《Python 语言程序设计》平时作业

>>1、

（1）

```
>>> print(a)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14] ——list
>>> print(b)
(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14)     ——tuple
```

（2）

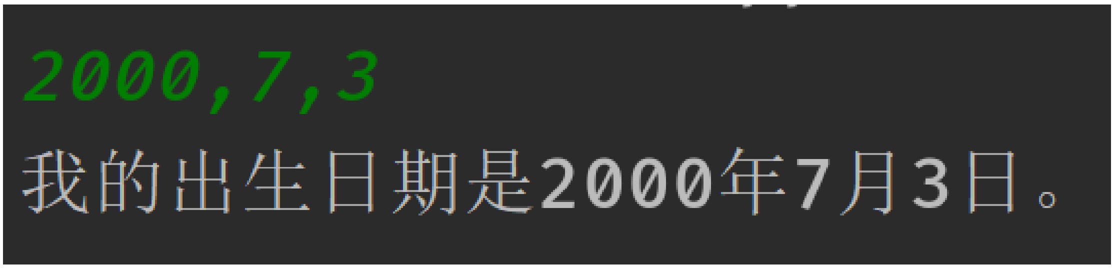
range()可以返回一系列连续增加的整数，也是一个迭代器。range(15)从 0 开始生成连续增加的一共 15 个整数，range(1,15)则是生成从 1 开始连续增加到 14 截止的整数。

（3）

```
>>> c = list(range(1,101,2))
>>> print(c)
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49,
51, 53, 55, 57, 59, 61, 63, 65, 67, 69, 71, 73, 75, 77, 79, 81, 83, 85, 87, 89, 91, 93, 95,
97, 99]
```

>>2、

```
a, b, c = map(str, input().split(','))
print("我的出生日期是"+a+"年"+b+"月"+c+"日。")
```

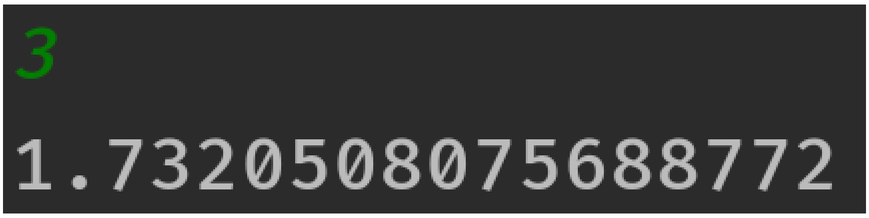


```
2000,7,3
我的出生日期是2000年7月3日。
```

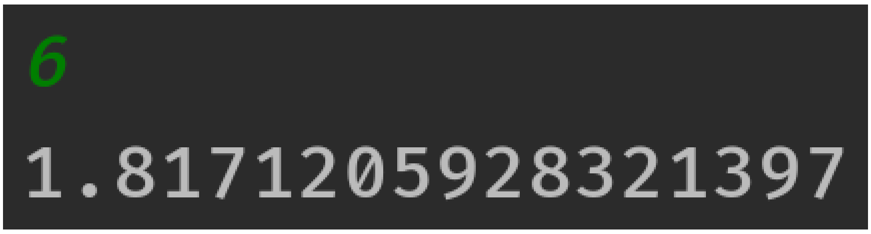
>>3、

```
m = int(input())
```

```
# 单分支
if m%2==1:
    print(m**(1/2))
if m%2==0:
    print(m**(1/3))
# 双分支
if m%2==1:
    print(m**(1/2))
else:
    print(m**(1/3))
# 条件运算
print(m**(1/2)) if m%2==1 else print(m**(1/3))
```



```
3
1.7320508075688772
```

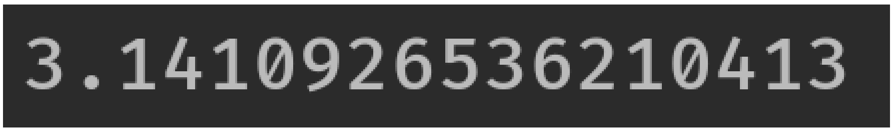


```
6
1.8171205928321397
```

>>4、

（1）

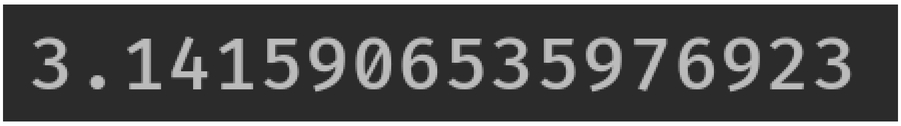
```
sum = 0
for i in range(1,1001):
    sum = sum + 1/(4*i-3) - 1/(4*i-1)
print(4*sum)
```



```
3.1410926536210413
```

（2）

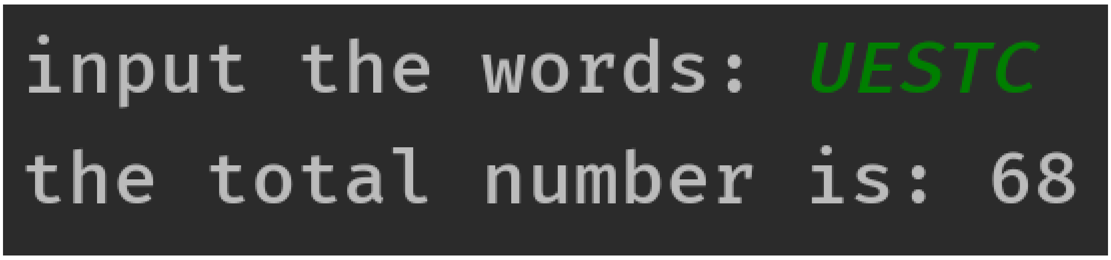
```
sum = 0
i = 1
while 1:
    n = 1/(4*i-1)
    sum = sum + 1/(4*i-3) - n
    i += 1
    if n<0.000001:
        print(4 * sum)
        exit(0)
```



```
3.1415906535976923
```

>>5、

```
total = 0
s = input("input the words: ")
s = s.upper()
for num in range(len(s)):
    total += ord(s[num]) - 64
print("the total number is:",total)
```

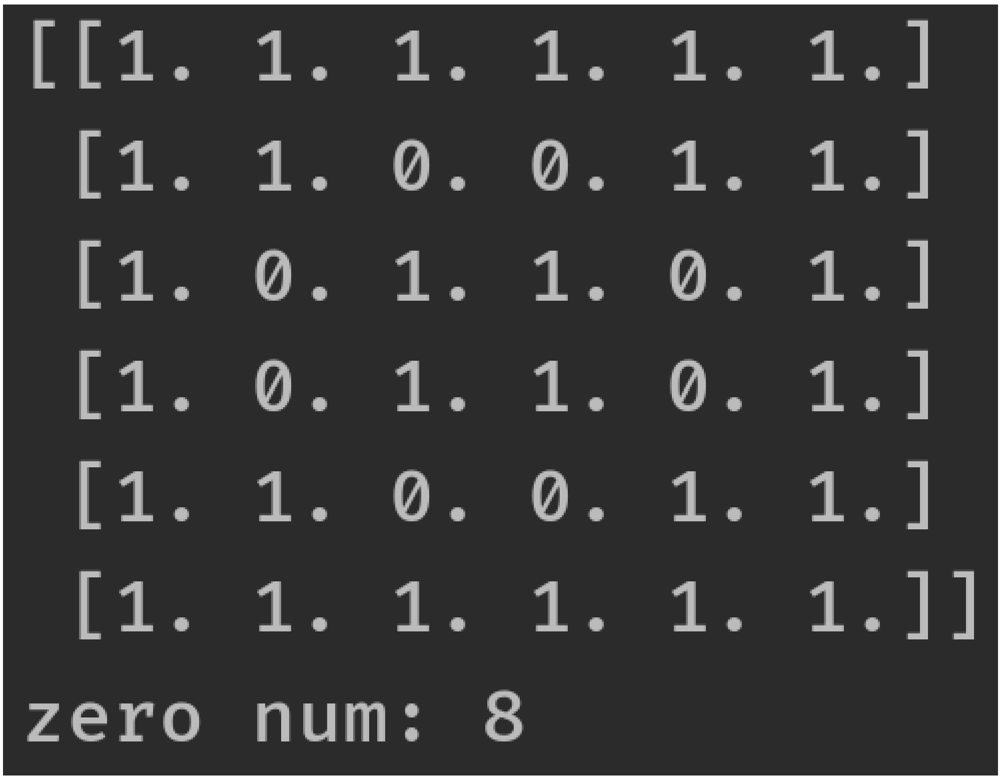


```
input the words: UESTC
the total number is: 68
```

>>6、

```
import numpy as np
x = y = 5
total = 36
a = np.zeros((x+1,y+1))
```

```
for x in range(0,6):
    for y in range(0,6):
        if x == 0 or y == 0 or x==5 or y==5 or x==y or (x+y) == 5:
            a[x][y] = 1
            total -= 1
print("\n")
print(a)
print("zero num:",total)
```

