《Python语言程序设计》2020第三次平时作业

平时作业说明：

（1）平时作业只需提供答案，不用抄题。

（2）平时作业答案可打印，也可手写。特别提醒：需写上姓名和学号。

（3）所有平时作业答案都等到期末与课程设计报告一起提交。具体提交办法期末时公布。

知识点：字符串与正则表达式、列表与元组、字典与集合

一、程序填空题

（为便于阅卷，不用抄题，只提交各小题答案）

1. 下面程序的功能是从列表中筛选出字符串元素，并将它们连接为一个字符串。程序运行后的输出情况如下图所示。请在横线处填写适当的语句或表达式将程序补充完整。



源程序：

s=''

t=[123,'AAA',3.14,'uestc',True,'BBB',456]

for i in t:

if \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:

s+=i

else:

print(s)

2. 下面程序的功能是从键盘读入一个字符串，在文本块中查找该字符串，并输出它在文本块中的位置（用索引下标表示）。程序运行后的输入输出情况如下图所示。请在横线处填写适当的语句或表达式将程序补充完整。



源程序：

x=input('x=')

s='xyzabcd'\*2

t=[]

for i in range(len(s)-len(x)+1):

if x==\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:

t+=[i]

else:

print(t)

3. 下面程序的功能是将字符串中的子串“bcd”全部替换为字符串“AA”。程序运行后的输入输出情况如下图所示。请在横线处填写适当的语句或表达式将程序补充完整。



源程序：

x='bcd'

t='AA'

s='xyzabcd'\*2

for i in range(len(s)-len(x)+1):

if x==s[i:i+len(x)]:

s=s[:i]+t+\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

else:

print(s)

4. 下面程序的功能是从键盘读入一个字符串，判断其是否是回文。程序运行后的输入输出情况如下图所示。请在横线处填写适当的语句或表达式将程序补充完整。注：回文是指一个字符串从左向右读与从右向左读相同。例如：noon，level，rotor，boob。



源程序：

s=input('s=')

b=True

for i in range(len(s)//2):

if s[i]!= \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:

b=False

break

if b:

print('Yes')

else:

print('No')

5. 下面程序的功能是从键盘读入一个由阿拉伯数字组成的字符串（中间可以包含一个小数点），如果它是正整数，就删去其左端的所有0字符，如果它是正小数，就删去其两端的所有0字符，最后再将它转换为数值型数据（int或float）。程序运行后的输入输出情况如下图所示。请在横线处填写适当的语句或表达式将程序补充完整。



源程序：

s=input('s=')

if '.' in s:

x=float(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

else:

x=int(s.lstrip('0'))

print(x)

6. 下面程序的功能是从键盘读入一个英文句子，将其转换为标题格式，即每个单词的首字母大写。程序运行后的输入输出情况如下图所示。请在横线处填写适当的语句或表达式将程序补充完整。



源程序：

s=input('s=')

s=s.split()

t=''

for i in s:

if not i.istitle():

i=chr(ord(i[0])-32)+ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

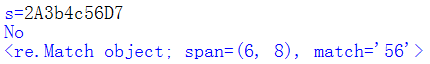
t+=i+' '

else:

t=t.strip()

print(t)

7. 下面程序的功能是从键盘读入一个由英文字母和阿拉伯数字组成的字符串，当该字符串满足英文字母和阿拉伯数字交替出现（即没有连续两个英文字母或连续两个阿拉伯数字出现）时，输出“Yes”，否则输出“No”。程序运行后的输入输出情况如下图所示。请在横线处填写适当的语句或表达式将程序补充完整。





源程序：

import re

s=input('s=')

z=re.search('\d{2,}|[a-z]{2,}',s, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

if(z):

print('No')

else:

print('Yes')

print(z)

8. 下面程序的功能是从键盘读入一个英语单词，判断该单词是否是以“er结尾”。如果该单词是以“er结尾”的，则输出“Yes”，否则输出“No”。程序运行后的输入输出情况如下图所示。请在横线处填写适当的语句或表达式将程序补充完整。



源程序：

import re

s=input('s=')

z=re.search(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_,s,re.I)

if(z):

print('Yes')

else:

print('No')

print(z)

9. 下面程序的功能是从键盘连续读入几个整数，整数之间用逗号分隔，求这些整数之和。程序运行后的输入输出情况如下图所示。请在横线处填写适当的语句或表达式将程序补充完整。



源程序：

import re

x=input('x=')

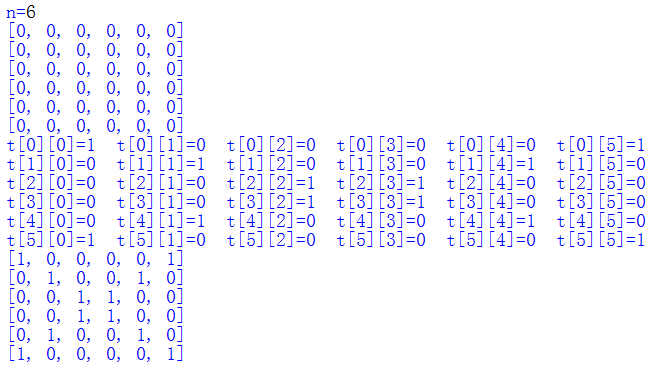
t=re.split(',',x)

y=[\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ for i in t]

s=sum(y)

print('s={}'.format(s))

