第一章

一．1.C

二．1、3.14159*49

2、3.14159*49*9

三．

1、

```
import turtle as t
import math
t.pensize(2)
t.right(90)
t.penup()
t.forward(200)
t.pendown()
t.left(90)
r=200
t.circle(r)
len=r*math.sqrt(3)
t.left(60)
t.forward(len)
t.left(120)
t.forward(len)
```

```
t.left(120)
t.forward(len)
```

2.

```
import turtle as t
t.color('red')
t.speed("fast")
for i in range(20):
    t.shape('turtle') #基础形状
    t.circle(100,360)
    t.left(18)
```

第二章

一、

1.D 2.C 3.C 4.B 5.A 6.C 7.A 8.B 9.A 10.D 11.A

12.B 13.B 14.C 15.B 16.A 17.C 18.D 19.B 20.C

二 .

1.

```
import math
a=math.sqrt(math.pi)
```

```
a=pow(a,1/2)

b=math.pow(13+math.e,1/2)

b=math.log(2*math.pi*b,10)

c=math.pi+1

c=math.log(c,3)

c=math.pow(math.tan(c),-1)

print(a,b,c)
```

2. (1)

```
r=2.11

v=3*(math.pi*r**2)/4

print(v)
```

(2)

```
r1=16.2

r2=9.4

s=math.pi*(r1**2-r2**2)

print(s)
```

(3)

```
r3=66

h=24.2
```

```
v1=math.pi*(r3**2)*h

s1=2*math.pi*(r3**2)+2*math.pi*h

print( “底面积：”,s1)

print( “体积：”,v1)
```

4.（1）

```
a=input()

num=0

for i in a:

    if i=='t':

        num+=1

print(num)
```

（2）

```
a=input()

num=0

for i in a:

    if i=='t':

        num+=1

#print(num)

list=[]

b=0
```

```
list=a.split('://')
list1=list[1].split('.')
del list[1]
list2=list+list1
for i in list2:
    if i=='com':
        continue
    b=b+1
print(b-1)
```

（3）

```
a=input()
str=a.replace('.','-')
print(str)
```

（4）

```
str=input()
str1=str[7:13]
print(str1)
str2=str[-19:-13]
```

```
print(str2)
```

（5）

```
str=input()
print("转换后的值为：",str.upper())
```

（6）

```
str=input()
n=0
for i in str:
    n+=1
print("总字符数为：",n)
```

（7）

```
str=input()
print("连接后为：",str+"index")
```

5.

```
a=float(input("数学:"))
```

```
b=float(input("语文:"))
c=float(input("英语:"))
sum1=a+b+c
sum2=a*0.3+b*0.5+c*0.2
abv=sum1/3
ma=max(a,b,c)
mi=min(a,b,c)
print(f"最大值是：{ma},最小值是{mi},总和为：{sum1},平均分为：{abv},测评综合为：
{sum2}")
```

6.

```
a=int(input())
b=a%10
c=(a-b)/10%10
d=(a-10*c-b)/100
print(b+c+d)
```

-

第三章

一、

1.D 2.C 3.B 4.B 5.B 6.B 7.A 8.A 9.C 10.A 11.B 12.D 13.D 14.C

15.B 16.D 17.D 18.A 19.D 20.A 21.B 22.A 23.B

二、

1.

```
for i in range(10):
    for j in range(9):
        if i<=j:
            print(f"{i+1}*{j+1}=",(i+1)*(j+1),end=" ")
    print()
```

2.

```
interviewees = [
    {"age":24, "exp":0, "major":"计算机"},
    {"age":32, "exp":4, "major":"电子"},
    {"age":36, "exp":8, "major":"电子"},
    {"age":26, "exp":2, "major":"通信"}
]

for idx, person in enumerate(interviewees, 1):
```

```
    age = person["age"]

    exp = person["exp"]

    major = person["major"]

    meet = (major == "计算机" and age < 25) or (major == "电子" and exp >= 4)

    print(f"序号{idx}面试者：{'获得面试机会' if meet else '抱歉，您不符合面试要求'}")
```

3.

```
scor=int(input("请输入成绩："))
if scor>=100&scor<=1:
    print("erro")
if scor>=90:
    grade="A"
    print(grade)
elif 89>=scor>=80:
    grade="B"
    print(grade)
elif 79>=scor>=70:
    grade="C"
    print(grade)
elif 69 >= scor >= 60:
     grade = "C"
     print(grade)
```

```
elif scor<=59:
        grade="E"
        print(grade)
```

4.

```
for i in range(2000,3001):
    year=i
    if year%4==0 and year%100!=0 or year%400==0:
        print("闰年")
    else:
        print("不是闰年")
```

5.

```
import math
for angle in range(0, 91, 5):   # 0 到 90，步长 5
    rad = math.radians(angle)   # 角度转弧度
    sin_val = math.sin(rad)
    cos_val = math.cos(rad)
```

```
    print(f"角度{angle}°：正弦={sin_val:.2f}，余弦={cos_val:.2f}")
```

6.

```
import math

n = int(input("请输入 n："))

for i in range(1, n+1):

    x = i   # 初始值

    for _ in range(10):   # 迭代 10 次足够精确

        x = (x + i / x) / 2

    sqrt_math = math.sqrt(i)

    print(f"整数{i}：牛顿迭代结果={x:.2f}，math.sqrt 结果={sqrt_math:.2f}")
```

7.

```
n=int(input())

a=0

for i in range(n):

    if n%7==0&a<i:

        a=i+1

print(a)
```

8.

```
import math
a=1
s=0
for i in range(1000):
    s+=math.pow(-1,i)*1/(i*2-1)
print(s)
```

第四章

一、

1~15：DCADADBADABBBAC

16~20：BBCDA

二、

1．

```
ls = [54, 36, 75, 28, 50]
ls.append(42)
idx = ls.index(28)
ls.insert(idx, 66)
val = ls.pop(ls.index(66))
print(val)
```

```
ls.sort(reverse=True)

ls.clear()
```

2.

```
list=[]
for i in range(100):
    if i%3==0:
        list.append(i)
print(list)
```

3.

(1)

```
years = [2006, 2007, 2008, 2009, 2010, 2011]
rates = [57, 56, 57, 62, 69, 72]
avg_rate = sum(rates) / len(rates)
print(f"平均取率: {avg_rate:.1f}%")
```

(2)

```
years = [2006, 2007, 2008, 2009, 2010, 2011]
rates = [57, 56, 57, 62, 69, 72]
```

```
max_rate = max(rates)

idx = rates.index(max_rate)

print(f"取率最高的年份: {years[idx]}年，取率: {max_rate}%")
```

4.

```
sentence = input("请输入英文句子: ")

words = sentence.split(“.”)

longest_length = max(len(word) for word in words)

print(f"最长单词的长度: {longest_length}")
```

5.

```
import random

list=[20]

list1=[]

for i in range(20):

    i+=1

    a=random.randint(1000,5000)

    list.append(a)

for i in list:
```

```
    if i%10!=0:
        list1.append(i)
print(list1)
```

6.

```
data = [
    ["王平",    "男", 1, 1, 0, ],
    ["李丽",    "女", 0, 1, 0, 1],
    ["陈小梅", "女", 0, 0, 1, 0],
    ["孙洪涛", "男", 0, 1, 1, 1],
    ["方亮",    "男", 1, 0, 1, 0]
]
projects = ["100m", "3000m", "跳远", "跳高"]
```

(1)

```
count_more_than_two = 0
for student in data:
    project_count = sum(student[2:])
    if project_count >= 2:
```

```
            count_more_than_two += 1

print(f"(1) 报名项目超过两项的学生人数: {count_more_than_two}")
```