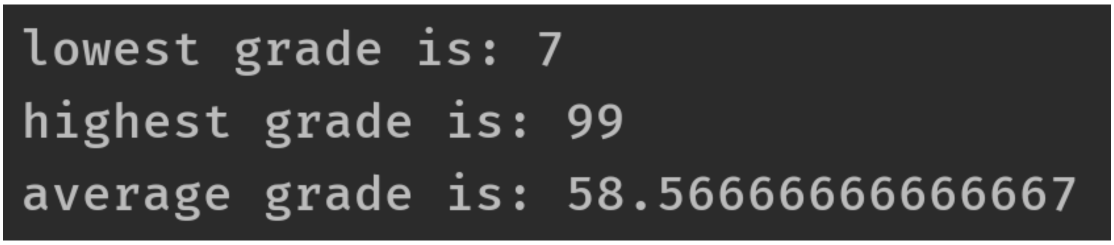


```
[[1. 1. 1. 1. 1. 1.]
 [1. 1. 0. 0. 1. 1.]
 [1. 0. 1. 1. 0. 1.]
 [1. 0. 1. 1. 0. 1.]
 [1. 1. 0. 0. 1. 1.]
 [1. 1. 1. 1. 1. 1.]]
zero num: 8
```

>>7、

```
grade = {'jason':45,'lollipop':55,'mike':30,'andy':70,'dengcong':62,
         'songjingshen':34,'anny':24,'zhouzhizhen':77,'dis':20,'op':13,
         'pop':68,'jack':99,'jamy':12,'dudu':88,'myself':59,'loip':88,
         'nxn':47,'dioasd':63,'coiajsc':78,'sjdaioji':99,'aciojsa':88,
         'axso':86,'uoca':67,'dioausd':88,'acioua':7,'xm':66,'iodsa':59,
         'xaio':21,'xjia':85,'casio':59}
low = 100
high = 0
```

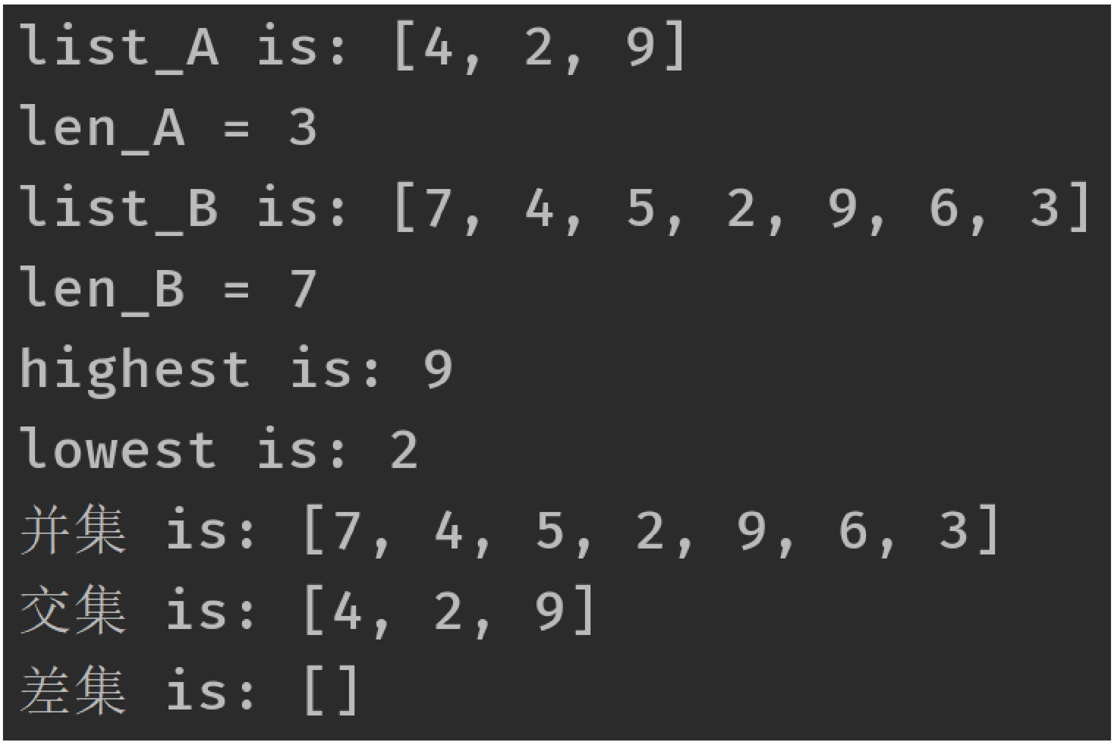
```
total = 0
for i in grade:
    if grade[i] < low:
        low = grade[i]
    if grade[i] > high:
        high = grade[i]
    total += grade[i]
aver = total / 30
print("lowest grade is: {0}\nhighest grade is: {1}\naverage grade is: {2}".format(low,high,aver))
```



```
lowest grade is: 7
highest grade is: 99
average grade is: 58.56666666666667
```

>>8、

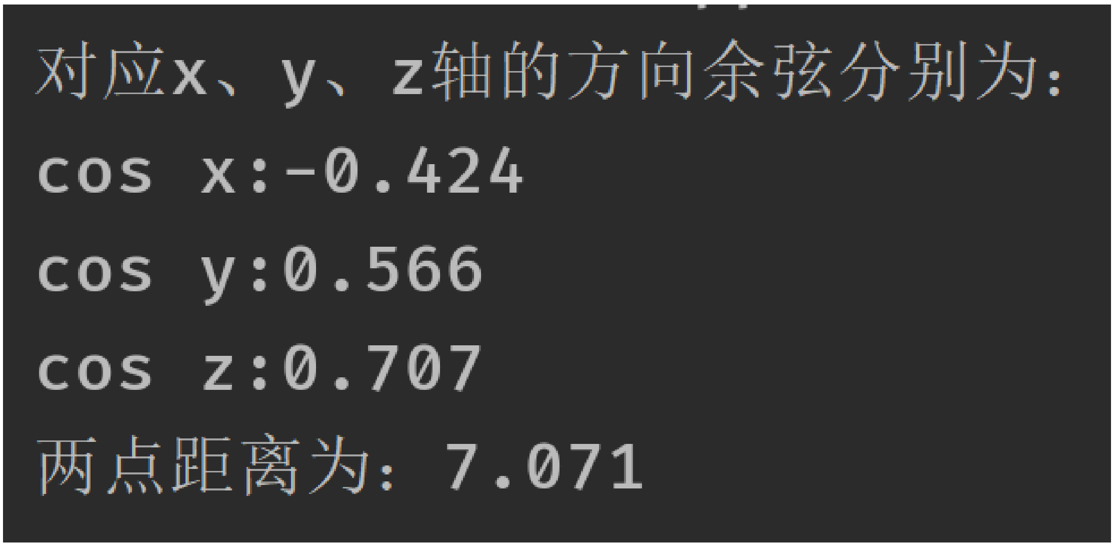
```
import random
len_a = random.randint(0,10)
len_b = 10 - len_a
list_a = [random.randint(0,10) for a in range(0,len_a)]
list_b = [random.randint(0,10) for b in range(0,len_b)]
print("list_A is:",list_a,"\nlen_A = {}".format(len_a))
print("list_B is:",list_b,"\nlen_B = {}".format(len_b))

high = 0
low = 10
list_c = []
list_d = []
list_e = []
for i in list_a:
    if i > high:
        high = i
    if i < low:
        low = i

    if i not in list_b and i not in list_c:
        list_c.append(i) # 并集

    if i in list_b and i not in list_d:
        list_d.append(i) # 交集

    if i not in list_b and i not in list_e:
        list_e.append(i) # 差集

for i in list_b:
    if i > high:
        high = i
    if i < low:
        low = i

    if i not in list_c:
        list_c.append(i) # 并集

print("highest is:",high)
print("lowest is:",low)
print("并集 is:",list_c)
print("交集 is:",list_d)
print("差集 is:",list_e)
```



```
list_A is: [4, 2, 9]
len_A = 3
list_B is: [7, 4, 5, 2, 9, 6, 3]
len_B = 7
highest is: 9
lowest is: 2
并集 is: [7, 4, 5, 2, 9, 6, 3]
交集 is: [4, 2, 9]
差集 is: []
```