10. 下面程序的功能是从键盘读入一个正方形矩阵的行数，用列表构造该方阵，将其两条对角线元素设置为1，其余元素设置为0，然后按照矩阵形状输出该矩阵。程序运行后的输入输出情况如下图所示。请在横线处填写适当的语句或表达式将程序补充完整。



源程序：

n=eval(input('n='))

t=[[0]\*n for i in range(n)]

for i in range(n):

print(t[i])

for i in range(n):

t[i][i]=1

t[i][n-i-1]=1

for i in range(n):

for j in range(n):

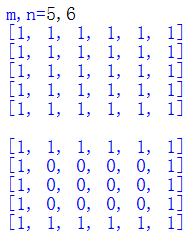
print('t[{}][{}]={:<3}'.format(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_,end='')

print()

for i in range(n):

print(t[i])

11. 下面程序的功能是从键盘读入一个矩阵的行数和列数，用列表构造该矩阵，将其所有靠边的元素设置为1，其余元素设置为0，然后按照矩阵形状输出该矩阵。。程序运行后的输入输出情况如下图所示。请在横线处填写适当的语句或表达式将程序补充完整。



源程序：

m,n=eval(input('m,n='))

t=[[1]\*n for i in range(m)]

for i in range(m):

print(t[i])

for i in range(1,m-1):

for j in range(1, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_):

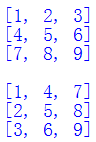
t[i][j]=0

print()

for i in range(m):

print(t[i])

12. 下面程序的功能是将一个矩阵转置，然后再输出它。程序运行后的输入输出情况如下图所示。请在横线处填写适当的语句或表达式将程序补充完整。提示：矩阵转置就是将矩阵中t[i][j]与t[j][i]元素互换。



源程序：

t=[[1,2,3],[4,5,6],[7,8,9]]

for i in range(len(t)):

print(t[i])

for i in range(len(t)):

for j in range(i):

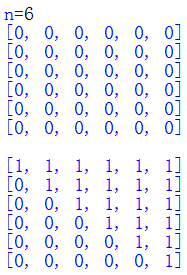
t[i][j],t[j][i]= \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

print()

for i in range(len(t)):

print(t[i])

13. 下面程序的功能是从键盘读入一个正方形矩阵的行数，用列表构造该方阵，将其右上三角元素设置为1，其余元素设置为0，然后按照矩阵形状输出该矩阵。程序运行后的输入输出情况如下图所示。请在横线处填写适当的语句或表达式将程序补充完整。



源程序：

n=eval(input('n='))

t=[[0]\*n for i in range(n)]

for i in range(n):

print(t[i])

for i in range(n):

for j in range(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_,n):

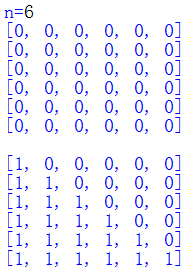
t[i][j]=1

print()

for i in range(n):

print(t[i])

14. 下面程序的功能是从键盘读入一个正方形矩阵的行数，用列表构造该方阵，将其左下三角元素设置为1，其余元素设置为0，然后按照矩阵形状输出该矩阵。程序运行后的输入输出情况如下图所示。请在横线处填写适当的语句或表达式将程序补充完整。



源程序：

n=eval(input('n='))

t=[[0]\*n for i in range(n)]

for i in range(n):

print(t[i])

for i in range(n):

for j in range(n):

if(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_):

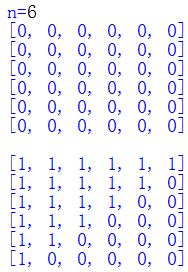
t[i][j]=1

print()

for i in range(n):

print(t[i])

15. 下面程序的功能是从键盘读入一个正方形矩阵的行数，用列表构造该方阵，将其左上三角元素设置为1，其余元素设置为0，然后按照矩阵形状输出该矩阵。程序运行后的输入输出情况如下图所示。请在横线处填写适当的语句或表达式将程序补充完整。



源程序：

n=eval(input('n='))

t=[[0]\*n for i in range(n)]

for i in range(n):

print(t[i])

for i in range(n):

for j in range(n):

if(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_<n):

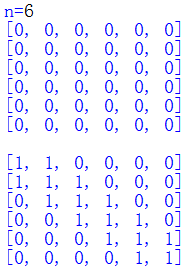
t[i][j]=1

print()

for i in range(n):

print(t[i])

16. 下面程序的功能是从键盘读入一个正方形矩阵的行数，用列表构造该方阵，将其主对角带元素设置为1，其余元素设置为0，然后按照矩阵形状输出该矩阵。程序运行后的输入输出情况如下图所示。请在横线处填写适当的语句或表达式将程序补充完整。



源程序：

n=eval(input('n='))

t=[[0]\*n for i in range(n)]

for i in range(n):

print(t[i])

for i in range(n):

if(i-1>=0):t[i-1][i]=1

t[i][i]=1

if(i+1<n):t[\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_][i]=1

print()

for i in range(n):

print(t[i])