(2)

```
print("\n(2) 学生报名情况:")

for student in data:

    name = student[0]

    signed_up = [projects[i] for i in range(4) if student[2+i] == 1]

    print(f"{name}: {', '.join(signed_up)}")
```

(3)

```
print("\n(3) 报名 3000m 的学生:")

for student in data:

    if student[3] == 1:   # 3000m 是第 4 列（索引 3）

        print(f"姓名: {student[0]}, 性别: {student[1]}")
```

第五章

1-10:ADDBCCCCCA

11-21:ACCDABABCDB

二、

```
dicTXL = { "小新": {"手机": "13915000001", "QQ": "18191220001"},"小亮": {"手机": "13915000002", "QQ": "19181220002"},"小刚": {"手机": "13915000003", "QQ": "18191220003"}}

dicOther = { "大刘": {"手机": "13914000001", "QQ": "1819120001"}, "大王": {"手机": "13914000002", "QQ": "1819120002"}, "大张": {"手机": "13914000003", "QQ": "1819120003"}}

dicWX = {"小新": "aaaa11","小刚": "bbbb12","大王": "cccc13","大刘": "dddd14"}

dicTXL.update(dicOther)

for name in dicTXL:
    if name in dicWX.keys():
        dicTXL[name]["微信号"] = dicWX[name]
    else:
        dicTXL[name]["微信号"] = dicTXL[name]["手机"]
```

(1)

```
dicTXL["大王"]["手机"] = "13914000004"
```

(2)

```
def search_contact(name):
```

```
    if name in dicTXL:
        return dicTXL[name]
    else:
        return "没有该同学的联系方式"
print(search_contact("小新"))
print(search_contact("张三"))
```

2.

```
data = [
    {"性别": "女", "书本": 10, "文具": 30, "服饰": 300, "零食": 150, "日用品": 600},
    {"性别": "女", "书本": 200, "文具": 10, "服饰": 300, "零食": 300, "日用品": 100},
    {"性别": "男", "书本": 200, "文具": 100, "服饰": 1000, "零食": 100, "日用品": 200},
    {"性别": "男", "书本": 50, "文具": 20, "服饰": 300, "零食": 100, "日用品": 200},
    {"性别": "男", "书本": 200, "文具": 50, "服饰": 400, "零食": 100, "日用品": 200},
    {"性别": "女", "书本": 100, "文具": 10, "服饰": 500, "零食": 150, "日用品": 800},
    {"性别": "女", "书本": 200, "文具": 100, "服饰": 500, "零食": 300, "日用品": 200},
    {"性别": "女", "书本": 300, "文具": 50, "服饰": 0, "零食": 10, "日用品": 50},
    {"性别": "男", "书本": 100, "文具": 10, "服饰": 500, "零食": 40, "日用品": 500},
    {"性别": "男", "书本": 200, "文具": 500, "服饰": 200, "零食": 100, "日用品": 100}
]
```

（1）

```
sum1=0
sum2=0
sum3=0
sum4=0
sum5=0
for i in range(10):
    sum1+=data[i]["书本"]
    sum2+= data[i]["文具"]
    sum3+= data[i]["服饰"]
    sum4+= data[i]["零食"]
    sum5+= data[i]["日用品"]
print("书本：",sum1,"文具",sum2,"服饰",sum3,"零食",sum4,"日用品",sum5)
```

（2）

```
a1=0
a2=0
av1=0
av2=0
boy=0
girl=0
for i in range(10):
```

```
    if data[i]["性别"]=="男":
        boy+=1
        a1+=data[i]["文具"]+data[i]["书本"]+data[i]["服饰"]+data[i]["零食"]+data[i]["日
用品"]
    if data[i]["性别"]== "女":
        girl+=1
        a2+= data[i]["文具"] + data[i]["书本"] + data[i]["服饰"] + data[i]["零食"] +
data[i]["日用品"]
av1=a1/10
av2=a2/10
print(av1,av2)
```

第六章

一、

DDCACABAABAABCCDDCBACCBDCADDCBB

二、

1.

```
import math
def area(r):
    s=math.pi*(r**2)
```

```
    return s
print("半径为 3.5 的圆面积为：",area(3.5))
print("半径为 2.9 的圆面积为：",area(2.9))
print("外圆半径为 6.2，内圆为 3.3 的圆环面积为：",area(6.2)-area(3.3))
```

2.

```
def showMsg(n, name):
    for _ in range(n):
        print(f"Happy Birthday {name}")
showMsg(3, "小明")
```

3.

```
def avg(a,b,c):
    return int((a+b+c)/3)
s={'小李': [77, 54, 57],'小张': [89, 66, 78],'小陈': [90, 93, 80],'小杨': [69, 58, 93]}
s1={}
for k,v in s.items():
    s1[k]=avg(v[0],v[1],v[2])
print(s1)
```

4.

```
def avg(lst):
    return int(sum(lst) / len(lst))
s = { '小李': [77, 54], '小张': [89, 66, 78, 99],'小陈': [90],'小杨': [69, 58, 93]}
result = {}
for name, scores in s.items():
    result[name] = avg(scores)
print(result)
```

5.

```
dict_score = {'01': [67, 88, 45],'02': [97, 68, 85],'03': [97, 98, 95],'04': [67, 48, 45],'05':
[82, 58, 75],'06': [96, 49, 65]}
for k, v in dict_score.items():
    if v[0]>=85:
        if v[1]>=85:
            if  v[2]>=85:
                print(k)
```

6.

```
def get_sequence(n):
    if n == 1 or n == 2 or n == 3:
        return 1
    return get_sequence(n-1) + get_sequence(n-2) + get_sequence(n-3)
sequence = [get_sequence(i) for i in range(1, 21)]
print(sequence)
```

第七章

一、

1-14 AABDDDCCABCBCD

二、

1.

```
import csv
dicTXL = { "小新": {"手机": "13915000001", "QQ": "18191220001"},"小亮": {"手机":
"13915000002", "QQ": "19181220002"},"小刚": {"手机": "13915000003", "QQ":
"18191220003"}}
dicOther = { "大刘": {"手机": "13914000001", "QQ": "1819120001"}, "大王": {"手机":
"13914000002", "QQ": "1819120002"}, "大张": {"手机": "13914000003", "QQ":
"1819120003"}}
```

```
dicWX = {"小新": "aaaa11","小刚": "bbbb12","大王": "cccc13","大刘": "dddd14"}
dicTXL.update(dicOther)
for name in dicTXL:
    if name in dicWX.keys():
        dicTXL[name]["微信"] = dicWX[name]
    else:
        dicTXL[name]["微信"] = dicTXL[name]["手机"]
dicTXL["大王"]["手机"] = "13914000004"
with open("通讯录.csv", "w", newline="", encoding="utf-8") as csvfile:
    fieldnames = ["姓名", "手机", "QQ", "微信"]
    writer = csv.DictWriter(csvfile, fieldnames=fieldnames)
    writer.writeheader()
    for name, info in dicTXL.items():
        row = {"姓名": name, "手机": info["手机"], "QQ": info["QQ"], "微信": info["微信
"]}
        writer.writerow(row)
with open("通讯录.csv", "r", encoding="utf-8") as csvfile:
    reader = csv.DictReader(csvfile)
    found = False
    for row in reader:
        if row["姓名"] == "大王":
            print(f"大王的手机号：{row['手机']}")
```

```
                print(f"大王的 QQ 号：{row['QQ']}")
                print(f"大王的微信号：{row['微信']}")
                found = True
                break
    if not found:
        print("未找到大王的联系方式")
```

