

苏州大学实验报告

院、系	计算机学院	年级专业	计算机科学	姓名	柯骅	学号	2027405033
课程名称	Python 程序设计					成绩	
指导教师	李正华	同组实验者		实验日期			

实验名称 实验三 列表

一. 实验目的

通过本次实验要达到如下目的：

1. 掌握列表这种数据结构的基本概念
2. 掌握列表的常用操作和常用方法，并理解相应的编程应用过程
3. 掌握序列数据的常用操作和常用方法，并理解相应的编程应用过程

二. 实验内容（Python 习题集中的题目、老师额外布置的题目，也可以写进来，会适当加分）

1. 求无序整数列表的中位数。如列表元素为偶数个，则取列表升序排列时中间两数中数值较小的元素为中位数。
2. 已知一个整数列表，判断列表内容是否为回文，即无论正序还是倒序，列表的内容是否相同。
3. 已知一个整数列表，筛选出该列表中不同的质数，并求出该列表中有多少个质数可以表达为该列表中另外两个质数的和。
4. 用筛法求 500 之内的所有质数，并打印输出所有的质数，每行输出 5 个质数。
5. 现有列表[35, 46, 57, 13, 24, 35, 99, 68, 13, 79, 88, 46]，请编写程序将其中重复的元素去除，并按从小到大的顺序排列后输出。35 46 57 13 24 35 99 68 13 79 88 46
6. 编写程序让用户输入两个字符串（一定是小写字母组成），判断两个字符串是否同构。如果有两个字符串，其中一个字符串的字符重新排列后，能变成另一个字符串，那么称为同构。
7. 编写程序对一个 4*4 的矩阵进行随机赋值，然后对该矩阵进行转置，并输出转置后的结果。
8. 现有 5 名同学期中考试高数和线代成绩如下：

姓名	高数	线代
张飞	78	75
李大刀	92	67
李墨白	84	88
王老虎	50	50
雷小米	99	98

编写程序按照总分从高到低进行排序后输出姓名和成绩。

9. 打印 n 阶魔方阵（n 为奇数）。魔方阵的每一行、每一列和两个对角线的和都相等。

三. 实验步骤和结果

1. 第一题程序如下:

```
l=list(map(int,input().split())) #用 map 函数一行读入一个列表
l.sort()
print(l[(len(l)-1)//2]) # (元素个数-1) 整除 2 得到中位数下标
```

运行结果如下:

6 5 4 2 1 3
3

5 6 4 8 9 7 2 3 1
5

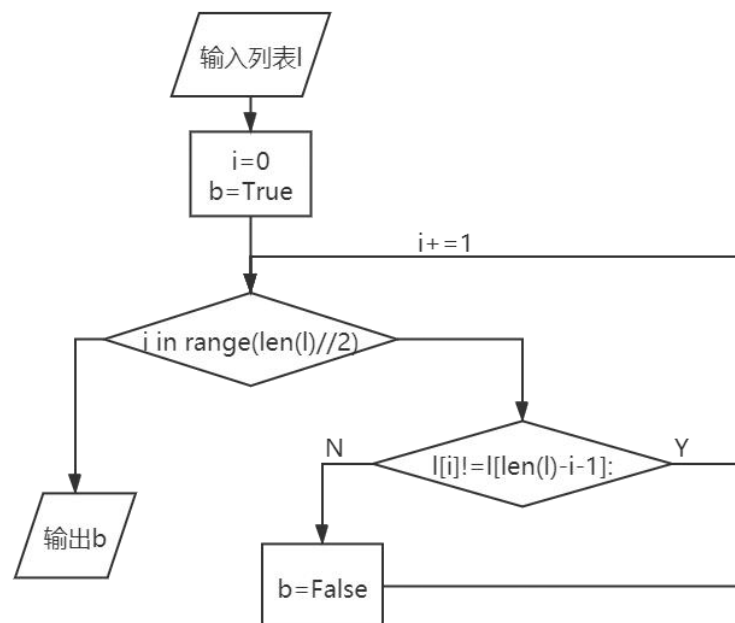
2. 第二题程序如下:

```
l=list(map(int,input().split())) #用 map 函数一行读入一个列表
b=True
for i in range(len(l)//2): #只需要用循环判断列表前半一半, 节省时间
    if l[i]!=l[len(l)-i-1]:
        b=False
print(b)
```

思路: 定义标记变量 **b**, 初值为 **True**, 然后扫描, 如果有不符合回文的数, 将 **b** 变为 **False**
运行结果如下:

