**Python题库**

# 填空题

1. Python安装扩展库常⽤用的⼯工具是\_\_\_\_pip\_\_\_\_\_\_和conda，其中后者需要安装

Python集成开发环境Anaconda3之后才可以使⽤用，⽽而前者是Python官⽅方推荐和标配的。

1. Python程序⽂文件扩展名主要有\_\_\_\_py\_\_\_\_\_\_和\_\_\_\_pyw\_\_\_\_\_两种，其中后者常⽤用于GUI程序。
2. Python源代码程序伪编译后的⽂文件扩展名为\_\_\_\_pyc\_\_\_\_\_\_。
3. 使⽤用pip⼯工具在线安装Excel⽂文件操作扩展库openpyxl的完整命令是

\_\_\_pip install openpyxl\_\_\_\_\_\_\_。

1. 使⽤用pip⼯工具在线升级科学计算扩展库numpy的完整命令是\_\_

\_pip install --upgrade numpy\_\_\_\_\_\_\_。

1. 使⽤用pip⼯工具把本机已安装的Python扩展库及版本信息输出到⽂文本⽂文件 requirements.txt中的完整命令是\_\_pip install –r requirements.txt\_\_\_\_\_\_\_\_。
2. 使⽤用pip⼯工具查看当前已安装的Python扩展库（不不含版本号）的完整命令是

\_\_pip list\_\_\_\_\_\_\_\_。

1. 表达式int('11111', 2)的值为\_\_\_\_31\_\_\_\_\_\_
2. 表达式chr(ord('D')+2)的\_\_F\_\_\_\_\_\_\_\_。
3. 运算符%\_\_\_\_\_可以\_\_\_\_\_（可以/不不可以）对浮点数进⾏行行求余数操作。
4. ⼀一个数字5\_\_\_\_\_是\_\_\_\_\_（是/不不是）合法的Python表达式。
5. 在Python 3.x中，内置函数input()把⽤用户的键盘输⼊入⼀一律律作为字符串串返回。正确
6. 在Python中关键字\_\_\_\_None\_\_\_\_\_\_表示空类型。
7. 列表、元组、字符串是Python的\_\_\_\_\_\_\_\_\_\_(有序、⽆无序？)序列。
8. 查看变量量类型的Python内置函数是\_\_\_\_type()\_\_\_\_\_\_。
9. 查看对象内存地址的Python内置函数是\_\_\_id()\_\_\_\_\_\_\_。
10. 以3为实部4为虚部，Python复数的表达形式为\_\_\_3+4J\_\_\_\_\_\_\_或\_\_\_\_3+4j\_\_\_\_\_\_。
11. Python运算符中⽤用来计算整商的是\_\_\_\_\_//\_\_\_\_\_。
12. Python运算符中⽤用来计算集合并集的是\_\_\_\_|\_\_\_\_\_\_。 20. Python运算符中⽤用来计算集合差集的是\_\_\_\_-\_\_\_\_\_\_。  &交集
13. Python运算符中⽤用来计算集合交集的是\_\_\_\_&\_\_\_\_\_\_。
14. 使⽤用运算符测试集合A是否为集合B的真⼦子集的表达式可以写作\_\_\_A>B\_\_\_\_\_\_\_。
15. \_\_\_del()\_\_\_\_\_\_\_命令既可以删除列列表中的⼀一个元素，也可以删除整个列列表或其他任意类型的Python对象。
16. 表达式int('123', 16)的值为\_\_\_1\*16^2+2\*16^1+3\*16^0\_\_\_\_\_\_\_。
17. 表达式int('123', 8)的值为\_\_\_\_\_\_\_\_\_\_。
18. 表达式int('123')的值为\_\_\_123\_\_\_\_\_\_\_。
19. 表达式int('101',2)的值为\_\_\_5\_\_\_\_\_\_\_。
20. 表达式abs(-3)的值为\_\_\_\_3\_\_\_\_\_\_。
21. Python 3.x语句句print(1, 2, 3, sep=':')的输出结果为\_1:2:3\_\_\_\_\_\_\_\_\_。
22. Python 3.x语句句print(1, 2, 3, sep=',')的输出结果为\_\_1,2,3\_\_\_\_\_\_\_\_。
23. 表达式int(4\*\*0.5)的值为\_\_\_2\_\_\_\_\_\_\_。
24. 表达式8 \*\* (1/3)的值为\_\_\_\_\_2.0\_\_\_\_\_。
25. Python内置函数\_\_\_\_len()\_\_\_\_\_\_可以返回列列表、元组、字典、集合、字符串串以及 range对象中元素个数。
26. Python内置函数\_\_\_\_max()\_\_\_\_\_\_⽤用来返回序列列中的最⼤大元素。
27. Python内置函数\_\_\_min()\_\_\_\_\_\_\_⽤用来返回序列列中的最⼩小元素。
28. Python内置函数\_\_\_\_sum()\_\_\_\_\_\_⽤用来返回数值型序列列中所有元素之和。
29. 已知x = 3，那么执⾏行行语句句x += 6之后，x的值为\_\_\_9\_\_\_\_\_\_\_。
30. 表达式13 / 4的值为\_\_\_\_3.25\_\_\_\_\_\_。
31. 表达式13 // 4的值为\_\_\_\_3\_\_\_\_\_\_。
32. 表达式-13 // 4的值为\_\_\_\_-4\_\_\_\_\_\_。
33. 表达式3 \*\* 2的值为\_\_\_\_9\_\_\_\_\_\_。
34. 表达式chr(ord('a')-32) 的值为\_\_\_\_A\_\_\_\_\_\_。
35. 表达式abs(3+4j) 的值为\_\_\_\_5\_\_\_\_\_\_。
36. 表达式16 \*\* 0.5的值为\_\_\_\_4.0\_\_\_\_\_\_。
37. 表达式type({3})的值为\_\_\_\_set\_\_\_\_\_\_。
38. 表达式type({3:3})的值为\_\_\_dict\_\_\_\_\_\_\_。
39. 表达式isinstance('Hello world', str)的值为\_\_\_\_True\_\_\_\_\_\_。
40. 表达式isinstance('abc', int)的值为\_\_\_False\_\_\_\_\_\_\_。
41. 表达式isinstance(4j, (int, float, complex))的值为\_\_\_\_True\_\_\_\_\_\_。
42. 表达式isinstance('4', (int, float, complex))的值为\_\_\_True\_\_\_\_\_\_\_。
43. 表达式type(3) in (int, float, complex)的值为\_\_\_True\_\_\_\_\_\_\_。
44. 表达式type('3') in (int, float, complex)的值为\_\_\_False\_\_\_\_\_\_\_。
45. 表达式type(3) == int的值为\_\_\_\_True\_\_\_\_\_\_。
46. Python 3.x的range()函数返回⼀一个\_\_\_\_\_\_列表\_\_\_\_\_\_\_。
47. 表达式“[3] in [1, 2, 3, 4]”的值为\_\_\_\_False\_\_\_\_\_\_\_。
48. 列表对象的sort()方法用来对列表元素进⾏原地排序，该函数返回值为