2.

```
comments = ["不满意", "一般", "满意", "很满意"]
result = "不满意,一般,很满意,一般,不满意,很满意,一般,一般,不满意,满意,满意,满意,满意,满意,一般,很满意,一般,满意,不满意,满意,一般,不满意,满意,很满意,不满意,满意,满意,很满意,一般,很满意,满意,很满意,不满意,满意,一般,很满意,不满,一般,很满意,满意,很满意,不满意,很满意,不满意,不满意,满意,很满意,很满意,一般,不满意,满意,满意,很满意,不满意,很满意,满意,很满意,满意,很满意,很满意,不满意,满意,满意,一般,很满意,满意,很满意,很满意,不满意,不满意,一般,很满意,满意,满意,一般,很满意,不满意,不满意,很满意,很满意,很满意,满意,很满意,满意,很满意,很满意,不满意,满意,满意"
# 拆分字符串为评语列表
comment_list = result.split(",")
counts = []
for c in comments:
    counts.append(comment_list.count(c))
```

```
max_count = max(counts)
max_comment = comments[counts.index(max_count)]
for i in range(len(comments)):
    print(f"{comments[i]}: {counts[i]}次")
print(f"出现次数最多的评语是{max_comment}，出现了{max_count}次")
f=open("C:/Users/65791/PycharmProjects/PythonProject2/.venv/xzh/result.txt","w",en
coding="utf-8")
f.write(result)
f=open("C:/Users/65791/PycharmProjects/PythonProject2/.venv/xzh/result.txt","r",enc
oding="utf-8")
m=0
bm=0
yb=0
hm=0
for i in f.readlines():
    for h in i.split(","):
        print(h)
        if h=="满意":
            m+=1
        if h=="不满意":
            bm+=1
        if h=="一般":
```

```
            yb+=1
        if h=="很满意":
            hm+=1
print(f"一般：{yb},满意：{m},不满意：{bm},很满意：{hm}")
```

第九章

二、

1.

四边形

```
import turtle as t
t.pensize(4)
t.color("black", "#8F653B")
t.begin_fill()
d=0
for i in range(4):
    t.forward(100)
    d+=90
    t.seth(d)
t.end_fill()
```

```
turtle.exitonclick()
```

五边形

```
import turtle as t
t.penup()
t.goto(0, -100)
t.pensize(5)
t.color("black","red")
t.pendown()
t.begin_fill()
t.right(36)
t.forward(100)
for i in range(4):
    t.right(72)
    t.forward(100)
t.end_fill()
t.exitonclick()
```

六边形

```
import turtle as t
t.penup()
t.goto(0, -100)
```

```
t.pensize(5)
t.color("black","red")
t.pendown()
t.begin_fill()
for i in range(6):
    t.forward(100)
    t.right(60)
t.end_fill()
t.exitonclick()
```

2.

```
import turtle
t = turtle.Turtle()
t.penup()
t.goto(0, -100)
t.color("black","red")
t.pendown()
t.begin_fill()
for i in range(5):
    t.forward(200)
```

```
    t.right(144)
t.end_fill()
turtle.exitonclick()
```

实训一

二、

1.运算器，控制器，存储器，输入设备，输出设备，存储程序与程序控制

2.机器语言，汇编语言，高级语言

3.交互式运行，文件式运行

三．

6.与上一个代码的区别在于 5 题是逆时针方向画图，而 6 题是以顺时针方向画图

实践二

二、

1.

字母或下划线；大小写；英文；英文

2.关键字

3.赋值语句；赋值语句

4.动态类型

5.链式；解包；解包

6.input（ ）

7.print（ ）

8.整数；浮点数；复数

9.*；/；整除；取余；幂运算；abs（ ）；round（ ）；max（ ）；min（ ）

10.复数；import math；sum（ ）；gcd（ ）；math.trunc() ；ceil（ ）；floor（ ）；math.pi

；math.e

11.10;I am student;1；1，2.5，6；int，float ，float

三、

1.

7，7；3，3；10，10；2.5，2.5；2，2；1，1；25，25；True，True；False，False；

2，2；5，5；False，False

2.

```
score1=int(input())
score2=int(input())
```

```
score3=int(input())
sum=score1+score2+score3
avg=sum/3
print(sum,avg)
```

3.

```
len1=23352
len2=23690
a1=24
len=len2-len1
s=a1/len*100
print(s)
```

4.

```
x=int(input("请输入三位数："))
a=x%10
b=(x-a)/10%10
c=(x-a-10*b)/100
m=a*100+b*10+c
print(m)
```

5.

```
import math
a=int(input("请输入三角形的第一条边长："))
b=int(input("请输入三角形的第二条边长："))
c=int(input("请输入三角形的第三条边长："))
l=(a+b+c)/2
s=math.sqrt(l*(l-a)*(l-b)*(l-c))
print(s)
```

6.

```
import math
r=int(input("请输入球的半径："))
s=4*math.pi*math.pow(r,2)
v=(3*math.pow(r,3)*math.pi)/4
print(s)
print(v)
```

7.

```
h=1
for i in range(365):
```

```
        h=h*(1+0.01)
print("每天都练功，一年后的武力值：",h)
m=1
for i in range(365):
        m=m*(1-0.01)
print("每天都不练功，一年后的武力值：",m)
```

8.

```
ts=int(input("跳绳成绩："))
run=int(input("跑步成绩："))
ty=int(input("跳远成绩："))
print("总分：",0.5*run+0.3*ts+0.2*ty)
```

实践三

二、

1.单引号，双引号，三引

2.不可以，0，递增，-1，递减

3.取全部

4.链接，重复，查找

5.len()，ord()

6.lower(),split()

7.我是张无忌，左手拿倚天剑，右手拿屠龙刀

3.1416

8.转为整数，转为浮点数，转为字符串

9.ab,cdef,gh

ab,cdef,gh,ab,cdef,gh

ab,cd

AB,CD

True

ab,cdef,gh,ab,cdef,gh 中出现字母 a 的位置 0

ab,cdef,gh,ab,cdef,gh 中出现字母 a 的位置 0

-1

['ab', 'cdef', 'ghab', 'cdef', 'gh']

三、

1.我喜欢 python

我喜欢我喜欢

PythonPythonPython

True

我

我喜

我喜欢

ho

欢喜我

Pto

yhn

True

3

6

2.

str="www.moe.gov.cn"

print("第一个字符是：",str[0])

print("前三个字符是：",str[:3])

print("输出后三个字符：",str[-4:-1])

```
print("字符总长度是：",len(str))

print("字符 o 所在位置是：",str.index('o'))

print("o 出现的次数：",str.count('o'))

print("替换后：",str.replace('.','-'))

print("转为大写字母：",str.upper())

print("拆分为四个字符：",str.split('.'))
```

3.

```
a=input("第一个室友的名字：")

b=input("第二个室友的名字：")

c=input("第三个室友的名字：")

d=input("第四个室友的名字：")

print("宿舍名字是：",a[-1]+b[-1]+c[-1]+d[-1])
```

4.

```
months = "JanFebMarAprMayJunJulAugSepOctNovDec"

month_num = int(input("请输入数字月份  1-12："))

start = (month_num - 1) * 3

month_abbr = months[start:start + 3]

print(f"{month_num}月份对应的英文缩写是：{month_abbr}")
```

5.

```
m=float(input("请输入金额："))
dm=m*0.1456
print("换为美元后：","%.2f"%dm)
```

6.

```
m=float(input("请输入金额："))
dm=m*6.868
print("换为人民币后：","%.2f"%dm)
```

实践四

二、

1.分支结构，单选择结构，双选择结构，多选择结构

2.if

3.if；else

4.if；elif；else

5.嵌套 if

6.<,<=,>,>=,==,!=;True,False;比较;and,or,not

7.(1)A、B，D，A、C、D，B

(2) a = eval(input("请输入 a 的值："))

b = eval(input("请输入 b 的值："))

```
if a > b:
    print("{} 比较大".format(a))
else:
    print("{} 比较大".format(b))
```

