实验4 问题的描述—自定义数据结构

实验报告

姓名：杨耕铭 班级：177717班 学号：17377194

# 1. 实验目的

（1）了解如何用较为复杂数据结构描述问题。

（2）掌握用 Python 语言实现数据结构并解决问题的方法。

# 2. 实验步骤（详述每一个子实验的关键语法及其实现）

（上机课未成功提交的可另附课下Accepted提交评测序号）

2.1 实验任务4-1

关键语法：

1. 学会使用from Stack import Stack导入提供的类。from Stack import Stack语句中第一个Stack对应的是导入的文件名，第二个Stack才是调用Stack类（如果文件名为abc，那么导入的语句就为from abc import Stack）。

2.学会引用已知程序

3.学会使用堆栈结构进行程序编写。

代码：

class Stack:

def \_\_init\_\_(self):

self.items = []

def push(self,item):

self.items.append(item)

def pop(self):

return self.items.pop()

def peek(self):

return self.items[len(self.items)-1]

def isEmpty(self):

return self.items == []

def size(self):

return len(self.items)

a=input()

zhan=Stack()

biao=list(a)

b=1

for c in biao:

if c =='}' or c ==')' or c ==']':

if zhan.isEmpty():

b=2

elif c =='}' and zhan.peek()!= '{':

b=2

elif c ==')' and zhan.peek()!='(':

b=2

elif c ==']' and zhan.peek()!='[':

b=2

else:

zhan.pop()

else :

zhan.push(c)

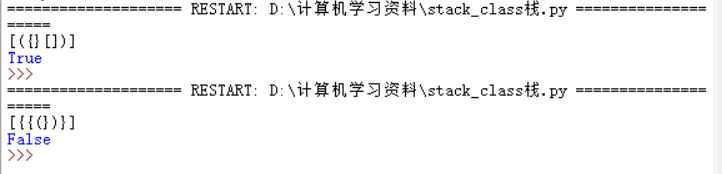
if zhan.isEmpty() and b==1:

print('True')

else :

print('False')

运行结果：

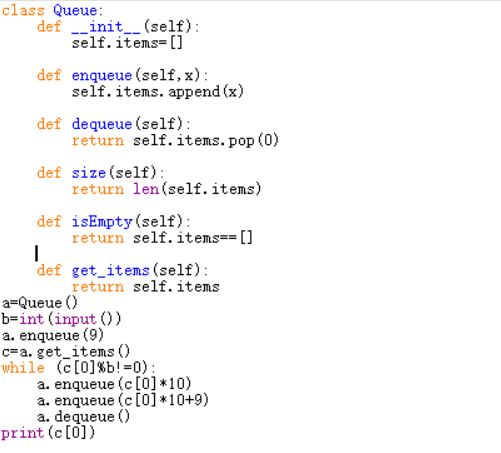


2.2 实验任务4-2（后来提交通过时的序列号是741297）

主要语法：

1. 学会使用from queue\_class import Queue导入提供的类
2. 学会使用队列结构解决问题。

代码：



运行结果：



* 1. 实验任务4-3

主要语法;学会使用Matrix中的八个方法：

\_\_init\_\_方法为构造方法，没有参数，目的是建立一个4x4的零矩阵； get\_pos方法有两个参数x，y，获取矩阵第x行第y列的元素，作为返回值；

set\_pos 方法有三个参数 x，y，value，目的是将矩阵第 x 行第 y 列的元素值改为 value；

initialize有一个参数matlist，为一个列表，保证该列表包含四个子列表，每个子列表包含四个整型数值；

output方法没有参数，目的是将矩阵以某种格式输出至标准输出流（屏幕）； trans方法没有参数，计算矩阵的转置矩阵，作为返回值； plus方法有一个参数m2，计算本矩阵与m2矩阵的和，作为返回值；

multiply方法有一个参数m2，计算本矩阵与m2矩阵的乘积，作为返回值。

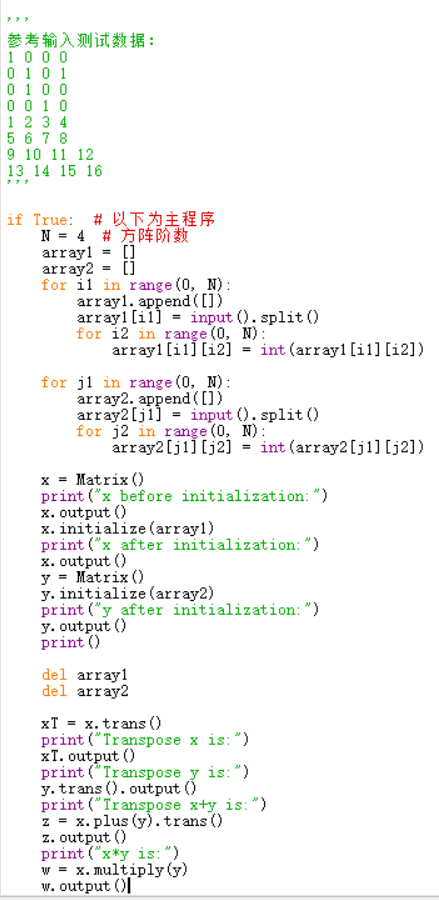
在本题中，你需要补全的为\_\_init\_\_，get\_pos，initialize和multiply四个方法。

为了让解释器在格式处理中不产生问题，需补全的方法函数体都使用了pass语句占位。

pass语句不进行任何运算，没有任何含义。

代码：





运行结果：



# 实验总结（实验心得及体会，以及对本课程的建议等）

这次实验相比上次难度再次大大提高，使得我做起来十分吃力。在课堂上虽然大致明白了什么是堆栈和队列，但在做起题目使用起来仍然不熟练，难以将堆栈队列转化为python中的程序语句。总的来说希望课程能适当降低点难度，使得我们这些零基础的同学能跟上老师的步伐，将题目得心应手的做出来。

# 4. 简答题

（1）简述栈与队列作为数据结构的异同（可举例说明）。

答：相同点：从"数据结构"的角度看，它们都是线性结构，即数据元素之间的关系相同。

不同点：栈是先进后出，而队列是先进先出。  
栈是限定只能在表的一端进行插入和删除操作的线性表。  
队列是只能在表的一端进行插入和在另一端进行删除操作的线性表

（2）类的属性和方法分别代表什么？如何在python中定义方法？\_\_init\_\_方法表示什么，应如何定义和使用？

答：属性：就是类对象的属性，存储某个值，这个值得代号可以称之为类的属性

方法：把每一项类中所有的功能封装起来称为方法，一般方法里的内容就是方法的执行过程。

在python中，方法通过函数的方法定义，其中，这些方法可能使用self作为特殊参数。

init是所有类所提供的构造方法：类似

definit（self， ）其后的括号给出了形参列表，其中self为特殊参数，该参数是指向对象自身的引用，必须为形参的第一项，但在调用时不会有实参的对应。