>>9、

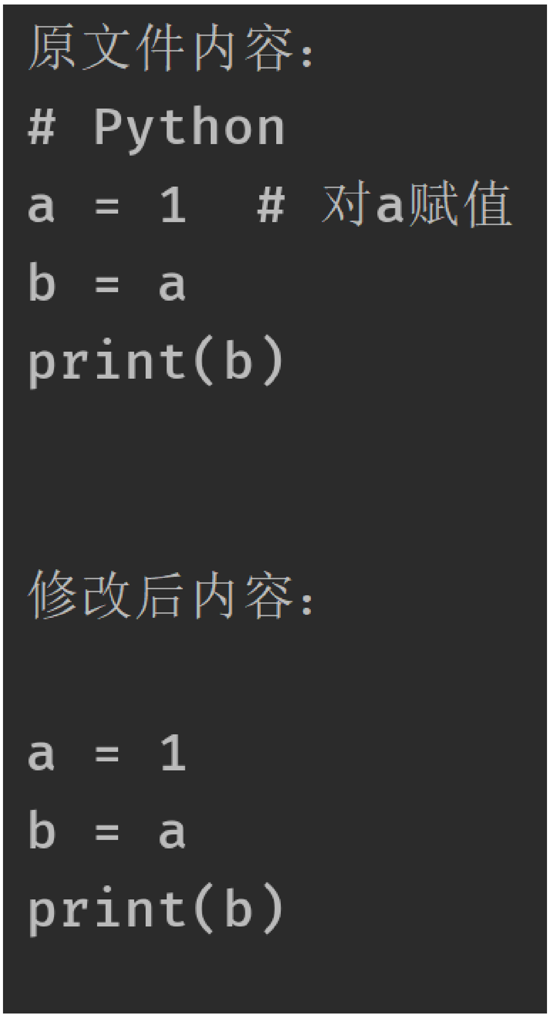
```
import math
class Point(object):
    def __init__(self,x,y,z):
        self.x = x
        self.y = y
        self.z = z
        return

def direction_cosine(a,b):
    x = a.x - b.x
    y = a.y - b.y
    z = a.z - b.z
    d = pow(math.pow(x,2) + math.pow(y,2) + math.pow(z,2),1/2)
    dx = x / d
    dy = y / d
    dz = z / d
    print("对应x、y、z轴的方向余弦分别为：\ncos x:{:.3f}\ncos y:{:.3f}\ncos z:{:.3f}".format(dx, dy, dz))
    return d

def main():
    p1 = Point(2,7,9)
    p2 = Point(5,3,4)
    d = direction_cosine(p1,p2)
    print('两点距离为：%.3f' % d)
    return
main()
```



```
对应x、y、z轴的方向余弦分别为:
cos x:-0.424
cos y:0.566
cos z:0.707
两点距离为: 7.071
```

>>10、

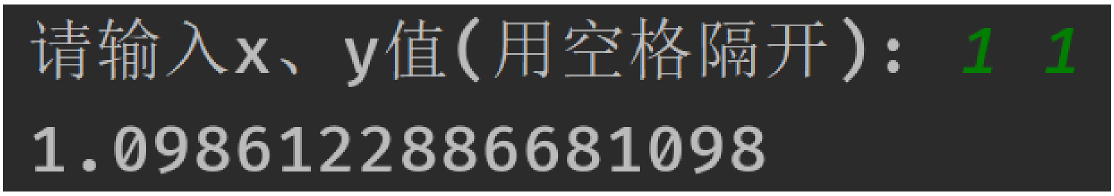
```
with open("a.py", 'r', encoding='utf-8') as fo_r:
    print("原文件内容：\n" + fo_r.read() + "\n")

with open("a.py", 'r', encoding='utf-8') as fo:
    lines = fo.readlines()

with open("a.py", 'w', encoding='utf-8') as fo_w:
    for line in lines:
        if len(line) != 1:
            s = ""
            if line[0] == "#":
                s = s + "\n"
                fo_w.write(s)
                continue
            for i in line:
                if i == "#":
                    s = s + "\n"
                    break
                s = s + i
            fo_w.write(s)
        else:
            fo_w.write(line)

with open("a.py", 'r', encoding='utf-8') as fo_r:
    print("修改后内容：\n" + fo_r.read())
```



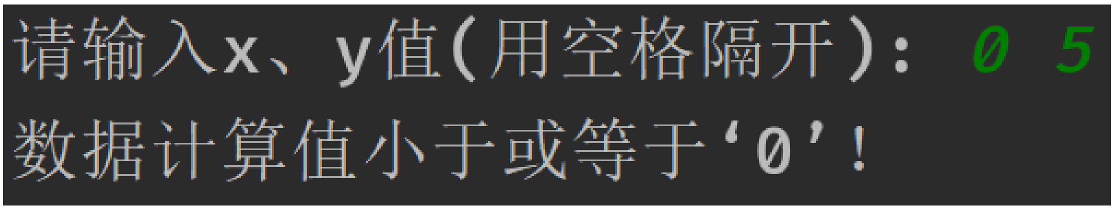
```
原文件内容:
# Python
a = 1  # 对a赋值
b = a
print(b)


修改后内容:

a = 1
b = a
print(b)
```

>>11、

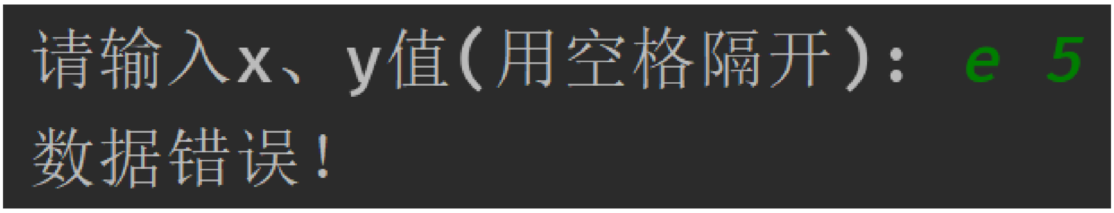
```
import math


def main():
    try:
        x, y = map(float, input('请输入 x、y 值(用空格隔开): ').split(" "))
        assert (3 * x - y + 1) > 0
        result = math.log(3*x-y+1, math.e)
    except AssertionError:
        print("数据计算值小于或等于‘0’！ ")
    except ValueError:
        print("数据错误！ ")
    except:
        print("程序发生异常！ ")
    else:
        print(result)


main()
```



```
请输入x、y值(用空格隔开): 1 1
1.0986122886681098
```



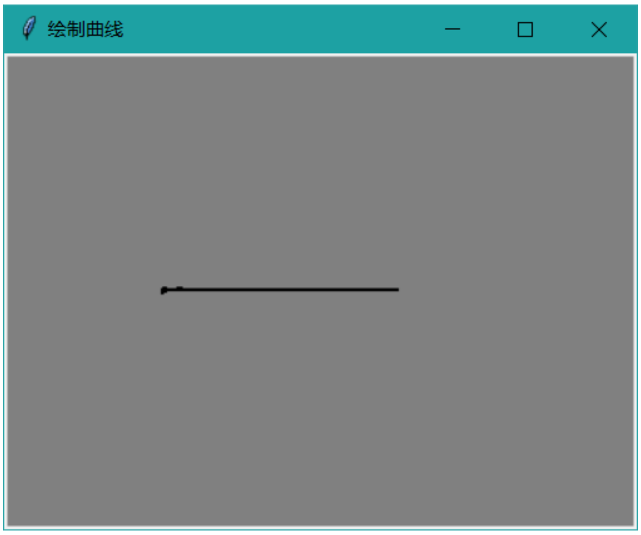
```
请输入x、y值(用空格隔开): 0 5
数据计算值小于或等于‘0’！
```



```
请输入x、y值(用空格隔开): e 5
数据错误！
```