（3）方法 3

三、

1.

```
n=int(input())
if n//3==0 or n//7==0:
    print("yes")
else:
    print("no")
```

2.

```
a = int(input("请输入第一个整数"))
b = int(input("请输入第二个整数"))
if a > b:
    a, b = b, a   # 交换两个变量的值
print(a, b)
```

3.

```
if 0<=ch<10:
elif ch.isalpha
```

4.

```
elif 3 < km <= 10:
cost = 10
cost = 10 + 7 * 1.2 + (km - 10) * 1.5
print("您需要支付{:.1f}元车费".format(cost))
print(cost)
```

5.

```
year=int(input("请输入年份："))
if year%400==0 or year%4==0 and year%100!=0:
    print("该年为闰年")
else :
    print("NO")
```

6.

```
x = float(input("请输入自变量 x："))
if x < 0:
    y = 0
elif 0 < x < 5:
    y = x
elif 5 <= x < 10:
    y = 3 * x - 5
elif 10 <= x < 20:
    y = 0.5 * x - 2
```

```
else:
    y = 0
print(f"因变量 y 的值为：{y}")
```

7.

```
a = float(input("请输入第一条边长："))
b = float(input("请输入第二条边长："))
c = float(input("请输入第三条边长："))
if a + b > c and a + c > b and b + c > a:
    perimeter = a + b + c
    s = perimeter / 2
    area = (s * (s - a) * (s - b) * (s - c)) ** 0.5
    print(f"三角形的周长为：{perimeter:.1f}")
    print(f"三角形的面积为：{area:.1f}")
else:
    print("三边边长有问题，无法构成三角形")
```

8.

```
score = float(input("请输入百分制分数: "))
if score >= 90:
    grade = "优"
elif 80 <= score < 90:
    grade = "良"
elif 70 <= score < 80:
    grade = "中"
elif 60 <= score < 70:
    grade = "及格"
else:
    grade = "不及格"
print(f"该分数对应的等级是: {grade}")
```

实践五

二、

1.选择结构；循环结构；for 循环，while 循环

2.for in :

3.while　:

4.for 用于已知迭代，while 用于未知迭代

5.0，1，2，3，，，n-

0，1，2，3，，，99

11，12，13，14，15

11，13，15

15，12，9，6

6.break，continue

7.进行，不进入

8.随机数，提供随机数

三、

1.·dcba

不可以，正向链接

2.16,4 次，1，7

3.死循环，CTRL+c

4.

```
sum=1
for i in range(1,11):
    sum=sum*i
print(sum)
```

5.

```
import random
T=int(random.randint(0,9))
m=False
N=0
while m!=True:
    N+=1
    t=int(input("请输入一个数字："))
    if t==T:
        print("yes,猜测了：",N)
        m=True
    elif t<T:
        print("猜小了")
    else:
        print("猜大了")
```

6.

```
import random
T=random.randint(0,100)
t=random.randint(0,100)
x=math.gcd(T,t)
y=T*t/x
print("最大公约数：",x,"最小公倍数：",y)
```

7.

```
import random
T=random.randint(100,1000)
a=0
b=0
for i in range(T):
    x=random.randint(0,1)
    if x==1:
        a+=1
    else:
        b+=1
if  a==b:
    print(“次数相同”)
else  :
```

```
        print(“此数不相同”)
print("为 0 的次数：",a,"为 1 的次数：",b)
```

8.

```
x=1
for i in range(1,366):
        if i%5<=4:
            x=x*1.1
        else:
            x=x*0.1
print("%.2f"%x)
```

9.

```
  k=63
for i in range(1,100):
        m=k*i
        if m%2==1 and m%3==0 and m%4==1 and m%5==1 and m%6==3    and
m%8==1 and m%9==0:
            print(m)
            break
```

10.

count_upper += 1

count_lower += 1

count_digit += 1

11.

for i in range(1,100)

实践六

二、

1.中括号，逗号，元素个数，[]

2.有序，位置编号，0，递加，列表名【位置编号】

3.正向索引，逆向索引，正数，负数，a【4】=5，a[-1]=5

4.可变的，列表名【位置编号】,append();insert();del;remove()

5.True;False;False;True

6.元素第一次出现位置编号，False；1

7.3，0

8.

```
for i in range(n):
    print(a[i])
for i in a:
    print(i)
```

9.sort();从小到大；正序，逆序；

10.sorted（ ）；不改变；【4，2，1，3，5】，【1，2，3，4，5】

11.切片；列表名【起始编号：终止编号：步长】

12.【a’,‘b’,‘c’,‘a’,‘b’,‘c’,‘a’,‘b’,‘c’】；【a’,‘b’,‘c’,‘d’,‘e’】，不改变；extend（ ）；mylist.extend（’d’,’e’）

13.浅拷贝；【1，2，3】；【1，4，3】，【1，2，3】，【7，5，6】；【1，2，3】

14.sum（ ），max（ ），min（ ），15；5；1

15.逗号，圆括号，不可以改变，可以改变

三、

1.

>>>myList[3]：列表索引从 0 开始，第 3 个索引（从 0 数）对应的元素是'4'。

>>>myList[-3]：负数索引从后往前数，-3 对应的元素是'7'。

>>>myList[2:5]：切片[2:5]表示从索引 2 到索引 4（不包含 5）的元素，即['3', '4', '5']。

>>>myList[2:8:2] :切片[2:8:2]表示从索引 2 到索引 7 ,步长为 2 的元素 ,即['3', '5', '7']。

>>>myList[1:]：从索引 1 到末尾的元素，即['2', '3', '4', '5', '6', '7', '8', '9']。

>>>myList[:3]：从开头到索引 2（不包含 3）的元素，即['1', '2', '3']。

>>>myList[1:-2] :从索引 1 到倒数第 3 个元素(不包含倒数第 2 个),即['2', '3', '4', '5', '6', '7']。

>>>myList[-4]：从后往前数第 4 个元素，即'6'。

>>>myList[-4:2]：从倒数第 4 个元素到索引 1（不包含 2），没有元素，结果为[]。

>>>myList[3] ,结果为'4' ;>>>myList[1:-2] ,结果为['2', '3', '4', '5', '6', '7'] ;>>>myList[-4]，结果为'6'；>>>myList[-4:2]，结果为[]。

>>>myList[1:-3:3]：从索引 1 到倒数第 4 个元素，步长为 3 的元素，即['2', '5']。

>>>myList[-3:3]：从倒数第 3 个元素到索引 2（不包含 3），没有元素，结果为[]。

>>>myList[0:-2] :从索引 0 到倒数第 3 个元素(不包含倒数第 2 个),即['1', '2', '3', '4', '5', '6', '7']。

>>>myList[1:-2]：结果为['2', '3', '4', '5', '6', '7']。

>>>myList[:-1]：从开头到倒数第 2 个元素（不包含最后一个），即['1', '2', '3', '4', '5', '6', '7', '8']

(1)索引是“点”操作：精准定位单个元素，返回单个值。

切片是“范围”操作：获取连续的多个元素，返回一个新的子序列。

(2)mylist[::2]

(3)Mylist[-1::2]

(4)9

2.