1 2 5 4 5 2 1
True

54 6 645 79 123
False



3. 第三题程序如下:

```
def func(n):#判断是否为素数的函数
    if n==1:    # 1 单独讨论
        return False
    for i in range(2,int(n**0.5)+1):#到根号 n 即可,这样不用导入 math
        if n%i==0:
            return False    #找到一个因子立即返回值
    return True

tot=0
l=list(map(int,input().split()))    #用 map 函数一行读入一个列表
ll=[]
for i in l:
    if func(i):
        ll.append(i)    #如果是素数就保留至新列表 ll

print(ll)
ll.sort()
for i in range(len(ll)):
    for j in range(i):    #range(i)以免 i 和 j 相同,同时节省时间
        if (ll[i]+ll[j]) in ll and i!=j:
            tot+=1
print(tot)
思路: 第一步, 找出素数, 保存至 ll 新列表
      第二步, 两层循环嵌套扫描有多少个符合题意
```

运行结果如下:

```
1 2 3 5 6 4 8 9 7 13 11
[2, 3, 5, 7, 13, 11]
3
```

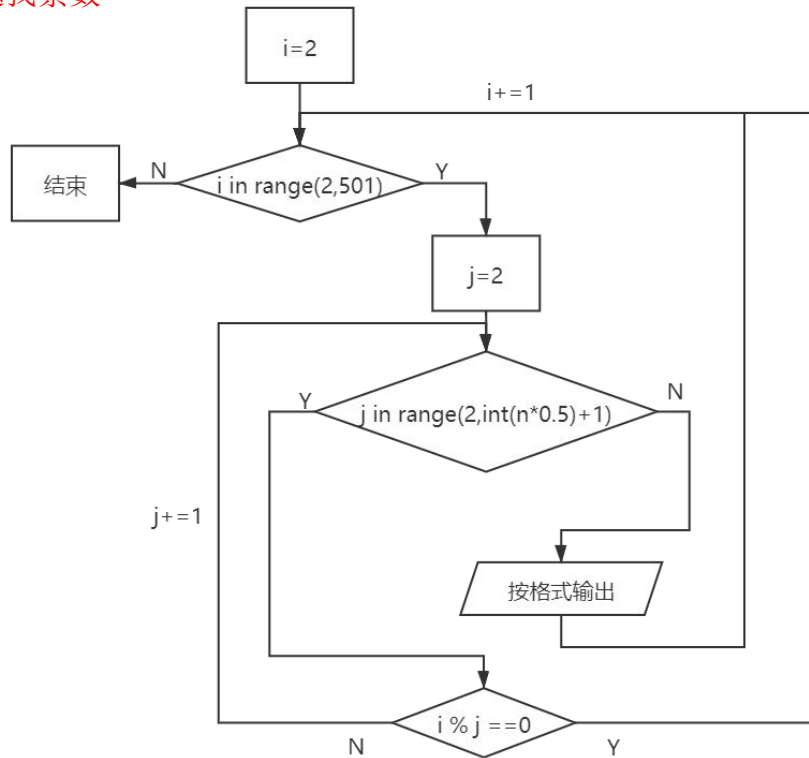
4. 第四题程序如下:

```
tot=0
for i in range(2,501):
    b=0    #用 b 来标记是否找到第三个因子
    for j in range(2,int(i**0.5)+1):
        if i % j==0:
            b=1
    if b:
        continue
    tot+=1
    print(i,end=' ')
    if tot%5==0:    #tot 标记输出格式
        print()
```

思路：类似于上一题找素数

运行结果如下：

```
2 3 5 7 11
13 17 19 23 29
31 37 41 43 47
53 59 61 67 71
73 79 83 89 97
101 103 107 109 113
127 131 137 139 149
151 157 163 167 173
179 181 191 193 197
199 211 223 227 229
233 239 241 251 257
263 269 271 277 281
283 293 307 311 313
317 331 337 347 349
353 359 367 373 379
383 389 397 401 409
419 421 431 433 439
443 449 457 461 463
467 479 487 491 499
```



5. 第五题程序如下：

```
l=list(map(int,input().split()))#用 map 函数一行读入一个列表
for i in l:
    for j in range(l.count(i)-1):#如果数量>=2 就会进入循环，否则跳过
        l.remove(i)

l.sort()
print(l)
```

思路：用 count 来查找出现次数，用得到的多出的次数循环 remove

运行结果如下：

```
35 46 57 13 24 35 99 68 13 79 88 46
[13, 24, 35, 46, 57, 68, 79, 88, 99]
```