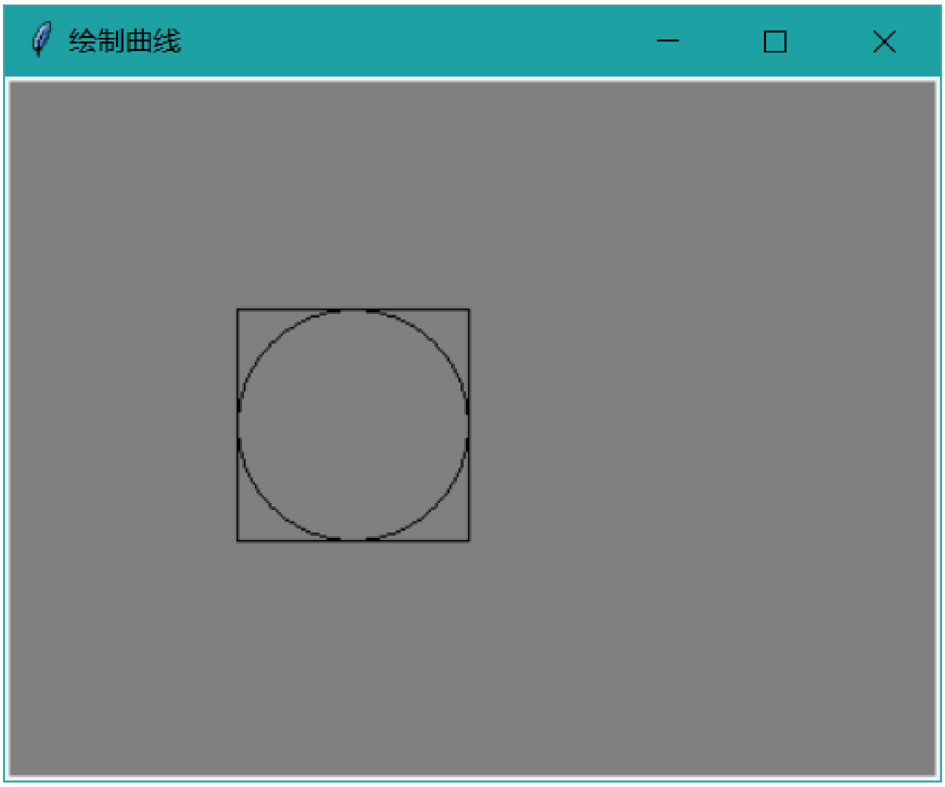
>>12、

```
import tkinter
import math
from tkinter import *
from math import *

w = Tk()
#  设置主窗口样式
w['width'] = 400
w['height'] = 300
w['bg'] = 'white'
w.title('绘制曲线')

#  创建画布对象
c = Canvas(w, width=400, height=300, bg='grey')
c.pack()

#  绘制图形
x = 0
while x <= 150:
    y = 2*pow(e, -0.5*x)*sin(2*pi*x)
    x = x+0.1
    c.create_rectangle(x+100, y+150, x+100.01, y+150.01)
mainloop()
```

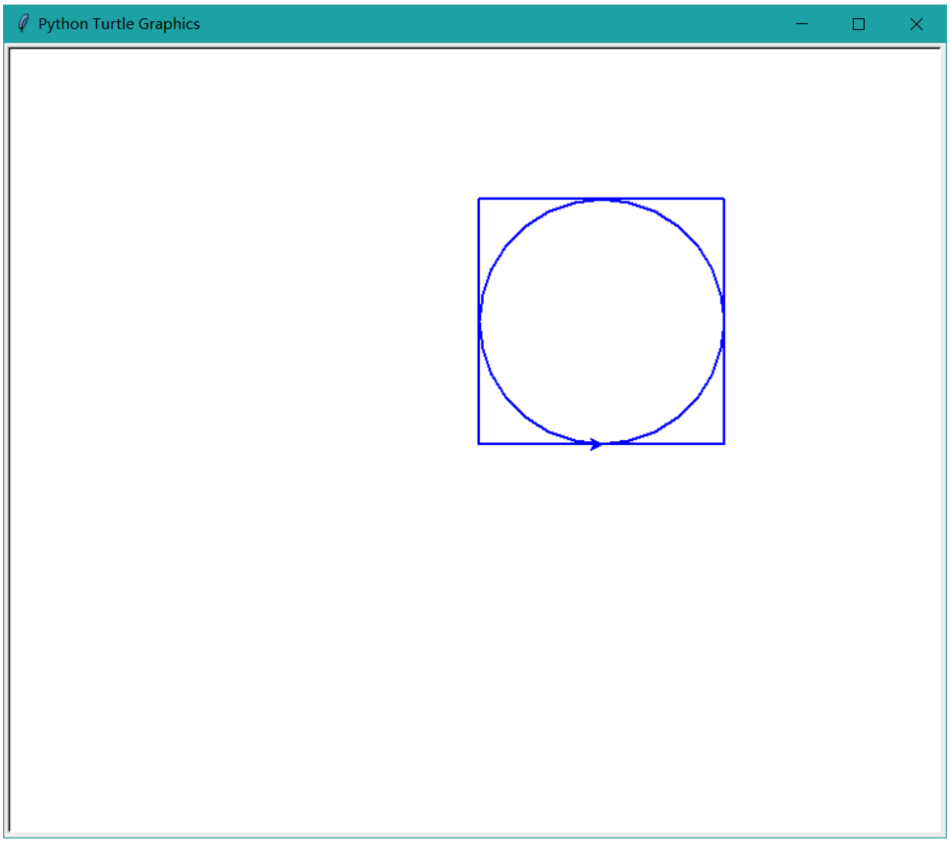


>>13、

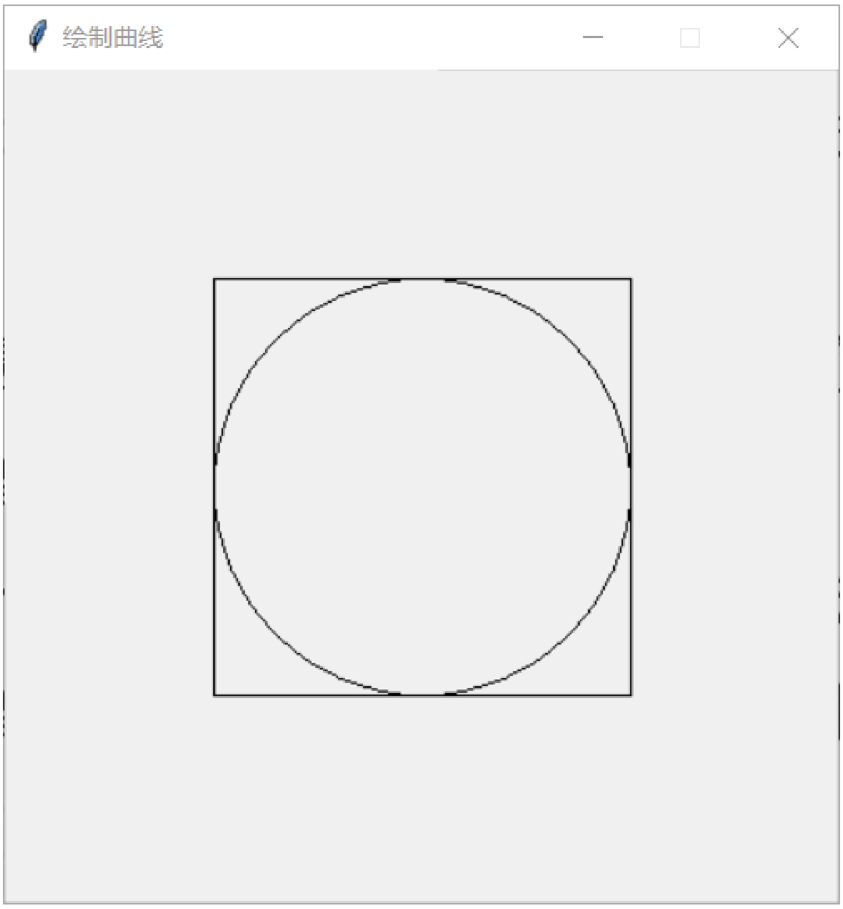
```
#   tkinter 绘制
import tkinter
from tkinter import *

w = Tk()
#   设置主窗口样式
w['width'] = 400
w['height'] = 300
w['bg'] = 'white'
w.title('绘制曲线')

#   创建画布对象
c = Canvas(w, width=400, height=300, bg='grey')
c.pack()

#   绘制图形
c.create_rectangle(100, 100, 200, 200)
c.create_oval(100, 100, 200, 200)
mainloop()
```



```
#   turtle 绘制
import turtle
from turtle import *

w = Turtle()
w.color("blue")
w.pensize(2)
w.speed(2)
for i in range(4):
        w.fd(200)
        w.lt(90)

w.up()
w.goto(100, 0)
w.down()
w.circle(100)
mainloop()
```

