《Python 程序设计(第 3 版)》
练习答案（部分）

黄锐军　编著

练习一

1.输入矩形的长与宽，计算矩形面积

2.输入一个时间值 s，它是距当日午夜的秒值，计算目前的时间，时间按 HH:MM:SS 格式输出。

3．如 a=1、b=2、c=3、d=0 写出下列的逻辑值：

（1）a>b and b>c or a+b<c

（2）a-b<c or b>c and not c

（3）not d or b>c+a or a

（4）d and b and c>d and a*b>c

（5）not（a>b and c>d）

（6）a*b>c or b+c>d and not d

（7）c+d<=b+d and d<c or 2*b>c

（8）d<b or c>a+b+d and b<c+a

4.有一个数 x 在区间[-5,0]内，写出其条件表达式。

5.写出下面表达式的值(设 a=1,b=2,c=3,x=4,y=3)

（1）a+b>c and b==c

（2）not a<b and b not =c or x+y<=3

（3）a+(b>=x+y) and c-a and y-x

（4）not（x=a）and（y=b）and 0

（5）not（a+b)+c-1 and b+c/2

（6）a or 1+'a' and b and 'c'

6. 校园贷是指在校学生向正规金融机构或者其他借贷平台借钱的行为，但是有些不法分子打着校园贷的幌子进行诈骗，坑害了部分学生，同学们要提高警惕，防止受骗。例如某同学看中了一款价格 8000 元的手机，但是苦于没有钱购买。偶然一天接触到了校园贷，可以借贷 10000 元，日利息是 0.5%，借贷时间是 150 天（三个月）。该同学开始觉得日利息很低，产生了想借贷的想法。同学们，我们帮这位同学算一算，150 天后连本带利他要还多少钱？

答：

```
loan=10000
interest=0.005
days=150
pay=loan*(1+interest)**days
print("Total payment is:",pay)
```

练习二

1. 输入 a、b、c 三个参数，求解 $ax^2+bx+c=0$ 的两个根，假定 $b^2-4ac>0$。

答：

```
a=input("a=")
b=input("b=")
c=input("c=")
a=float(a)
b=float(b)
c=float(c)
if a!=0:
    d=b*b-4*a*c
    if d>0:
        d=math.sqrt(d)
        x1=(-b+d)/2/a
        x2 = (-b - d) / 2 / a
        print("x1=",x1,"x2=",x2)
    elif d==0:
        print("x1,x2=", -b/2/a)
    else:
        print("无实数解")
else:
    print("不是一元二次方程")
```

2．输入 a、b、c 三个参数，以它们作为三角形的三边，判断是否可以构成一个三角形，如能则进一步计算其面积。三角形的面积 s 可以用以下公式计算：

s=sqrt(p*(p-a)*(p-b)*(p-c))

其中 p=(a+b+c)/2。

答：

```
import math
a=input("a=")
b=input("b=")
c=input("c=")
a=float(a)
b=float(b)
c=float(c)
if a+b>c and a+c>b and b+c>a:
    print("Triangle is valid")
    p=(a+b+c)/2
    s=math.sqrt(p*(p-a)*(p-b)*(p-c))
    print("Area of triangle is:",s)
```

```
else:
    print("Triangle is not valid")
```

3．输入一个字母，判断它是否为小写英文字母。

答：

```
c=input("c=")
if c>="a" and c<="z":
    print("c is a lowercase letter")
else:
    print("No")
```

4．从键盘输入 5 个字符，统计’0’字符出现的次数。

答：

```
count=0
for i in range(5):
    c=input("c=")
    if c=="0":
        count+=1
print("count=",count)
```

5．输入两个整数，判断哪个大并输出结果。

答：

```
x=input("x=")
y=input("y=")
x=float(x)
y=float(y)
if x>y:
    print(x,"is greater than",y)
elif x<y:
    print(y,"is greater than",x)
else:
    print(x,"is equal to",y)
```

6．输入一个字母，如它是一个小写英文字母，则把它变为对应大写字母输出。

答：

```
x=input("x=")
if x>="a" and x<="z":
    x=chr(ord(x)-ord("a")+ord("A"))
    print(x)
```

7．输入一个年份，判断它是否为闰年。

答：

```
y=input("year=")
```

```
y=int(y)
if y%4==0 and y%100!=0 or y%400==0:
    print("leap year")
else:
    print("not a leap year")
```

8. 从键盘输入 a、b 两个数，按大小顺序输出它们。

答：

```
a=input("a=")
b=input("b=")
if a>b:
    print(a,b)
else:
    print(b,a)
```

9. 输入 a、b、c 三个整数，找出最小的数。

答：

```
a=input("a=")
b=input("b=")
c=input("c=")
a=float(a)
b=float(b)
c=float(c)
if a>b:
    a,b=b,a
if a>c:
    a,c=c,a
if b>c:
    b,c=c,b
print(a,b,c)
```

