姓名：马永龙

手机号：18717891157

（注：请独立完成，发现抄袭情况直接fail）

1. 实现一个数组排序（20分）

#在工作中一般用这个

# list1 = [6, 3, 5, 2, 12, 1, 9]

# print(sorted(list1))

def bubbleSort(nums):

n = len(nums)

# 遍历所有数组元素

for i in range(n):

# Last i elements are already in place

for j in range(0, n - i - 1):

if nums[j] > nums[j + 1]:

nums[j], nums[j + 1] = nums[j + 1], nums[j]

list1 = [6, 3, 5, 2, 12, 1, 9]

bubbleSort(list1)

1. 如下表展示了部分学生的各科目学习成绩信息，请写出sql语句找出所有科目平均分大于60分的学生ID：（20分）

|  |  |  |  |
| --- | --- | --- | --- |
| Id | student\_id | course\_id | score |
| 1 | 2 | 1 | 89 |
| 2 | 4 | 2 | 56 |
| 3 | 6 | 1 | 91 |
| 4 | 4 | 3 | 80 |
| 5 | 2 | 1 | 61 |
| ... | ... | ... | ... |

表名：info

select \* from info;

select student\_id,avg(score) from info group by course\_id having avg(score) > 60;

#最近工作中没有遇到sql，本地没有环境，随便写了一下，没有验证

1. 找出一个长字符串中连续重复次数最多的字符，如：在“nddemdereeeffetcssssfggddregg” 中‘s’重复出现次数（4次）最多，因此“s”为连续重复出现最长的字符。（20分）

s = 'nddemdereeeffetcssssfggddregg'

def maxRepeatStr(str1):

if not str1:

return str1

# 去重后的每个元素的重复字符串可能的最大长度

len\_up\_border = len(str1) - len(set(str1)) + 1

# 缓存遍历，最大长度和当前元素（分别初始化为1和最大ASCII码对应的字符）

max\_len = 1

res = {}

# 长度反向遍历的跳出标识

flag = False

for i in range(len\_up\_border, 0, -1):

for s in set(str1):

spt = s \* i

splited = str1.split(spt)

if len(splited) > 1:

max\_len = max(max\_len, i)

res[i] = s

flag = True

if flag:

break

print(res)

output = [v \* k for k, v in res.items()]

print(output[0])

maxRepeatStr(s)

1. (**用递归的方式实现**，40分)

现有一个json，在某一维度中可能会出现‘score’键，请用任意语言写一个check方法判断该json是否满足要求。要求如下：

a. 任意一个键为score的值预期是一个整型数字且大于60，则检查通过

b. 若不存在键为score的值，则检查通过

json参考样例：



import json

path = '/home/myl/workspace/JavascriptDemo/test.json'

with open(path,'r',encoding='utf-8') as f:

content = json.load(f)

print(content)

def get\_dict\_value(now\_dict, target\_key, results=[]):

for key in now\_dict.keys(): # 当前迭代的字典

data = now\_dict[key] # 当前key所对应的value赋给data

if isinstance(data, list): # 如果是data是list，就遍历list，再递归处理

for i in data:

get\_dict\_value(i, target\_key, results=results)

if isinstance(data, dict): # 如果data是一个字典，就递归遍历

get\_dict\_value(data, target\_key, results=results)

if key==target\_key and isinstance(data, dict) != True: # 找到了目标key，并且它的value不是一个字典

results.append(now\_dict[key])

return results

res = get\_dict\_value(content, 'score', results=[])

print(res)

if len(res) == 0:

print("pass")

for i in res:

if i > 60:

print("pass")

break