1、 简单解释Python基于值的自动内存管理方式？（Python采用的是基于值得内存管理方式，在Python中可以为不同变量赋值为相同值，这个值在内存中只有一份，多个变量指向同一个内存地址；Python具有自动内存管理功能，会自动跟踪内存中所有的值，对于没有任何变量指向的值，Python自动将其删除。）   
2、 写出Python运算符&的两种功能？（1）数字位运算；2）集合交集运算。）   
3、 在Python中导入模块中的对象有哪几种方式？（1)import 模块名 [as 别名]；2)from 模块名 import 对象名[ as 别名]；3)from math import \*）   
4、 解释Python脚本程序的“**name**”变量及其作用？（每个Python脚本在运行时都有一个“**name**”属性。如果脚本作为模块被导入，则其“**name**”属性的值被自动设置为模块名；如果脚本独立运行，则其“**name**”属性值被自动设置为“**main**”。利用“**name**”属性即可控制Python程序的运行方式。）   
5、 为什么应尽量从列表的尾部进行元素的增加与删除操作？（当列表增加或删除元素时，列表对象自动进行内存扩展或收缩，从而保证元素之间没有缝隙，但这涉及到列表元素的移动，效率较低，应尽量从列表尾部进行元素的增加与删除操作以提高处理速度。）   
6、 分析逻辑运算符“or”的短路求值特性？（假设有表达式“表达式1 or 表达式2”，如果表达式1的值等价于True，那么无论表达式2的值是什么，整个表达式的值总是等价于True。因此，不需要再计算表达式2的值。）   
7、 简单解释Python中短字符串驻留机制？（对于短字符串，将其赋值给多个不同的对象时，内存中只有一个副本，多个对象共享改副本。）   
8、 异常和错误有什么区别？（异常是指因为程序执行过程中出错而在正常控制流以外采取的行为。严格来说，语法错误和逻辑错误不属于异常，但有些语法错误往往会导致异常，例如由于大小写拼写错误而访问不存在的对象，或者试图访问不存在的文件，等等。）   
9、 使用pdb模块进行Python程序调试主要有哪几种用法？（1）在交互模式下使用pdb模块提供的功能可以直接调试语句块、表达式、函数等多种脚本。2）在程序中嵌入断点来实现调试功能。在程序中首先导入pdb模块，然后使用pdb.set\_trace()在需要的位置设置断点。如果程序中存在通过该方法调用显式插入的断点，那么在命令提示符环境下执行该程序或双击执行程序时将自动打开pdb调试环境，即使该程序当前不处于调试状态。3）使用命令行调试程序。在命令行提示符下执行“python –m pdb 脚本文件名”，则直接进入调试环境；当调试结束或程序正常结束以后，pdb将重启该程序。）   
10、 阅读下面的代码，并分析假设文件“D:\test.txt”不存在的情况下两段代码可能发生的问题。   
代码1：

try:   
fp = open(r’d:\test.txt’)   
print(‘Hello world!’, file=fp)   
finally:   
fp.close()   
代码2：   
try:   
fp = open(r’d:\test.txt’, ‘a+’)   
print(‘Hello world!’, file=fp)   
finally:   
fp.close()   
答：   
假设文件“D:\test.txt”不存在，那么第一段代码会抛出异常，提示fp没有定义；第二段代码执行正常。原因是第二段代码使用内置函数open()打开指定文件时如果不存在则会创建该文件，从而不会抛出异常。   
11、   
四、 编程题   
1、编写程序，在D盘根目录下创建一个文本文件test.txt，并向其中写入字符串hello world。   
答：   
fp = open(r’D:\test.txt’, ‘a+’)   
print(‘hello world’, file=fp)   
fp.close()   
2、写出下面代码的优化版本，提高运行效率。   
x = list(range(500))   
for item in x:   
t = 5\*\*5   
print(item+t)   
答：   
x = list(range(500))   
t = 5\*\*5   
for item in x:   
print(item+t)   
3、编写程序，生成一个包含20个随机整数的列表，然后对其中偶数下标的元素进行降序排列，奇数下标的元素不变。（提示：使用切片。）   
答：   
import random   
x = [random.randint(0,100) for i in range(20)]   
print(x)   
y = x[::2]   
y.sort(reverse=True)   
x[::2] = y   
print(x)   
4、写出下面代码的执行结果。   
def Join(List, sep=None):   
return (sep or ‘,’).join(List)   
print(Join([‘a’, ‘b’, ‘c’]))   
print(Join([‘a’, ‘b’, ‘c’],’:’))   
答：   
a,b,c   
a:b:c   
5、写出下面代码的运行结果。   
def Sum(a, b=3, c=5):   
return sum([a, b, c])   
print(Sum(a=8, c=2))   
print(Sum(8))   
print(Sum(8,2))   
答：   
13   
16   
15   
6、写出下面代码的运行结果。   
def Sum(\*p):   
return sum(p)   
print(Sum(3, 5, 8))   
print(Sum(8))   
print(Sum(8, 2, 10))   
答：   
16   
8   
20   
7、编写函数，判断一个数字是否为素数，是则返回字符串YES，否则返回字符串NO。   
答：   
import math   
def IsPrime(v):   
n = int(math.sqrt(v)+1)   
for i in range(2,n):   
if v%i==0:   
return ‘No’   
else:   
return ‘Yes’   
8、编写函数，模拟Python内置函数sorted()。   
答：   
def Sorted(v):   
t = v[::]   
r = []   
while t:   
tt = min(t)   
r.append(tt)   
t.remove(tt)   
return r   
9、编写程序，生成包含20个随机数的列表，然后将前10个元素升序排列，后10个元素降序排列，并输出结果。   
答：   
import random   
x = [random.randint(0,100) for i in range(20)]   
print(x)   
y = x[0:10]   
y.sort()   
x[0:10] = y   
y = x[10:20]   
y.sort(reverse=True)   
x[10:20] = y   
print(x)   
10、编写程序，运行后用户输入4位整数作为年份，判断其是否为闰年。如果年份能被400整除，则为闰年；如果年份能被4整除但不能被100整除也为闰年。   
答：   
x = input(‘Please input an integer of 4 digits meaning the year:’)   
x = eval(x)   
if x%400==0 or (x%4==0 and not x%100==0):   
print(‘Yes’)   
else:   
print(‘No’)   
11、编写程序，实现分段函数计算，如下表所示。   
x y   
x<0 0   
0<=x<5 x   
5<=x<10 3x-5   
10<=x<20 0.5x-2   
20<=x 0