10. 某企业发放的奖金根据利润提成。利润低于或等于 10 万元时，奖金可提 12%；利润高于 10 万元，低于 20 万元时，高于 10 万元的部分，可提成 8.5%；20 万到 40 万之间时，高于 20 万元的部分，可提成 6%；40 万到 60 万之间时高于 40 万元的部分，可提成 4%；60 万到 100 万之间时，高于 60 万元的部分，可提成 2.5%，高于 100 万元时，超过 100 万元的部分按 1%提成，从键盘输入当月利润，求应发放奖金总数？

答：

```
x=input("输入利润(万元):")
x=float(x)
y=0
if x<10:
    y=0.012*x
elif x<20:
```

```
        y=10*0.12+(x-10)*0.085
elif x<40:
        y=10*0.12+10*0.085+(x-20)*0.06
elif x<60:
        y=10*0.12+10*0.085+10*0.06+(x-40)*0.04
elif x<100:
        y=10*0.12+10*0.085+10*0.06+10*0.04+(x-60)*0.025
else:
        y=10*0.12+10*0.085+10*0.06+10*0.04+10*0.025+(x-100)*0.001
print("奖金为：%.2f 万元"%y)
```

11．平面上有四个圆，圆心分别为(2，2)，(-2，2)，(-2，-2)，(2，-2)，圆半径为 1。今输入任一点的坐标，判断该点是否在这四个圆中，如在则给出是在哪一个圆中。

答：

```
import math
x=input("x=")
y=input("y=")
x=float(x)
y=float(y)
flag=False
for a in range(-1,2):
    for b in range(-1,2):
        if a!=0 and b!=0:
            p=a*2
            q=b*2
            print("Checking circle (",p,",",q,")")
            if math.sqrt((x-p)**2+(y-q)**2)<1:
                print("Point is in the circle (",p,",",q,")")
                flag=True
                break
if not flag:
    print("Point is not in any circle")
```

练习三

1. 计算 1+2+4+……+100 的和。

答：

```
s=0
for i in range(1,101):
    s=s+i
print(s)
```

2. 计算 1+1/3+1/5+……+1/99 的和。

答：

```
s=0
for i in range(1,100,2):
    s=s+1/i
print(s)
```

3. 从键盘输入一个字符串，直到回车结束，统计字符串中的大小写英文字母各有多少个。

答：

```
s=input()
a=0
b=0
for x in s:
    if x>="a" and x<="z":
        a+=1
    elif x>="A" and x<="Z":
        b+=1
print("小写：",a," 大写:",b)
```

4. 有一分数序列：2/1，3/2，5/3，8/5，13/8，21/13…，求出这个数列的前 20 项之和。

答：

```
s=0
a=1
b=2
t=""
for i in range(20):
    t=t+str(b)+"/"+str(a)
    if i<19:
        t=t+"+"
    else:
        t=t+"="
    s=s+b/a
    c=a+b
    a=b
    b=c
print(t+str(s))
```

6. 输入若干个同学的成绩，计算平均成绩，输入的成绩为负数或大于 100 时表示结束输入。

答：

```
s=0
n=0
while True:
    m=float(input())
    if m<0 or m>100:
        break
    s=s+m
    n=n+1
print("平均:",s/n)
```

7. 输入三个正整数 a、b、n，精确计算 a/b 的结果到小数后 n 位。

答：

```
a=input("a=")
b=input("b=")
n=input("n=")
a=int(a)
b=int(b)
n=int(n)
s=str(a//b)+"."
a=a%b
for i in range(n-1):
     a*=10
     s+=str(a//b)
     a=a%b
print(s)
```

8. 一个猴子第一天摘下若干个桃子，当即吃了一半，还不过瘾，又多吃了一个。第二天早上又将剩下的桃子吃掉一半，又多吃了一个。以后每天早上都吃了前一天剩下的一半零一个。到第 10 天早上想再吃时，见只剩下一个桃子了。求第一天共摘了多少。

答：

```
n=10
while True:
    m=n
    flag=True
    for i in range(9):
        if m%2==0:
            m=m-m//2-1
        else:
            flag=False
            break
```

```
    if m==1 and flag:
        break
    n=n+1
print("桃子数：",n)
for i in range(9):
    print(i+1,n,"->",n-n//2-1)
    n=n-n//2-1
    结果：
    桃子数： 1534
    1 1534 -> 766
    2 766 -> 382
    3 382 -> 190
    4 190 -> 94
    5 94 -> 46
    6 46 -> 22
    7 22 -> 10
    8 10 -> 4
    9 4 -> 1
```

9. 有一序列：1，3，5，8，13，21...，用 while 循环求出这个数列的前 20 项之和。

答：

```
a=3
b=5
s=1+3+5
t="1+3+5+"
i=0
while i<17:
    c=a+b
    s=s+c
    t=t+str(c)
    if(i<16):
        t=t+"+"
    a=b
    b=c
    i=i+1
print(t+"="+str(s))
```

