

中山大学计算机学院本科生实验报告

(2021 学年第 1 学期)

年级	2020	专业 (方向)	网络空间安全
学号	20337251	姓名	伍建霖
电话	13692259882	Email	773542531@qq.com
开始日期	2022.2.27	完成日期	2022.2.27

1. 实验题目

Task 1 ★

给定一个字符串并判断这个字符串是否是回文字符串。回文是一个向前和向后读取相同的字符串，本题中只考虑字母和数字字符。如果是，输出 `True`；否则输出 `False`。

测试用例：

用例1: A man, a plan, a canal: Panama

用例2: race a car

用例3: 1111111111111111111121111111111111111111

2. 实验目的

考察是否能实现 python 基本的流程控制

3. 程序设计

先用 `input` 读入数据并将字符串转成列表 `ans`，接着遍历列表来去掉符号等非
法输入，然后用 `lower` 函数将其全部转成小写，最后判断 `ans` 是否等于 `ans[::-1]`

4. 程序运行与测试

问题	输出	调试控制台	终端	问题	输出	调试控制台	终端
			PS D:\CodeField\AI> conda activa PS D:\CodeField\AI> & 'C:\Users hon-2022.0.1814523869\pythonFile 1a!+*4,a@%,/1 True PS D:\CodeField\AI> d:; cd 'd:\ e\extensions\ms-python.pytho eField\AI\EXP1\1.py' ASDdsa True PS D:\CodeField\AI> █				PS D:\CodeField\AI> d:; cd e\extensions\ms-python.pytho eField\AI\EXP1\1.py' 123123 False PS D:\CodeField\AI> d:; cd e\extensions\ms-python.pytho eField\AI\EXP1\1.py' 213asa312 True PS D:\CodeField\AI> █
问题	输出	调试控制台	终端	问题	输出	调试控制台	终端
			PS D:\CodeField\AI> d:; cd e\extensions\ms-python.pytho eField\AI\EXP1\1.py' /saddas/ True PS D:\CodeField\AI> d:; cd e\extensions\ms-python.pytho eField\AI\EXP1\1.py' ////////1 True PS D:\CodeField\AI> █				PS D:\CodeField\AI> d:; cd e\extensions\ms-python.pytho eField\AI\EXP1\1.py' ;[]'./ True PS D:\CodeField\AI> d:; cd e\extensions\ms-python.pytho eField\AI\EXP1\1.py' qwerewq True PS D:\CodeField\AI> █

1. 实验题目

Task 2 ★

制作一个两人石头剪刀布游戏。（提示：使用 `input` 输入两个玩家的选择，比较它们，打印出祝贺获胜者的消息，并询问玩家是否想开始新游戏，输入 `y` 则重新开始游戏，输入 `n` 退出游戏，石头：Rock，剪刀：Scissors，布：Paper）

示例输入：

```
Player1 input: Rock
Player2 input: Paper
```

示例输出：

```
Congratulate Player2!
Try new game?
```

2. 实验目的

考察是否能实现 python 基本的流程控制

3. 程序设计

一开始先用一个 `while` 循环来判断是否要重新开始游戏，循环内先用一个 `while` 来判断读入的两个变量是否合法，接着判断是平局还是 `s1` 或 `s2` 赢，最后判断是否要重开一把

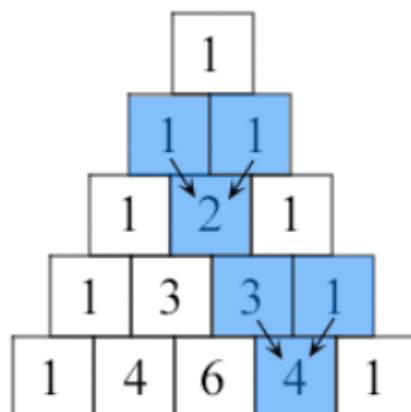
4. 程序运行与测试

问题	输出	调试控制台	终端	问题	输出	调试控制台	终端
			<pre>eField\AI\EXP1\1.py' qwerewq True PS D:\CodeField\AI> d:; cd ' e\extensions\ms-python.pytho eField\AI\EXP1\2.py' Player1 input:Rock Player2 input:rock Player2请输入正确的单词 Player1 input:█</pre>				<pre>PS D:\CodeField\AI> d:; cd e\extensions\ms-python.pytho eField\AI\EXP1\2.py' Player1 input:Rock Player2 input:rock Player2请输入正确的单词 Player1 input:Rock Player2 input:Rock Draw Try new game?y/n █</pre>
问题	输出	调试控制台	终端	问题	输出	调试控制台	终端
			<pre>Player2请输入正确的单词 Player1 input:Rock Player2 input:Rock Draw Try new game?y/n y Player1 input:Rock Player2 input:Scissors Congratulate Player1! Try new game?y/n █</pre>				<pre>Player1 input:Rock Player2 input:Scissors Congratulate Player1! Try new game?y/n y Player1 input:Paper Player2 input:Scissors Congratulate Player2! Try new game?y/n n PS D:\CodeField\AI> █</pre>