>>>myList = ['a', 'b', 'c', 'd', 'e']

>>>myList[0] = "hello" → myList 变为 ['hello', 'b', 'c', 'd', 'e']

>>>myList[-2] = False → myList 变为 ['hello', 'b', 'c', False, 'e']

>>>myList[2:4] = [1,2] → 替换索引 2 到 3 的元素，myList 变为 ['hello', 'b', 1, 2, 'e']

>>>myList[1:] = "world" → 将索引 1 及以后的元素替换为字符串"world"的每个字符，myList 变为 ['hello', 'w', 'o', 'r', 'l', 'd']

>>>myList[1:] = ["world"] → 将索引 1 及以后的元素替换为列表["world"]，myList 变为 ['hello', 'world']

1）

myList[1:] = "world" 中，字符串"world"会被自动拆分为单个字符的列表（即['w','o','r','l','d']），因此索引 1 及以后的元素被替换为这些字符；而 myList[1:] = ["world"] 中，右侧是列表["world"]，因此索引 1 及以后的元素被替换为该列表的单个元素"world"。

（2）

会报错。因为列表切片赋值时，右侧必须是可迭代对象（如列表、字符串等），而 32 是单个整数（不可迭代）。需修改为 myList[2] = 32；修改后 myList 的值为 [1,2,32,4,5]。

（3）

myList[1:2] = [6,7,8]

myList[0:3] = [6,7,8]。

myList[0:3] = [{6,7,8}]

3.

(1) [1,2,3,4,5]。[2,3,4,5,6]。一个是通过长度索引，一个是直接遍历，起始编号不一样

（2）5，求出列表长度，有，len（）

（3）3，计算元素出现的次数，有，count（）

4.

["张英","女","计算机专业", [1992,8,31]]

["张英","女","计算机专业", [1992,8,31], 90, 86, 92]

["1001","张英","女","计算机专业", [1992,8,31], 90, 86, 92]

["1001","张英","女","计算机专业", [1992,8,31], 90, 86, 92, 95, 88]

append()：将单个对象（如列表、元素）添加到列表末尾，例如 append([1,2]) 会把[1,2]作为一个元素添加。

extend()：将可迭代对象（如列表、元组）的每个元素拆分后添加到列表末尾，例如 extend([1,2]) 会添加 1 和 2 两个元素。

insert()：在指定索引位置插入单个对象，语法为 insert(索引, 对象)。

运算符 +：拼接两个列表，返回新列表，不修改原列表；而前三者是直接修改原列表。

5.myStr.sort()

前一个是根据 ASCLL 编码进行排序，后一个是直接逆序

6.

[["香蕉","黄色"],["草莓","红色"],["葡萄","紫色"]]

[["香蕉","黄色"],["草莓","红色"],["葡萄","紫色"]]

[["香蕉","黄色"],["草莓","红色"],["葡萄","紫色"]]

[["西瓜","红色"],["草莓","红色"],["葡萄","紫色"]]；

[["香蕉","黄色"],["草莓","红色"],["葡萄","紫色"]]；

[["香蕉","黄色"],["草莓","红色"],["葡萄","紫色"]]

[["西瓜","红色"],["草莓","红色"],["葡萄","绿色"]]；

[["香蕉","黄色"],["草莓","红色"],["葡萄","绿色"]]；

[["香蕉","黄色"],["草莓","红色"],["葡萄","绿色"]]

(1) newList = myList：newList 是 myList 的引用，修改 myList 的顶层元素（如 myList[0])会同步影响 newList ;但修改嵌套列表的内部元素(如 myList[2][1])时，newList 也会受影响（因引用的是同一嵌套列表）。copyList = myList.copy()：copyList 是 myList 的浅拷贝，仅复制顶层列表，嵌套列表仍为引用。因此修改 myList 的顶层元素不影响 copyList，但修改嵌套列表的内部元素会影响 copyList。sliceList = myList[:]：sliceList 是通过切片实现的浅拷贝，与 copy() 效果一致，仅复制顶层列表，嵌套列表为引用。

(2) [["香蕉","黄色"],["草莓","红色"],["葡萄","红色"]]原因：myList 中的元素是元组（不可变类型），但题目中实际应为列表嵌套（如之前的 myList 定义），若为列表嵌套，则 myList[2][1] 是嵌套列表的元素，修改后所有引用该嵌套列表的变量（newList、copyList、sliceList）都会受影响

7.

```
import random

lst_who = ["小马", "小羊", "小鹿"]

lst_where = ["草地上", "电影院", "家里"]

lst_what = ["看电影", "听故事", "吃晚饭"]

index_who = random.randint(0, 2)

index_where = random.randint(0, 2)

index_what = random.randint(0, 2)

sentence = f"{lst_who[index_who]}在{lst_where[index_where]}{lst_what[index_what]}"

print(sentence)
```

8.

```
month_days = [31,28,31,30,31,30,31,31,30,31,30,31]
while True:
    month = int(input("请输入月份（0 结束）: "))
    if month == 0:
        break
    elif 1 <= month <= 12:
        print(f"{month}月有{month_days[month-1]}天")
```

```
    else:
        print("输入月份无效，请重新输入")
```

9.

```
fib = [1,1]
for i in range(2,30):
    lst_fib.append(fib[i-1] +fib[i-2])
print("斐波那契数列前 30 项：", fib)
```

10.

```
student = [["001","李梅",19],["002","刘祥",20],["003","张武",18]]
student.append(["004","刘宁",20])
student.append(["006","梁峰",19])
print(student)
student.append(["005","林歌",20])
student.sort(key=lambda x:x[1])
print(student)
print(student[2])
for i in range(5):
    print(student[i][1])
```

```
age=0
for i in range(5):
    age+=student[i][2]
print("平均年纪：",age/5)
```

11.

```
list = ["甲", "乙", "丙", "丁"]
for x in list:
    count = (x != "甲") + (x == "丙") + (x != "丁")
    if count == 3:
        print(f"做好事的人是：{x}")
        break
```

12.

```
import random
ls = ["A", "B", "C", "D", "E"]
found = False
```

```
while not found:
    random.shuffle(ls)  # 随机打乱列表
    cond1 = (ls[1] == "D") + (ls[2] == "B") == 1
    cond2 = (ls[1] == "C") + (ls[3] == "E") == 1
    cond3 = (ls[0] == "E") + (ls[4] == "A") == 1
    cond4 = (ls[2] == "C") + (ls[3] == "A") == 1
    cond5 = (ls[1] == "B") + (ls[4] == "D") == 1
    if cond1 and cond2 and cond3 and cond4 and cond5:
        print("真实名次（第 1 到第 5 名）: ", ls)
        found = True
```

13.

```
lst = [["李玉","男",25],["金忠","男",23],["刘妍","女",21],["莫心","女",24],["沈冲","男",28]]
name=str(input("请输入名字："))
while name!='0':
    for i in range(5):
        if lst[i][0]==name:
            del lst[i]
        else :
```

```
                print("不存在")
                brak
```

14.

```
lst_odd = [1,3,5,7,9]
lst_even = [2,4,6,8,10]
new_lst = lst_odd + lst_even
new_lst.sort(reverse=True)
print("合并并降序排序后的列表：", new_lst)
```

15.

```
s = input("请输入英文字符串：").lower()
letters = []
for char in s:
    if char.isalpha() and char not in letters:
        letters.append(char)
letters.sort()
print("".join(letters))
```

16.

```
import random

myList = [random.randint(10,99) for _ in range(20)]

myList[0:10] = sorted(myList[0:10])

print("处理后的列表：", myList)
```

17.

```
s = input("请输入英文语句：")

words = s.split()

reversed_words = words[::-1]

print(" ".join(reversed_words))
```

18.

```
scores = []

for i in range(20):

    score = float(input(f"请输入第{i+1}位学生的成绩："))

    scores.append(score)
```

max_score = max(scores)

min_score = min(scores)

avg_score = sum(scores) / len(scores)

print(f"最高分：{max_score}，最低分：{min_score}，平均分：{avg_score:.2f}")

实践七

二、

1. 大括号；逗号；键值；键；值

2. 映射；键；无序

3. 唯一；不允许；允许

4. 不可变；可以

5. {}；dict()

6. 不能；删除后重新创建

7. 字典名[键] = 值

8. keys()；values()；items()

9. clear()清空字典内所有元素，del 命令是删除字典本身

10. 键

11. 没有；sorted()