结果：

```
1+3+5+8+13+21+34+55+89+144+233+377+610+987+1597+2584+4181+6765+10946+17711=4636
4
```

10.一个数如正好等于它的所有因子之和，则称为完数，例如 6 的因子有 1、2、3，而 6=1+2+3，因此 6 是一个完数。编程序找出 1000 之内的所有完数。

答：

```
def isOK(n):
```

```
    s=0
    t=""
    for m in range(1,n):
        if n%m==0:
            s=s+m
            t=t+str(m)+"+"
    if s==n:
        return t.strip("+")
    return ""

for n in range(1,1001):
    t=isOK(n)
    if t:
        print(n,"=",t)
```

结果：

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

11．有近千名学生排队，7 人一行余 3 人，5 人一行余 2 人，3 人一行余 1 人，编写程序求学生人数。

答：

```
n=800
while True:
    if n%7==3 and n%5==2 and n%3==1:
        break
    n=n+1
print(n)
```

12．小华今年 12 岁，他妈妈比他大 20 岁，编写程序计算多少年后他妈妈年龄比他大一倍。

答：

```
xh=12
mm=12+20
y=0
while True:
    y=y+1
    xh+=1
    mm+=1
    if mm==2*xh:
        break
print(y,"年后:",xh,mm)
```

13．两个乒乓球队进行比赛，各出三人。甲队为 a, b, c 三人，乙队为 x, y, z 三人。已抽签决定比赛名单。有人向队员打听比赛的名单。a 说他不和 x 比，c 说他不和 x, z 比，请编程序

找出三队赛手的名单。

答：

```
for a in ['x','y','z']:
    for b in ['x','y','z']:
        for c in ['x','y','z']:
            if a!=b and a!=c and b!=c and a!='x' and c!='x' and c!='z':
                print("a-",a,"b-",b,"c-",c)
```

结果：

```
a- z b- x c- y
```

14. 目前世界人口是 60 亿，如每年按 1.5%的比例增长，则多少年后是 80 亿？

答：

```
p=60
y=0
while True:
    y=y+1
    p=p*1.015
    if p>=80:
        break
print(y)
```

15. 一球从 80 米高度自由下落，每次落地后返回原高度的一半，再落下。求它在第 10 次落地时共经过多少米？第 10 次反弹多高？

答：

```
h=80
s=h
t=str(h)+"+"
for i in range(9):
    s=s+h
    t=t+str(h)+"+"
    h=h/2
print(t.strip("+"),"=",s)
print("第 10 次：",h)
```

结果：

```
80+80+40.0+20.0+10.0+5.0+2.5+1.25+0.625+0.3125 = 239.6875
第 10 次： 0.15625
```

练习四

1. 能直接修改字符串的某个字符吗？例如 s="abc",s[0]= "1"可以吗？
   答： 不行
2. 输入一个字符串，输出它所包含的所有数字，例如输入"23me3e"，输出"233"
   答：

```
    s=input()
    for x in s:
        if x>="0" and x<="9":
            print(x,end="")
print()
```

3. 设计一个字符串函数 reverse(s)，它返回字符串 s 的反串，例如 reverse("abc")返回"cba"。
   答：

```
    def reverse(s):
        t=""
        for e in s:
            t=e+t
    return t
```

4. 元祖与列表有什么不同？
5. 一个列表中的元素类型要求一致吗？例如 list=[1, "a"]是正确的吗？
6. 列表是否还可以嵌套别的列表？列举一个例子说明。
7. 用一个字典描述一个日期，包含年 year、月 month、日 day 的键字。
8. Python 的字典数据类型与 JSON 数据类有很多相似的地方，说明有哪些共同点。
9. 写出下列程序执行的结果：

```
d={"students":[{"name":"A","sex":"M"},{"name":"B","sex":"C"}]}
for k1 in d.keys():
    for k2 in d[k1]:
        for k3 in k2.keys():
            print(k3,k2[k3])
```

   答：

```
name A
sex M
name B
sex C
```

10. 如果使用字典描述一个时间，例如 t={" hour":12, "minute":23, "second":34}表示时间"12:23:34",设计一个函数 interval(t1,t2)，计算时间 t1 与 t2 的时间差，返回相同结构的一个字典时间。

答：

```
def interval(t1,t2):
    a=t1["hour"]*3600+t1["minute"]*60+t1["second"]
    b=t2["hour"]*3600+t2["minute"]*60+t2["second"]
    if a>b:
        c=a-b
    else:
        c=b-a
    t={}
    t["hour"]=c//3600
    t["mimute"]=(c-t["hour"]*3600)//60
    t["second"]=c-c-t["hour"]*3600-t["minute"]*60
return t
```

