第11周 Python实验2--参考答案

组合数据类型之列表

```
班级：                姓名：                学号：
```

要求：

1. 在文件开头部分填写自己的信息；
2. 在每个题下方的代码块中书写该题的代码，并运行出结果；
3. 在2节课的时间内完成前6个题；打印为pdf文件并提交，文件名改为“Python实验2+班级姓名.pdf”。

一、实验目的：

- 掌握列表的创建及各种操作方法；
- 掌握列表的一些常用内置函数；
- 掌握列表推导式的用法；
- 能在程序设计中合理使用列表解决问题。

二、实验内容（在各题后书写程序和运行结果）

1. 已知两个列表 eng=['morning', 'noon', 'afternoon', 'evening'], chn=['上午','中午', '下午', '晚上'], 将这两个列表中的元素依次（中文在前，英文在后）追加到一个空列表中，形成一个新的列表words。

In [1]:
```
# 以循环形成新列表
eng=['morning', 'noon', 'afternoon', 'evening']
chn=['上午','中午', '下午', '晚上']
words = []
for i in range(len(chn)):
    words.append(chn[i])
    words.append(eng[i])
words
```

Out[1]: ['上午', 'morning', '中午', 'noon', '下午', 'afternoon', '晚上', 'evening']

2. 用list()函数能分离字符串的特点，求所有的水仙花数：即一个3位数的各位数字的立方和等于该数本身。

In [3]:
```
for k in range(100,1000):
    nums = [int(i) for i in list(str(k))]
    sum_nums = sum([x**3 for x in nums])
    if k==sum_nums:
        print(k,'是水仙花数')
```

```
153 是水仙花数
370 是水仙花数
371 是水仙花数
407 是水仙花数
```

In [6]:
```
# 因字符串本身可以作为迭代对象，所以上面代码的第2行可以简化，即不需要list函数
for k in range(100,1000):
    nums = [int(i) for i in str(k)]
    sum_nums = sum([x**3 for x in nums])
    if k==sum_nums:
        print(k,'是水仙花数')
```

```
153 是水仙花数
370 是水仙花数
371 是水仙花数
407 是水仙花数
```

3. 对任意输入的一个正整数，选出是奇数的各位数字。

In [5]:
```
n = input('请输入一个任意的正整数n=')
nums = [int(k) for k in n]  # 分离各位数
odd = [x for x in nums if x%2]
odd
```

```
请输入一个任意的正整数n=4873698734346816
```

Out[5]: [7, 3, 9, 7, 3, 3, 1]

4. 回文数是指一个数字从左边读和从右边读的结果是一模一样的,比如12321。从键盘任意输入一个正整数，判断其是否为回文数，使用列表完成该问题。

In [13]:
```
# 以下纯用循环完成判断
n = int(input('请输入一个正整数，n='))
nums = []
n1 = n  # 记录n的值，以备后续使用
while n>0:
    k = n%10  # 取n的末位数
    nums.append(k)  # 将n的各位数从后往前存入nums中
    n = (n-k)/10  # n中去掉末位数
flag = True
for i in range(len(nums)):
    if nums[i]!=nums[-i-1]:  # 前后对比
        flag = False
        break
if flag:
    print(n1, '是回文数')
else:
    print(n1, '不是回文数')
```

```
请输入一个正整数，n=23532
23532 是回文数
```

In [11]:
```
n = input('请输入一个正整数，n=')
n1 = [int(x) for x in n]  # 取出正整数n的各位数字
n2 = n[::-1]  # 列表n1的逆序
if n1==n2:
    print(n + '是回文数')
else:
    print(n + '不是回文数')
```

```
请输入一个正整数，n=23532
23532是回文数
```

In [10]:
```
n = input('请输入一个正整数，n=')
if n==n[::-1]:   # 字符串n的逆序
    print(n + '是回文数')
else:
    print(n + '不是回文数')
```

```
请输入一个正整数，n=23532
23532是回文数
```

In [3]:
```
n = input('请输入一个正整数，n=')
s = '是回文数' if n==n[::-1] else '不是回文数'
print(n + s)
```

```
请输入一个正整数，n=23532
23532是回文数
```

5. 矩形法是数值方法求定积分的最基本方法。即 $\int_a^b f(x)dx \approx \sum_{k=0}^{n-1} f(x_k)\cdot\frac{b-a}{n}$，这里 n 为区间$[a,b]$ 的等分数。试用该方法求 $\int_0^\pi \sin x dx$的近似值，取 $n=100.$