12. 大括号；不能；无序

三、

1、

>>>type(myDict)：输出 <class 'dict'>

>>>myDict["鸡翅"]：输出 10

>>>myDict（修改汉堡后）：输出 {'汉堡': 15.5, '鸡翅': 10, '薯条': 6}

>>>myDict（添加奶茶后）：输出 {'汉堡': 15.5, '鸡翅': 10, '薯条': 6, '奶茶': 12}

>>>"鸡块" in myDict：输出 False

>>>"鸡翅" in myDict：输出 True，解析：判断键"鸡翅"是否在字典中，之前操作中字典包含该键。

>>>myDict.pop("薯条")：输出 6

>>>myDict：输出 {'汉堡': 15.5, '鸡翅': 10, '奶茶': 12}

>>>myDict.get("鸡翅")：输出 10

>>>myDict.get("鸡翅","抱歉,无此商品")：输出 10

>>>myDict.clear() 后 >>>myDict：输出 {}

（1） 会产生 KeyError 错误。原因：字典的索引方式是通过键访问，而非整数下标，0 不是该字典中的键，因此会报错。

（2）使用 字典名[键] 查询时，若键不存在会抛出 KeyError 错误；而 get() 方法查询时，若键不存在，可返回默认值（若未指定默认值则返回 None），不会报错。

（3）del myDict["薯条"]

（4）del myDict

2.

(1)a b c;97 98 99; a:97 b:98 c:99

(2)运行结果不受影响。

3.

>>>len(myDict)：输出 3

>>>list(myDict.keys())：输出 ['color', 'shape', 'pattern']

>>>myDict["shape"]：输出 ['circle', 'square', 'triangle']

>>>myDict["pattern"][1]：输出 "line"

>>>myDict["color"][1:]：输出 ['green', 'blue']

4.

>>>myDict["pattern"]：输出 {'dot':24, 'line':26}

>>>myDict["shape"]["square"]：输出 20

>>>colorDict=myDict["color"] 后 >>>colorDict：输出 {'red':12, 'green':23, 'blue':15}

>>>sorted(colorDict)：输出 ['blue', 'green', 'red']

- >>>sorted(colorDict.items())：输出 [('blue', 15), ('green', 23), ('red', 12)]

>>>lst=[(v, k) for k, v in colorDict.items()] 后 >>>lst :输出 [(12, 'red'), (23, 'green'), (15, 'blue')]

>>>lst.sort() 后 >>>lst：输出 [(12, 'red'), (15, 'blue'), (23, 'green')]

>>>dict(lst)：输出 {12: 'red', 15: 'blue', 23: 'green'}

(1) 键；列表；没有。

(2)将字典 colorDict 中的键值对反转

(3)将包含元组的列表 lst 转换为字典

5.

myDict.items():

s = sum(v)

length = len(v)

print("{}:{:.1f}".format(k, s / length))

6.

>>>set1 & set2：输出 {2,3,5}

>>>set1 | set2：输出 {1,2,3,4,5,6}

>>>set1 ^ set2：输出 {1,4,6}

>>>set1 - set2：输出 {1,4}

>>>set2 - set1：输出 {6}

>>>set1 >= set3：输出 True

>>>set2 < set3：输出 False

>>>set3.add((7,8)) 后 >>>set3：输出 {2,3,5,(7,8)}

>>>set3.remove(2) 后 >>>set3：输出 {3,5,(7,8)}

>>>set3.clear() 后 >>>set3：输出 set()

7.

```
dic_student={}
for i in range(5):
    name=input(f"请输入第{i+1}个学生的名字：")
    dic_student[i]=name
    age=input(f"请输入第{i+1}个学生的年龄：")
    dic_student[name]=age
```

```
for k,v in dic_student.items():
    print((k,v))
```

8.

```
myDict={"方糖":9,"鸡蛋":49,"魔盒":39,"曲奇":29}
sum=0
max=0
for k,v in myDict.items():
    print(f"{k}.....{v}元")
    sum=sum+v
    print(sum)
    if max<v:
        max=v
print(f"平均价格为：{sum/4},最大值为：{max}")
```

9.

```
dic_student = {}
students = [ ("王建", [18, "172cm", "80kg"]), ("张云", [19, "165cm", "55kg"]), ("张秋雨",
[18, "178cm", "82kg"]),("刘欢", [17, "169cm", "75kg"]),("姜宇", [19, "170cm", "70kg"])]
for name, info in students:
```

```
    dic_student[name] = info
for name, info in dic_student.items():
    age, height, weight = info
    print(f"{name}\t{age}\t{height}\t{weight}")
```

10

```
dic_student = {}
students = [ ("王建", [18, "172cm", "80kg"]), ("张云", [19, "165cm", "55kg"]), ("张秋雨",
[18, "178cm", "82kg"]),("刘欢", [17, "169cm", "75kg"]),("姜宇", [19, "170cm", "70kg"])]
for name, info in students:
    dic_student[name] = info
for name, info in dic_student.items():
    age, height, weight = info
    print(f"{name}\t{age}\t{height}\t{weight}")
```

11.

```
dic_country = {"China": "Beijing", "Amarica": “Huashengdun", "Norway": "Oslo",
“Japan”:”Tokyo, "Germany": “Berlin","Canada": “Ottawa", “France":  “Paris",
"Thailand":  “Bangkok", }
country = input("请输入国家名：")
if country in dic_country:
    print(f"{country}的首都是{dic_country[country]}")
else:
    print("抱歉，未找到该国家的首都信息")
```

12.

```
dict={"John":123,"Marry":111,"Tommy":123456}
name=input("请输入用户名：")
mima = input("请输入密码：")
if name in dict.keys():
    if dict[name]==mima:
        print("登陆成功")
    else:
        print("密码不正确")
else:
    print("用户名错误")
```

13.

```
lst_staff = ["李梅", "张富", "付妍", "赵诺", "刘江"]

dic_award = {"张富": 10000, "赵诺": 15000}

for staff in lst_staff:

    if staff in dic_award:

        print(f"{staff} 应发年终奖金额：{dic_award[staff]} 元")

    else:

        print(f"{staff} 应发年终奖金额：5000 元")
```

14.

```
dic_score = {"徐丽": {"语文": 88, "数学": 90, "英语": 98, "计算机": 95},"张兴": {"语文": 85, "数学": 92, "英语": 95, "计算机": 98} , "刘宁": {"语文": 89, "数学": 89 ， "英语": 90, "计算机": 92} , "张旭": {"语文": 82, "数学": 86, "英语": 89, "计算机": 90}}

for name, scores in dic_score.items():

    total = sum(scores.values())

    average = total / len(scores)
```

```
        dic_score[name]["平均分"] = average
for name, info in dic_score.items():
        print(f"姓名：{name}")
        for subject, score in info.items():
                print(f"{subject}：{score}")
        print("-" * 20)
```

15.

```
s = input("请输入英文字符串：")
myDict = {}
for char in s:
        if char.isalpha():
                char_lower = char.lower()
                myDict[char_lower] = myDict.get(char_lower, 0) + 1
for key, value in myDict.items():
        print(f"字母 {key} 出现了 {value} 次")
```

16.

```
s = "Artificial Intelligence: From Beginning to Date covers a wide range of topics in artificial intelligence with three distinct features. Artificial Intelligence: From Beginning to Date is expected to be welcomed by readers.It is a comprehensive manual and practical guide for artificial intelligence research and development personnel to conduct research on related artificial intelligence projects. It is also a valuable reference for undergraduate and graduate students to learn artificial intelligence."

s = s.lower().replace(":", "").replace(".", "").replace(",", "")

words = s.split()

word_dict = {}

for word in words:

    word_dict[word] = word_dict.get(word, 0) + 1

print("单词出现次数统计：")

for word, count in word_dict.items():

    print(f"{word}: {count}")

sorted_words = sorted(word_dict.items(), key=lambda x: x[1], reverse=True)

print("\n 出现次数排前五名的单词：")

for i in range(min(5, len(sorted_words))):

    word, count = sorted_words[i]

    print(f"{word}: {count}")
```