\_\_\_\_\_从小到大的列表\_\_\_。

1. 列表对象的\_\_\_\_\_remove（）\_\_\_\_\_\_⽅方法删除⾸首次出现的指定元素，如果列列表中不不存在要删除的元素，则抛出异常。
2. 假设列列表对象aList的值为[3, 4, 5, 6, 7, 9, 11, 13, 15, 17]，那么切⽚片aList[3:7] 得到的值是\_\_\_\_\_\_【6,7,9,11】\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。
3. 在Python中，字典和集合都是⽤用⼀一对\_\_\_\_\_{}\_\_\_\_\_\_\_作为界定符，字典的每个元素有两部分组成，即\_\_\_\_键\_\_\_\_和\_\_\_\_值\_\_\_\_\_\_，其中\_\_\_\_键\_\_\_\_不不允许重复。
4. 使⽤字典对象的\_\_\_\_items（）\_\_\_\_\_\_方法可以返回字典的“键-值对”，使用字典对象的\_\_\_\_\_keys（）\_\_\_\_\_\_\_\_⽅法可以返回字典的“键”，使⽤字典对象的\_\_\_\_values（）\_\_\_\_\_\_方法可以返回字典的“值”。
5. 假设有列表a = ['name', 'age', 'sex']和b = ['Dong', 38, 'Male']，请使⽤一个语句将这两个列表的内容转换为字典，并且以列表a中的元素为“键”，以列表b中的元素为“值”，这个语句句可以写为\_\_\_\_\_dict(zip（a,b）)\_\_\_\_\_\_\_\_。
6. 假设有⼀个列表a，现要求从列表a中每3个元素取1个，并且将取到的元素组成新的列表b，可以使⽤用语句句\_\_\_b=a[0::3]\_\_\_\_\_\_\_\_\_\_\_\_\_。
7. 使⽤用列列表推导式⽣生成包含10个数字5的列列表，语句句可以写为

\_\_arr=[5 for i in range(10)]\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。

1. \_\_\_不可以\_\_\_（可以、不可以）使⽤用del命令来删除元组中的部分元素。
2. 表达式[1, 2, 3]\*3的值为\_\_\_\_[1,2,3,1,2,3,1,2,3]\_\_\_\_。
3. 表达式list(map(str, [1, 2, 3]))的值为\_[‘1’,’2’,’3’]\_\_\_\_\_\_\_\_\_。
4. 语句句x = 3==3, 5执⾏行行结束后，变量量x的值为\_\_ (True, 5)\_\_。
5. 已知x = 3，并且id(x)的返回值为496103280，那么执⾏行行语句句x += 6之后，表达式id(x) == 496103280的值为\_\_\_\_False\_\_\_\_\_\_。
6. 已知x = 3，那么执⾏行行语句句x \*= 6之后，x的值为\_\_\_18\_\_\_\_\_\_\_。
7. 表达式3 in [1, 2, 3, 4]的值为\_\_\_True\_\_\_\_\_\_\_。
8. 使⽤用列列表与整数的乘法⽣生成包含10个数字5的列列表，可以写为\_\_\_arr=[5]\*10\_\_\_\_。
9. 任意⻓长度的⾮非空列列表、元组和字符串串中最后⼀一个元素的下标为\_\_-l\_
10. 表达式list(range(1, 10, 3))的值为\_\_\_[1,4,7]\_\_\_\_\_\_\_。
11. 表达式list(range(10, 1, -3))的值为\_\_\_[10,7,4]\_\_\_\_\_\_\_。
12. 表达式list(range(5))的值为\_\_\_[0,1,2,3,4]\_\_\_\_\_\_\_\_\_\_\_\_\_。
13. 已知a = [1, 2, 3]和b = [1, 2, 4]，那么id(a[1])==id(b[1])的执⾏行行结果为

\_\_\_True\_\_\_\_\_\_\_。

1. 切⽚片操作list(range(6))[::2]执⾏行行结果为\_[0,2,4]\_\_\_\_\_\_\_\_\_。
2. 使用切片操作在列表对象x的开始处增加⼀个元素3的代码为\_\_\_x[0:0]=3\_\_。
3. 语句句sorted([1, 2, 3], reverse=True) == reversed([1, 2, 3])执⾏行行结果为

