”结束的01字符串。

输出“True”或“False”。

输入要求

1. 我们用的是char来存字符，所以输入只能为Ascii码内的字符。

2. 要考虑到溢出问题，所以要用字符串来存，扩大适用性。

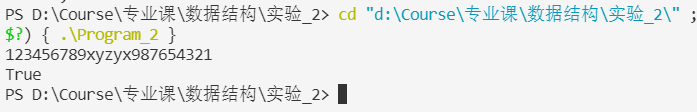
3. 我们不能输入中文字符，因为中文编码一般占2个字节，各个字节存的东西不一样，在

判断时会造成差错。

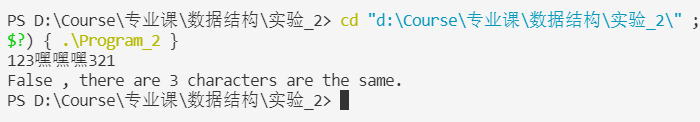
4. 在输入时我们要清除前导0。

5. 字符串中不能出现除01外的字符。

输入输出的正确形式是



输入输出的错误形式是



## 剖析程序

1. 正确处理输入数据。

2. 用字符数组串 ( class stack\_input ) 来存。

3. 将01串从字符数组串 ( class stack\_input ) 导入八进制字符数组 ( class stack\_output )。

4. 输出八进制字符串类的信息，头加0。

# 二: 概要设计

## 问题解决思路

1. 栈先入后出，一次性获得反串；队列先入先出，一次性获得正串。

2. 然后比较正反串，输出T或F并且输出前后可正确匹配的字符数。

## 主函数伪代码

main()

{

字符串存输入 ;

用栈获得反串word\_s ;

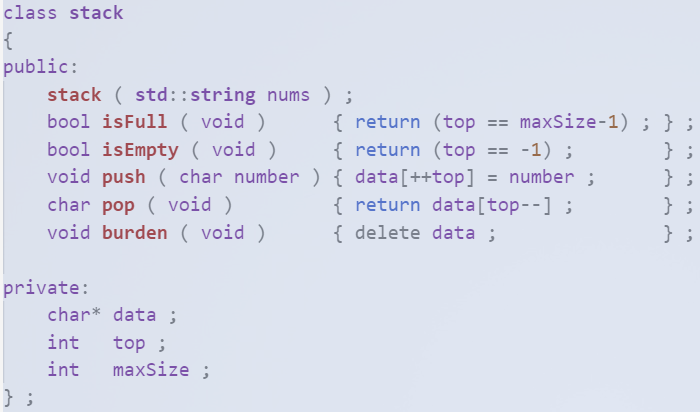
用队列获得正串word\_q ;

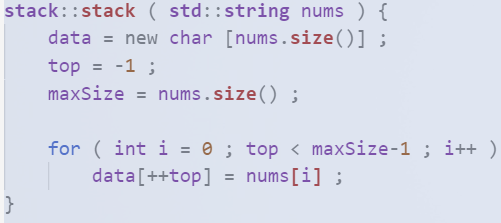
匹配字符串 ;

输出结果 ;

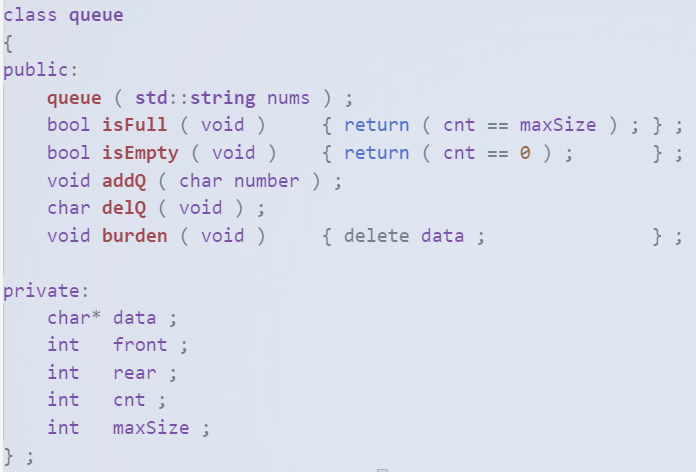
}

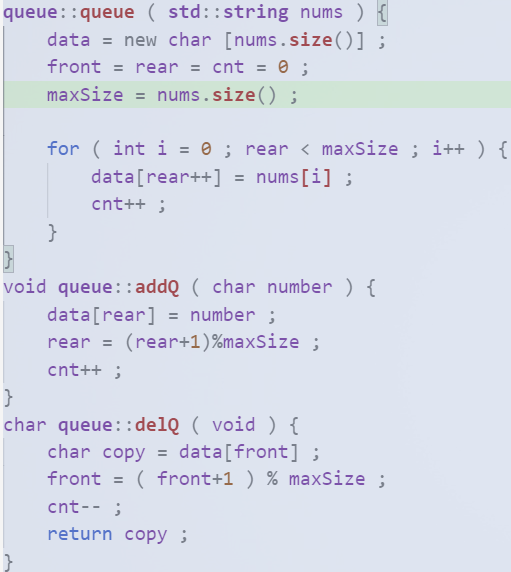
## class stack 定义





## class queue 定义





## 函数调用关系

Main

stack.push()