17. 下面程序的功能是从键盘读入一串字符，统计其中各个字符的出现次数，并输出出现频次最高的字符及其出现的次数。程序运行后的输入输出情况如下图所示。请在横线处填写适当的语句或表达式将程序补充完整。



源程序：

s=input('s=')

t=list(s)

d={}.fromkeys(set(t), \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

for i in t:

d[i]=d[i]+1

print(d)

n=max(d.values())

for k,v in d.items():

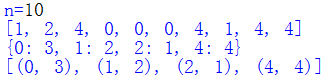
if(v==n):

m=k

break

print("d['{}']={}".format(m,n))

18. 下面程序的功能是从键盘读入正整数n，生成n个在[0,5]范围内的随机数，统计不同数字出现的次数，并按照升序输出之。程序运行后的输入输出情况如下图所示。请在横线处填写适当的语句或表达式将程序补充完整。



源程序：

import random

n=eval(input('n='))

s=[random.randint(0,5) for i in range(n)]

print(s)

t=set(s)

d={}

for i in t:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_=s.count(i)

print(d)

print(sorted(d.items()))

19. 下面程序的功能是从键盘读入一串字符，删去重复字符（即重复字符只保留一个），最后输出之。程序运行后的输入输出情况如下图所示。请在横线处填写适当的语句或表达式将程序补充完整。



源程序：

s=input('s=')

t=list(s)

x=\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

y=''.join(x)

print(y)

20. 下面程序的功能是从键盘读入一串字符，删去重复字符（即重复字符只保留一个），最后按照各个字符首次出现的次序输出之。程序运行后的输入输出情况如下图所示。请在横线处填写适当的语句或表达式将程序补充完整。



源程序：

s=input('s=')

t=list(s)

x=''

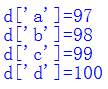
for i in t:

if i not in \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:

x+=i

print(x)

21. 下面程序的功能是将26个英文小写字母及其ASCII码组成一个字典，并输出之。程序运行后的输入输出情况如下图所示（由于篇幅限制，在此只显示前几个词条）。请在横线处填写适当的语句或表达式将程序补充完整。



源程序：

import string

c=string.ascii\_letters

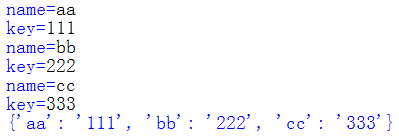
k=[(i, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) for i in c if(i>='a')]

d=dict(k)

for k,v in d.items():

print("d['{}']={}".format(k,v))

22. 下面程序的功能是从键盘读入某系统的5个登录账号和密码，并构建成字典，最后输出之。程序运行后的输入输出情况如下图所示。请在横线处填写适当的语句或表达式将程序补充完整。



源程序：

d=\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

for i in range(3):

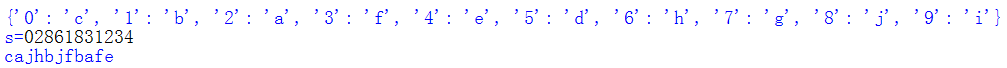
n=input('name=')

k=input('key=')

d[n]=k

print(d)

23. 下面程序的功能是实现数字加密功能。将10个阿拉伯数字分别用10个英文字母表示。读入一个数字字符串，将其转换为密文并输出之。程序运行后的输入输出情况如下图所示。请在横线处填写适当的语句或表达式将程序补充完整。



源程序：

m=list('0123456789')

n=list('cbafedhgji')

d=dict(zip(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_))

print(d)

s=input('s=')

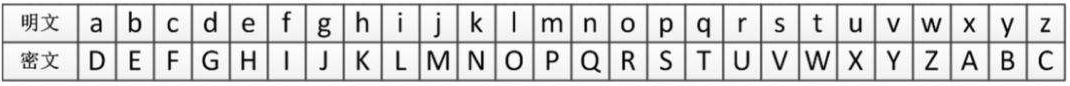
t=[d[i] for i in s]

x=''.join(t)

print(x)

24. 编写一段程序实现下面的功能：从键盘读入秘钥数字和明文，使用凯撒密码算法，将明文加密为密文。

提示：恺撒密码是一种替换加密的技术，明文中的所有字母都在英文字母表上向后（或向前）按照一个固定数目（秘钥）进行偏移后被替换成密文。例如，当偏移量是3（即key=3）的时候，所有的字母A将被替换成D，B变成E，以此类推，W变成Z，X变成A，Y变成B，Z变成C。



例如：明文=hello，则对应的密文=khoor。

25. 编写一段程序实现下面的功能：随机产生一个长度为10的任意字符串（见下面的代码段），将其中的英文字母、阿拉伯数字和其他字符筛选出来各自单独组成一个字符串。例如：

随机字符串是：'a@B23#c&4dE\*?5'，将被拆分为三个字符串：'aBcdE'，'2345'，'@#&\*?'。

import re,string,random

c=string.ascii\_letters+string.digits+string.punctuation

t=[random.choice(c) for i in range(10)]

s=''.join(t)

print(s)