\_\_\_True\_\_。

1. 表达式sorted([111, 2, 33], key=lambda x: len(str(x)))的值为

\_\_\_\_[111,33,2]\_\_\_\_\_\_\_\_\_\_\_\_。

1. 表达式sorted([111, 2, 33], key=lambda x: -len(str(x)))的值为\_[111,33,2]\_\_。
2. 表达式max([111, 22, 3], key=str)的值为\_\_\_\_3\_\_\_\_\_\_。
3. 语句句x = (3,)执⾏行行后x的值为\_\_(3,)\_\_\_\_\_\_\_\_。
4. 语句句x = (3)执⾏行行后x的值为\_\_\_\_\_3\_\_\_\_\_。
5. 已知x=3和y=5，执⾏行行语句句 x, y = y, x 后x的值是\_\_\_\_5\_\_\_\_。
6. 可以使⽤用内置函数\_\_\_globals()\_\_\_\_\_\_\_查看包含当前作⽤用域内所有全局变量量和值的字典。
7. locals():…当前作用域内所有局部变量和值的字典
8. 字典中多个元素之间使⽤用\_\_\_\_,\_\_\_\_\_\_分隔开，每个元素的“键”与“值”之间使⽤用 \_\_\_:\_\_\_\_\_\_\_分隔开。
9. 字典对象的\_\_\_\_get()\_\_\_\_\_\_⽅方法可以获取指定“键”对应的“值”，并且可以在指定“键”不不存在的时候返回指定值，如果不不指定则返回None。
10. 表达式{1:'a',2:'b',3:'c'}.get[4:’d’]的值为\_\_\_\_None\_\_\_\_\_\_。
11. 已知x={1:2},那么执⾏行行语句句x[2]=3之后，x的值为\_\_\_\_{1:2,2:3}\_\_\_\_\_\_
12. 表达式{1,2,3,4}-{3,4,5,6}的值为\_\_\_\_{1,2}\_\_\_\_\_\_。
13. 表达式set([1,1,2,3])的值为\_\_\_{1,2,3}\_\_\_\_\_\_\_。
14. 使⽤用列列表推导式得到100以内所有能被13整除的整数的代码可以写作

\_\_\_\_arr=[I for I in range(1,101) if (i%13)==0]\_\_\_\_\_\_。

1. 已知x=[3,5,7],那么表达式x[10:]的值为\_ []\_\_\_\_\_\_\_\_\_。
2. 已知x=[3,5,7],那么执⾏行行语句句x[len(x):]=[1,2]之后，x的值为\_[3,5,7,1,2]\_。
3. 已知x=[3,7,5]，那么执⾏行行语句句x.sort(reverse=True)之后，x的值为 \_\_\_\_\_[7,5,3]\_\_\_\_\_[降序排列]。
4. 已知x=[3,7,5]，那么执⾏行语句

x.sort(key=lambda x:len(str(x)),reverse=True)之后，x的值为\_\_\_\_\_\_\_\_\_\_。

1. 表达式list(zip([1,2],[3,4]))的值为\_\_[(1,3),(2,4)]\_\_\_。
2. 已知x=[1,2,3,2,3]，执⾏行行语句句x.pop()已知之后，x的值为\_\_[1,2,3,2]\_\_。
3. 已知x=[1,2,3,2,3]，执⾏行行语句句x.pop(0)已知之后，x的值为\_[2,3,2,3]\_\_。
4. 表达式list(map(list,zip(\*[[1,2,3],[4,5,6]])))的值为\_ [[1, 4], [2, 5], [3, 6]]\_\_\_。
5. 表达式[x for x in [1,2,3,4,5] if x<3]的值为\_\_\_[1,2]\_\_\_\_。
6. 表达式[x for x in [1,2,3,4,5] if x%2==1]的值为\_\_\_[3,5]\_\_\_\_。
7. 表达式[index for index,value in enumerate([3,5,7,3,7]) if value==max([3,5,7,3,7])]的值为\_\_\_\_[2,4]\_\_\_\_\_\_。
8. 已知x=[3,5,3,7],那么表达式[x.index(i) for i in x if i==3]的值为\_\_\_[0,0]\_\_\_\_\_\_\_。
9. 已知列列表x=[1,2],那么表达式list(enumerate(x))的值为\_\_\_[(0,1),(1,2)]\_\_\_\_\_\_\_。
10. 已知vec=[[1,2],[3,4]]，则表达式[col for row in vec for col in row]的值为

\_\_\_\_\_\_\_\_\_\_。

1. 已知vec=[[1,2],[3,4]]，则表达式[[row[i] for row in vec] for i in range(len(vec[0]))]的值为\_\_\_\_\_\_\_\_\_\_。
2. 已知x=list(range(10)),则表达式x[-4:]的值为\_\_\_[6,7,8,9]\_\_\_\_\_\_\_。
3. 已知x=[3,5,7]，那么执⾏语句x[1:]=[2]之后，x的值为\_\_\_[3,2]\_\_\_\_。
4. 已知x=[3,5,7]，那么执⾏语句句x[:3]=[2]之后，x的值为\_\_\_[2]\_\_\_\_\_\_\_。
5. 已知x为非空列表，那么执⾏语句y=x[:]之后，id(x[0])==id(y[0])的值为 \_\_\_\_True （浅拷贝）\_\_\_\_\_\_。
6. 已知x=[1,2,3,2,3],执行语句x.remove(2)之后，x的值为\_\_[1,3,2,3]\_\_\_\_\_\_\_\_。
7. 表达式len([i+5 for i in range(10)])的值为\_\_\_10\_\_\_\_\_\_\_。
8. 表达式len(range(1,10))的值为\_\_\_9\_\_\_\_。
9. 表达式range(10)[-1]的值为\_\_\_\_9\_\_\_\_。
10. range(10,20)[4]的值为\_\_\_\_\_\_14\_\_\_\_。
11. 表达式round(3.4)的值为\_\_\_\_\_3\_\_\_\_\_。
12. 表达式round(3.7)的值为\_\_\_\_\_4\_\_\_\_\_。
13. 已知x=(3),那么表达式x\*3的值为\_\_\_9\_\_\_\_\_。
14. 已知x=(3,),那么表达式x\*3的值为\_\_\_(3,3,3)\_\_\_。
15. 假设列列表对象x=[1,1,1],那么表达式id(x[0])==id(x[2])的值为\_\_\_True\_\_\_。
16. 已知列列表x=list(range(10)),那么执⾏行行语句句del x[::2]之后，x的值为