练习五

1. 定义一个数学中的复数类 Complex，它有一个构造函数与一个显示函数，建立一个 Complex 对象并调用该显示函数显示。

答：

```
    class Complex:
        def __init__(self,x,y):
            self.x=x
            self.y=y
        def show(self):
            print("("+str(self.x)+","+str(self.y)+")")

    c=Complex(1,2)
c.show()
```

2.定义一个计算机类 MyComputer，它包含 CPU 类型(String 类型)、RAM 内存大小(Integer 类型)、HD 硬盘大小(Integer 类型)，设计它的构造函数，并设计一个显示函数，建立一个 MyComputer 对象并调用该显示函数显示。

答：

```
class Computer:
    def __init__(self,CPU,RAM,HD):
        self.CPU=CPU
        self.RAM=RAM
        self.HD=HD
    def show(self):
        print(self.CPU,self.RAM,self.HD)

c=Computer("Intel i-7",8,500)
c.show()
```

2. 设计一个整数类 MyInteger，它有一个整数变量，并有一个 Value 属性，可以通过为 Value 存取该变量的值，还有一个转二进制字符串的成员函数 toBin 及转十六进制字符串的成员函数 toHex。

答：

```
    class MyInteger:
        def __init__(self,v):
            self.Value=v
        def show(self):
            print(self.Value)
        def toHex(self):
            return hex(self.Value)
        def toBin(self):
```

```
            return bin(self.Value)

        i=MyInteger(10)
        i.show()
        print(i.toHex())
    print(i.toBin())
```

4. 建立一个普通人员类 Person，包含姓名(m_name)、性别(m_sex)、年龄(m_age)成员变量。

(1) 建立 Person 类，包含 Private 成员 m_name、m_sex、m_age 成员变量；

(2) 建立 Person 的构造函数；

(3) 建立一个显示过程 Show()，显示该对象的数据；

(4) 派生一个学生类 Student，增加班级(m_class)，专业(m_major)，设计这些类的构造函数；

(5) 建立 m_class、m_major 对应的属性函数 sClass()、sMajor()；

(6) 建立显示成员函数 Show()，显示该学生对象所有成员数据；

答：

```
class Person:
    def __init__(self,name,sex,age):
        self.__m_name=name
        self.__m_sex=sex
        self.__m_age=age
    def show(self,end="\n"):
        print(self.__m_name,self.__m_sex,self.__m_age,end=end)

class Student(Person):
    def __init__(self,name,sex,age,mclass,major):
        Person.__init__(self,name,sex,age)
        self.__m_class=mclass
        self.__m_major=major
    def sClass(self):
        return self.__m_class
    def sMajor(self):
        return self.__m_major
    def show(self):
        Person.show(self,end=" ")
        print(self.__m_class,self.__m_major)

s=Student("A","M",20,"Class A","Computer")
s.show()
```

5. 根据本章建立的 MyDate 类与 MyTime 类，设计一个日期时间类 MyDateTime：

```
class MyDateTime:
    def __init__(self,year,month,day,hour,minute,second):
        self.date=MyDate(year,month,day)
```

```
        self.time=MyTime(hour,minute,second)
```