答：   
x = input(‘Please input x:’)   
x = eval(x)   
if x<0 or x>=20:   
print(0)   
elif 0<=x<5:   
print(x)   
elif 5<=x<10:   
print(3\*x-5)   
elif 10<=x<20:   
print(0.5\*x-2)   
12、阅读下面的程序，判断其是否可以正常运行，如果可以运行则写出执行结果，如果不能运行则写出理由。   
class Test:   
def **init**(self, value):   
self.\_\_value = value   
@property   
def value(self):   
return self.\_\_value

t = Test(3)   
t.value = 5   
print(t.value)   
答：   
不能运行。程序中定义的是只读属性，不能修改属性的值。   
13、下面代码的功能是，随机生成50个介于[1,20]之间的整数，然后统计每个整数出现频率。请把缺少的代码补全。   
import random   
x = [random.***\_\_***(1,20) for i in range(**\_**)]   
r = dict()   
for i in x:   
r[i] = r.get(i, **\_**)+1   
for k, v in r.items():   
print(k, v)   
答：   
分别填写randint、50、0   
14、假设有Python程序文件demo.py，代码如下：   
def main():   
if **name** == ‘**main**‘:   
print(1)   
else:   
print(2)

main()   
将该程序文件直接运行时输出结果为***\_***，作为模块导入时得到结果**\_\_\_**-。（1、2）   
15、下面程序的执行结果是***\_\_\_\_\_\_\_\_***。（1）   
s = 0   
for i in range(1,101):   
s += i   
else:   
print(1)   
16、下面程序的执行结果是***\_\_\_\_***。（1275）   
s = 0   
for i in range(1,101):   
s += i   
if i == 50:   
print(s)   
break   
else:   
print(1)   
17、下面的程序是否能够正常执行，若不能，请解释原因；若能，请分析其执行结果。   
from random import randint

result = set()   
while True:   
result.add(randint(1,10))   
if len(result)==20:   
break   
print(result)   
答：无法正确执行，因为该程序的功能是从[1,10]区间中选择20个不同的随机整数，而该区间并没有这么多整数，所以程序死循环。   
18、下面的代码是否能够正确运行，若不能请解释原因；若能，请分析其执行结果。

x = list(range(20))   
for i in range(len(x)):   
del x[i]   
答：无法正确执行，因为删除列表元素时会影响其他元素在列表中的索引，上面的代码会抛出下标越界的异常。   
19、阅读下面的代码，解释其功能。   
x = list(range(20))   
for index, value in enumerate(x):   
if value == 3:   
x[index] = 5   
答：将列表x中值为3的元素修改为5。   
20、阅读下面的代码，解释其功能。   
x = [range(3\*i, 3\*i+5) for i in range(2)]   
x = list(map(list, x))   
x = list(map(list, zip(\*x)))   
答：首先生成一个包含列表的列表，然后模拟矩阵转置。   
21、阅读下面的代码，解释其功能。   
import string   
x = string.ascii\_letters + string.digits   
import random   
print(”.join(random.sample(x, 10)))   
答：输出由英文字母大小写或数字组成的长度为10且不重复的随机字符串。   
22、阅读下面的代码，分析其执行结果。   
def demo(\*p):   
return sum(p)   
print(demo(1,2,3,4,5))   
print(demo(1,2,3))   
答：输出结果为   
15   
6   
23、阅读下面的代码，分析其执行结果。   
def demo(a, b, c=3, d=100):   
return sum((a,b,c,d))   
print(demo(1, 2, 3, 4))   
print(demo(1, 2, d=3))   
答：输出结果为   
10   
9   
24、下面的代码输出结果为***\_\_\_***。（3）   
def demo():   
x = 5   
x = 3   
demo()   
print(x)   
25、下面函数的功能为***\_\_\_\_\_***。（将序列循环左移k位，得到新序列并返回）   
def demo(lst, k):   
if k