In [1]:
```
# 以下用循环完成
from math import pi, sin

a,b = 0, pi
n = 100
h = (b-a)/n
S = 0
for k in range(n):
    S += sin(a+h*k)
S *= h
print('定积分的近似值为:', S)
```

```
定积分的近似值为: 1.9998355038874436
```

In [3]:
```
# 以下用列表推导式
from math import pi, sin

a,b = 0, pi
n = 100
h = (b-a)/n
sinx = [sin(a+k*h) for k in range(n)]
S = sum(sinx)*h
print('定积分的近似值为:', S)
```

```
定积分的近似值为: 1.9998355038874436
```

6. 若一个数等于它的各个真因子之和，则称该数为完数，如6=1+2+3，所以6是完数。编写程序求[1,500]之间的全部完数，并按如下格式输出：

   6 = 1+2+3 是完数

In [4]:

```
# 以下的程序同时输出完数的所有真因子，这里用列表保存真因子

for n in range(1,501):
    factors = []
    for f in range(1,n//2+1):
        if n%f==0:
            factors.append(f)
    if sum(factors)==n:
        print(n,end=' = ')
        for f in factors[:-1]:
            print(f, end='+')
        print(factors[-1],'是完数') # 最后一个单独输出
```

```
6 = 1+2+3 是完数
28 = 1+2+4+7+14 是完数
496 = 1+2+4+8+16+31+62+124+248 是完数
```

In [5]:

```
# 使用列表推导式求真因子

for n in range(1,501):
    factors = [f for f in range(1,n//2+1) if n%f==0]
    if sum(factors)==n:
        print(n,end=' = ')
        for f in factors[:-1]:
            print(f, end='+')
        print(factors[-1],'是完数') # 最后一个单独输出
```

```
6 = 1+2+3 是完数
28 = 1+2+4+7+14 是完数
496 = 1+2+4+8+16+31+62+124+248 是完数
```

In [6]:

```
# 如果只是求全部的完数，则用嵌套的列表推导式比较简单

[n for n in range(1,501) if n==sum([f for f in range(1,n//2+1) if n%f==0])]
```

Out[6]: [6, 28, 496]

7. （选做题）使用random模块的randint函数产生[1,10]上的100个随机整数，放到一个列表中，并输出各不同的数字及其出现次数。

In [7]:

```
# 以下使用循环记录列表中的不同数字的出现频数
from random import randint

rands = []
for k in range(100): # 产生100个随机数
    rands.append(randint(1,10))
print(rands)
nums = [0*x for x in range(10)] # 记录频数，初始化为0
for k in rands:
    nums[k-1]+=1
for i in range(1,11):
    print('数字%d在列表中出现了%d次'%(i, nums[i-1]))
```

```
[2, 3, 7, 8, 2, 4, 8, 2, 9, 1, 5, 6, 10, 1, 4, 2, 5, 10, 5, 4, 2, 8, 10, 2, 6, 2, 6, 2, 4, 5, 3, 3,
7, 10, 7, 7, 5, 5, 7, 3, 9, 3, 4, 10, 2, 4, 4, 10, 10, 8, 9, 10, 7, 8, 1, 10, 8, 8, 3, 6, 5, 4, 9, 6,
10, 8, 10, 2, 4, 3, 6, 4, 6, 4, 1, 6, 10, 7, 8, 6, 2, 1, 6, 8, 7, 6, 4, 3, 4, 5, 3, 10, 4, 7, 10, 8,
2, 3, 8, 8]
数字1在列表中出现了5次
数字2在列表中出现了12次
数字3在列表中出现了10次
数字4在列表中出现了14次
数字5在列表中出现了8次
数字6在列表中出现了11次
数字7在列表中出现了9次
数字8在列表中出现了13次
数字9在列表中出现了4次
数字10在列表中出现了14次
```

