

# 习题课

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。

```
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)
```

从random库中选取相应的函数满足并实现下列条件

(1) 随机生成100以内的10个整数

(2) 随机选取0到100间的奇数

(3) 从字符串' abcdefghij'中随机选取4个字符

(4) 随机选取列表['apple','pear','orange','banana']中的1个字符串

```
import random
for i in range(10):
    print(random.randint(0,100))
random.randrange(1,101,2)
seq = 'abcdefghij'
n = random.sample(seq,4)
print(n)
import random
list = ['apple','pear','orange','banana']
n = random.choice(list)
print(n)
```

请用pandas模块完成以下任务

1) 创建DataFrame结构的二维数组df, 其行名、列名以及元素值如下

	animal	fruit	number
a	cat	apple	3
b	dog	orange	5

2) 将行名重命名, 将行名'a','b'修改为'C','D'

3) 将'C'行 'number'列元素3修改为NaN (即缺失值)

4) 请用缺失值填充方法fillna()将所有缺失值填充为0

```
import pandas as pd
import numpy as np
df =
pd.DataFrame({'animal':['cat','dog'],'fruit':['apple','orange'],'number':[3,5]},index=['a','b'])
df.index=['C','D']
df = df.replace(3,np.nan)
df = df.fillna(0)
```

有关于某班级学生考试成绩的excel文件，文件名为考试成绩，文件保存在当前文件夹下，文件前几行内容如下：

	A	B	C	D	E	F	G	H	I
1	姓名	学号	python	高数	线代	概率论	大学英语		
2	张三	2010001	80	76	78	91	92		
3	李四	2010002	87	72	98	100	75		
4	王五	2010003	66	55	69	60	77		
5	赵六	2010004	54	49	60	59	58		
6	钱二	2010005	65	71	86	80	82		
7	孙七	2010006	93	89	98	100	75		

设本班级没有重名的同学，请用pandas模块做以下任务

- 1) 读文件，并将原始数据保存成DataFrame格式的数组df
- 2) 编程找出所有科目总分数低于320分的所有同学
- 3) 编程找出各科考试成绩中有59分的所有同学
- 4) 编程找出所有科目总分数最高的前三位同学

```
df= pd.read_excel(r'C:\Users\user\Desktop\ks.xls')
```

```
df['总分']=df.iloc[:,2:7].sum(axis=1)
```

```
df[df['总分']<320]
```

```
df.loc[(df['python']==59) | (df['线代']==59) | (df['概率论']==59) | (df['大学
```

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'],':'))
```

a,b,c

a:b:c

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))
```

13

16

15

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.____randint____(1,20) for i in range(____50____)]
r = dict()
for i in x:
    r[i] = r.get(i, ____0____)+1
for k, v in r.items():
    print(k, v)
```



下面程序的执行结果是

```
s = 0
```

```
for i in range(1,6):
```

```
    s += i
```

```
    if i == 5:
```

```
        print(s,end="")
```

```
        break
```

```
else:
```

```
    print(1)
```

下面程序的执行结果是

```
from random import randint
```

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

无限循环

下面程序的执行结果是

```
def func(x):  
    return x.index(max(x))
```

```
x=[2,4,5,6,75,3,2]  
print(func(x))
```

```
demo_arr = np.array(['2018',88],['2019',99])  
df_obj = pd.DataFrame(demo_arr, ['000','001'],['year','data01'])  
print(df_obj)  
其中['year','data01']是 ( )
```

- A. 列标签columns
- B. 行标签index
- C. 第一维数据
- D. 第二维数据

**A**

以下程序的输出结果是( )

```
1.     for i in range(3):  
2.         for s in "abcd":  
3.             if s=="c":  
4.                 break  
5.             print(s,end="")
```

- A. abcabcabc
- B. aaabbbbccc
- C. aaabbbb
- D. ababab

**D**

a是一个Series对象，共4个元素，请问， $a > 0$ 的结果是( )

- A. 一个Series对象，每个值是True或False
- B. 一个Series对象，每个值0或者原值
- C. 一个值，0或NaN
- D. 一个布尔值，True或False

**A**

下面程序的执行结果是（ ）

```
import numpy as np
from sklearn.linear_model import LinearRegression

X = np.array([1, 2, 3, 4]).reshape(-1, 1) # 转为 2D 数组
y = np.array([10,13,16,19])

# 训练模型
model = LinearRegression()
model.fit(X, y)
x_new = np.array([[8]]) # 注意输入必须是 2D 数组
y_pred = model.predict(x_new)
print(y_pred)
```

[31.]

输入一个整数列表Names，其中有3个2，请返回第2个2的索引值

```
names = eval(input())  
i = names.index(2)  
new_names = names[i+1:]  
j = new_names.index(2)  
print(i+j+1)
```



判断黑洞数。黑洞数是指这样的整数：由这个数字每位上的数字组成的最大数减去每位数字组成的最小数仍然得到这个数自身。例如3为黑洞数是495，因为 $954-459=495$

```
i = input()
lst = list(str(i))
x = int( ''.join(sorted(lst)))
print(x)
y = int( ''.join(sorted(lst,reverse=True)))
print(y)
if y-x==i:
    print('yes')
else:
    print('no')
```

- 1、使用numpy库random子库随机产生四门课的成绩，随机种子数取：0x1010
- 2、每门课程40个成绩，分布范围为50-100分之间，要符合正态分布规则， $\mu=75$ ， $\sigma=8$ ，学生的成绩要为整数。
- 3、用此成绩创建一个DataFrame对象，学生的学号范围为1001-1040，四门课程的列标签分别为 'A','B','C','D'.行索引为学生的学号
- 4、输出该班级前面五名学生成绩，效果如下。

	A	B	C	D	
1001	83	73	93	60	
1002	78	74	81	83	
1003	64	73	64	82	
1004	82	76	83	81	
1005	73	83	74	73	

```
import numpy as np
import pandas as pd
np.random.seed(0x1010)
ar=np.random.normal(75,8,160).reshape(40,4)
ar=ar.astype(np.int)
id=[1000+i for i in range(1,41)]
df=pd.DataFrame(ar,columns=list('ABCD'),index=id)
print(df.head())
```