Python程序设计

实验 函数（二）

```
# 打印你的学号和姓名
print('学号：2018302100035')

print('姓名：梁皓然')
```

```
学号：2018302100035
姓名：梁皓然
```

1. 尝试运行下面的语句，观察结果，理解相关代码。如果有错，则思考原因。

```
print(eval("5/2+5%2+5//2"))
```

```
5.5
```

```
counter = 1; num = 0
def testVariables():
    global counter
    for i in (1,2,3):
        counter += 1
    num = 10

testVariables()
print(counter, num)
```

```
4 0
```

```
print(type(lambda x: x*x))
```

```
<class 'function'>
```

```
f = lambda x,y:x*y
f(12,34)
```

```
408
```

```
f1 = lambda x:x*2
f2 = lambda x:x**2
print(f1(f2(2)))
```

```
8
```

```
def f3(a,b,c):print(a+b)
nums = (1,2,3)
f3(*nums)
```

```
3
```

```
d = lambda p: p*2; t = lambda p: p*3
x = 2; x = d(x); x = t(x); x = d(x)
print(x)
```

```
24
```

```
i = map(lambda x:x**2, (1,2,3))
for v in i: print(v, end=' ')
```

```
1 4 9
```

```
def f4(vlist):
    if len(vlist)==1: return vlist[0]
    else: return vlist[0] + f4(vlist[1:])

print(f4([1,2,3,4,5,6,7,8]))
```

```
36
```

2. 编写程序，定义一个查找函数search(dataset, k)，在 dataset 列表中查找指定的数 k 出现的所有位置，然后编写测试代码，产生一组随机整数，由用户输入要找的数，使用search函数查找，并输出查找结果。

```
# 随机生成一个包含10个0-9整数的数组
import numpy as np
lst = np.random.randint(10, size=10)

# 将数值列表转换成字符串列表，即给每个元素加上单引号
dataset=[]
for i in lst:
    dataset.append(str(i))

print(dataset)
```

```
['6', '1', '6', '2', '9', '8', '8', '7', '0', '1']
```

```
# 借助enumerate函数来获取指定的数 k 出现的所有位置
def search(dataset=None, k=' '):
    return [index for (index,value) in enumerate(lst) if value == k]
```

```
# 测试代码
search(dataset,8)
```

```
[5, 6]
```

3. 编写并测试一个递归函数mymax，它找出列表中最大的数字。

提示：mymax是第一个数据项和所有其他数据项的最大值中较大的一个。例如，mymax([23,12,64,8,39])的返回值为64。

```
def mymax(list):
    if len(list) == 2:
        if list[0]>list[1]:
            return list[0]
        else:
            return list[1]
    sub_max = max(list[1:])
    if list[0] > sub_max:
        return list[0]
    else:
        return sub_max
mymax([1,2,5,3,0])
```

```
5
```

4. 你可以编写更多程序。（选做）

（1）“回文”是顺着读或倒着读含有相同顺序的字母的句子，一个典型的例子是“Able was I, ere I saw Elba”。写一个函数来检测一个字符串是不是回文。

提示：如果打算用递归方法，其基本思想是检查字符串的第一个和最后一个字母是否相同；如果相同，那么如果这两个字母之间的所有内容都是回文，它就是回文。

有两种特殊情况要检查。如果字符串的第一个或最后一个字符不是字母，你可以检查该字符串删除该字符，其余部分是不是回文。此外，在比较字母时，请确保不区分大小写。

在程序中使用你的函数，提示用户输入短语，然后指出它是不是回文。另一个经典的测试是“A man, a plan, a canal, Panama!”

```
# 定义一个函数判断一个字符串是不是回文
def is_palindrom(s):
    """判断回文数，递归法"""
    if len(s) < 2:
        return True
    if s[0] == s[-1]:
        return is_palindrom(s[1:-1])
    else:
        return False
```

```
s = input('请输入字符串：')
```

```
请输入字符串：A man, a plan, a canal, Panama!3322
```

```
# 去除字符串中的标点符号
s1 = ''.join(e for e in s if e.isalnum())
print(s1)
```

```
AmanaplanacanalPanama3322
```

```
# 去除字符串中的数字
from string import digits
remove_digits = str.maketrans('', '', digits)
res = s1.translate(remove_digits)
print(res)
```

```
AmanaplanacanalPanama
```

```
# 将字符串中的所有大写字母转换为小写字母
res1 = str.lower(res)
print(res1)
```

```
amanaplanacanalpanama
```

```
# 测试函数
print(is_palindrom(res1))
```

```
True
```

（2）计算机科学家和数学家经常使用 10 以外基数的进制系统。编写一个程序，允许用户输入一个数字和一个基数，然后打印出新基数中的数字。使用递归函数baseConversion(num, base)打印数字。

提示：考虑基数 10。要获得基数 10 时最右边的数字，只需除以 10 后查看余数。例如，153 % 10 是 3。要获取剩余的数字，你可以对 15 重复该过程， 15 是 153 // 10。这个过程适用于任何基数。唯一的问题是要以相反的顺序得到数字（从右到左）。当 num 小于 base 时会发生递归的基本情况，输出就是 num。在一般情况下，函数（递归）打印 num // base 的数字，然后打印 num % base。你应该在连续输出之间放置一个空格，因为基数大于 10 时，会打印出多个字符的“数字”。例如，baseConversion(1234, 16)应打印 4 13 2。

```
# 递归实现进制转换:10进制~任意(2-16)进制
def baseConversion(nums, base):
    digits = '0123456789ABCDEF'
    if nums < base:
        return digits[nums]
    else:
        return baseConversion(nums // base, base) + digits[nums % base]
```

```
# 测试函数
str = baseConversion(1234, 16)
```

```
# 在连续输出之间放置一个空格
import re
text_list = re.findall(".{1}",str)
new_text = " ".join(text_list)
print(new_text)
```

```
4 D 2
```

```
# 打印结果
print(new_text.replace('A', '10').replace('B', '11').replace('C',
'12').replace('D', '13').replace('E', '14').replace('F', '15'))
```

```
4 13 2
```