它包含日期属性 date 与时间属性 time，请继续编写下面函数：

(1) MyDateTime 的显示函数 show(self)；

(2) MyDateTime 转字符串函数 toString(self)；

(3) 把字符串转为 MyDateTime 对象的静态函数 fromString(s)；

(4) MyDateTime 的比较函数 compare(self,dt)。

答：

```
class MyDate:
    def __init__(self,y,m,d):
        months=[0,31,28,31,30,31,30,31,31,30,31,30,31]
        if y<0:
            raise Exception("无效年份")
        if m<1 or m>12:
            raise Exception("无效月份")
        if y%400==0 or y%4==0 and y%100!=0:
            months[2]=29
        else:
            months[2]=28
        if d<1 or d>months[m]:
            raise Exception("无效日期")
        self.year=y
        self.month=m
        self.day=d

    def show(self,end='\n'):
        print("%04d-%02d-%02d" %(self.year,self.month,self.day),end=end)

    @staticmethod
    def fromString(s):
        y,m,d=s.split('-')
        return MyDate(int(y),int(m),int(d))

    def toString(self):
        return "%04d-%02d-%02d" %(self.year,self.month,self.day)

    def compare(self,d):
        if self.year<d.year:
            return -1
        elif self.year>d.year:
            return 1
        else:
            if self.month<d.month:
                return -1
            elif self.month>d.month:
```

```
                return 1
            else:
                if self.day<d.day:
                    return -1
                elif self.day>d.day:
                    return 1
                else:
                    return 0


class MyTime:
    def __init__(self,hour,mimute,second):
        if hour>=0 and hour<24 and mimute>=0 and mimute<60 and second>=0 and
second<60:
            self.hour=hour
            self.minute=mimute
            self.second=second
        else:
            raise Exception("无效时间")

    def show(self,end='\n'):
        print("%02d:%02d:%02d"%(self.hour,self.minute,self.second),end=end)

    def toString(self):
        return "%02d:%02d:%02d"%(self.hour,self.minute,self.second)

    def toSeconds(self):
        return self.hour*3600+self.minute * 60 + self.second

    @staticmethod
    def fromString(string):
        h,m,s=string.split(":")
        return MyTime(int(h),int(m),int(s))

    def compare(self,t):
        a=self.toSeconds()
        b=t.toSeconds()
        if a>b:
            return 1
        elif a<b:
            return -1
        else:
            return 0
```

```
class MyDateTime:
    def __init__(self,year,month,day,hour,minute,second):
        self.date=MyDate(year,month,day)
        self.time=MyTime(hour,minute,second)

    def show(self,end='\n'):
        self.date.show(end=' ')
        self.time.show(end=end)

    def toString(self):
        return self.date.toString() + " " + self.time.toString()

    @staticmethod
    def fromString(s):
        a,b=s.split(" ")
        date=MyDate.fromString(a)
        time=MyTime.fromString(b)
        return MyDateTime(date.year,date.month,date.day,
                          time.hour,time.minute,time.second)

    def compare(self,dt):
        if self.date.compare(dt.date)==0:
            return self.time.compare(dt.time)
        else:
            return self.date.compare(dt.date)
try:
    a=MyDateTime(2024,6,30,23,59,50)
    b=MyDateTime.fromString("2024-05-19 10:00:06")
    a.show()
    b.show()
    sa=a.toString()
    sb=b.toString()
    c=a.compare(b)
    if c==0:
        print(sa,"is equal to ",sb)
    elif c==-1:
        print(sa,"is earlier than ",sb)
    else:
        print(sa,"is later than ",sb)
except Exception as e:
    print(e)
```

练习六

1. 编写一个程序建立一个文本文件 abc.txt，向其中写入”abc\n”并存盘，查看 abc.txt 是几个字节的文件，说明为什么。

   答：4 字节

2. 用 Windows 记事本编写一个文本文件 xyz.txt，在其中存入”123”后打”Enter”键换行，存盘后查看文件应是 5 个字节长，用 read(1)读该文件，看看要读 5 次还是 4 次就把文件读完，为什么？编写程序验证。

   答：5 字节

3. 编写程序查找某个单词（键盘输入）所出现的行号及该行的内容，并统计该单词在文件共出现多少次。

答：

```
def scanWord(s):
    words=[]
    i=0
    w=""
    while i<len(s):
        if s[i].lower()>="a" and s[i].lower()<="z":
            w=w+s[i]
        else:
            if w:
                words.append(w)
                w=""
        i=i+1
    if w:
        words.append(w)
    return words

def countWord(word,words):
    n=0
    for w in words:
        if w==word:
            n+=1
    return n

def countWordInFile(word,fileName):
    count=0
    f=open(fileName,"r")
    rows=f.readlines()
    no=0
    for row in rows:
        no+=1
```

```
            words=scanWord(row)
            if word in words:
                print(no,row,end="")
                count+=countWord(word,words)
        f.close()
        print("Total ",count)

fileName=input("输入文件：")
word=input("输入单词：")
countWordInFile(word,fileName)
```

4．设一个文本文件 marks.txt 中存储了学生的姓名及成绩如下：

张三　96

李四　95

……

每行为学生姓名与成绩，编写一个程序读取这些学生的姓名与成绩并按成绩的从高到低的顺序输出到另外一个文件 sorted.txt 中。

答：

```
def read(fileName):
    st=[]
    f=open(fileName,"r")
    rows=f.readlines()
    for row in rows:
        s=row.strip("\n").split(" ")
        if len(s)==2:
            d={"Name":s[0],"Mark":float(s[1])}
            st.append(d)
    f.close()
    return st

def sort(st):
    for i in range(len(st)):
        for j in range(i+1,len(st)):
            if st[j]["Mark"]<st[i]["Mark"]:
                t=st[i]
                st[i]=st[j]
                st[j]=t
    return st

def save(st):
    f=open("f:\\sorted.txt","w")
    for s in st:
        f.write(s["Name"]+" "+str(s["Mark"])+"\n")
    f.close()
```

```
fileName="f:\\marks.txt"
st=read(fileName)
st=sort(st)
save(st)
```

5．用二进制方式创建文件 abc 并写入”abc\r\nxyz\n\r123\r456\n”的字符串，查看该文件有多少个字节，再用文本文件的方式打开此文件并用 fgetc 来读，看看每次读出的是什么字符。