\_\_\_\_\_\_\_\_\_\_。

1. 已知列列表x=[1,2,3,4],那么执⾏行行语句句del x[1]之后，x的值为\_【1，3，4】\_\_。
2. 表达式[1,2]\*2的值为\_\_\_1,2,1,2\_\_\_\_\_\_\_。
3. 已知列列表x=[1,2,3],那么执⾏行行语句句x.insert(1,4)之后，x的值为\_\_\_\_\_1,4,2,3\_\_\_\_\_。
4. 已知列列表x＝[1,2,3]，那么执⾏行行语句句x.insert(0,4）之后，x的值为\_\_\_\_\_4,1,2,3\_\_\_\_\_。
5. 已知x=[[1]]\*3，那么执⾏行行语句句x[0][0]=5之后，变量量x的值为\_\_\_\_\_[[5],[1],[1]]\_\_\_\_\_。
6. 已知x=[1,2,3]，那么执⾏行行语句句x[len(x)-1:]=[4,5,6]之后，变量量x的值为

\_\_\_\_\_\_[1,2,4,5,6]\_\_\_\_。

1. 已知x是一个列表对象，那么执行语句y=x[:]之后，表达式id(x)==id(y)的值为

\_\_\_\_True\_\_\_\_\_\_。

1. 表达式sorted([13,1,237,89,100],key=Iambda x:Ien(str(x)))的值为 \_\_\_\_\_\_\_\_\_\_。
2. 已知x={1:2,2:3}，那么表达式x.get(3,4)的值为\_\_\_\_\_\_\_\_\_\_。
3. 已知x={1:2,2:3}，那么表达式x.get(2,4)的值为\_\_\_\_\_\_\_\_\_\_。
4. 表达式{1,2,3}|{3,4,5}的值为
5. 表达式{1,2,3}&{3,4,5}的值为\_\_\_\_\_\_\_\_\_\_。
6. 表达式{1,2,3}-{3,4,5}的值为\_\_\_\_\_\_\_\_\_\_。
7. 表达式{1,2,3}<{3,4,5}的值为\_\_\_\_\_\_\_\_\_\_。
8. 表达式{1,2,3}<{1,2,4}的值为\_\_\_\_\_\_\_\_\_\_。
9. 表达式[1,2,3].count(4)的值为\_\_\_\_\_\_\_\_\_\_。
10. Python标准库random中\_\_\_\_\_\_\_\_\_\_函数的作⽤用是从序列列中随机选择1个元素
11. Python标准库random中sample(seq,k）函数从序列列中选择\_\_\_\_\_\_\_\_\_\_ (重复、

不不重复？)的k个元素

1. random模块中\_\_\_\_\_\_\_\_\_\_函数的作⽤用是将列列表中的元素随机乱序。
2. 执⾏行行代码x,y,z=sorted([1,3,2]）之后，变量量y的值为\_\_\_\_\_\_\_\_\_\_。
3. 表达式(1,2,3)+(4,5)的值为\_\_\_\_\_\_\_\_\_\_。
4. 表达式dict(zip([1,2],[3,4]))的值为\_\_\_\_\_\_\_\_\_\_。
5. 语句句x,y,z=[1,2,3]执⾏行行后，变量量y的值为\_\_\_\_\_\_\_\_\_\_。
6. 已知x={1,2,3}，那么执⾏行行语句句x.add(3)之后，x的值为\_\_\_\_\_\_\_\_\_\_。
7. 已知x={1:1}，那么执⾏行行语句句x[2]=4之后，Ien(x）的值为\_\_\_\_\_\_\_\_\_\_。
8. 已知x={1:1,2:2}，那么执⾏行行语句句x[2]=4之后，len(x)的值为\_\_\_\_\_\_\_\_\_\_。 152. 已知列列表x=[1,2]，那么连续执⾏行行命令y=x和y.append(3）之后，x的值为 \_\_\_\_\_\_\_\_\_\_。
9. 已知列列表x=[1,2]，那么连续执⾏行行命令v=xl:l和v.append(3）之后，x的值为

\_\_\_\_\_\_\_\_\_\_。

1. 已知列列表x=[1,2]，执⾏行行语句句y=x[:]后，表达式id(x)==id(y）的值为\_\_\_\_\_\_\_\_\_\_
2. 已知列列表x=[1,2]，执⾏行行语句句y=x后，表达式id(x)==id(y）的值为\_\_\_\_\_\_\_\_\_\_。
3. 已知列列表x=[1,2]，执⾏行行语句句y=x后，表达式x is y的值为\_\_\_\_\_\_\_\_\_\_。
4. 已知列列表x=[1,2]，执⾏行行语句句y=x[:]后，表达式x is not y的值为\_\_\_\_\_\_\_\_\_\_。
5. 表达式[I for I in range(10) if i>8]的值为\_\_\_\_\_\_\_\_\_\_。
6. 已知x=[[1,2,3],[4,5,6]]，那么表达式[[ row[i] for row in x] for I in range(len (x[0]))] 值为\_\_\_\_\_\_\_\_\_\_。
7. 执⾏行行语句句x,y,z=map(str,range(3))之后，变量量y的值为\_\_\_\_\_\_\_\_\_\_。
8. 已知列列表x=[1,2]，那么执⾏行行语句句x.extend([3])之后，x的值为\_\_\_\_\_\_\_\_\_\_。
9. 已知列列表x=[1,2]，那么执⾏行行语句句x.append([3])之后，x的值为\_\_\_\_\_\_\_\_\_\_。
10. 执⾏行行语句句x,y,z='123'之后，y的值为\_\_\_\_\_\_\_\_\_\_。
11. 表达式print(0b10101）的值为\_\_\_\_\_\_\_\_\_\_。
12. 已知x=[1,2,3,4,5]，那么执⾏行行语句句del x[:3]之后，x的值为\_\_\_\_\_\_\_\_\_\_。
13. 已知x=range(1,4）和y=range(4,7)，那么表达式sum([i\*j for i,j in zip(x,y)])的值为