17.

```
food_dict = {
    "蔬菜": {"菠菜": "绿色", "胡萝卜": "橙色", "茄子": "紫色", "毛豆": "绿色"},
    "水果": {"山竹": "紫色", "香蕉": "黄色", "橘子": "橙色", "草莓": "红色"},
    "饮料": {"椰子汁": "白色", "西瓜汁": "红色", "玉米汁": "黄色", "葡萄汁": "紫色"}
}
color_count = {}
for category, items in food_dict.items():
    for food, color in items.items():
        color_count[color] = color_count.get(color, 0) + 1
for color, count in color_count.items():
    print(f" {color} 的食物有 {count} 个")
```

18.

```
dic_house = {"李刚":93, "陈静":78, "张金柱":88, "赵启山":91, "李鑫":65, "黄宁":83}
sorted_students = sorted(dic_house.items(), key=lambda x: x[1], reverse=True)
for student, score in sorted_students:
```

```
    print(student)
```

19.

```
dic_house = {"001": {"房型": "3 室 1 厅", "面积": 68.69, "朝向": "南北", "装修情况": "简装
", "挂牌价": 37124, "关注人数": 35},"002": {"房型": "2 室 1 厅", "面积": 87.16, "朝向": "
南西", "装修情况": "精装", "挂牌价": 37375, "关注人数": 148},
    "003": {"房型": "3 室 1 厅", "面积": 61.72, "朝向": "南北", "装修情况": "精装", "挂牌
价": 37266, "关注人数": 146},
    "004": {"房型": "3 室 2 厅", "面积": 68.18, "朝向": "南北", "装修情况": "精装", "挂牌
价": 68496, "关注人数": 79},
    "005": {"房型": "3 室 2 厅", "面积": 71.67, "朝向": "南西", "装修情况": "简装", "挂牌
价": 33487, "关注人数": 105},
    "006": {"房型": "3 室 1 厅", "面积": 84.78, "朝向": "南北", "装修情况": "简装", "挂牌
价": 51782, "关注人数": 34}
}
dic_house1=sorted(dic_house.items(), key=lambda x: x[1]["挂牌价"])
print("价格最低的三套房：")
for i in range(3):
    print(dic_house1[i])
dic_house2=sorted(dic_house.items(), key=lambda x: x[1]["关注人数"])
```

```
print("人气值最高的三套房：")
for i in range(3):
    print(dic_house2[i])
```

20.

```
dic_repertory={"酱油":50,"醋":60,"盐":100,"糖":120,"鸡精":20,"麻油":60}
dic_repertory["糖"]=80
dic_repertory["鸡精"]=50
dic_repertory["麻油"]=60
dic_repertory=sorted(dic_repertory.items(), key=lambda x: x[1], reverse=False)
print("价格最高的是：",dic_repertory[0])
print("价格最低的是：",dic_repertory[5])
```

21.

```
s = "Whether the weather be fine, or whether the weather be not. Whether the
weather be cold, or whether the weather be hot. We will weather the weather whether
we like it or not."
s = s.lower().replace(",", "").replace(".", "")
```

```
words = s.split()

unique_words = set(words)

count = len(unique_words)

print(f"不区分大小写且不重复的英文单词个数为：{count}")
```

22.

```
set_highjump = {"李朋", "王宇", "张锁", "刘松山", "白旭", "李晓亮"}

set_longjump = {"王宇", "唐英", "刘松山", "白旭", "刘小雨", "宁成"}

all_students = set_highjump.union(set_longjump)

print("参加比赛的所有学生名单：", all_students)

both = set_highjump.intersection(set_longjump)

print("两项比赛都参加的学生名单：", both)

only_highjump = set_highjump - set_longjump

print("仅参加跳高比赛的同学名单：", only_highjump)

only_longjump = set_longjump - set_highjump

print("仅参加跳远比赛的同学名单：", only_longjump)

only_one = only_highjump.union(only_longjump)

print("仅参加一项比赛的学生名单：", only_one)
```

实践八

二．

1. (1) 处理；(2) 内置函数、自定义函数、匿名函数（lambda 函数）、递归函数

2. 函数定义语法格式：def 函数名(参数列表):

(1) 函数名；标识符

(2) 冒号（:）

(3) 逗号（,）

(4) return

(5) 可以

3. 函数调用

(1) 函数名(实际参数列表)

(2) 实际参数（实参）；顺序

(3) 表达式

(4) 语句

4. 函数的参数

(1) 实参

(2) 实际参数（实参）

(3) 位置；名称；默认值

5. 默认参数值

(1) 默认；默认

(2) 末尾

6. 可变数量参数

两个星号（**）；字典

7. 全局变量与局部变量

(1) 局部

(2) 全局

(3) 局部；全局

8. lambda()函数

一行内；匿名函数

9. 递归函数

(1) 它自己

(2) 基线条件（终止条件）；递归条件

10.

d1= 3.6055

d2= 4.4721

d3 = 5

三．

1、

```
def copy(n):
    for i in range(n):
        print("I want nobody nobody but you")


copy(1)
print("How can I be with other")
print("I do not want any other")
copy(3)
```

2.

```
def isOdd(n):
    if type(n) == int:
        if n % 2 != 0:
            return True
        else:
            return False
    else:
        print("数据错误")
```

3.

```
def calc_age(last_digit, birth_year):
    result = (last_digit * 2 + 5) * 50 + 1766 - birth_year
    return result % 100   # 取最后两位作为年龄
print(calc_age(3, 1990))
```

4.

```
def isLeap(year):
    # 判断是否为闰年
    return (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0)

# 主程序统计 1990-2020 年的闰年
count = 0
for year in range(1990, 2021):
    if isLeap(year):
        print(year, end=' ')
        count += 1
        if count % 5 == 0:
```

```
                print()  # 每行输出 5 个后换行
if count % 5 != 0:
    print()  # 确保最后一行输出后换行
```

5.

```
def sum_digits(n):
    s = str(n)
    total = 0
    for c in s:
        total += int(c)
    return total
num = int(input("请输入一个正整数："))
print(f"{num}的各位数字之和为：{sum_digits(num)}")
```

6.

```
def sum_digits(n):
    s = str(n)
```

```
    total = 0
    for c in s:
        total += int(c)
    return total
for x in range(100, 1000):
    sums = []
    for i in range(3, 8):
        product = x * i
        sums.append(sum_digits(product))
    if len(set(sums)) == 1:
        print(f"x={x}: ", end="")
        for i in range(3, 8):
            print(f"x*{i}={x*i}", end=", " if i < 7 else "\n")
```

7.

```
def is_perfect(n):
    if n <= 0:
        return 0
```

```
    divisors = []
    for i in range(1, n):
        if n % i == 0:
            divisors.append(i)
    return 1 if sum(divisors) == n else 0
num = int(input("请输入一个整数："))
print(f"该数是完数" if is_perfect(num) else "该数不是完数")
```

8.

```
def sum_series(n, m):
    total = 0
    for i in range(1, n+1):
        total += i ** m
    return total
s = sum_series(100, 1) + sum_series(50, 2) + sum_series(10, -1)
print(f"s 的值为：{s}")
```

9.

```
def sum_aaa(a, n):
    total = 0
    current = 0
    for _ in range(n):
        current = current * 10 + a
        total += current
    return total
a = int(input("请输入 a（1~9 之间）："))
n = int(input("请输入 n："))
print(f"当 a={a}, n={n}时，s 的值为：{sum_aaa(a, n)}")
```

10.