二、单项选择题

说明：为便于阅卷，请按下表格式给出单选题的答案。

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 题号 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 答案 |  |  |  |  |  |  |  |  |  |  |
| 题号 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 答案 |  |  |  |  |  |  |  |  |  |  |
| 题号 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 答案 |  |  |  |  |  |  |  |  |  |  |
| 题号 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 答案 |  |  |  |  |  |  |  |  |  |  |
| 题号 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 |
| 答案 |  |  |  |  |  |  |  |  |  |  |
| 题号 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |
| 答案 |  |  |  |  |  |  |  |  |  |  |
| 题号 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 |
| 答案 |  |  |  |  |  |  |  |  |  |  |
| 题号 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |
| 答案 |  |  |  |  |  |  |  |  |  |  |
| 题号 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 |
| 答案 |  |  |  |  |  |  |  |  |  |  |
| 题号 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 |
| 答案 |  |  |  |  |  |  |  |  |  |  |

单项选择题：

1. 运行下面程序段后的输出结果是（ ）

s=[]

t=['ABC',{1,2,3},[4,5],['EFG','XYZ']]

for i in t:

s+=i

else:

print(len(s))

A. 4 B. 5 C. 10 D. 14

2. 运行下面程序段后的输出结果是（ ）

s=[]

t=['ABC',{1,2,3},[4,5]]

for i in t:

s+=[i]

else:

print(len(s))

A. 3 B. 5 C. 6 D. 8

3. 运行下面程序段后的输出结果是（ ）

print(len('\\\t\"\101\x30\u0041\b\n'))

A. 8 B. 9 C. 15 D. 24

4. 运行下面程序段后的输出结果是（ ）

uestc='成电';成电='电子科技大学';print(eval(uestc))

A. uestc B. 成电 C. 电子科技大学 D. '电子科技大学'

5. 运行下面程序段后的输出结果是（ ）

s='good good study';s=s.title();print(s)