In [8]:

```
# 以下利用列表的方法 .count()对列表元素计数
from random import randint

rands = [randint(1,10) for k in range(100)] # 产生100个随机数
print(rands)
for i in range(1,11):
    print('数字%d在列表中出现了%d次'%(i, rands.count(i)))
```

```
[4, 10, 8, 3, 3, 5, 6, 6, 9, 4, 10, 5, 9, 5, 4, 9, 5, 8, 5, 1, 2, 8, 3, 6, 3, 10, 1, 8, 4, 9, 7, 3,
5, 4, 3, 6, 9, 7, 6, 10, 2, 7, 8, 4, 2, 5, 1, 2, 8, 2, 7, 2, 10, 10, 4, 3, 8, 10, 4, 2, 7, 4, 5, 5,
6, 7, 2, 10, 10, 2, 1, 8, 9, 8, 10, 8, 8, 5, 6, 1, 1, 4, 1, 4, 1, 8, 6, 9, 7, 7, 7, 3, 1, 9, 5, 4, 5,
4, 5, 3]
数字1在列表中出现了9次
数字2在列表中出现了9次
数字3在列表中出现了9次
数字4在列表中出现了13次
数字5在列表中出现了13次
数字6在列表中出现了8次
数字7在列表中出现了9次
数字8在列表中出现了12次
数字9在列表中出现了8次
数字10在列表中出现了10次
```

8. （选做题）今天是11月12日，星期二。编写程序，对本月的任一日期，输出对应的星期。

In [11]:

```
days = '一二三四五六日'
today = (12, '二')
r0 = days.index(today[1])- today[0]%7  # 计算初始日(即第0天)对应的星期序号
for date in range(1,31):
    r = (date+r0)%7
    print(date,'日：','星期'+days[r])
```

```
1 日： 星期五
2 日： 星期六
3 日： 星期日
4 日： 星期一
5 日： 星期二
6 日： 星期三
7 日： 星期四
8 日： 星期五
9 日： 星期六
10 日： 星期日
11 日： 星期一
12 日： 星期二
13 日： 星期三
14 日： 星期四
15 日： 星期五
16 日： 星期六
17 日： 星期日
18 日： 星期一
19 日： 星期二
20 日： 星期三
21 日： 星期四
22 日： 星期五
23 日： 星期六
24 日： 星期日
25 日： 星期一
26 日： 星期二
27 日： 星期三
28 日： 星期四
29 日： 星期五
30 日： 星期六
```

In [27]:

```
days = '一二三四五六日'
for date in range(1,31):
    ind = (date-(12-1))%7  # 12日对应到days中的序号1（星期二）
    print(date, '日：', '星期'+days[ind])
```

```
1 日： 星期五
2 日： 星期六
3 日： 星期日
4 日： 星期一
5 日： 星期二
6 日： 星期三
7 日： 星期四
8 日： 星期五
9 日： 星期六
10 日： 星期日
11 日： 星期一
12 日： 星期二
13 日： 星期三
14 日： 星期四
15 日： 星期五
16 日： 星期六
17 日： 星期日
18 日： 星期一
19 日： 星期二
20 日： 星期三
21 日： 星期四
22 日： 星期五
23 日： 星期六
24 日： 星期日
25 日： 星期一
26 日： 星期二
27 日： 星期三
28 日： 星期四
29 日： 星期五
30 日： 星期六
```

9. （选做题）16位银联信用卡的最后一位为校验码，采用的是LUHN算法，也叫模10算法。其计算方法为：

（1）从卡号最后一位数字开始，逆向将奇数位(1、3、5等等)相加。

（2）从卡号最后一位数字开始，逆向将偶数位数字，先乘以2，再将其各位数求和。

（3）将奇数位总和加上偶数位总和，结果能被10整除。

请你编程产生10个信用卡号：卡号的前6位固定为625965，接下来的9位从0~9中随机取值，然后依LUHN算法产生最后一位检验码。输出这10个16位卡号。

In [4]:

```python
from random import randint

for i in range(10):
    s1 = [6,2,5,9,6,5]
    s2 = [randint(0, 9) for k in range(9)]
    s = s1 + s2  # 前15位
    p1 = sum(s[-2::-2]) # 16位卡号（去掉末位）的逆向奇数位相加
    p2 = 2*sum(s[-1::-2]) # 16位卡号的逆向偶数位相加
    t = (-p1-p2) % 10 # 使 p1+p2+t能被10整除
    cardid = ''.join(map(str, s1 + s2 + [t])) # 卡号
    print(cardid)
```

```
6259659664705907
6259658572677430
6259653921892803
6259650096639681
6259657599560074
6259654670519574
6259656762192854
6259658782697749
6259659895962061
6259657828505189
```

In [19]:

```python
# 上面的第9行可以如下验证：

print((-16-8)%10)
print((-16-24)%10)
```

```
6
0
```

In [ ]:

```python

```