6. 第六题程序如下：

```
s1=list(map(str,input()))
s2=list(map(str,input()))#用 map 函数一行读入一个列表
s1.sort()
s2.sort()
if s1==s2:
    print(True)
else:
    print(False)
```

思路：转换成列表，排序后直接比较

运行结果如下：

```
sdasfadsgsfg      qpptt
adssafgfsdgsd     qqptt
True                False
```

7. 第七题程序如下：

```
import numpy
arr=numpy.random.randint(100,500,(4,4))#numpy 自带的生成矩阵函数
arr=arr.transpose()
print(arr)
```

思路：熟练掌握 `numpy` 中关于矩阵的函数

运行结果如下：

```
[[323 257 476 322]  [[380 445 239 100]
 [446 471 160 331]  [180 425 272 101]
 [160 411 194 245]  [152 346 297 405]
 [100 320 125 239]] [322 266 211 150]]
```

8. 第八题程序如下：

```
t=[[ '张飞',78,75],[ '李大刀',92,67],[ '李墨白',84,88],[ '王老虎',50,50],[ '雷小米',99,98]]
t.sort(key=lambda x:x[1]+x[2],reverse=True)#用两项总分逆序排序
print(t)
```

思路：使用 `lambda` 表达式使程序简洁

运行结果如下：

```
[['雷小米', 99, 98], ['李墨白', 84, 88], ['李大刀', 92, 67], ['张飞', 78, 75], ['王老虎', 50, 50]]
```

9. 第九题程序如下：

```
n=int(input())
a=[[None]*(n+1) for _ in range(n+1)]#开拓一个下标最大为 n 的二维列表

i=2
j=(n-1)//2
for t in range(1,n*n+1):
    x=1 if i==n else i+1
    y=j # (x,y)是按照第二种规则的正下方位置
    i=n if i==1 else i-1
    j=1 if j==n else j+1 # (i,j)是按照第一种规则的位置
    if a[i][j]==None:
        a[i][j]=t
```

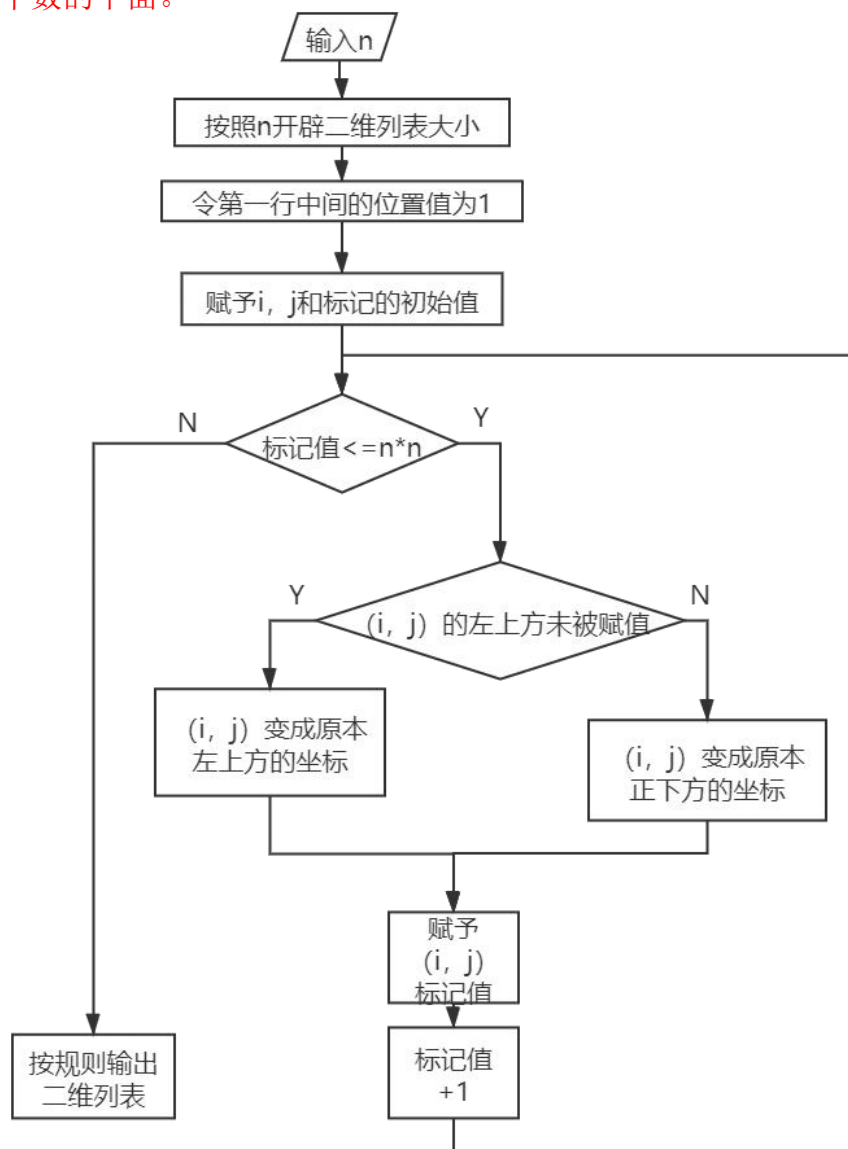
```

else:
    a[x][y]=t
    i=x
    j=y                #现在(i,j)是当前以填好数的点
for i in range(1,n+1):
    for j in range(1,n+1):
        print(a[i][j],end=' ')
    print()           #指定输出格式