答：

```
def save(s):
    f=open("f:\\data.txt","wb")
    #wb: write binary
    f.write(s.encode())
    f.close()

def read():
    f=open("f:\\data.txt","rb")
    #rb: read binary
    data=f.read()
    s=data.decode()
    f.close()
    print(s)

s="abc\r\nxyz\n\r123\r456\n"
save(s)
read()
```

6．用文本文件方式创建文件并写入”abc\nxyz\r\n123\r”字符串，再用二进制方式打开读取每个字节，看看每个字节是什么。

答：

```
def save(s):
    f=open("f:\\data.txt","w")
    f.write(s)
    f.close()

def read():
    f=open("f:\\data.txt","rb")
    #rb: read binary
    data=f.read()
    for d in data:
        print(hex(d),end=" ")
    f.close()
```

```
s=" abc\nxyz\r\n123\r "
save(s)
read()
```

7. 编写一个程序，它能统计一篇英文文章中的 a~z 各个字母出现的次数（不分大小写），并按出现次数的多少按顺序输出。

答：

```
def count(fileName):
    no={}
    for i in range(26):
        x=chr(ord('a')+i)
        no[x]=0
    f=open(fileName,"r")
    rows=f.readlines()
    for row in rows:
        s=row.strip("\n")
        for x in s:
            y=x.lower()
            if y>='a' and y<='z':
                no[y]=no[y]+1
    f.close()
    print(no)

fileName=input( "输入文件：")
count(fileName)
```

练习七

1.简述 Python 连接 MySQL 数据库的方法？
2.fetchone()怎么样判断读到数据库末尾？
3.fetchall()与 fetchone()有什么区别？
4.rowcount()返回值代表什么？
5.MySQL 中怎么样使用带参数的 SQL 命令？
6.数据库操作完成后为什么要调用 commit()函数？
7.什么是游标 cursor，它有什么作用？
8.Python 怎么样连接 Sqlite 数据库？
9. 把学生成绩管理实践项目改成 Sqlite 数据库。
答：

```
import sqlite3
import os

class MarkDB:
    def open(self):
        self.con = sqlite3.connect("marks.db")
        self.cursor = self.con.cursor()
        try:
            sql = "create table marks (pNo varchar(16) primary key,pName
varchar(16) ,pChinese float,pMath float,pEnglish float)"
            self.cursor.execute(sql)
        except:
            pass

    def close(self):
        self.con.commit()
        self.con.close()

    def clear(self):
        try:
            self.cursor.execute("delete from marks")
        except Exception as err:
            print(err)

    def show(self):
        try:
            self.cursor.execute("select pNo,pName,pChinese,pMath,pEnglish from
marks")
            print("%-16s%-16s%-8s%-8s%-8s" % ("学号", "姓名", "语文", "数学","
英语"))
```

```
            rows = self.cursor.fetchall()
            for row in rows:
                print("%-16s%-16s%-8.0f%-8.0f%-8.0f" % (row[0], row[1], row[2],
row[3],row[4]))
        except Exception as err:
            print(err)

    def __insert(self, pNo, pName, pChinese, pMath,pEnglish):
        try:
            sql = "insert into marks (pNo,pName,pChinese,pMath,pEnglish) values
(?,?,?,?,?)"
            self.cursor.execute(sql, (pNo, pName, pChinese, pMath,pEnglish))
            print(self.cursor.rowcount, " row inserted")
        except Exception as err:
            print(err)

    def __update(self, pNo, pChinese, pMath,pEnglish):
        try:
            sql = "update marks set pChinese=?,pMath=?,pEnglish=? where pNo=?"
            self.cursor.execute(sql, ( pChinese, pMath, pEnglish, pNo))
            print(self.cursor.rowcount, " row updated")
        except Exception as err:
            print(err)

    def __delete(self, pNo):
        try:
            sql = "delete from marks where pNo=?"
            self.cursor.execute(sql, (pNo,))
            print(self.cursor.rowcount, " row deleted")
        except Exception as err:
            print(err)

    def enterMark(self,s):
        while True:
            m=input(s)
            try:
                m=float(m)
                if m>=0 and m<=100:
                    break
            except Exception as err:
                print(err)
        return m

    def insert(self):
```

```
        pNo=input("学号:").strip()
        pName=input("姓名:").strip()
        if pNo!="" and pName!="":
            pChinese=self.enterMark("语文:")
            pMath = self.enterMark("数学:")
            pEnglish = self.enterMark("英语:")
            self.__insert(pNo,pName,pChinese,pMath,pEnglish)
        else:
            print("学号、姓名不能空")

    def update(self):
        pNo = input("学号:").strip()
        if pNo != "":
            pChinese = self.enterMark("语文:")
            pMath = self.enterMark("数学:")
            pEnglish = self.enterMark("英语:")
            self.__update(pNo, pChinese, pMath, pEnglish)
        else:
            print("学号不能空")

    def delete(self):
        pNo = input("学号:").strip()
        if pNo != "":
            self.__delete(pNo)
        else:
            print("学号不能空")

    def export(self):
        try:
            f=open("marks.txt","wt",encoding="utf-8")
            self.cursor.execute("select pNo,pName,pChinese,pMath,pEnglish from
marks")
            f.write("学号,姓名,语文,数学,英语\n")
            rows = self.cursor.fetchall()
            for row in rows:
                f.write(row[0]+","+row[1]+","+str(row[2])+","+str(row[3])+","+
str(row[4])+"\n")
            f.close()
            print("成绩导出完毕")
        except Exception as err:
            print(err)

    def __parseMark(self,s):
        m=0
```

```
        try:
            m=float(s)
            if m<0 or m>100:
                m=0
        except:
            pass
        return m

    def load(self):
        try:
            s=input("输入导入文件路径与名称:")
            if os.path.exists(s):
                f=open(s,"rt",encoding="utf-8")
                s=f.readline().strip("\n")
                st=s.split(",")
                if len(st)==5 and st[0]=="学号" and st[1]=="姓名" and st[2]=="
语文" and st[3]=="数学" and st[4]=="英语":
                    s="going"
                    while s!="":
                        s=f.readline().strip("\n")
                        if s!="":
                            st=s.split(",")
                            if len(st)==5:
                                pNo=st[0].strip()
                                pName=st[1].strip()
                                pChinese=self.__parseMark(st[2])
                                pMath = self.__parseMark(st[3])
                                pEnglish = self.__parseMark(st[3])
                                if pNo!="" and pName!="":
                                    self.__insert(pNo,pName,pChinese,pMath,pEn
glish)
                    print("成绩转载完毕")
                else:
                    print("文件格式不正确")
                f.close()
            else:
                print(s+"文件不存在")
        except Exception as err:
            print(err)

    def process(self):
        self.open()
        while True:
            s=input(">")
```

```
            if s=="show":
                self.show()
            elif s=="insert":
                self.insert()
            elif s=="update":
                self.update()
            elif s=="delete":
                self.delete()
            elif s=="export":
                self.export()
            elif s=="load":
                self.load()
            elif s=="exit":
                break
            else:
                                                print("Accept     commands:
show/insert/update/delete/export/load/exit")
                print("show   --- 显示所有记录")
                print("insert ---  增加一条记录")
                print("update --- 修改一条记录")
                print("delete ---  删除一条记录")
                print("export ---  导出到文件")
                print("load  ---  从文件导入")
                print("exit   ---  退出系统")
        self.close()

db=MarkDB()
db.process()
```

