

专业： 年级： 学号： 姓名： 成绩：

草 稿 区

得 分

一、选择题（本题共 60 分，每小题 3 分）

1-12 为 python 试题，13-20 为 R 试题

1. Python 中如何声明一个空集合（）

A、myset = {} B、myset = [] C、myset = () D、myset = set()

2. 已知 x = [[1,3,3], [2,3,1],[3,3,1]] ，那么表达式 sorted(x, key=lambda x:(x[1], -x[2]))的值为（）

A、[[2, 3, 1], [1, 3, 3], [3, 3, 1]] B、[[1, 3, 3], [2, 3, 1], [3, 3, 1]] C、[[3, 3, 1], [1, 3, 3], [2, 3, 1]] D、报错

3. 针对元组(1, 2, [1, 2, ‘1’ , ‘2’])的说法正确的是（）

A、长度为 6 B、属于二维元组
C、元组的元素可变 D、嵌入的列表的值可变

4. 右边代码输出的最后一个数为（）

A、19 B、20 C、18 D、17

```
for num in range(10,20):
    for i in range(2,num):
        if num % i == 0:
            j=num/i
            break
    else:
        print(num)
```

5. a = dict(one=1, two=2, three=3); akeys=a.keys(); del a['two']; print(list(akeys)); 以上代码输出为（）

A、['one', 'two', 'three'] B、['two', 'three'] C、['one', 'three'] D、程序报错

6. 关于 python 中的函数参数传递，以下说法正确的是（）

- A、如果参数是不可变类型，则在函数中不能改变对参数进行赋值
- B、即使参数是可变类型，在函数中修改参数的值也不会影响函数外该参数的值
- C、如果参数是可变类型，则函数中修改参数的值会影响函数外该参数的值
- D、无论参数是可变类型还是不可变类型，参数传递的都只是拷贝

7. 右边关于 sounds 的代码的输出是（）

- A、程序报错
- B、Bob 会叫；Alice 会“喵喵”叫
- C、Bob 会叫；Alice 会叫
- D、Bob 会“喵喵”叫；Alice 会“喵喵”叫

```
def fun1():  
    return "fun1"  
def fun2():  
    return "fun2"  
def fun3():  
    return "fun3"  
funcs=[fun1,fun2,fun3]  
print(funcs[1])
```

```
class Animal():  
    def __init__(self, name):  
        self.name = name  
    def sounds(self):  
        print(self.name, '会叫')  
class Cat(Animal):  
    def __init__(self,name):  
        super().__init__(name)  
    def sounds(self):  
        print(self.name, '会“喵喵”叫')  
def sounds(all):  
    try:  
        all.sounds()  
    except AttributeError:  
        print('不会叫')  
a=Animal("Bob")  
b=Cat("Alice")  
a.sounds()  
b.sounds()
```

8. 上面关于函数的代码的输出是（）

- A、fun1 B、fun2 C、fun3 D、以上都不对

9. 在 numpy 中，下面哪条语句可以将两个列数相同的矩阵拼接在一起()

- A、np.concatenate(array1, array2)
- B、np.append(array1, array2)
- C、np.stack(array1, array2)
- D、np.vstack((array1, array2))

10. 以下代码的输出是 ()

```
import pandas as pd
import numpy as np
obj = pd.Series(np.arange(4), index=['a', 'b', 'c', 'd'])
test=obj[['b', 'a', 'd']];test[0]=1000;print(obj.values)
test=obj[1:3];test[0]=100;print(obj.values)
```

A、 [0 1 2 3][0 100 2 3] B、 [0 1000 2 3][0 100 2 3]

C、 [0 1 2 3][0 1 2 3] D、 [0 1000 2 3][0 1 2 3]

11. 以下代码输出是 ()

```
import pandas as pd
data = pd.DataFrame(np.arange(16).reshape((4, 4)),
index=['Ohio', 'Colorado', 'Utah', 'New York'],
columns=['one', 'two', 'three', 'four'])
print(data.loc['Colorado', ['two']].values)
```

A、 4 B、 9 C、 5 D、 程序报错

12. 在文件打开方式中, () 是只读模式

A、 a B、 b C、 w D、 r

13. 右边关于 dataframe 的代码输出是 ()

A、 1 B、 2 C、 3 D、 4

```
df <- data.frame(col1 = c(1,2,3,4,5),
col2 = c("A","B","C","D","E"),
col3 = c(0.1,0.2,0.3,0.4,0.5))
new_df <- df[df$col1 > 3,]
print(nrow(new_df))
```

14. 下列哪个函数用于连接两个向量 ()

A、 concat() B、 c() C、 merge() D、 +

15. 以下代码输出是 ()

```
df <- data.frame(col1 = c(1,2,3,4,5),
col2 = c("A","B","C","D","E"),
col3 = c(0.1,0.2,0.3,0.4,0.5))
df$col4 <- ifelse(df$col1 > 2, "yes", "no")
print(df$col4)
```

A、no no yes yes yes

B、no yes no yes no

C、yes yes yes yes yes

D、no no no no no

```
x <- c(1,2,3,4,5)
y <- sapply(x, function(i) {
  if (i %% 2 == 0) return(i^2)
  else return(i^3)})
print(y)
```

16. 右边关于 sapply 代码输出是 ()

A、10 B、30 C、1 4 27 16 125 D、1 4 9 16 25

17. 右边关于函数 f 的代码输出是 ()

A、报错 B、NULL C、NA D、45

```
f=function(a,b){
  print(a)
}
f(45)
```

18. 下列说法正确的是()

A、NaN 表示没有定义 B、Inf 表示没有定义 C、NaN 表示缺失 D、NA 表示缺失

19. xvals=seq(-4,4,length=50), fx=dnorm(xvals,mean=0,sd=1) 则 fx[2]>fx[47]的值为 ()

A、TRUE B、FALSE C、不能确定 D、程序报错

20. 设 foo <- list(matrix(data=1:4,nrow=2,ncol=2),c(T,F,T,T),"hello"),

则 class(foo[1])和 class(foo[[1]])为 ()

A、matrix, matrix B、list, list C、list, matrix D、matrix, list

二、(R) 大部分的统计分布函数在 R 里都有实现, 这些函数名都有一套统一的前缀, 分别解释 d、p、q、r 代表什么?。(10 分)

三、(Python) 右边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。(10 分)

```
from random import randint
result = set()
while True:
    result.add(randint(1,10))
    print(result)
    print(len(result))
    if len(result)==20:
        break
print(result)
```

四、请简述箱线图和小提琴图的用法以及它们的特点（R 和 Python 都可以）（10 分）

五、(Python) 请写出并解释右边关于 scope_test 的代码的输出结果。(10 分)

```
def scope_test():
    spam = "test spam"
    def do_local():
        spam='local spam'
        print(spam) # = "local spam"

    def do_nonlocal():
        nonlocal spam
        spam = "nonlocal spam"

    def do_global():
        global spam
        spam = "global spam"
    do_local()
    print("After local assignment:", spam)
    do_nonlocal()
    print("After nonlocal assignment:", spam)
    do_global()
    print("After global assignment:", spam)
scope_test()
print("In global scope:", spam)
```