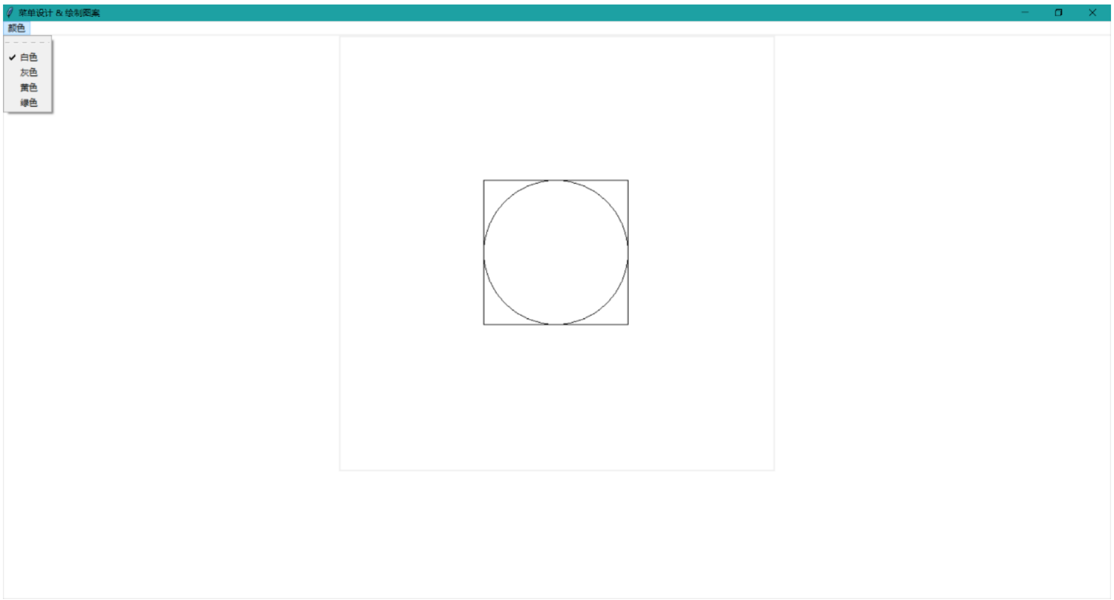


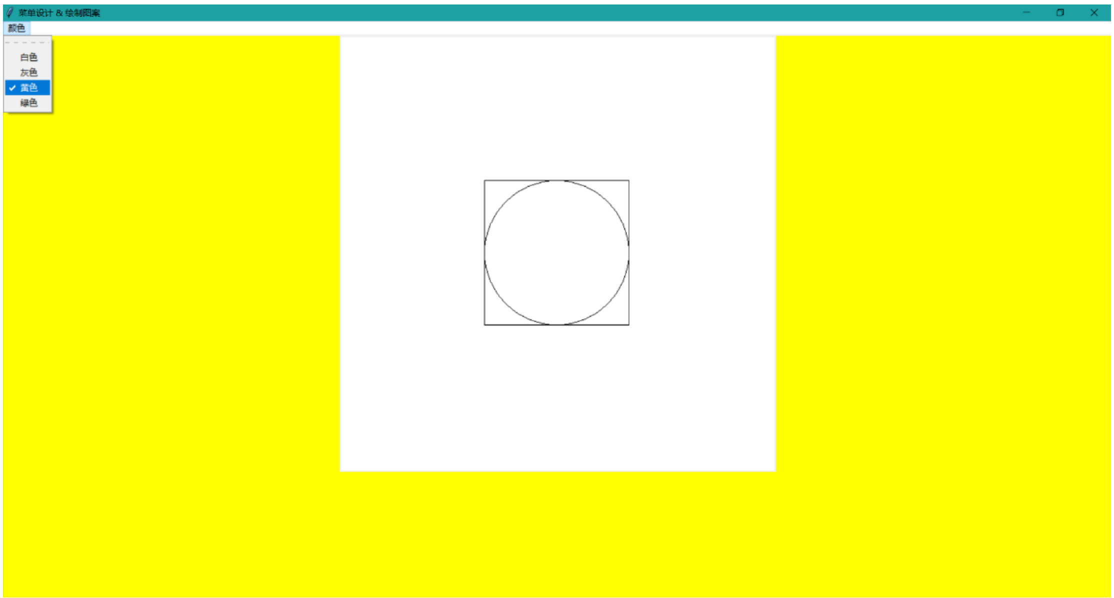
```
#   graphics 绘制
import graphics
from graphics import *

#   绘制正方形
w = GraphWin("绘制曲线", 400, 400)
p1 = Point(100, 100)
p2 = Point(300, 300)
r = Rectangle(p1, p2)
r.draw(w)

#   绘制同心圆
p3 = Point(200, 200)
c = Circle(p3, 100)
c.draw(w)

w.mainloop()
```



>>14、

```
from tkinter import *



def func1():
    w['bg'] = 'white'



def func2():
    w['bg'] = 'grey'



def func3():
    w['bg'] = 'yellow'



def func4():
    w['bg'] = 'green'



#  创建主窗口
w = Tk()

#  设置主窗口样式
w['width'] = 1000
w['height'] = 1000
w['bg'] = 'white'
w.title('菜单设计 & 绘制图案')

#  菜单
m = Menu(w)
w.config(menu=m)
fm = Menu(m)
m.add_cascade(label='颜色', menu=fm)
fm.add_radiobutton(label='白色', command=func1)
fm.add_radiobutton(label='灰色', command=func2)
fm.add_radiobutton(label='黄色', command=func3)
fm.add_radiobutton(label='绿色', command=func4)

#  创建画布对象
c = Canvas(w, width=600, height=600, bg='white')
c.pack()

#  绘制图形
c.create_rectangle(200, 200, 400, 400)
c.create_oval(200, 200, 400, 400)

w.mainloop()
```

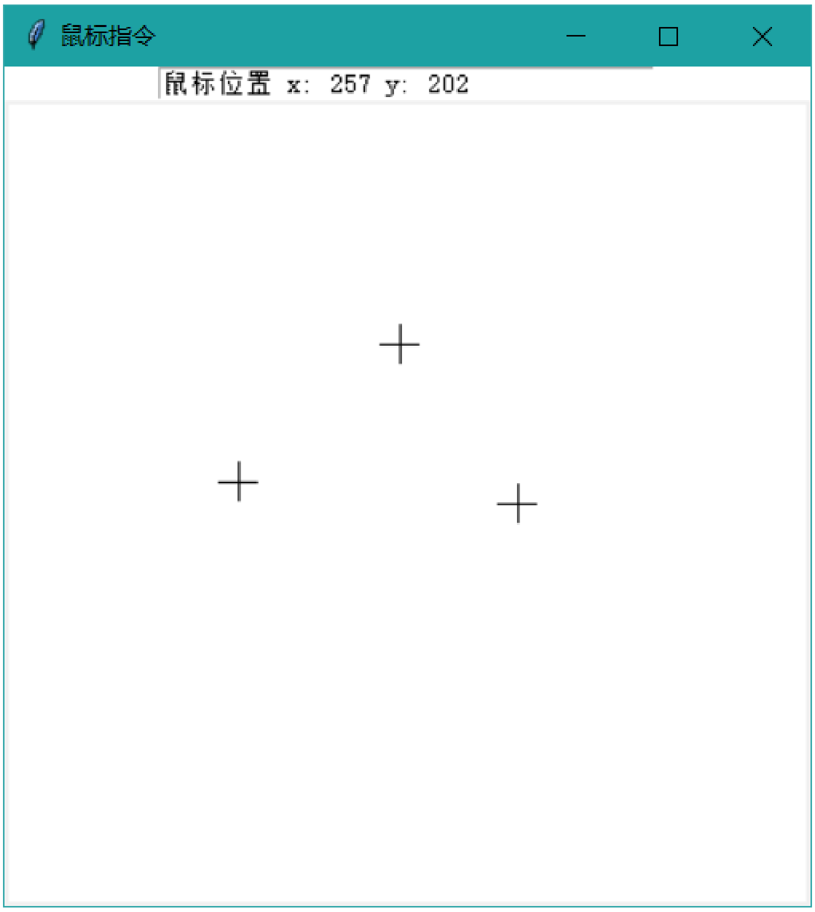




>>15、

```
from tkinter import *



def callback1(event):
    output.delete(1.0, END)
    x1 = event.x-10
    x2 = event.x+10
    y1 = event.y-10
    y2 = event.y+10
    c.create_line(x1, event.y, x2, event.y)
    c.create_line(event.x, y1, event.x, y2)
    posstr = "鼠标位置  x: "+str(event.x)+"\ty: "+str(event.y)
    output.insert(1.0, posstr)



def callback2(event):
    c.delete(ALL)



#   创建主窗口
w = Tk()
```

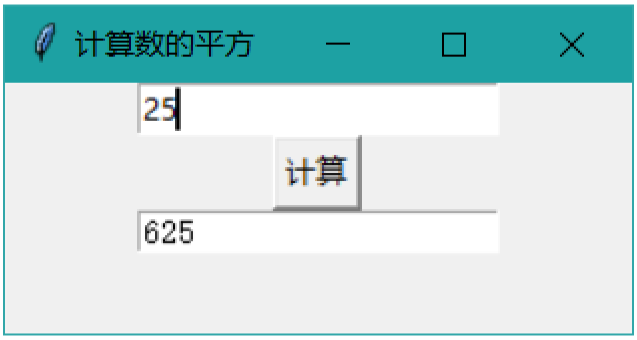
```
# 设置主窗口样式
w['width'] = 400
w['height'] = 400
w['bg'] = 'white'
w.title('鼠标指令')

# 界面设计
output = Text(w, width=35, height=1)
output.pack()
c = Canvas(w, width=400, height=400, bg='white')
c.pack()

# 鼠标指令
c.bind("<Button-1>", callback1)
c.bind("<Double-Button-1>", callback2)

w.mainloop()
```