\_\_\_\_\_\_\_\_\_\_。

1. 表达式[5 for I in range(3)]的值为\_\_\_\_\_\_\_\_\_\_。
2. 表达式{1,2,3}=={1,3,2}的值为\_\_\_\_\_\_\_\_\_\_。
3. 表达式[1,2,3]==[1,3,2]的值为 \_\_\_\_\_\_\_\_\_\_。
4. 已知x=[1,2,1].)，那么表达式id(x[0])==id(x[2)的值为\_\_\_\_\_\_\_\_\_\_。
5. 表达式3 not in[1,2,3]的植为\_\_\_\_\_\_\_\_\_\_。
6. 已知x=[1,2]，那么执⾏行行语句句x[0:0]＝[3,3］之后，x的值为\_\_\_\_\_\_\_\_\_\_。
7. 已知x=[1,2]．那么执⾏行行语句句x[0:1]＝[3,3］之后，x的值为\_\_\_\_\_\_\_\_\_\_。
8. 已知x=[1,2,3,4,5]，那么执⾏行行语句句del x[1:3]之后，x的值为\_\_\_\_\_\_\_\_\_\_。
9. 已知x=[[1,2,3],[4,5,6]]．那么表达式sum([i\*j for i,j in zip(\*x)])的值为

\_\_\_\_\_\_\_\_\_\_。

1. 已知列列表x=[1,2,3]和y=[4,5,6]，那么表达式［(i,j) for i,j in zip(x,y) if i==3]的值为

\_\_\_\_\_\_\_\_\_\_。

1. 已知列列表x=[1.0,2.0,3.0］，那么表达式sum(x)/len(x)的值为\_\_\_\_\_\_\_\_\_\_。
2. 已知x={1:2,2:3,3:4}，那么表达式sum(x)的值为\_\_\_\_\_\_\_\_\_\_。
3. 已知x={1:2,2:3,3:4}．那么表达式sum(x.values())的值为\_\_\_\_\_\_\_\_\_\_。
4. 已知x=[3,2,3,3,4]，那么表达式[index for index,value in enumerate(x) if value==3]的值为\_\_\_\_\_\_\_\_\_\_。
5. 表达式1234%1000//100的值为\_\_\_\_\_\_\_\_\_\_。
6. 表达式3//5的值为\_\_\_\_\_\_\_\_\_\_。
7. 表达式[1,2]＋[3]的值为\_\_\_\_\_\_\_\_\_\_。
8. 表达式(1,)+(2,)的值为\_\_\_\_\_\_\_\_\_\_。
9. 表达式(1)＋(2)的值为\_\_\_\_\_\_\_\_\_\_。
10. 已知x,y=map(int,['1",'2"]),那么表达式x+y的值为\_\_\_\_\_\_\_\_\_\_。
11. 已知x,y=map(str,range(1,3))．那么表达式x+y的值为\_\_\_\_\_\_\_\_\_\_。
12. 已知列列表x=list(range(5)),那么执⾏行行语句句x.remove(3)之后，表达式x.index(4)的值为\_\_\_\_\_\_\_\_\_\_。
13. 已知列列表x＝[1,3,2]，那么执⾏行行语句句x.reverse()之后，x的值为\_\_\_\_\_\_\_\_\_\_。

190 . 已知列列表x=[1,3,2]，那么执⾏行行语句句x=x.reverse()之后，x的值为\_\_\_\_\_\_\_\_\_\_。

1. 已知x为⾮非空列列表，那么表达式x.reverse()==Iist(reversed(x))的值为

\_\_\_\_\_\_\_\_\_\_。

1. 已知x为⾮非空列列表，那么表达式x.sort()==sorted(x)的值为\_\_\_\_\_\_\_\_\_\_。
2. 已知列列表x＝[1,3,2]．那么执⾏行行语句句y=Iist(reversed(x))之后，x的值为 \_\_\_\_\_\_\_\_\_\_。
3. 已知列列表x＝[1,3,2]，那么执⾏行行语句句y=Iist(reversed(x))之后，x的值为

\_\_\_\_\_\_\_\_\_\_。

1. 已知列列表x中包含超过5个以上的元素，那么表达式x==x[:5]＋x[5:]的值为

\_\_\_\_\_\_\_\_\_\_。

1. 已知列列表x=[1,3,2],那么表达式[value for index,value in enumerate(x) if index==2]的值为\_\_\_\_\_\_\_\_\_\_。
2. 已知l列列表x=[1,3,2]．那么执⾏行行语句句a,b,c=sorted(x)之后，b的值为\_\_\_\_\_\_\_\_\_\_
3. 已知列列表x=[1,3,2]，那么执⾏行行语句句a,b,c=map(str,sorted(x))之后，c的值为

\_\_\_\_\_\_\_\_\_\_。

1. 表达式set([1,2,3])=={1,2,3}的值为\_\_\_\_\_\_\_\_\_\_。
2. 表达式set([1,2,2,3])=={i,2,3}的值为\_\_\_\_\_\_\_\_\_\_。
3. 表达式'%c'%65==Str(65)的值为\_\_\_\_\_\_\_\_\_\_。
4. 表达式'%s'%65==str(65)的值为\_\_\_\_\_\_\_\_\_\_。
5. 表达式chr(ord('b")^32)的值为\_\_\_\_\_\_\_\_\_\_。
6. 表达式'abc'in'abdcefg'的值为\_\_\_\_\_\_\_\_\_\_。
7. 已知x='abcd'和y='abcde'，那么表达式[i==j for i,j in zip(x,y)]的值为