A. Good good study B. 'Good good study'

C. Good Good Study D. 'Good Good Study'

6. 若已知：s='uestc'; 则下列选项中与其余三项的输出结果不同的是（ ）

A. print(s[::-1]) B. print(s[-1::-1]) C. print(s[-1:0:-1]+s[0]) D. print(s[-1:-1:-1])

7. 若已知：s='uestc'; 则下列选项中与其余三项的输出结果不同的是（ ）

A. for i in s[::-1]:print(i,end='') B. for i in s:print(s[len(s)-s.find(i)-1],end='')

C. print(s[len(s)::-1]) D. print(s[-1:-len(s):-1])

8. 若已知：s='uestc'; 则下列选项中与其余三项的运算结果不同的是（ ）

A. s.find('e') B. s.rfind('e') C. s.index('e') D. s.lindex('e')

9. 下列选项中与其余三项的运算结果不同的是（ ）

A. 'aB'.swapcase() B. 'aB'.capitalize() C. 'aB'.title() D. 'aB'. upper()

10. 在字符串s中，若0≤i<len(s)，则下列各项中与s[i]字符相同的是（ ）

A. s[len(s)-i] B. s[i-len(s)-1] C. s[i-len(s)] D. s[i-len(s)+1]

11. 在字符串s中，若0≤i<len(s)，则下列各项中与s[-i]字符相同的是（ ）

A. s[len(s)-i] B. s[len(s)-i-1] C. s[i-len(s)] D. s[len(s)-i+1]

12. 表达式：'23'\*1+2\*'45'+'67'\*0的值是（ ）

A. '234545' B. '232345454567' C. 234545 D. 232345454567

13. 下列选项中与其余三项的运算结果不同的是（ ）

A. 'abc'.ljust(5,'\*') B. '\*\*'.join(['abc','']) C. 'abc'.join('\*\*') D. 'abc'.join(('','\*\*'))

14. 下列选项中与其余三项的运算结果不同的是（ ）

A. 'abc'.join('\*\*') B. 'abc'.center(5,'\*')

C. '\*\*'.join('abc') D. ('abc'.zfill(4)+'0').replace('0','\*')

15. 若已知：t='0120'; 则表达式：t.lstrip('0') if t.startswith('0') else t的值是（ ）

A. '0120' B. '120' C. '012' D. '12'

16. 若已知：s='012.30'; 则表达式：s.rstrip('0') if s.endswith('0') else s的值是（ ）

A. '012.30' B. '12.30' C. '012.3' D. '12.3'

17. 下列表达式中与其余三项的值不同的是（ ）

A. 'et' in 'uestc' B. 'ab '>'ab'

C. len('abc')==len("""abc""") D. str(3.14)==repr(3.14)

18. 若已知：s='10203040'; 则表达式：s.replace('0','',s.count('0')-2)的值是（ ）

A. '1234' B. '1203040' C. '123040' D. '12340'

19. 若已知：s='10203040'; 则表达式：s.split(sep='0',maxsplit=2)的值是（ ）

A. ['1', '2', '3', '4', ''] B. ['1', '2', '3040'] C. ['10203', '4', ''] D. ['1020', '3', '4']

20. 若已知：s='10203040'; 则表达式：s.partition('0')的值是（ ）

A. ('1', '0', '203040') B. ('1', '2', '3040')

C. ('1', '0', '2', '0', '3', '0', '4', '0') D. ('1', '2', '3', '4')

21. 已知：s='10203040'; 若想得到：'1234'; 则不正确的表达式是（ ）

A. ''.join([i for i in s if i!='0']) B. s.strip('0')

C. ''.join(s.split('0')) D. s.replace('0','')

22. 若已知：s='uestc'; 则下列表达式中与其余三项的值不同的是（ ）

A. s.isalnum() B. s.isalpha() C. s.islower() D. s.isdigit()

23. 若已知：s='2A3B'; 则下列选项中与其余三项的运算结果不同的是（ ）

A. s.isalnum() B. s.isupper() C. s.istitle() D. s.isalpha()

24. 下列选项中，属于bytes类型的是（ ）

A. r'\101' B. '\101' C. 0b101 D. b'101'

25. 若已知：s=b'abcd'; 则下列表达式中与其余三项的值不同的是（ ）

A. [i for i in s] B. list(s)

C. [ord(i) for i in str(s)] D. [ord(i) for i in s.decode()]

26. 下列选项中，可以用作字典关键字的数据类型是（ ）

A. list B. set C. bytearray D. bytes

27. 若已知：s='uestc'; 则下列表达式中与其余三项的值不同的是（ ）

A. [(i,s[i]) for i in range(5)] B. [(eval(i[0]),i[1]) for i in zip('01234',s)]

C. list(enumerate(s)) D. [(i,j) for i,j in zip(b'01234',s)]

28. 若已知：s='uEstC'; 则与下面表达式功能相同的是（ ）

''.join([chr(ord(i)+32) if i.isupper() else chr(ord(i)-32) for i in s])

A. s.swapcase() B. s.title() C. s.capitalize() D. s.upper()

29. 若已知：s='1020304'; 则与下面表达式功能相同的是（ ）

''.join([i if i!='0' else '-' for i in s])

A. s.replace('-','0') B. s.replace('0','-') C. s.swap('-','0') D. s.swap('0','-')

30. 运行下面程序段后的输出结果是（ ）

x='uestc'

y='UESTC'

t=''.maketrans(x,y)

s='uEsTC'

z=s.translate(t)

print(z)

A. UESTC B. uestc C. Uestc D. UeStc

31. 若已知：s='1020304'; 则与下面表达式功能相同的是（ ）

s.translate(''.maketrans('0','-'))

A. s.replace('-','0') B. s.replace('0','-') C. s.swap('-','0') D. s.swap('0','-')

32. 若已知：s='abcdaxyz'; 则表达式：s.translate(''.maketrans('abcd','1234'))的值是（ ）

A. '1234axyz' B. '12341xyz' C. 1234axyz D. 12341xyz

33. 在标准库string中，由26个小写英文字母组成的字符串常量是（ ）

A. string.ascii\_lowercase B. string.ascii\_uppercase

C. string.ascii\_letters D. string.punctuation

34. 在Python交互式命令窗口中运行下面程序段后的输出结果是（ ）

import re;re.findall('[a-z]+','uEstC')

A. ('u', 'st') B. ['u', 'st'] C. ('u', 's', 't') D. ['u', 's', 't']

35. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. import re;re.findall('[0-9]+','a2bc34d5')

B. import re;re.findall('[^a-z]+','a2bc34d5')

C. import re;re.findall('[^A-Z]+','a2bc34d5',re.I)

D. import re;re.findall('[^abc]+','a2bc34d5')

36. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. import re;re.findall('[abcd]','a2bc34d5')

B. import re;re.findall('[^0-9]','a2bc34d5')

C. import re;re.findall('[a-z]','a2bc34d5')

D. import re;re.findall('[a-z]+','a2bc34d5')

37. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. import re;re.findall('[a-z]{2,3}','a2bc34def5ghij6')

B. import re;re.findall('[\D]+','a2bc34def5ghij6')

C. import re;re.findall('bc|def|ghi','a2bc34def5ghij6')

D. import re;re.findall('[^aj\d]+','a2bc34def5ghij6')

38. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. import re;re.sub('ab','-','ab2ba34bab567aabb890aaabbb')

B. import re;re.sub('a+b','-','ab2ba34bab567aabb890aaabbb',count=4)

C. import re;re.sub('[a]{1}[b]{1}','-','ab2ba34bab567aabb890aaabbb',count=4)

D. import re;re.sub('(ab){1}','-','ab2ba34bab567aabb890aaabbb')

39. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. import re;re.sub('b+\d{0,}','-','ab2ba34bab567aabb890aaabbb')

B. import re;re.sub('b{1,}\d\*','-','ab2ba34bab567aabb890aaabbb')

C. import re;re.sub('b+|b+\d|b+\d\d|b+\d\d\d','-','ab2ba34bab567aabb890aaabbb')

D. import re;re.sub('bb\*\d\*','-','ab2ba34bab567aabb890aaabbb')

40. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. import re;re.sub('\d\*','','a2b33c444d5555e666f77g8')

B. import re;''.join(re.findall('[a-z]','a2b33c444d5555e666f77g8'))

C. import re;z=re.compile('\d{0,}');re.sub(z,'','a2b33c444d5555e666f77g8')

D. import re;z=re.compile('\d+');re.subn(z,'','a2b33c444d5555e666f77g8')

41. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. import re;z=re.match('^-?[1-9]\d\*$','-3');True if z else False