鼠标位置 x: 257 y: 202

>>16、

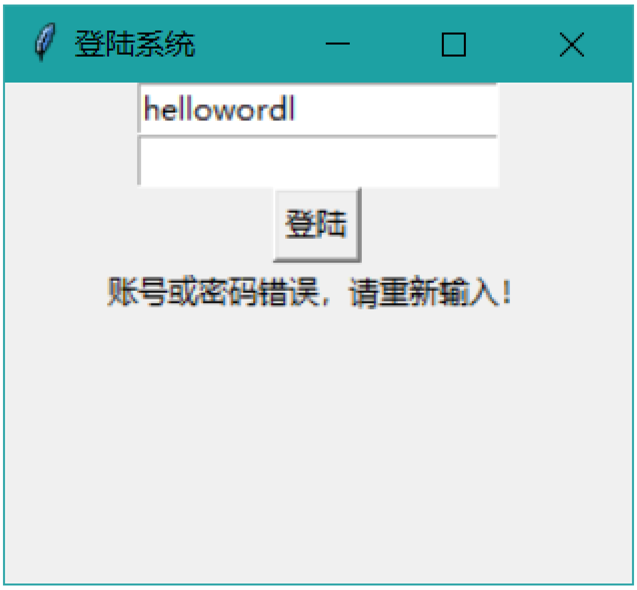
```
from tkinter import *


def callback():
    outPut.delete(1.0, END)
    txt = v.get()
    num = int(txt)
    result = num*num
    outPut.insert(1.0, result)


#  创建主窗口
w = Tk()

#  设置主窗口样式
w.geometry("250x100")
w.title('计算数的平方')

#  界面设计
v = StringVar()
inPut = Entry(w, textvariable=v)
inPut.pack()
v.set("在此处输入数字")
b = Button(w, text='计算', command=callback)
b.pack()
outPut = Text(w, width=20, height=1)
outPut.pack()

w.mainloop()
```



>>17、

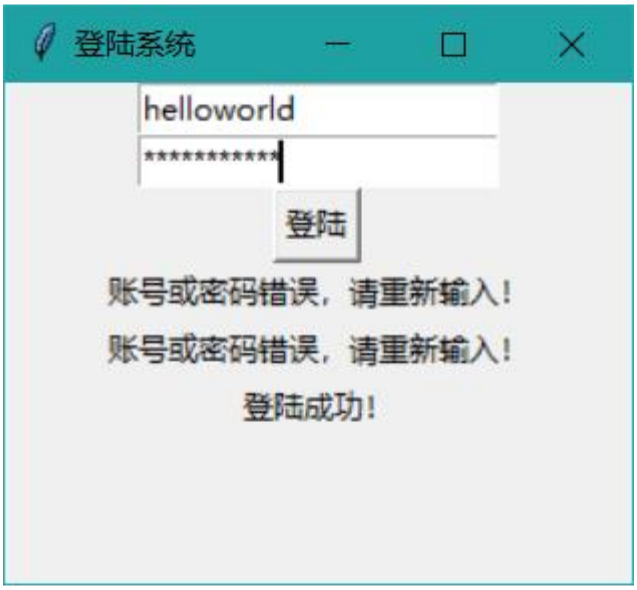
```
from tkinter import *
import random


def callback():
    account = ac.get()
    code = co.get()
    if account == 'helloworld' and code == 'hellopython':
        sstr = '登陆成功！'
    else:
        sstr = '账号或密码错误，请重新输入！'
        co.set("")
    Label(w, text=sstr).pack()


#  创建主窗口
w = Tk()

#  设置主窗口样式
w.geometry("250x200")
w.title('登陆系统')

#  界面设计
ac = StringVar()
inPut_A = Entry(w, textvariable=ac)
inPut_A.pack()
```

```
ac.set("在此处输入账号")
co = StringVar()
inPut_C = Entry(w, textvariable=co)
inPut_C.pack()
co.set("")
inPut_C.config(show='*')
Button(w, text='登陆', command=callback).pack()



w.mainloop()
```



hellowordl

账号或密码错误，请重新输入！



helloworld

****

账号或密码错误，请重新输入！

账号或密码错误，请重新输入！

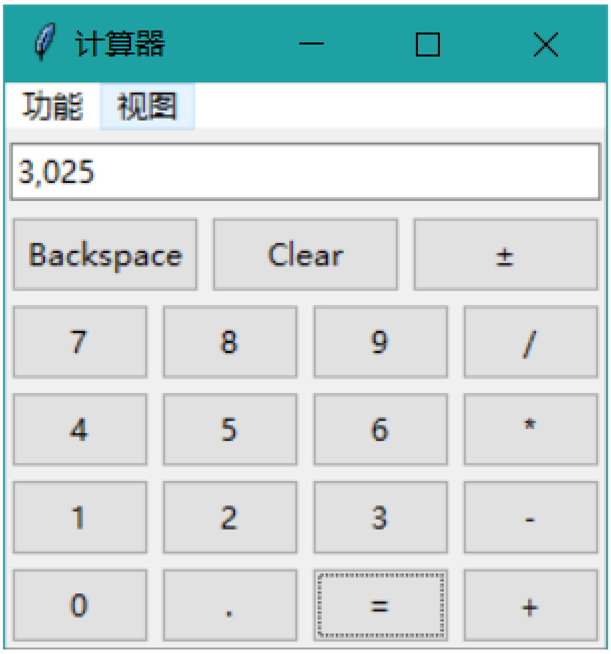
登陆成功！

>>18、

```
from tkinter import *
from tkinter.ttk import *


def my_frame(master):
    w = Frame(master)
    w.pack(side=TOP, expand=YES, fill=BOTH)
    return w


def my_button(master, text, command):
    w = Button(master, text=text, command=command, width=6)
    w.pack(side=LEFT, expand=YES, fill=BOTH, padx=2, pady=2)
    return w


def back(text):
    if len(text) > 0:
        return text[:-1]
    else:
        return text


def calc(text):
    try:
        if sep_flag.get() == 0:
            return eval(del_sep(text))
        else:
            return add_sep(str(eval(del_sep(text))))
    except (SyntaxError, ZeroDivisionError, NameError):
        return 'Error'


def add_sep(text):
    #  如果已经添加了千位分隔符，则返回删除千位分隔符的文本
    added_index = text.find(',')
    if added_index > 0:
        return text.replace(',', '')

    #  如果没有添加，那么返回添加了千位分隔符的文本
    else:
        dot_index = text.find('.')
        if dot_index > 0:
            text_head = text[:dot_index]
            text_tail = text[dot_index:]
        elif dot_index < 0:
            text_head = text
            text_tail = ''
        else:
            text_head = ''
            text_tail = text
        list_ = [char for char in text_head]
        length = len(list_)
        tmp_index = 3
        while length-tmp_index > 0:
            list_.insert(length-tmp_index, ',')
            tmp_index += 3
        list_.extend(text_tail)
        new_text = ''
        for char in list_:
            new_text += char
        return new_text



def del_sep(text):
    return text.replace(',', '')



#  创建主窗口
```

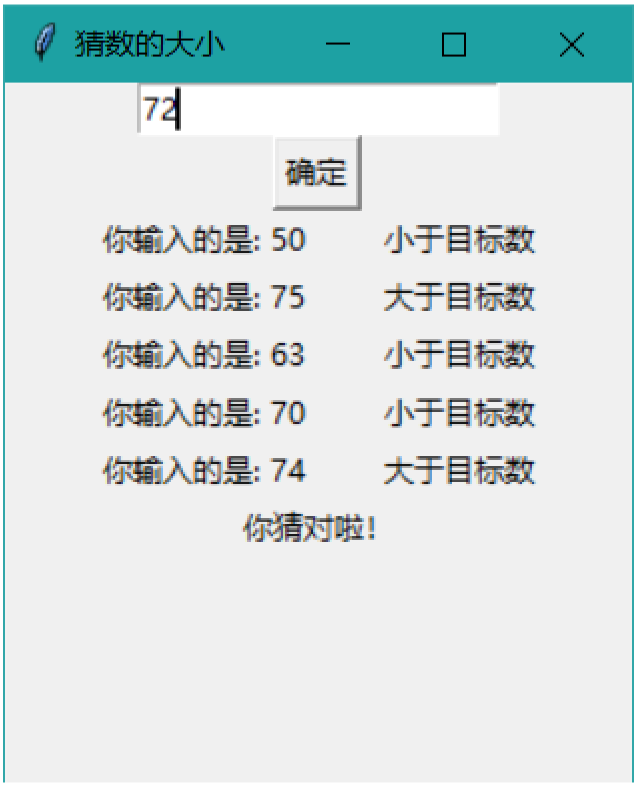
```
wind = Tk()

#   设置主窗口样式
wind.title('计算器')
main_menu = Menu(wind)

#   创建'功能'菜单
calc_menu = Menu(main_menu, tearoff=0)
calc_menu.add_command(label='退出', command=lambda: exit())
main_menu.add_cascade(label='功能', menu=calc_menu)

#   创建'视图'菜单
text = StringVar()
sep_flag = IntVar()
sep_flag.set(0)
view_menu = Menu(main_menu, tearoff=0)
view_menu.add_checkbutton(label='显示千位分隔符', variable=sep_flag,
command=lambda t=text: t.set(add_sep(t.get())))
main_menu.add_cascade(label='视图', menu=view_menu)
wind['menu'] = main_menu

#   创建文本框
Entry(wind, textvariable=text).pack(expand=YES, fill=BOTH, padx=2, pady=4)

#   创建 Style 对象
style = Style()
style.configure('TButton', padding=3)

#   创建第一行的按钮
fedit = my_frame(wind)
my_button(fedit, 'Backspace', lambda t=text: t.set(back(t.get())))
my_button(fedit, 'Clear', lambda t=text: t.set(''))
my_button(fedit, '±', lambda t=text: t.set('-('+t.get()+')'))

#   创建下面的按钮
```

```
for key in ('789/', '456*', '123-', '0.=+'):
    fsymb = my_frame(wind)
    for char in key:
        if char == '=':
            my_button(fsymb, char, lambda t=text: t.set(calc(t.get())))
        else:
            my_button(fsymb, char, lambda t=text, c=char: t.set(t.get()+c))

wind.mainloop()
```