\_\_\_\_\_\_\_\_\_\_。

1. 已知x=Iist(range(20))。那么表达式x[-1]的值为\_\_\_\_\_\_\_\_\_\_。
2. 已知x=3+4j和v=5+6j，那么表达式x+y的值为\_\_\_\_\_\_\_\_\_\_。
3. 表达式int('11',2)的值为\_\_\_\_\_\_\_\_\_\_。
4. 表达式chr(ord('A')+1)的值为\_\_\_\_\_\_\_\_\_\_。
5. 表达式int(str(34))==34的值为\_\_\_\_\_\_\_\_\_\_。
6. 表达式Iist(str([3,4]))==[3,4]的值为\_\_\_\_\_\_\_\_\_\_。
7. 表达式15//4的值为\_\_\_\_\_\_\_\_\_\_。
8. 表达式sorted({'a':3,'b':9,'c':75})的值为\_\_\_\_\_\_\_\_\_\_。
9. 表达式sorted({'a':3,'b':9,'c':75}.vaIues())的值为\_\_\_\_\_\_\_\_\_\_。
10. 已知x=[3,2,4,1]．那么执⾏行行语句句x=x.sort()之后，x的值为\_\_\_\_\_\_\_\_\_\_。
11. 已知x=list(range(20))，那么语句句print(x[100:200])的输出结果为\_\_\_\_\_\_\_\_\_\_。
12. 已知x=Iist(range(20))，那么执⾏行行语句句x[:18]=[]后列列表x的值为\_\_\_\_\_\_\_\_\_\_。
13. 已知x=([1],[2])，那么执⾏行行语句句x[0].append(3)后x的值为\_\_\_\_\_\_\_\_\_\_。
14. 已知x={1:1,2:2}，那么执⾏行行语句句x.update({2:3,3:3})之后，表达式 sorted(x.items())的值为\_\_\_\_\_\_\_\_\_\_。
15. 已知x={1:1,2:2}，那么执⾏行行语句句x[3]=3之后，表达式sorted(x.items())的值为

\_\_\_\_\_\_\_\_\_\_。

1. 已知x=[1,2,3]，那么表达式not(set(x\*100)-set(x))的值为\_\_\_\_\_\_\_\_\_\_。
2. 表达式[1,2,3]>[1,3,2]的值为＿\_\_\_\_\_\_\_\_\_\_。
3. 已知x=[1,2,3]，那么表达式set(x\*100)==set(x)的值为\_\_\_True\_\_\_\_。
4. 已知x=[1,2,3,4,5]，那么执⾏行行语句句x[::2]=range(3)之后，x的值为\_\_\_\_\_\_\_\_\_\_。
5. 已知x＝[1,2,3,4,5]，那么执⾏行行语句句x[1::2]=sorted(x[i::2]，reverse=True)之后，x

的值为\_\_\_\_\_\_\_\_\_\_。

1. 表达式type({})＝=dict的值为\_\_\_\_True\_\_\_\_\_\_。
2. 表达式type({})==Set的值为\_\_\_\_\_False\_\_\_\_\_。
3. Python提供了了两种基本的循环结构：\_\_\_\_while\_\_\_\_\_\_\_和\_\_\_\_\_for\_\_\_\_\_\_。

# 判断题

1. Python是⼀一种跨平台、开源、免费的⾼高级动态编程语⾔言。
2. Python 3.x完全兼容Python 2.x。
3. 在Windows平台上编写的Python程序⽆无法在Unix平台运⾏行行。
4. 不不可以在同⼀一台计算机上安装多个Python版本。
5. pip命令⽀支持使⽤用扩展名为whl的⽂文件离线安装Python扩展库。
6. 下载whl⽂文件进⾏行行离线安装扩展库时，因为whl⽂文件的名字都⽐比较⻓长，可以改成短⼀一些的名字再使⽤用pip进⾏行行安装。
7. ⽤用来安装Python扩展库的pip命令应该在命令提示符环境下运⾏行行，如果安装了了多个版本的Python，最好切换⾄至相应版本的Python安装⽬目录下运⾏行行。
8. 已知x = 3，那么赋值语句句x = 'abcedfg'是⽆无法正常执⾏行行的。
9. Python变量量使⽤用前必须先声明，并且⼀一旦声明就不不能在当前作⽤用域内改变其类型了了。
10. Python不不允许使⽤用关键字作为变量量名，但是允许使⽤用内置函数名作为变量量名，不不过这会改变函数名的含义，所以不不建议这样做。
11. 在Python中可以使⽤用if作为变量量名。
12. 在Python 3.x中可以使⽤用中⽂文作为变量量名。
13. Python变量量名必须以字⺟母或下划线开头，并且区分字⺟母⼤大⼩小写。
14. 加法运算符可以⽤用来连接字符串串并⽣生成新字符串串。
15. x = 9999\*\*9999这样的语句句在Python中⽆无法运⾏行行，因为数字太⼤大了了超出了了整型变量量的表示范围。
16. 3+4j不不是合法的Python表达式。
17. 0o12f是合法的⼋八进制数字。
18. 只有Python扩展库才需要导⼊入以后才能使⽤用其中的对象，Python标准库不不需要导⼊入即可使⽤用其中的所有对象。
19. 在Python中0xad是合法的⼗十六进制数字表示形式。
20. 4j 是合法Python数字类型。
21. Python使⽤用缩进来体现代码之间的逻辑关系，对缩进的要求⾮非常严格。
22. Python代码的注释只有⼀一种⽅方式，那就是使⽤用#符号。
23. 放在⼀一对三引号之间的任何内容将被认为是注释。
24. 尽管可以使⽤用import语句句⼀一次导⼊入任意多个标准库或扩展库，但是仍建议每次只导⼊入⼀一个标准库或扩展库。
25. 为了了让代码更更加紧凑，编写Python程序时应尽量量避免加⼊入空格和空⾏行行。
26. 在Python 3.5中运算符+不不仅可以实现数值的相加、字符串串连接，还可以实现列列表、元组的连接和集合的并集运算。
27. 在Python中可以使⽤用 for 作为变量量名。
28. 在Python中可以使⽤用 id 作为变量量名，但是不不建议这样做。
29. ⼀一个数字5也是合法的Python表达式。
30. 执⾏行行语句句from math import sin之后，可以直接使⽤用sin()函数，例例如 sin(3)。
31. ⼀一般来说，Python扩展库没有通⽤用于所有版本Python的，安装时应选择与已安装Python的版本对应的扩展库。
32. Python变量量名区分⼤大⼩小写，所以student和Student不不是同⼀一个变量量。
33. 在Python 3.x中reduce()是内置函数。
34. 如果只需要math模块中的sin()函数，建议使⽤用from math import sin来导⼊入，⽽而不不要使⽤用import math导⼊入整个模块。
35. 表达式pow(3, 2) == 3 \*\* 2的值为True。
36. 已知x = 3，那么执⾏行行语句句x+=6之后，x的内存地址不不变。
37. 安装Python扩展库时只能使⽤用pip⼯工具在线安装，如果安装不不成功就没有别的办法了了。
38. 列列表、元组、字符串串属于有序序列列，⽽而字典和集合属于⽆无序序列列。 39、同⼀一个集合中的元素都是唯⼀一的，不不会存在重复的元素。