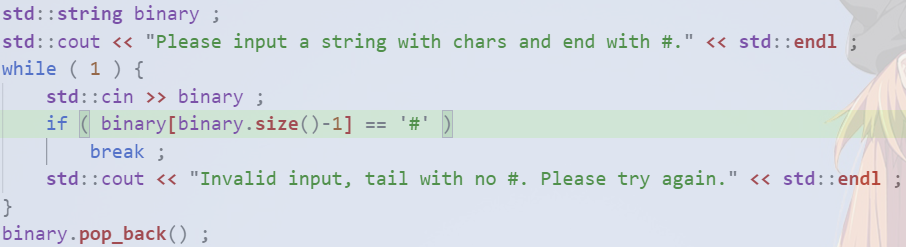
queue.push()

queue.push()

stack.pop()

字符串输入 输出反串 字符串输入 输出正串

## 输入处理



检测末尾是否有’#’，没有，则重新输入。

# 三：调试分析报告

1. 要正确处理string类型，注意字符串的结尾是’\0’。

2. 永远不要相信用户输入，要加入检测机制：结尾必须有’#’。

3. 匹配时下标处理要正确，当不能完全匹配时方便输出可匹配字符数。

# 四：设计实验的回顾讨论

1. 输入

形式：01字符串，以’#’结尾。

正串：队列先入先出，获得所需。

反串：栈先入后出，获得所需。

2. 栈的初始化

采用vector<char> , 有效减少存储空间。

把字符串字符逐个存入。

先入后出，返回反串。

3. 队列初始化注意事项：

采用vector<char> , 有效减少存储空间。

把字符串字符逐个存入。

先入先出，返回正串。

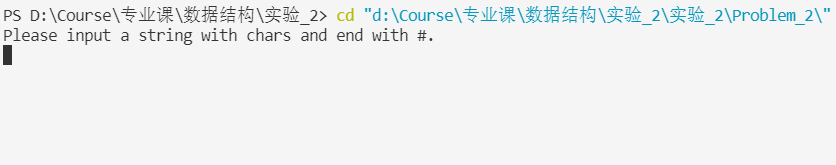
4. 字符串前后能匹配个数没有调整好，一开始输出总是大1。

5. 该程序的时间复杂度为O(n)。

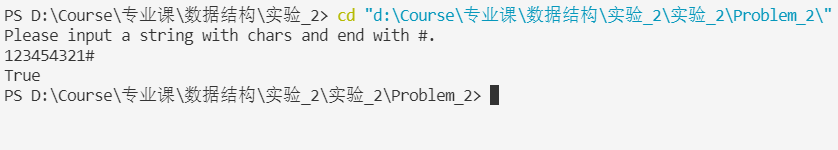
6. 该程序的空间复杂度为O(n)。

# 五：用户使用说明

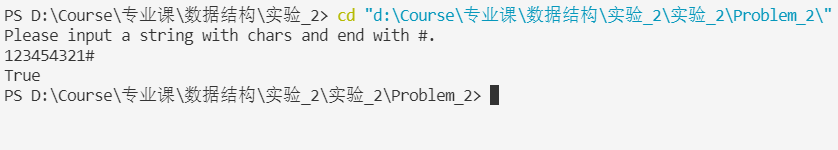
打开程序，首先会出现如下界面：



用户输入合法的n后将会继续显示：



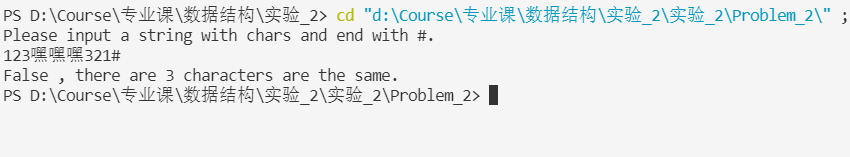
当输入不合法时，会出现：



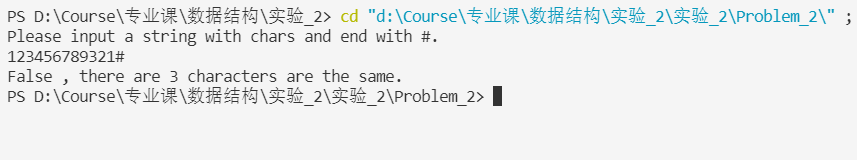
用户可根据提示，重新输入相应数值并回车即可。

# 六：测试结果

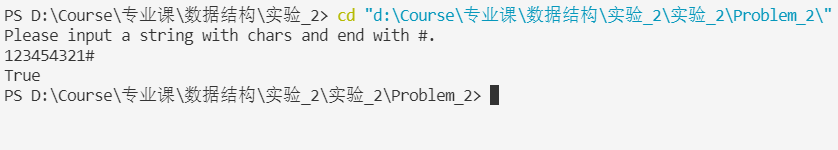
中文



False



True



# 七：实验结果

**实验结论：**

1. 当考虑反序顺序问题时，栈和队列往往是我们重要的工具。

2. 在考虑问题时要全面，永远不要信任用户的输入。

**心得体会：**

通过此次实验，实现了理论与实践的结合，通过解决现实问题，加深了对栈与队伍基本操作的理解，并且借助这两个工具解决了判断回文字符这一问题，然而我们在此实验过程中也遇到了字符串前后个数不匹配问题，但在不断的调试中解决了此问题，并且在随后的小组讨论中，通过交流彼此的思路，开拓了每个人的视野，在彼此的探讨中，还发现了代码中不合理的地方，使每个人都受益匪浅，且程序也得到了进一步的优化，每个人的能力也都得到了一定的提高。