>>19、

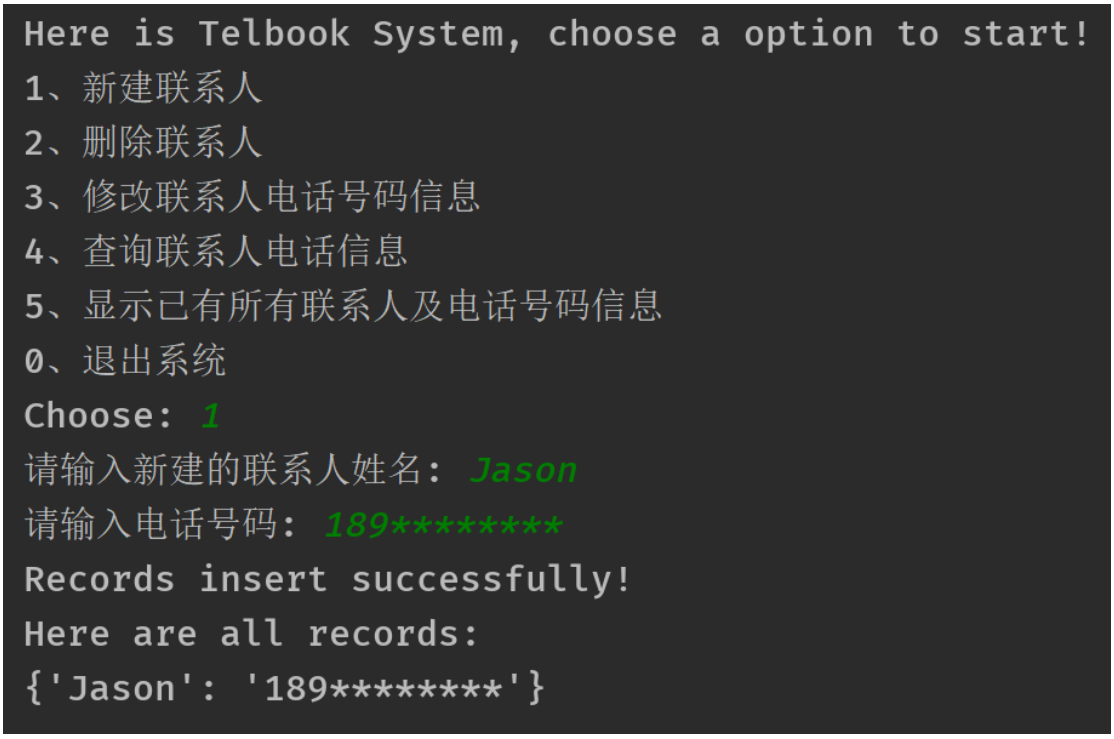
```
from tkinter import *
import random


def callback():
    txt = v.get()
    num = int(txt)
    if num > n:
        sstr = "你输入的是: " + txt + "\t 大于目标数"
    if num < n:
        sstr = "你输入的是: " + txt + "\t 小于目标数"
    if num == n:
        sstr = "你猜对啦！ "
    Label(w, text=sstr).pack()



#   随机数
n = random.randint(0, 100)

#   创建主窗口
w = Tk()

#   设置主窗口样式
w.geometry("250x500")
w.title('猜数的大小')

#   界面设计
v = StringVar()
inPut = Entry(w, textvariable=v)
inPut.pack()
v.set("在此处输入数字(0-100)")
b = Button(w, text='确定', command=callback)
b.pack()



w.mainloop()
```



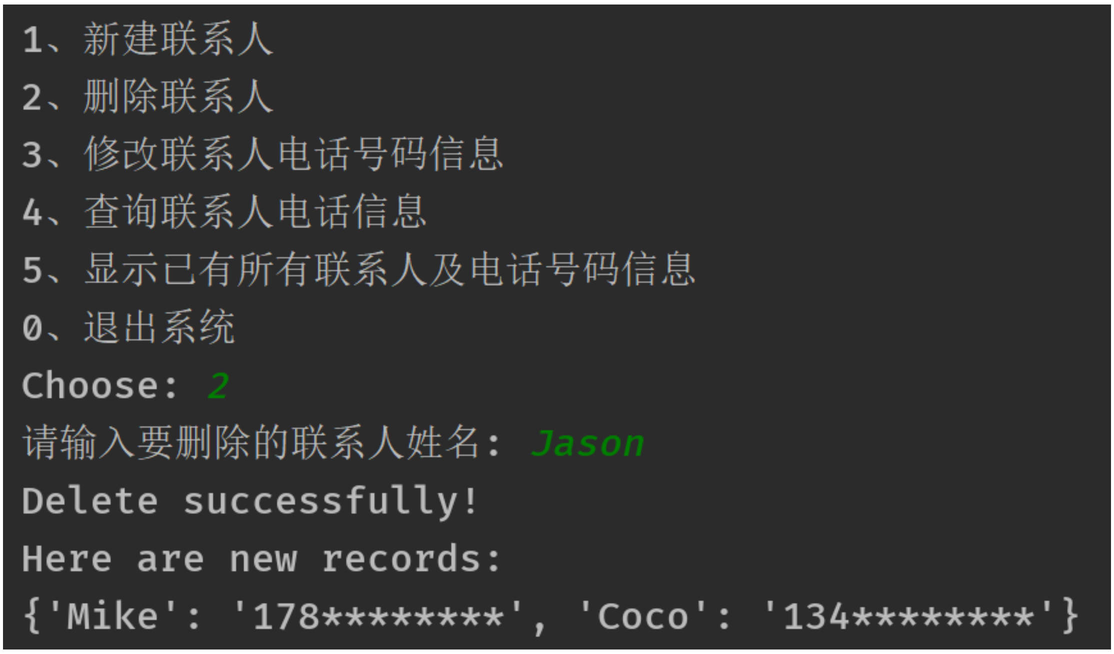
你输入的是: 50　小于目标数
你输入的是: 75　大于目标数
你输入的是: 63　小于目标数
你输入的是: 70　小于目标数
你输入的是: 74　大于目标数
你猜对啦!