1. 实验题目

Task 3 ★★★

实现一个可以打印出帕斯卡三角形前 n 行的函数。（帕斯卡三角形，又称杨辉三角，每个数等于它上方两数之和）



2. 实验目的

考察能否在 python 中实现一个函数

3. 程序设计

杨辉三角形 $L[n][i] = L[n-1][i-1] + L[n-1][i]$

先读入行数 n 并把它转成 `int` 类型，接着一个 `for` 循环，循环 n 次，循环内先打印出 `lisnow`，然后将 `lisnow` 赋给 `lispre` 并清空 `lisnow`，接着再开一个循环，将 `lispre[i-1]+lispre[i]` `append` 进 `lisnow` (i 等于 0 或当前行数时 `append 1`)

```

1  n = int(input())
2  lis_now = [1]
3  for i in range(n):
4      print(lis_now)
5      lis_pre = lis_now
6      lis_now = []
7      for j in range(i+2):
8          if(j != 0 and j != i+1):
9              lis_now.append(lis_pre[j] + lis_pre[j-1])
10         else:
11             lis_now.append(1)

```

4. 程序运行与测试

问题 输出 调试控制台 终端

```

PS D:\CodeField\AI> d:; cd e\extensions\ms-python.pytho
eField\AI\EXP1\3.py'
5
[1]
[1, 1]
[1, 2, 1]
[1, 3, 3, 1]
[1, 4, 6, 4, 1]
PS D:\CodeField\AI>

```

问题 输出 调试控制台 终端

```

e\extensions\ms-python.pytho
eField\AI\EXP1\3.py'
7
[1]
[1, 1]
[1, 2, 1]
[1, 3, 3, 1]
[1, 4, 6, 4, 1]
[1, 5, 10, 10, 5, 1]
[1, 6, 15, 20, 15, 6, 1]
PS D:\CodeField\AI>

```

1. 实验题目

Task 4 ★★

实现一个关于列表的类 MyList，该类包括以下方法：

1. 添加元素：push(args)，参数元素类型为字符串或整型，将 args 添加到列表末尾；
2. 获取元素：get(num)，参数为整型，从列表中随机获取 num 个元素并打印；
3. 获取列表长度：len()，打印列表长度；
4. 删除元素：del()，删除列表第一个元素并打印；
5. 清空列表：clear()，清空列表元素。

在代码中实例化类并测试每个方法，并打印每次操作后的列表。

示例：

```
temp_list = MyList([123, 'abc'])
temp_list.add(456)
print(temp_list.memory) # 这里假设类 MyList 有一个属性 memory 来记录列表
temp_list.get(3)
...
```

2. 实验目的

考察能否在 python 中实现类

3. 程序设计

Push 函数：借助 list 列表的 append 函数

Get 函数：先判断 num 是否大于 len(self.mylist)，接着输出前 num 个元素

Len 函数：借助列表的 len 函数

Del 函数：利用 list 的切片操作，self.mylist = self.mylist[1:]

Clear 函数：可以用切片来清空，也可以直接赋为空列表

4. 程序运行与测试

```
eField\AI\EXP1\4.py'  
after __init__: [[1, 2, 3, 'asd']]  
after push_list_str: [[1, 2, 3, 'asd'], 'iop']  
after push_list_int: [[1, 2, 3, 'asd'], 'iop', 999]  
after get:[1, 2, 3, 'asd'] iop 999  
after len_list:3  
['iop', 999]  
after del_list: ['iop', 999]  
after clear_list: []  
PS D:\CodeField\AI> □
```

实验总结与心得

从这次实验中，我学习到了如何实现 python 的基本流程控制，python 中的基本数据结构（列表，元组，字典，集合）以及其中的一些常用函数比如 append，还有切片操作等等，了解 python 和 c++ 的差异（数组，遍历等等操作在 python 有哪些不同），还实现了类及其函数