10.

```
import random
def red_packet(total=100, num=15):
    amounts = []
    remaining = total * 100  # 转为分方便计算
    for i in range(num - 1):
```

```
            max_amount = remaining - (num - i - 1)
            amount = random.randint(1, max_amount)
            amounts.append(amount / 100)
            remaining -= amount
        amounts.append(remaining / 100)
        return amounts


print("默认值测试（总金额 100，数量 15）: ")
packets = red_packet()
print(packets)
print(f"总金额：{sum(packets):.2f}")
print("\n 自定义值测试（总金额 200，数量 10）: ")
packets_custom = red_packet(200, 10)
print(packets_custom)
print(f"总金额：{sum(packets_custom):.2f}")
print("\n（1）单个红包最多接近总金额（如总金额 100，数量 15 时，最多接近
100-0.01*14=99.86 元）。")
print("（2）特点：金额随机，满足最少 0.01 元且总和等于总金额；公平性：理论上随
```

机，但实际因分配顺序可能存在差异。")

print("（3）修改规则后，单个红包最大值限制更严格，金额分布更均匀，极端大额红包概率降低。")

print("（4）可设计为基于正态分布的算法，使金额更集中在平均值附近，提升公平性。")

11.

```
def isdif(n):
    s = str(n)
    for c in s:
        if s.count(c) > 1:
            return 0
    return 1
num = int(input("请输入一个正整数："))
if isdif(num):
    print(f"{num}的各位数字互不相同")
else:
    print(f"{num}中有重复数字")
```

12.

```
def sum_positive(lst):
    total = 0
    for num in lst:
        if num > 0:
            total += num
    return total
test_lst = [-1, 2, 3, -4, 5]
print(f"列表{test_lst}的正数之和为：{sum_positive(test_lst)}")
```

13.

```
def has_duplicate(lst):
    for elem in lst:
        if lst.count(elem) > 1:
            return True
    return False
def has_duplicate_set(lst):
    return len(set(lst)) < len(lst)
lst1 = [1,2,1,3,4]
lst2 = [1,2,3,4,5]
```

```
print(f"列表{lst1}是否有重复元素：{has_duplicate(lst1)}")

print(f"列表{lst2}是否有重复元素：{has_duplicate_set(lst2)}")
```

14.

```
import random
def transfer(a,low,hight):
    list=[]
    min_value=min(a)
    max_value=max(a)
    for i in a:
        print(i)
        x=low+(i-min_value)*(hight-low)/(max_value-min_value)
        x=round(x,4)
        list.append(x)
    return list
a=[]
for i in range(10):
    x=random.randint(1, 100)
    print(x)
    a.append(x)
b=transfer(a,10,100)
print(b)
```

15.

```
def judge(password):
    level = 0
    has_digit = any(ch.isdigit() for ch in password)
    if has_digit:
        level += 1
    has_upper = any(ch.isupper() for ch in password)
    if has_upper:
        level += 1
    has_lower = any(ch.islower() for ch in password)
    if has_lower:
        level += 1
    if len(password) >= 8:
        level += 1
    return level
while True:
    password = input("请输入测试密码：")
```

```
    if not password:
        break
    level = judge(password)
    print(f"{password} 的密码强度为 {level} 级")
```

实践九

二、

1.ASCLL，Unicode

2.

r	读取文件内容	错误	√	×
w	清空文件内容	创建	×	√
a	保留文件内容（追加）	创建	×	√
r+	读取文件内容	错误	√	√
w+	清空文件内容	创建	√	√
a+	保留文件内容（追加）	创建	√	√

3.

（1）read（）；readline（）；字符串；readlines（）；列表

（2）write（）

（3）列表

（4）close（ ）

（5）writerow（ ）；逗号；writerows（ ）

4.

（1）try。。。except

（2）except（ ）

三、

1.（1）仅读模式打开文件

（2）分割

（3）因为 B 行将 a，b 经过分割，转换成了字符串，需要重新转换为整数

（4）关闭文件

（5）如果该文件不存在可以直接创建

（6）以打开模式访问文件

（7）不能

2.

（1）ZeroDivisionError 异常相关

c_3=c/a（当用户输入 a=0 时，该语句执行除法 c/0，触发除零异常）。

（2）ValueError 异常相关

假设 a、c 接收的数据可正常转为整数，可能产生 ValueError 的语句：无（因题目假设数据可正常转整数，而 ValueError 通常在类型转换失败时触发，此处无相关语句）。

（3）IndexError 异常相关

- IndexError 表示 列表索引超出范围异常（当尝试访问列表中不存在的索引时触发）。
- 产生 IndexError 的语句：print(ls[3])（列表 ls=[a,b,c]只有 3 个元素，索引为 0、1、2，访问 ls[3]超出范围）。
- 产生 IndexError 的原因：列表 ls 的长度为 3，有效索引是 0、1、2，而语句 print(ls[3]) 试图访问索引 3，该索引不存在于列表中。

3.

```
content = """慈母手中线，游子身上衣。
临行密密缝，意恐迟迟归。
谁言寸草心，报得三春晖。"""

with open("yzy.txt", "w", encoding="utf-8") as f:
    f.write(content)
for i in f.readline:
```

```
        print(i)
l=f.readlines()
```

5.

```
total_cost = 0
with open("a.txt", "r") as f:
        for line in f:
                parts = line.strip().split()
                price = float(parts[1])
                quantity = int(parts[2])
                total_cost += price * quantity
print(f"总费用：{total_cost}")
```

6.

```
import csv
program_design = []
cell_bio = []
physiology = []
```

```
with open("score.csv", "r", newline="") as f:
    reader = csv.reader(f)
    next(reader)  # 跳过表头
    for row in reader:
        program_design.append(int(row[1]))
        cell_bio.append(int(row[2]))
        physiology.append(int(row[3]))
def calc_stats(lst):
    return {"平均分": sum(lst)/len(lst), "最高分": max(lst), "最低分": min(lst)}
stats_pd = calc_stats(program_design)
stats_cb = calc_stats(cell_bio)
stats_phy = calc_stats(physiology)
print("程序设计课程统计：", stats_pd)
print("细胞生物课程统计：", stats_cb)
print("生理学课程统计：", stats_phy)
```

7.

```
line_count = 0
```

```
word_count = 0

with open("zen.txt", "r") as f:

    for line in f:

        line_count += 1

        word_count += len(line.strip().split())

print(f"行数：{line_count}，单词个数：{word_count}")
```

8.

```
import random

nums = [random.randint(10, 99) for _ in range(100)]

with open("number.txt", "w") as f:

    for i in range(0, 100, 10):

        line = " ".join(map(str, nums[i:i+10])) + "\n"

        f.write(line)
```

9.

```
import random

nums = [random.randint(10, 99) for _ in range(100)]
```

```
with open("number.txt", "w") as f:
    for i in range(0, 100, 10):
        line = " ".join(map(str, nums[i:i+10])) + "\n"
        f.write(line)
```

10.

```
with open("data.txt", "r") as f:
    nums = list(map(int, f.read().split(",")))
nums.sort()
with open("sorted_data.txt", "w") as f:
    f.write(",".join(map(str, nums)))
```

11.

```
students = []
with open("score.txt", "r") as f:
    next(f)  # 跳过表头
    for line in f:
        sid,平时,期末  = line.strip().split()
```

```
            平时 = int(平时)
            期末 = int(期末)
            总评 = round(平时 * 0.4 + 期末 * 0.6)
            students.append((sid, 平时, 期末, 总评))
students.sort(key=lambda x: x[3], reverse=True)
with open("sorted_score.txt", "w") as f:
    f.write("学号,平时成绩,期末成绩,总评成绩\n")
    for stu in students:
        f.write(f"{stu[0]},{stu[1]},{stu[2]},{stu[3]}\n")
```