练习八

1. 套接字是什么？为什么在服务器端宇客户端建立了套接字就可以通信了？
2. 整数怎么样转为二进制数据？二进制数据又怎么样转为整数？
3. 字符串怎么样转为二进制数据？二进制数据又怎么样转为字符串？转化与编码有关吗？
4. 你还能不能设计出其它的字符串传输的格式完成字符串的网络传输？
5. 设计一个服务器程序 server.py，它能每隔 1 秒的时间通过 TCP 协议发送[-30,100]之间的一个整数，设计一个客户端程序 client.py 连接这个服务器，并获取与显示服务器发送的每个整数。

   答：

   (1) server.py

```
import socket
import struct
import random
import time

def readInt(socket):
    size=struct.calcsize("@i")
    data=socket.recv(size)
    n=struct.unpack("@i",data)[0]
    return n

def writeInt(socket,n):
    data=struct.pack("@i",n)
    socket.send(data)

try:
    s = socket.socket()
    host = "127.0.0.1"
    port = 2345
    s.bind((host, port))
    print(host,"is listening...")
    s.listen()
    c, addr = s.accept()
    m=0
    while m<20:
        n=random.randint(-30,100)
        writeInt(c,n)
        time.sleep(1)
        m+=1
    c.close()
    s.close()
except Exception as e:
```

```
    print(e)
```

(2) client.py

```
import socket
import struct

def readInt(socket):
    size=struct.calcsize("@i")
    data=socket.recv(size)
    n=struct.unpack("@i",data)[0]
    return n

def writeInt(socket,n):
    data=struct.pack("@i",n)
    socket.send(data)

try:
    s = socket.socket()
    host = "127.0.0.1"
    port = 2345
    s.connect((host, port))
    print("Connected to server")
    m=0
    try:
        while True:
            m+=1
            n=readInt(s)
            print("Turn",m,"Received",n)
    except Exception as e:
        print(e)
    s.close()
except Exception as e:
    print(e)
```

练习九

1. 设计一个汉字编码表查询程序，程序可以使用 seek 命令查找任何一个字符的 GBK,Unicode,UTF8 的编码，执行的命令格式是“seek 字符”，例如执行：

