1. 实现一个二叉树类,并使用递归和非递归算法实现前序遍历、中序遍历、后序遍历。

设计思想：实现节点类，实现二叉树类，设计节点插入，并实现递归和非递归的遍历算法

时间复杂度：O(n)

实验结果：截图如下

设计输入二叉树为：

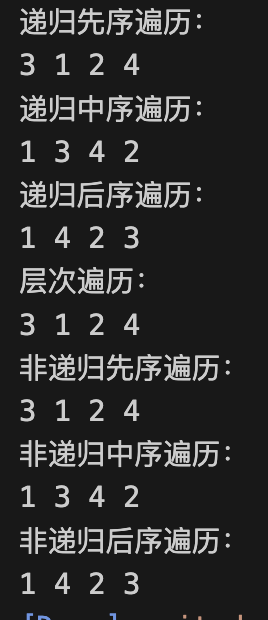
3

/ \

1. 2

\

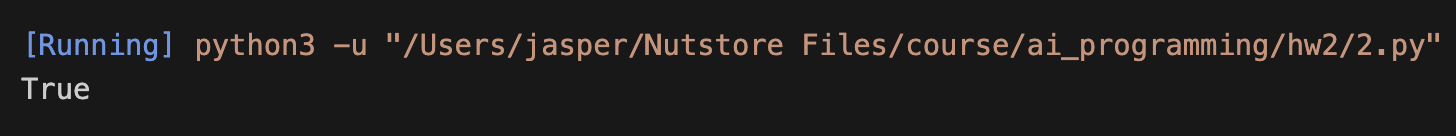
4



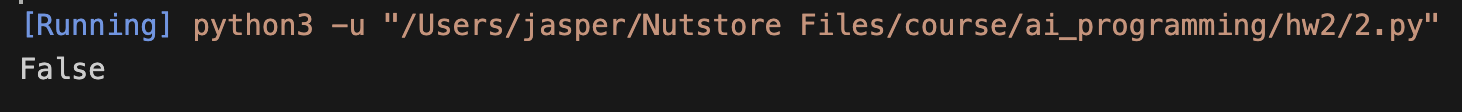
1. 找出一个字符串中是否含有连续的 6 个数字(正则表达式)。

设计思想：使用正则表达式\d判断是否包含连续的6个数字

实验结果：字符串string = 'abc123def4561233'



字符串string = 'abc123def453'

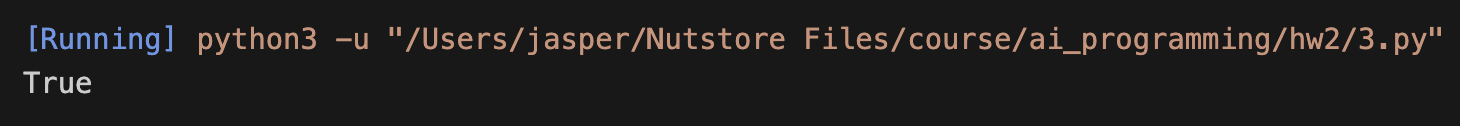


1. 验证输入的字符串是否为一个合法的用户密码,合法的用户密码长度在 6~18 之间,且只能包含英文(大小写均可)和数字。

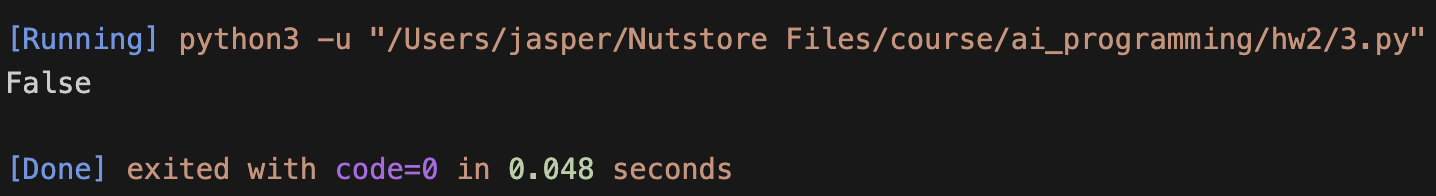
设计思想：使用're.match('^[0-9a-zA-Z]{6,18}$', s):

判断是否在输入最前面到最后面只有英文（无论大小写）和数字并且长度在6-18之间

实验结果：string = 'abc123def4561233'



String = ‘\_abc123def4561233’



4. 请使用装饰器设计模式,编写程序模拟奶茶点单的过程。

1) 用户可以要求买中杯,大杯和超大杯的奶茶,每种杯型价格分别为

6 元, 8 元以及 10 元;

2) 茶底可以选择乌龙茶、绿茶、红茶,价格分别为 2 元、3 元、4 元;

3) 加料可以选择珍珠、椰果、奶盖或者不加,价格分别为 1 元、2 元、

3 元和 0 元,加料可以多选,例如可以同时选择珍珠和椰果;

4) 用户完成奶茶定制后系统会输出用户点的奶茶的杯型、茶底和小料

信息并输出总价。

设计思想：使用字典将名称和价格保存起来，判断输入，找到输入对应的价格，使用装饰器对最后的order\_milk\_tea函数进行装饰，传入price参数，在customize\_milk\_tea执行函数func

实验结果：截图所示

