

## Project 1 封面页

姓名		课堂 编号	68
如果你 100%确信在完成本项目的时候,自己没有任何涉嫌 <b>抄袭</b> , <b>作弊</b> 和 <b>违反北京理工大学的学术规范</b> 的行为,请用笔在上面签名.			

报告文件名	p168.doc	压缩包文件名	p168src.zip
报告文件的 sha1sum		压缩包文件的 sha1sum	
报告文件 上传时间		压缩包文件 上传时间	
上传文件的 计算机的 MAC address	70-18-8B-DB-18-D4		

1. (python version) 在 console 或者 terminal 里显示自己的 python 解释器版本号。

```
C:\Users\Administrator>python -U
Python 3.7.2
```

2. (run python interpreter quietly) 在 console 或者 terminal 里运行 python, 但只给出提示符>>>, 在提示符后输入代码, 打印 zen of python。

```
>>> import this
The Zen of Python, by Tim Peters
```

3. (display python's keyword (关键字)) 在 console 或者 terminal 里运行 python, 但只给出提示符>>>, 多次在提示符后输入代码, 打印 python 的关键字的个数及全部关键字。

```
>>> import keyword
>>> keyword.kwlist
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'cla
ss', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from
', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pas
s', 'raise', 'return', 'try', 'while', 'with', 'yield']
>>> len(keyword.kwlist)
35
```

4. (BE vs LE) 请查阅文献和 Internet 资源, 简述自己对“大小端”的理解, 并用 python 代码查出自己的计算机到底是“big”还是“little”。

大端模式, 指数据的高字节存在内存的低地址中, 数据的低字节存在内存的高地址中。地址由小向大增加, 而数据从高位往低位放。

小端模式, 指数据的高字节存在内存的高地址中, 数据的低字节存在内存的低地址中, 高地址部分权值高, 低地址部分权值低。

完整代码:

```
1. import sys
2. print(sys.byteorder)
```

运行截图:

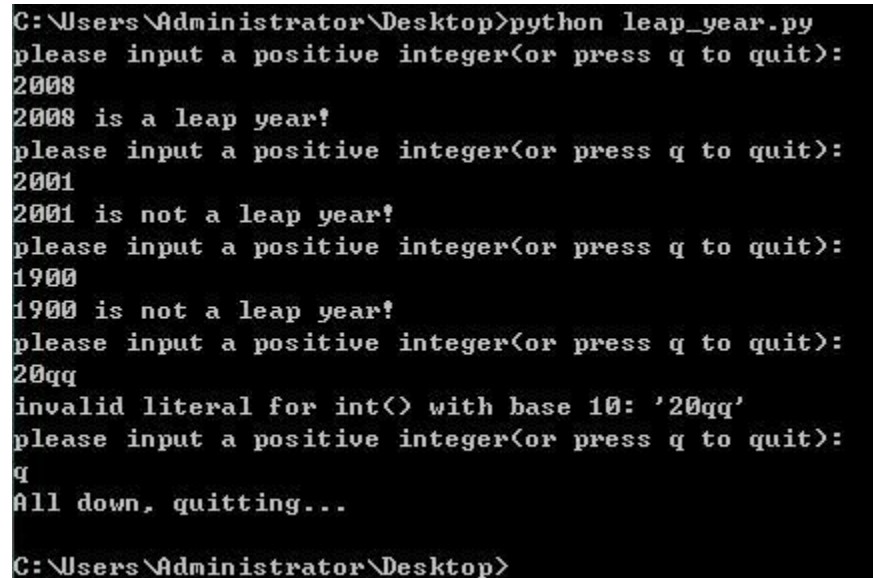
```
C:\Users\Administrator\Desktop>python endian.py
little
```

5. (leap year) 要求用户输入一个年份，通过程序判断并输出是否为闰年。注意，程序应当检测输入是否合法，就算输入非法，程序还可以继续，直到用户输入 q 才能退出。

完整代码：

```
1. tip='please input a positive integer(or press q to quit):\n'
2. while True:
3.     year=input(tip)
4.     if year=='q':
5.         print('All down, quitting...')
6.         exit()
7.     try:
8.         year=int(year)
9.     except ValueError as e:
10.        print(str(e))
11.    else:
12.        if year<=0:
13.            print('11')
14.        elif year%400==0 or (year%4==0 and year%100!=0):
15.            print('%d is a leap year!'%year)
16.        else:
17.            print('%d is not a leap year!'%year)
```