B. import re;z=re.match('^\w+\W+$','abc@#\*5');z==None

C. import re;z=re.match('(\d[a-z]\d)+','2a3cd45');z!=None

D. import re;z=re.match('\d+','2a3cd45');z==None

42. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. import re;z=re.search('\W','a2b@3cd#4\*5%6e');print(z)

B. import re;z=re.search('[^a-zA-Z0-9]','a2b@3cd#4\*5%6e');print(z)

C. import re,string;z=re.search('['+string.punctuation+']','a2b@3cd#4\*5%6e');print(z)

D. import re,string;z=re.search('[^'+string.digits+']','a2b@3cd#4\*5%6e');print(z)

43. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. import re,string;z=re.finditer('['+string.digits+']','a2b3cd');[i.group() for i in z]

B. import re,string;z=re.finditer('[^'+string.ascii\_letters+']','a2b3cd');[i.group() for i in z]

C. import re;z=re.findall('\D','a2b3cd');z

D. import re;z=re.finditer('[^a-z]','a2b3cd',re.I);[i.group() for i in z]

44. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. import re;re.split('\W+','ab?@c#$def&\*%gh/\ijk')

B. import re;re.split('\W\W\*','ab?@c#$def&\*%gh/\ijk')

C. import re;re.findall('\W+','ab?@c#$def&\*%gh/\ijk')

D. import re;re.findall('\w+','ab?@c#$def&\*%gh/\ijk')

45. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. s=[['abcd'],('1234')];s[0][0]+s[1][0]

B. s=[('abcd',),['1234']];s[0][0]+s[1][0]

C. s=[('abcd',),['1234']];s[0][0]+s[1][0][0]

D. s=['abcd','1234'];s[0]+s[1][0]

46. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. ['abc','efg','xyz']<['b'] B. ('abc',456)<('d',123)

C. 'a' in ['abcd'] D. 'a' in ((('abc')))

47. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. len([0 for i in range(3) for j in range(4)])

B. len([i for i in range(3) for j in range(4)])

C. len([j for i in range(4) for j in range(3)])

D. len([i\*j for i in range(5) for j in range(4)if i>j])

48. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. s=list('uestc');[(i,j) for i,j in enumerate(s)]

B. s=list('uestc');[i for i in enumerate(s)]

C. s=list('uestc');t=enumerate(s);list(zip(\*t))

D. s=list('uestc');list(zip(range(5),s))

49. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. s=list('uestc');sorted(s,key=ord,reverse=False)

B. s=list('uestc');s.sort(key=ord,reverse=True);s

C. s=list('uestc');s.sort();s

D. s=list('uestc');sorted(s)

50. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. s=['abc','x','uestc'];s.sort();s

B. s=['abc','x','uestc'];sorted(s)

C. s=['abc','x','uestc'];sorted(s,key=len)

D. s=['abc','x','uestc'];sorted(s,key=max)

51. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. s='uestc';''.join(list(reversed(s)))

B. s='uestc';t=sorted(list(s));''.join(t)

C. s='uestc';t=list(s);t.reverse();''.join(t)

D. s='uestc';s[::-1]

52. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. s=[(),{},set(),None,'',0,[],False];any(s)

B. s=[(()),"''",[[]]];all(s)

C. s=[({}),False,[()]];all(s)

D. s=[({},),' ',[False]];all(s)

53. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. x=[1,2,3,4];a,b,c,d=x;a,b,c,d

B. x=(1,2,3,4);a,b,c,d=x;a,b,c,d

C. x=dict(a=1,b=2,c=3,d=4);a,b,c,d=x;a,b,c,d

D. x={1,2,3,4};a,b,c,d=x;a,b,c,d

54. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. x,\*y,z=1,2,3,4;x,y,z B. x,\*y,z=(1,2,3,4);x,y,z

C. x,\*y,z=([1,2,3,4]);x,y,z D. x,\*y,z=({1,2,3,4},);x,y,z

55. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. list(i for i in 'uestc') B. [i for i in 'uestc']

C. {i for i in 'uestc'} D. list((i for i in 'uestc'))

56. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. t=((i,ord(i)) for i in 'uestc');dict(t)

B. t=enumerate('uestc');dict(t)

C. t=[(i,ord(i)) for i in 'uestc'];dict(t)

D. t={(i,ord(i)) for i in 'uestc'};dict(t)

57. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. s='a2bc34d5';str([i for i in s if i.isdigit()])

B. s='a2bc34d5';''.join([i for i in s if i.isnumeric()])

C. s='a2bc34d5';t=[i for i in s if i.isalpha()];''.join([i for i in s if i not in t])

D. s='a2bc34d5';t=[];x=[t.append(i) for i in s if i.isdecimal()];''.join(t)

58. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. s='a2bc34d5';t=list(s);x=[t.remove(i) for i in s if i.isdigit()];t