>>20、

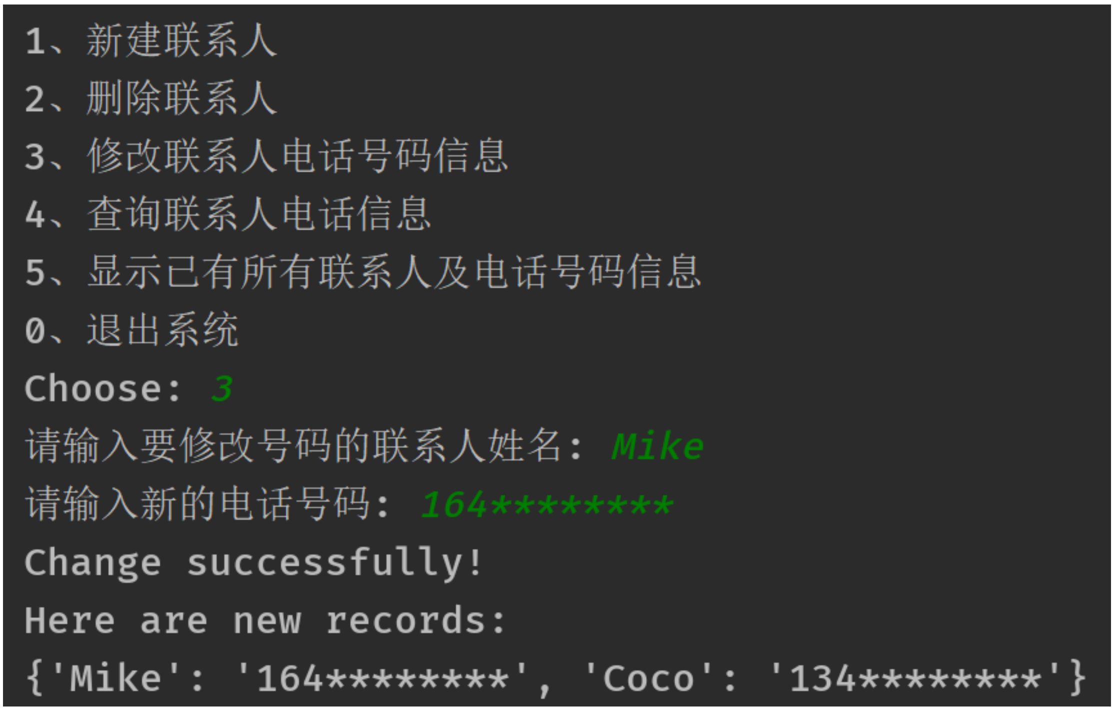
```
def insert():
    name = input('请输入新建的联系人姓名: ')
    phone = input('请输入电话号码: ')
    telbook[str(name)] = str(phone)
    print("Records insert successfully!\nHere are all records:")
    print(telbook)



def delete():
    name = input('请输入要删除的联系人姓名: ')
    del telbook[name]
    print('Delete successfully!\nHere are new records:')
    print(telbook)



def change():
    name = input('请输入要修改号码的联系人姓名: ')
    for key in sorted(telbook.keys()):
        if str(name) == key:
            phone = input('请输入新的电话号码: ')
            telbook[str(key)] = str(phone)
            print('Change successfully!\nHere are new records:')
            print(telbook)
            return
    print('该联系人不存在，请查证！ ')


def show_all():
    print('Here are all records:')
    print(telbook)


def search():
    name = input('请输入联系人姓名: ')
    for key in sorted(telbook.keys()):
        if str(name) == key:
            print('联系人 '+key+' 的电话号码为: '+telbook[key])
            return
    print('该联系人不存在，请查证！ ')


telbook = {}
print('Here is Telbook System, choose a option to start!')
while 1:
    choice = input('1、新建联系人\n2、删除联系人\n3、修改联系人电话号码信息\n4、查询联系人电话信息\n5、显示已有所有联系人及电话号码信息\n0、退出系统\nChoose: ')
    if int(choice) == 1:
        insert()
    elif int(choice) == 2:
        delete()
    elif int(choice) == 3:
        change()
    elif int(choice) == 4:
        search()
    elif int(choice) == 5:
        show_all()
    elif int(choice) == 0:
        exit(1)
    else:
        print('请输入正确选项序号!')
    print('\n')
```



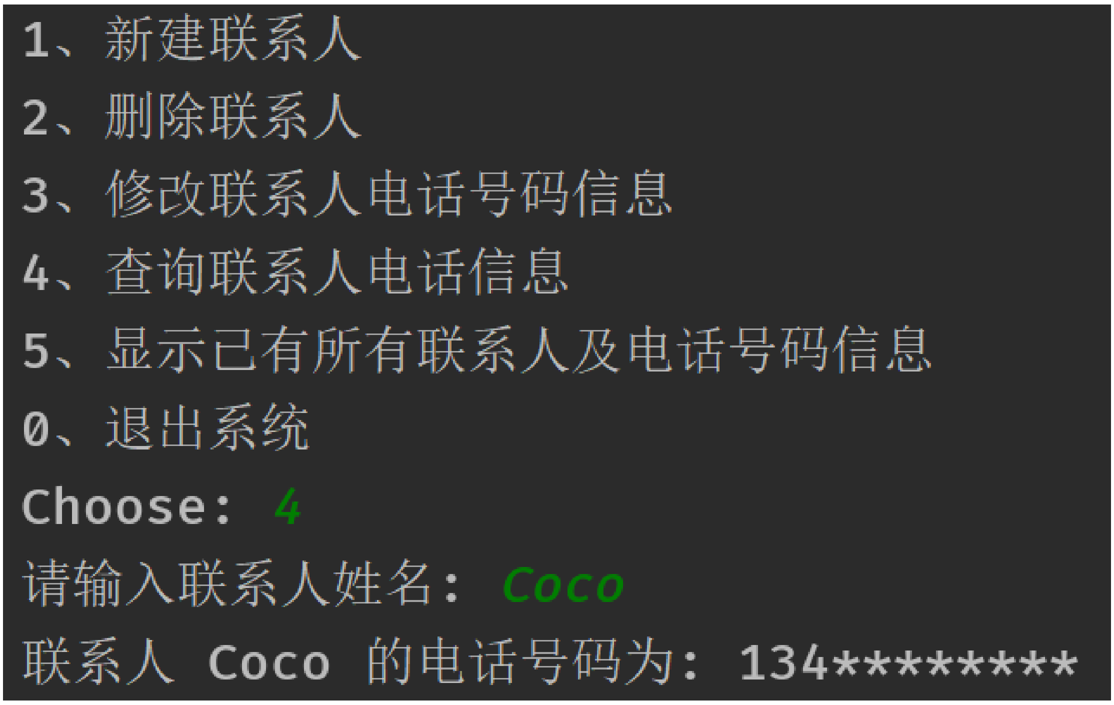
```
Here is Telbook System, choose a option to start!
1、新建联系人
2、删除联系人
3、修改联系人电话号码信息
4、查询联系人电话信息
5、显示已有所有联系人及电话号码信息
0、退出系统
Choose: 1
请输入新建的联系人姓名: Jason
请输入电话号码: 189********
Records insert successfully!
Here are all records:
{'Jason': '189********'}
```



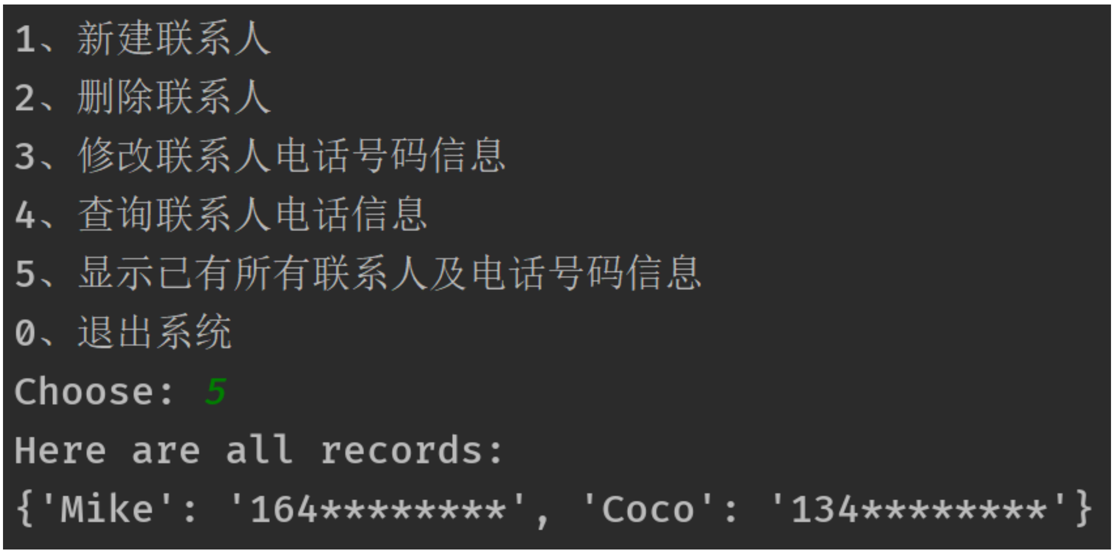
```
1、新建联系人
2、删除联系人
3、修改联系人电话号码信息
4、查询联系人电话信息
5、显示已有所有联系人及电话号码信息
0、退出系统
Choose: 2
请输入要删除的联系人姓名: Jason
Delete successfully!
Here are new records:
{'Mike': '178********', 'Coco': '134********'}
```



```
1、新建联系人
2、删除联系人
3、修改联系人电话号码信息
4、查询联系人电话信息
5、显示已有所有联系人及电话号码信息
0、退出系统
Choose: 3
请输入要修改号码的联系人姓名: Mike
请输入新的电话号码: 164********
Change successfully!
Here are new records:
{'Mike': '164********', 'Coco': '134********'}
```



```
1、新建联系人
2、删除联系人
3、修改联系人电话号码信息
4、查询联系人电话信息
5、显示已有所有联系人及电话号码信息
0、退出系统
Choose: 4
请输入联系人姓名: Coco
联系人 Coco 的电话号码为: 134*********
```



```
1、新建联系人
2、删除联系人
3、修改联系人电话号码信息
4、查询联系人电话信息
5、显示已有所有联系人及电话号码信息
0、退出系统
Choose: 5
Here are all records:
{'Mike': '164*********', 'Coco': '134*********'}
```