40、同⼀一个列列表中的元素都是唯⼀一的，不不允许存在相同的元素。

41、列列表、元组、字符串串⽀支持双向索引，-1表示最后⼀一个元素的下标。

42、集合⽀支持双向索引，-1表示最后⼀一个元素的下标。

43、python⽀支持使⽤用字典的“键”作为下标来访问字典中的值。

44、列列表可以作为字典的“键”。

45、元组可以作为字典的“键”。

46、字典的“键”必须是不不可变的。

47、对于关键字in⽽而⾔言，集合的测试速度⽐比列列表快很多。

48、已知x为⾮非空列列表，那么表达式sorted(x,reverse=True)==list(reversed(x))的值⼀一定是True。

49、已知x为⾮非空列列表，那么x.sort(reverse=True)和x.reverse()的作⽤用是等价的。

50、⽣生成器器推导式⽐比列列表推导式具有更更⾼高的效率，推荐使⽤用。

51、python集合可以包含相同的元素。

52、python字典中的“键”不不允许重复。

53、python字典中的“值”不不允许重复。

54、python集合中的元素可以是元组。

55、python集合中的元素可以是列列表。

56、已知A和B是两个集合，并且表达式A<B的值为False，那么表达式A>B的值⼀一定为

True。

57、列列表对象的append()⽅方法属于原地操作，⽤用于在列列表尾部追加⼀一个元素。 58、对于列列表⽽而⾔言，在尾部追加元素⽐比在中间位置插⼊入元素速度更更快⼀一些，尤其是对于包含⼤大量量元素的列列表。

59、假设有⾮非空列列表x，那么x.append(3)、x=x+[3]与x.insert(0,3)在执⾏行行时间上基本没有太⼤大区别。

60、使⽤用python列列表的⽅方法inset()为列列表插⼊入元素时会改变列列表中插⼊入位置之后元素

的索引。

61、假设x为列列表对象，那么x.pop()和x.pop(-1)的作⽤用是⼀一样的。

62、使⽤用del命令或者列列表对象的remove()⽅方法删除列列表中⾮非尾部元素时会影响列列表中部分元素的索引。

63、使⽤用列列表对象的remove()⽅方法可以删除列列表中⾸首次出现的指定元素，如果列列中不不存在要删除的指定元素则抛出异常。

64、元组是不不可变的，不不⽀支持列列表对象的insert()、remove()等⽅方法，也不不⽀支持del命令删除其中的元素，但可以使⽤用del命令删除整个元组对象。 65、⽆无法删除集合中指定位置的元素，只能删除特定值的元素。

66、元组的访问速度⽐比列列表要快⼀一些，如果定义了了⼀一系列列常量量值，并且主要⽤用途仅仅是对其遍历⽽而不不需要进⾏行行任何修改，建议使⽤用元组⽽而不不使⽤用列列表。

67、当以指定”键“为下标给字典对象赋值时，若该”键“存在则表示修改该”键“对应的”值

“，若不不存在则表示为字典对象添加⼀一个新的”键：值对“。

68、假设x是含有5个元素的列列表，那么切⽚片操作x[10:]是⽆无法执⾏行行的，会抛出异常。 69、假设x是含有5个元素的列列表，那么使⽤用print(x[10])是⽆无法执⾏行行的，会抛出异常。

70、只能对列列表进⾏行行切⽚片操作，不不能对元组和字符串串进⾏行行切⽚片操作。

71、只能通过切⽚片访问列列表中的元素，不不能使⽤用切⽚片修改列列表中的元素。

72、只能通过切⽚片访问元组中的元素，不不能使⽤用切⽚片修改元组中的元素。

73、python集合不不⽀支持使⽤用下标访问其中的元素。

74、已知列列表x中包含超过5个以上的元素，那么语句句x=x[5:]+x[:5]可以实现将列列表x中的元素循环左移5位。

75、对于⽣生成器器对象x=(3 for i in range(5))，连续两次执⾏行行list(x)的结果是⼀一样的。

76、对于⼤大量量列列表的连接，extend()⽅方法⽐比运算符+具有更更⾼高的效率。

77、表达式{1,3,2}>{1,2,3}的值为True。

78、列列表对象的extend()⽅方法属于原地操作，调⽤用前后列列表对象的地址不不变。

79、对于数字n，如果表达式0 not in [n%d for d in range(2,n]的值为True，则说明n是素数。

80、表达式list('[1,2,3]')的值是[1 , 2,3]. 81、已知x为⾮非空列列表，那么执⾏行行语句句x[0]=3之后，列列表对象x的内存地址不不变。 82、列列表对象的pop()⽅方法默认删除并返回最后⼀一个元素，如果列列表已空则抛出异常。