B. s='a2bc34d5';[i for i in s if i.isalpha()]

C. s='a2bc34d5'; t=[];x=[t.append(i) for i in s if i.isalpha()];t

D. import re;s='a2bc34d5';re.findall('[a-z]+',s,re.I)

59. 在Python交互式命令窗口中运行下面程序段，会出错的是（ ）

A. s=(2,[3,4,5],6);s[1][0]=7;s B. s=[(2),(3),(4)];s[1]=5;s

C. s=['abc','ijk','mn'];s[1][0]='x';s D. s=['abc','ijk','mn'];s[1]='x';s

60. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. s=[1,2,3];t=[4,5,6];s+=t;s

B. s=[1,2,3];t=[4,5,6];s.extend(t);s

C. s=[1,2,3];t=[4,5,6];x=[s.insert(0,i) for i in t];s

D. s=[1,2,3];t=[4,5,6];x=[s.append(i) for i in t];s

61. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. s=[1,2,3];t=s;t[1]=4;s B. s=[1,2,3];t=s;s[1]=4;t

C. s=[1,2,3];t=s.copy();t[1]=4;s D. s=[1,2,3];t=s.copy();t[1]=4;t

62. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. d=dict(a=1,b=2,c=3);d

B. d={'a': 1, 'b': 2, 'c': 3};d

C. k=['a','b','c'];v=[1,2,3];d=dict.fromkeys(k,v);d

D. k=['a','b','c'];v=[1,2,3];d=dict(zip(k,v));d

63. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. k=['a','b','c'];d=dict(enumerate(k));d

B. d={};d[0]='a';d[1]='b';d[2]='c';d

C. d={0:'a',1:'b'};d.update({2:'c'});d

D. d=dict(0='a',1='b',2='c')

64. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. d=dict(a=2,b=3,c=4);d.get('c')

B. d=dict(a=2,b=3,c=4);max(d)

C. d=dict(a=2,b=3,c=4);max(d.values())

D. d=dict(a=2,b=3,c=4);d.get('d',4)

65. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. d=dict(a=2,b=3,c=4);[k for k,v in d.items()]

B. d=dict(a=2,b=3,c=4);sorted(d)

C. d=dict(a=2,b=3,c=4);list(d.keys())

D. d=dict(a=2,b=3,c=4);t=list(zip(\*(d.items())));t[0]

66. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. s=dict(a=2);t=dict(a=1,b=3);s.update(t);s

B. s=dict(a=2,b=1);t=dict(b=3);s.update(t);s

C. s=dict(a=1,b=3);t=dict(a=2);s.update(t);s

D. s=dict(a=1);t=dict(a=2,b=3);s.update(t);s

67. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. d=dict(a=2,b=3,c=4);t=d.copy();[t.pop(k) for k in d.keys()]

B. d=dict(a=2,b=3,c=4);[d[k] for k in d.keys()]

C. d=dict(a=2,b=3,c=4);d.values()

D. d=dict(a=2,b=3,c=4);[v for k,v in d.items()]

68. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. s={2,3,4};s B. s={4,2,3,3,2,4};s

C. t=[3,2,4,3,3,2,4,4,3,2];set(t) D. d=dict(a=2,b=3,c=4);set(d)

69. 如果s={2,3,4};t={3,4,5};则在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. x=s.difference(t);y=t.difference(s);x.union(y)