运行截图：



```
C:\Users\Administrator\Desktop>python leap_year.py
please input a positive integer(or press q to quit):
2008
2008 is a leap year!
please input a positive integer(or press q to quit):
2001
2001 is not a leap year!
please input a positive integer(or press q to quit):
1900
1900 is not a leap year!
please input a positive integer(or press q to quit):
20qq
invalid literal for int() with base 10: '20qq'
please input a positive integer(or press q to quit):
q
All down, quitting...

C:\Users\Administrator\Desktop>
```

## 6. binomial coefficient

6.1 Write a function that computes the binomial coefficient  $C(n, k)$  efficiently.

程序请命名为 cnk. py;

完整代码：

```

1. import sys
2. def cnk(n,k):
3.     s1=1
4.     s2=1
5.     for i in range(k):
6.         s1=s1*(n-i)
7.         s2=s2*(i+1)
8.     return int(s1/s2)
9. n=int(sys.argv[1])
10. k=int(sys.argv[2])
11. print(cnk(n,k))

```

运行截图：

```

C:\Users\Administrator\Desktop>python cnk.py 4 2
6

```

6.2 了解一下中国福利彩票“双色球”的玩法。假设你购买了一注该彩票，中一等奖的概率为  $1/n$ ，改写上面的 cnk.py 为 lottery.py，lottery.py 的作用是算出正整数  $n$ ；

完整代码：

```

1. def cnk(n,k):
2.     s1=1
3.     s2=1
4.     for i in range(k):
5.         s1=s1*(n-i)
6.         s2=s2*(i+1)
7.     return int(s1/s2)
8. n=cnk(33,6)*cnk(16,1)
9. print(n)

```

运行截图：

```

C:\Users\Administrator\Desktop>python lottery.py
17721088

```

6.3 打印出杨辉三角，程序请命名为 yanghui.py。

完整代码：

```

1. import sys
2. def cnk(n,k):
3.     s1=1
4.     s2=1
5.     for i in range(k):
6.         s1=s1*(n-i)

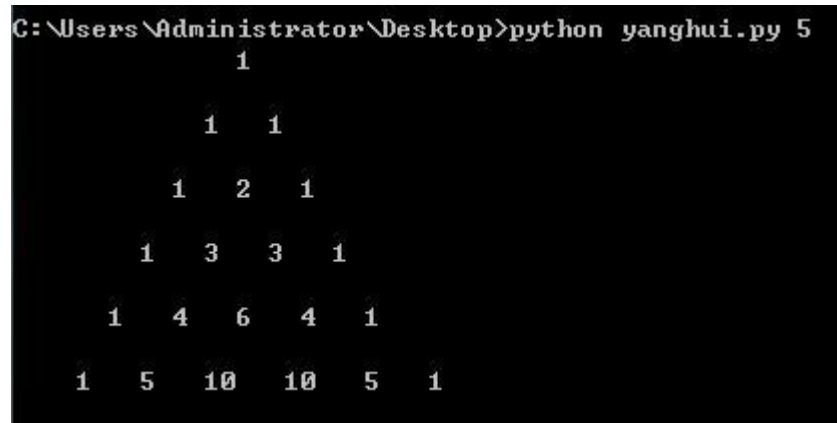
```

```

7.         s2=s2*(i+1)
8.     return int(s1/s2)
9. n=int(sys.argv[1])
10. for i in range(0,n+1):
11.     for j in range(1,n-i+3):
12.         print(' ',end='')
13.     for k in range(0,i+1):
14.         print(cnk(i,k), ' ',end='')
15.     print('\n')

```

运行截图:



```

C:\Users\Administrator\Desktop>python yanghui.py 5
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1

```

## 7. ltds exercise 1

(ltds 代表 list, tuple, dict, set 及其它的 python 基本数据类型.)

7.1 随机产生一个 7 到 14 之间的正整数 (包含 7 或者 14) n, 打印 n;

7.2 产生一个长度为 n 的随机整数的 list, list 名字为 L1, 每个整数在 -10 到 10 之间 (包含 -10 或者 10), 打印 L1;

7.3 新建 dict D1, D1 中的 key 为 L1 中的不同元素, value 为该元素在 L1 中出现的次数, 打印 D1;

7.4 分 3 次拷贝 L1 为 L2, 第 1 次对 L2 从小到大排序后输出 L2; 第 2 次对 L2 从大到小排序后输出 L2; 第 3 次对 L2 按绝对值从小到大排序后输出 L2; 整个过程 L1 保持不变;

7.5 反向输出 L1, 且 L1 变为了原来的反向.