83、表达式{1,2}\*2的值为{1,2,1,2}。

84、假设random模块已导⼊入，那么表达式random.sample(range(10),20)的作⽤用是⽣生成20个不不重复的整数。

85、假设random模块已导⼊入，那么表达式random.sample(range(10),7)的作⽤用是⽣生成 7个不不重复的整数。

86、使⽤用random模块的函数randint(1,100)获取随机数时，有可能会得到100。

87、已知x=[1,2,3,4]，那么执⾏行行x[0]=5之后，x的值为(5,2,3,4)。

88、已知x=3，那么执⾏行行x+=6语句句前后x的内存地址是不不变的。

89、内置函数len()返回指定序列列的元素个数，适⽤用于列列表、元组、字符串串、字典、集合以及range、zip等迭代对象。

90、已知x和y是两个等⻓长的整数列列表，那么表达式sum((i\*j for i,j in zip(x,y)))的作⽤用是

计算这两个列列表所表示的向量量的内积。

91、已知x和y是两个等⻓长的整数列列表，那么表达式[i+j for i,j in zip(x,y)]的作⽤用是计算这

两个列列表所表示的向量量的和。

92、表达式int['1'\*64,2]与sum(2\*\*i for i in range(64))的计算结果是⼀一样的，但是前者更更快⼀一些。

93、已知x=list(range(20))，那么语句句del x[::2]可以正常执⾏行行。

94、已知x=list(range(20))，那么语句句x[::2]=[]可以正常执⾏行行。

95、已知x=list(range(20))，那么语句句print(x[100:200])⽆无法正常执⾏行行。

96、已知x是个列列表对象，那么执⾏行行语句句y=x之后，对y所做的任何操作都会同样作⽤用到 x上。

97、已知x是个列列表对象，那么执⾏行行语句句y=x[:]之后，对y所做的任何操作都会同样作⽤用到x上。

98、在python中，变量量不不直接存储值，⽽而是存储值的引⽤用，也就是值在内存中的地址。

99、表达式(i\*\*2 for i in range(100))的结果是个元组。

100、在python中元组的值是不不可变的，因此，已知x=([1],[2])，那么语句句

x[0].append(3)是⽆无法正常执⾏行行的。

101、已知x={1:1,2:2}，那么语句句x[3]=3⽆无法正常执⾏行行。

102、已知列列表x=[1,2,3,4]，那么表达式x.find(5)的值应为-1。

103、列列表对象的排序⽅方法sort()只能按元素从⼩小到⼤大排列列，不不⽀支持别的排序⽅方式。

104、表达式‘a’+1的值为‘b’

105、创建只包含⼀一个元素的数组时，必须在元素后⾯面加⼀一个逗号，例例如(3 , )

# 问答题

1. Python程序的\_\_name\_\_的作⽤用是什什么？
2. 为什什么应尽量量从列列表的尾部进⾏行行元素的增加与删除操作？
3. 简单解释Python基于值的内存管理理模式。
4. 简单解释运算符/和//的区别。 5. 分析逻辑运算符“or”的短路路求值特性。

# 编程题

1. 编写程序，⽣生成包含1000个0到100之间的随机整数，并统计每个元素的出现次

数。

1. 编写程序，⽤用户输⼊入⼀一个列列表和2个整数作为下标，然后使⽤用切⽚片获取并输出列列表中介于2个下标之间的元素组成的⼦子列列表。例例如⽤用户输⼊入[1, 2, 3, 4, 5, 6]和2,5，程序输出[3, 4, 5, 6]。
2. 设计⼀一个字典，并编写程序，⽤用户输⼊入内容作为“键”，然后输出字典中对应的

“值”，如果⽤用户输⼊入的“键”不不存在，则输出“您输⼊入的键不不存在！”

1. 编写程序，⽣生成包含20个随机数的列列表，然后将前10个元素升序排列列，后10个元素降序排列列，并输出结果。
2. 编写程序，运⾏行行后⽤用户输⼊入4位整数作为年年份，判断其是否为闰年年。如果年年份能被

400整除，则为闰年年；如果年年份能被4整除但不不能被100整除也为闰年年。

1. 编写程序，⽣生成⼀一个包含50个随机整数的列列表，然后删除其中所有奇数。（提示：从后向前删）
2. 编写程序，⽣生成⼀一个包含20个随机整数的列列表，然后对其中偶数下标的元素进⾏行行降序排列列，奇数下标的元素不不变。（提示：使⽤用切⽚片）
3. 编写程序，⽤用户从键盘输⼊入⼩小于1000的整数，对其进⾏行行因式分解。例例如，10=2\*5，

60=2\*2\*3\*5。

1. 编写程序，⾄至少使⽤用2种不不同的⽅方法计算100以内所有奇数的和。
2. 编写程序，输出所有由1，2，3，4这四个数字组成的素数，并且在每个素数中每个数字只使⽤用⼀一次。
3. 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 | 3 x -5 |
| 10<=x<20 | 0.5x-2 |
| 20<=x | 0 |

1. 编写程序，输⼊入任意⼤大的⾃自然数，输出各位数字之和。
2. 编写程序，输⼊入两个集合setA和setB，分别输出它们的交集、并集和差集。
3. 编写程序，输⼊入⼀一个⾃自然数，输出它的⼆二进制、⼋八进制、⼗十六进制表示形式。 15. 编写程序，输⼊入⼀一个包含若⼲干整数的列列表，输出⼀一个新列列表，要求新列列表中只包含原列列表中的偶数。
4. 编写程序，输⼊入两个分别包含若⼲干整数的列列表lstA和lstB，输出⼀一个字典，要求使⽤用列列表lstA中的元素作为键，列列表lstB中的元素作为值，并且最终字典中的元素数量量取决于lstA和lstB中元素最少的列列表的数量量。
5. 编写程序，输⼊入⼀一个包含若⼲干整数的列列表，输出新列列表，要求新列列表中的所有元素来⾃自于输⼊入的列列表，并且降序排列列。
6. 编写程序，输⼊入⼀一个包含若⼲干整数的列列表，输出列列表中所有整数连乘的结果。 19. 编写程序，输⼊入两个各包含2个整数的列列表，分别表示城市中两个地点的坐标，输出两点之间的曼哈顿距离。

20. 编写程序，输⼊入包含若⼲干集合的列列表，输出这些集合的并集。要求使⽤用reduce()函数和lambda表达式完成。