B. x=s.union(t);x.difference(s.intersection(t))

C. s.update(t);s.difference(x.intersection(t))

D. s.symmetric\_difference(t)

70. 如果s={2,3,4};则在Python交互式命令窗口中运行下面程序段，最后s的值与其余三项不同的是（ ）

A. s.remove(2) B. s.discard(2)

C. s.difference({2}) D. s.difference\_update({2})

71. 如果s={2,3,4};t={3,4,5};则在Python交互式命令窗口中运行下面程序段，最后s的值与其余三项不同的是（ ）

A. s.intersection(t) B. s.intersection\_update(t)

C. s.difference(s.intersection(t)) D. s.symmetric\_difference(t)

72. 如果s={2,3,4};t={3,4,5};则在Python交互式命令窗口中运行下面程序段，最后s的值与其余三项不同的是（ ）

A. s.difference\_update(t)

B. s.difference\_update(s.intersection(t))

C. s.symmetric\_difference\_update(s.intersection(t))

D. s.discard(s.intersection\_update(t))

73. 如果s={2,3,4};t={3,4,5};则在Python交互式命令窗口中运行下面程序段，最后s的值与其余三项不同的是（ ）

A. s.symmetric\_difference\_update(t)

B. s.symmetric\_difference(t)

C. s=set().union(s.difference(t),t.difference(s))

D. s=(s.union(t)).difference(s.intersection(t))

74. 如果s={2,3,4};t={3,4,5};则在Python交互式命令窗口中运行下面程序段，最后s的值与其余三项不同的是（ ）

A. s.union(t) B. s.update(t)

C. s.union({6}) D. s.symmetric\_difference(t)

75. 如果x,y,z都是集合，则下列表达式非法的是（ ）

A. x.intersection(y,z) B. x.union(y,z)

C. x.difference(y,z) D. x.symmetric\_difference(y,z)

76. 如果x={1,2,3,4,5};y={1};z={2};则在Python交互式命令窗口中运行下面程序段，最后x的值与其余三项不同的是（ ）

A. x=x.difference(y,z)

B. x.difference(y,z)

C. x=(x.symmetric\_difference(y)).symmetric\_difference(z)

D. x.difference\_update(y,z)

77. 如果x={1,2,3,4};y={1,3,5};z={2,3,4};则在Python交互式命令窗口中运行下面程序段，最后x的值与其余三项不同的是（ ）

A. x.intersection(y,z)

B. x=(x.intersection(y)).intersection(z)

C. x.intersection\_update(y.intersection(z))

D. x.difference\_update(y.symmetric\_difference(z))

78. 如果x={1,2,3,4};y={1,3,5};z={2,3,4};则在Python交互式命令窗口中运行下面程序段，最后x的值与其余三项不同的是（ ）

A. x.intersection\_update(y,z)

B. x.difference\_update(y,z)

C. x=(x.union(y,z)).difference(y.symmetric\_difference(z))

D. x.difference\_update(y.symmetric\_difference(z))

79. 如果s={1,3,5};t={2,4};则下列表达式含义与其余三项不同的是（ ）

A. s.isdisjoint(t)

B. s.intersection(t)==set()

C. (s-t)==set()

D. (s.symmetric\_difference(t))==(s.union(t))

80. 如果s={2,3,4};t={2,4};则下列表达式含义与其余三项不同的是（ ）

A. (s-t).issubset(s)

B. s.union(t)==s

C. (s.symmetric\_difference(t)).issubset(s)

D. (s.difference(t)).issubset(s)

81. 下列各类型数据中，能作字典关键字的数据类型是（ ）

A. list B. set C. dict D. float

82. 向集合s中添加元素2（假设s中原来没有该元素），错误的是（ ）

A. s.add(2) B. s.update({2})

C. s.union({2}) D. s.symmetric\_difference\_update({2})

83. 从集合s中删去元素2（假设s中原来有该元素），错误的是（ ）

A. s.difference\_update({2}) B. s.intersection\_update(s.difference({2}))

C. s.discard({2}) D. s.symmetric\_difference\_update({2})

84. 如果s={2,3,4};则下列表达式含义与其余三项不同的是（ ）

A. s.clear() B. s.difference(s)

C. s.difference\_update(s.copy()) D. s.symmetric\_difference\_update(s)

85. 下列各类型数据中，不能作字典关键字的数据类型是（ ）

A. frozenset({2}) B. str(2) C. bytearray(b'2') D. b'2'

86. 如果s={2,3,4};t={3,4,5};则在Python交互式命令窗口中运行下面程序段，最后s的值与其余三项不同的是（ ）

A. s|t-s&t B. s-s&t C. s-s^t D. s-=t

87. 如果s={2,3,4};t={3,4,5};则在Python交互式命令窗口中运行下面程序段，最后s的值与其余三项不同的是（ ）

A. s-=s^t B. s=(s|t)-(s^t) C. s-=(s-t)|(t-s) D. s&t

88. 如果s={2,3,4};欲删去s中的元素3，则下列表达式不能实现该操作的是（ ）

A. s-={3} B. s^={3} C. s&=s-{3} D. s|=s^{3}

89. 如果s={2,3,4};欲向s中添加元素5，则下列表达式不能实现该操作的是（ ）

A. s+{5} B. s|={5} C. s|=s^{5} D. s^={5}

90. 如果s={1,3,5};t={2,3,4};则下列表达式的值与其余三项不同的是（ ）

A. s B. s-t|s&t C. s^t|s-(t-s) D. (s|t)-s^t|s-t

91. 下列集合运算符优先级最高的是（ ）

A. & B. | C. - D. ^

92. 下列集合运算符优先级最低的是（ ）

A. & B. | C. - D. ^

93. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. {2&3} B. {2|3} C. ({2}|{3})-{2} D. {2,3}^set([2])

94. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. {7&3}^{4} B. {3}^{4}&{7} C. {3}-{7}^{4} D. {3^7}|{3}

95. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. {3&2&1} B. {3^2^1} C. {3}^{2}&{1}^{0} D. {3}&{2}&{1}|{0}

96. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. {3}>{2} B. {3}>={2} C. [3]>=[2] D. {2}>={2,0}

97. 在Python交互式命令窗口中运行下面程序段，输出结果与其余三项不同的是（ ）

A. {3,2}>{2,3} B. {1,2,3}>{1,2}>{1} C. {1,2,3}=={3,1,2} D. {2}<{2,0}

98. 在Python交互式命令窗口中运行下面程序段，输出的结果是（ ）

{3,2,1}-{1,5,7}|{3,2,6}^{2,4,3}&{2}

A. set() B. {1, 3, 5} C. {2, 3, 6} D. {1,2, 3, 4,5,6,7}

99. 在Python交互式命令窗口中运行下面程序段，输出的结果是（ ）

{3,2,1}^{1,5,7}&{3,2,6}-{2,4,3}|{2}

A. {1, 2, 3} B. set() C. {4,5,6} D. {1,4,7}

100. 在Python交互式命令窗口中运行下面程序段，输出的结果是（ ）

{3,2,1}|{1,5,7}^{3,2,6}^{2,4,3}-{2}

A. set() B. {1, 3, 5} C. {2, 3, 6} D. {1,2, 3, 4,5,6,7}