完整代码:

```

1. import random
2. n=random.randint(7,14)
3. print(n)
4. L1=[random.randint(-10,10) for i in range(n)]
5. print(L1)
6. D1_list=[(i,L1.count(i)) for i in L1]

```

```

7. D1=dict(D1_list)
8. print(D1)
9. L2=L1[:]
10. L2.sort()
11. print(L2)
12. L2.sort(reverse=True)
13. print(L2)
14. L2.sort(key=abs)
15. print(L2)
16. L1.reverse()
17. print(L1)

```

运行截图：

```

C:\Users\Administrator\Desktop>python ltds1.py
10
[-6, -10, -4, -1, -4, 8, -3, -7, 1, 6]
<-6: 1, -10: 1, -4: 2, -1: 1, 8: 1, -3: 1, -7: 1, 1: 1, 6: 1>
[-10, -7, -6, -4, -4, -3, -1, 1, 6, 8]
[8, 6, 1, -1, -3, -4, -4, -6, -7, -10]
[1, -1, -3, -4, -4, 6, -6, -7, 8, -10]
[6, 1, -7, -3, 8, -4, -1, -4, -10, -6]

```

## 8. ltds exercise 2.

用如下语句创建 tuple T1:

```
T1=([1,2,3], {'A':90, 'B': 80}, {-100,0,100})
```

8.1 打印 T1;

8.2 更改 T1 里内嵌的 list 的第 1 个位置的元素为 20 (位置从 0 开始计), 打印 T1;

8.3 在 T1 里内嵌的 list 的尾部增加一个元素 4, 打印 T1;

8.4 更改 T1 里内嵌的 dict 的 'A' 对应的值为 99, 打印 T1;

8.5 删除 T1 里内嵌的 set 中的元素 -100, 打印 T1.

完整代码:

```

1. T1=([1,2,3], {'A':90, 'B': 80}, {-100,0,100})
2. print(T1)
3. T1[0][1]=20
4. print(T1)
5. T1[0].append(4)
6. print(T1)
7. T1[1]['A']=99
8. print(T1)
9. T1[2].remove(-100)
10. print(T1)

```

运行截图：

```
C:\Users\Administrator\Desktop>python ltds2.py
<1, 2, 31, {'A': 90, 'B': 80}, <0, 100, -100>>
<1, 20, 31, {'A': 90, 'B': 80}, <0, 100, -100>>
<1, 20, 3, 41, {'A': 90, 'B': 80}, <0, 100, -100>>
<1, 20, 3, 41, {'A': 99, 'B': 80}, <0, 100, -100>>
<1, 20, 3, 41, {'A': 99, 'B': 80}, <0, 100>>
```

### 9. ltds exercise 3.

9.1 用 list comprehension 产生学生姓名 list L1, 内含学生姓名“u19f34”至“u19f72” (不包括引号), 共 39 个;

9.2 用 tuple comprehension 产生学生的语文成绩的 tuple T1, T1 的每个元素为 0-100 之间的随机整数, 共 39 个元素;

用 tuple comprehension 产生学生的数学成绩的 tuple T2, T2 的每个元素为 0-100 之间的随机整数, 共 39 个元素;

用 tuple comprehension 产生学生的英语成绩的 tuple T3, T3 的每个元素为 0-100 之间的随机整数, 共 39 个元素;

9.3 用 dict comprehension 产生成绩单 dict D1 (显然要从上面的 list 和 tuples 创建此 dict);

9.4 漂亮打印出原始成绩单 D1;

9.5 拷贝 D1 为 D2, 按语文成绩从低到高漂亮打印出 D2;

9.6 拷贝 D1 为 D3, 按三门课的总成绩从高到低漂亮打印出 D3;