```

思路:

- (1)将 1 放在第一行中间一列;
- (2)从 2 开始直到 $n \times n$ 止各数依次按下列规则存放: 每一个数存放的行比前一个数的行数减 1, 列数加 1 (例如上面的三阶魔方, 5 在 4 的上一行后一列);
- (3)如果上一个数的行数为 1, 则下一个数的行数为 n (指最下一行); 例如 1 在第一行, 则 2 应放在最下一行, 列数同样加 1;
- (4)当上一个数的列数为 n 时, 下一个数的列数应为 1, 行数减去 1。
- (5)如果按上面规则确定的位置上已有数, 或上一个数是第一行第 n 列时, 则把下一个数放在上一个数的下面。



运行结果如下：

5	9
17 24 1 8 15	47 58 69 80 1 12 23 34 45
23 5 7 14 16	57 68 79 9 11 22 33 44 46
4 6 13 20 22	67 78 8 10 21 32 43 54 56
10 12 19 21 3	77 7 18 20 31 42 53 55 66
11 18 25 2 9	6 17 19 30 41 52 63 65 76
	16 27 29 40 51 62 64 75 5
	26 28 39 50 61 72 74 4 15
	36 38 49 60 71 73 3 14 25
	37 48 59 70 81 2 13 24 35

拓展题目：

1. 实现 reverse 功能

程序如下：

```
l=list(map(int,input().split()))
for i in range(len(l)//2):
    t=l[i]
    l[i]=l[len(l)-i-1]
    l[len(l)-i-1]=t      #互换
print(l)
```

思路：循环一般长度，和后一半分别互换

运行结果如下：

1 2 3 4 5	8 9 5 4 2
[5, 4, 3, 2, 1]	[2, 4, 5, 9, 8]

2. 二分查找

程序如下：

```
lst=list(map(int,input().split()))
n=int(input())
lst.sort()
l=0
r=len(lst)-1
while l<r:
    mid=(l+r)//2
    if lst[mid]>=n:
        r=mid        #在左边
    else:
        l=mid+1      #在右边
print(l+1)
```

思路：先排序，然后用 **while** 每次判断左标记和右标记的中间值和目标值的大小
最后输出目标值在列表中排序后为第几小

运行结果如下：

8 9 7 4 5 6	1 2 3 6 5 8 7 4 9
9	8
6	8

3. 选择排序：

程序如下：

```
l=list(map(int,input().split()))
for i in range(1,len(l)):
    for j in range(i):
        if l[i]<l[j]:
            t=l[i]
            l[i]=l[j]
            l[j]=t        #交换值
print(l)
```

思路：每次循环找出第 k 小

运行结果如下：

1 2 3 5 4 1	35321 465 123 9789 123
[1, 1, 2, 3, 4, 5]	[123, 123, 465, 9789, 35321]

四. 实验总结（包括对老师的建议）

掌握了列表这种数据结构的基本概念

掌握了列表的常用操作和常用方法，并理解了相应的编程应用过程

掌握了序列数据的常用操作和常用方法，并理解了相应的编程应用过程

在写循环语句和 `range` 的结合时，要考虑清楚何时开始和停止，`range()`里究竟该写啥

调试程序时，中途输出是一种较好的调试方法

开拓列表时，注意是深拷贝还是浅拷贝，输出的出现重复，一般就是拷贝出错

及时写注释

平方根可以不用导入 `math` 用 `int(n**0.5)`即可