$seek 我

输出：

我 GBK: CED2 Unicode: 6211 UTF-8: E68891

答：

```
def byteToHex(d):
    s=""
    for b in d:
        t=hex(b)
        t=t[2:]
        if len(t)!=2:
            t="0"+t
        s=s+t
    return s.upper()

def decToHex(d):
    s=hex(d)
    return s[2:].upper()

while True:
    s=input("$")
    s=s.split(" ")
    if len(s)==2 and s[0]=="seek":
        t=s[1][0]
        gbk=byteToHex(t.encode("gbk"))
        unicode=decToHex(ord(t))
        utf8=byteToHex(t.encode("utf-8"))
        print("GBK:",gbk," Unicode:",unicode," UTF-8:",utf8)
    elif s[0]=="exit":
        break
```

2. 有一个人要上一个 6 级的楼梯，他一次可以跨 1、2、3 个台阶，求出他有多少种不同的上楼方法。编程计算后发现有下面 24 种不同的上楼方法，请编写递归程序验证。

1: [0, 1, 2, 3, 4, 5, 6]

2: [0, 1, 2, 3, 4, 6]

3: [0, 1, 2, 3, 5, 6]

4: [0, 1, 2, 3, 6]

5: [0, 1, 2, 4, 5, 6]
6: [0, 1, 2, 4, 6]
7: [0, 1, 2, 5, 6]
8: [0, 1, 3, 4, 5, 6]
9: [0, 1, 3, 4, 6]
10: [0, 1, 3, 5, 6]
11: [0, 1, 3, 6]
12: [0, 1, 4, 5, 6]
13: [0, 1, 4, 6]
14: [0, 2, 3, 4, 5, 6]
15: [0, 2, 3, 4, 6]
16: [0, 2, 3, 5, 6]
17: [0, 2, 3, 6]
18: [0, 2, 4, 5, 6]
19: [0, 2, 4, 6]
20: [0, 2, 5, 6]
21: [0, 3, 4, 5, 6]
22: [0, 3, 4, 6]
23: [0, 3, 5, 6]
24: [0, 3, 6]
Total number of paths: 24

例如方法 18: [0, 2, 4, 5, 6]，他从 0 台阶开始，先来到台阶 2、然后台阶 4、然后台阶 5、然后台阶 6。

答：

```
def goUpstairs(steps):
    global ways
    p=steps[-1]
    for s in [1,2,3]:
        if p+s<6:
            steps.append(p+s)
            goUpstairs(steps)
        elif p+s==6:
            steps.append(p+s)
            ways+=1
            print("%2d:"%ways,end=" ")
            print(steps)
            steps.pop()
    if steps:
        steps.pop()

ways=0
steps=[0]
goUpstairs(steps)
print("Total ways:",ways)
```

3. 线性回归（Linear Regression）是人工智能中最基础的方法之一。例如表所示是某车间生产的零件尺寸 x 与耗材 y 之间的关系，尺寸 x 越大耗材 y 越大。

x	1.	1.5	2.	2.5	3.	3.5	4.	4.5	5.	5.5	6.	6.5
y	5.6	5.5	6.7	8.3	9.	9.5	9.3	12.6	13.4	12.3	15.4	15.9

同学们可以画出 x 与 y 的散点图，猜想 y 与 x 基本上有一定的线性关系，即 y=ax+b，其中 a、b 是两个常数。数学上可以根据最小二乘法原理推导出 a 与 b 的值：

$$a = \frac{n\sum_{i=1}^{n} x_i y_i - (\sum_{i=1}^{n} x_i)(\sum_{i=1}^{n} y_i)}{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}$$

$$b = \frac{1}{n}\sum_{i=1}^{n} y_i - a\frac{1}{n}\sum_{i=1}^{n} x_i$$

其中 n 是样本数，这里 n=12，x_i 是各个 x 序列值，y_i 是各个 y 序列值。根据这个公式编写程序计算 a、b 的值，并画出 y=ax+b 的直线图。

答：

```
import matplotlib.pyplot as plt

def mul(x,y):
    s=0
    if len(x)==len(y):
        for i in range(len(x)):
            s=s+x[i]*y[i]
    return s

def sum(x):
    s=0
    for e in x:
        s+=e
    return s

x=[1.,1.5,2.,2.5,3.,3.5,4.,4.5,5.,5.5,6.,6.5]
y=[5.6,5.5,6.7,8.3,10.,9.5,10.3,12.6,13.4,12.3,15.4,15.9]
n=len(x)
xs=sum(x)
ys=sum(y)
xys=mul(x,y)
x2s=mul(x,x)
a=(n*xys-xs*ys)/(n*x2s-(xs*xs))
b=ys/n-a*xs/n
print("a=",a)
print("b=",b)
```

```
plt.scatter(x,y)
Y=[]
for i in range(n):
    Y.append(a*x[i]+b)
plt.plot(x,Y)
plt.show()
```

a= 1.9279720279720245

b= 3.2284382284382422