9.7 拷贝 D1 为 D4, 按语文成绩从低到高 (若语文成绩相同, 则按数学成绩从高到低) 漂亮打印出 D4。

完整代码:

```
1. import random
2. L1=['u19f%s'%i for i in range(34,73)]
3. T1=[random.randint(0,100) for i in range(1,40)]
4. T1=tuple(T1)
5. T2=[random.randint(0,100) for i in range(1,40)]
6. T2=tuple(T2)
7. T3=[random.randint(0,100) for i in range(1,40)]
8. T3=tuple(T3)
9. D1_list=[(1,(t1,t2,t3)) for l,t1,t2,t3 in zip(L1,T1,T2,T3)]
10. D1=dict(D1_list)
11. print(' 姓名      语文  数学  英语')
12. for i in range(39):
13.     print(D1_list[i][0].ljust(6),'|',str(D1_list[i][1][0]).ljust(4),
14.           str(D1_list[i][1][1]).ljust(4),str(D1_list[i][1][2]).ljust(4))
```

```

15. print('\n')
16. D2_list=D1_list[:]
17. D2_list.sort(key=lambda x:x[1][0])
18. print(' 姓名    语文  数学  英语')
19. for i in range(39):
20.     print(D2_list[i][0].ljust(6),'|',str(D2_list[i][1][0]).ljust(4),
21.           str(D2_list[i][1][1]).ljust(4),str(D2_list[i][1][2]).ljust(4))
22. print('\n')
23. D3_list=[(l,t1+t2+t3) for l,t1,t2,t3 in zip(L1,T1,T2,T3)]
24. D3_list.sort(reverse=True,key=lambda x:x[1])
25. print(' 姓名    总成绩')
26. for i in range(39):
27.     print(D3_list[i][0].ljust(6),'|',str(D3_list[i][1]).ljust(4))
28. print('\n')
29. D4_list=D1_list[:]
30. D4_list.sort(key=lambda x:(x[1][0],x[1][1]))
31. print(' 姓名    语文  数学  英语')
32. for i in range(39):
33.     print(D4_list[i][0].ljust(6),'|',str(D4_list[i][1][0]).ljust(4),
34.           str(D4_list[i][1][1]).ljust(4),str(D4_list[i][1][2]).ljust(4))

```

运行截图（部分）：

```

姓名    语文  数学  英语
u19f39 : 0    57   65
u19f56 : 1    89   27
u19f70 : 7    84   95
u19f42 : 11   84   95
u19f54 : 14   28   55
u19f55 : 18   58   46
u19f45 : 21   13   78
u19f61 : 22    6   69
u19f37 : 26   48   12
u19f64 : 28   44   75
u19f38 : 32   89   76
u19f58 : 32   22   38
u19f66 : 34   73   29
u19f62 : 37   90   14
u19f72 : 38   91   40

```

10. (finding primes) 改写或者全新地写 lecture 中讲过的计算素数的程序，找出 1-10,000,000 之间的素数，报告该算法所花的时间。

完整代码：

```

1. import time
2. t0=time.time()
3. n=10000000
4. L=[True]*n
5. for i in range(2,int(n**0.5)+1):

```



```

6.     if L[i]:
7.         for j in range(i*i,n,i):
8.             L[j]=False
9.     for x in range(2,n):
10.        if L[x]:
11.            print(x)
12.    print('所用时间: %fs'%(time.time()-t0))

```

运行截图（部分）：

```

9999907
9999929
9999931
9999937
9999943
9999971
9999973
9999991
所用时间：18.658067s
[Finished in 19.1s]

```

11. (fibonacci numbers) 写一个程序，打印出小于 10,000,000 的所有 fibonacci 数。

第 0 个和第 1 个 fibonacci 数均为 1，从第 0 个开始打印，逗号加空格隔开，但最后一个数后面无逗号 or 空格，格式在下面。

完整代码：

```

1.  n=10000000
2.  print('1, 1',end='')
3.  t1=1
4.  t2=2
5.  while t2<n:
6.      print(', ',end='')
7.      print(t2,end='')
8.      t=t2
9.      t2=t2+t1
10.     t1=t

```

运行截图：

```

C:\Users\Administrator\Desktop>python fibonacci.py
1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181,
6765, 10946, 17711, 28657, 46368, 75025, 121393, 196418, 317811, 514229, 832040,
1346269, 2178309, 3524578, 5702887, 9227465

```

12. (ansi vs utf8 encoding) 找出自己的汉语名字的 ansi (gb2312) 和 utf8 编码，每个字分别找。

完整代码:

```
1. name=['张','翼','鹏']
2. L1=[str(i.encode('gbk')).replace(r'\x','').replace('\\','').upper() for i in name]
3. L2=[str(i.encode('utf-8')).replace(r'\x','').replace('\\','').upper() for i in name]
4. for i in range(len(name)):
5.     print('%s 的 ansi 编码为%s %s,utf8 编码为%s %s %s'%(name[i],
6.         L1[i][1:3],L1[i][3:],L2[i][1:3],L2[i][3:5],L2[i][5:])))
```

运行截图:

```
C:\Users\Administrator\Desktop>python myutf8.py
张的ansi编码为D5 C5,utf8编码为E5 BC A0
翼的ansi编码为D2 ED,utf8编码为E7 BF BC
鹏的ansi编码为C5 F4,utf8编码为E9 B9 8F
```