Q. 파이썬 외장함수 random을 이용하여 아래사항들을 출력하세요.

- 0.0<= x <1.0 사이의 float를 랜덤하게 선택
- 10에서 20사이의 정수중에서 난수값 선택
- 10보다 적은 int 선택
- 20과 30사이의 float 선택
- [6,7,8,9,10]리스트 내부에 있는 요소를 랜덤하게 선택
- [6,7,8,9,10]리스트 내부에 있는 요소중 k개 선택

```
import random

print (random.random())
print (random.randint(10, 20))
print (random.randrange(10))
print (random.uniform(20, 30))
print (random.choice([6,7,8,9,10]))
print (random.sample([6,7,8,9,10], k=2))
```

```
0.29852642093292314
12
5
21.89955645729318
6
[7, 9]
```

Q. 주어진 데이터에서 랜덤으로 하나를 선택하여 꺼낸 다음 그 값을 출력하고 꺼내진 요소를 제외시

data = [1,2,3,4,5,6,7,8,9,10]

```
data = [1,2,3,4,5,6,7,8,9,10]
random.shuffle(data)
print (data.pop())
print (sorted(data))
```

```
2
[1, 3, 4, 5, 6, 7, 8, 9, 10]
```

Q. 넘파이 random을 이용하여 아래사항들을 출력하세요.

- 0부터 10까지 랜덤한 숫자 1개
- 0부터 1사이의 균일분포에서 난수 5개로 구성되는 배열 생성
- 평균 0, 표준편차 1의 가우시안 표준정규분포에서 난수 10개로 구성되는 배열 생성

```
import numpy as np
print (np.random.randint(1, 10))
print (np.random.rand(5))
print (np.random.randn(10))
```

```
9
[3.14217457e-04 7.68468347e-01 6.77429146e-01 4.95385723e-01
 8.91905984e-01]
[ 0.90552964 -0.74571398 -0.73277102  0.30040138  1.44176289  0.88948358
  1.23661487  0.93477923 -0.20275066  0.08665786]
```

Code Text

Q. 넘파이 random을 이용하여 아래사항들을 출력하세요.

- [6,7,8,9,10] 리스트의 데이터의 순서 바꾸기

```
arr = [6,7,8,9,10]
np.random.shuffle(arr)
print (arr)
```

```
[9, 8, 6, 10, 7]
```

Q. 넘파이 random을 이용하여 아래사항들을 출력하세요.

- 10보다 적은 정수에서 랜덤으로 중복되지 않게 5개를 선택 배열을 생성하세요.
- c1 = [6,7,8,9,10] 배열에서 랜덤으로 중복 선택이 가능하게 5개를 선택, 배열을 생성하세요.

```
print (np.random.choice(10, 5, replace=False))
c1 = [6,7,8,9,10]
print (np.random.choice(c1, 5))
```

```
[6 3 4 9 0]
[ 6  7 10  9  9]
```

Q. 넘파이 random을 이용하여 2.0보다 크거나 3.0보다 작은 5개의 수를 출력하세요.

```
print (np.random.uniform(2.0, 3.0, 5))
```

```
[2.78528131 2.94019464 2.30228088 2.08392709 2.98506673]
```

Q. 넘파이 random을 이용하여 0.0보다 크거나 1.0보다 작은 (3,3) 2차원 배열을 2가지 방식으로

```
print (np.random.rand(3,3),'\n')
print (np.random.uniform(0.0,1.0,(3,3)),'\n')
print (np.random.random_sample((3,3)))
```

```
[[0.12773351 0.9616301  0.39786691]
```

Q. 앞(head) 또는 뒤(tail) (n=1) 가 나올 확률이 각각 50%(p=0.5)인 동전 던지기를 20번(size=2(

```
print (np.random.binomial(1, 0.5, size=20))
```

```
[1 0 1 1 1 0 1 0 1 1 1 1 1 1 0 1 1 1 1 0]
```

Q. (2,3,4) 3차원 배열 형태로 정규분포(np.random.normal)로 부터 무작위 샘플을 생성

```
print (np.random.normal(0.0, size=(2, 3, 4)))
```

```
[[[-0.98563329  0.79109744  1.32862223 -0.46211299]
  [-0.19897257 -0.45616672 -0.37004468 -0.40040919]
  [ 0.28814635 -3.08512496 -0.6199405  -0.01724624]]

 [[-1.11105286  0.64117183  0.13284961  0.15814423]
  [-0.77106216 -0.6751964   1.23089826 -0.67129015]
  [ 0.30528408 -0.93996078 -0.03648123 -0.06028254]]]
```

Q. [4 7 7 1 9]에서 중복되지 않는 원소 배열과 각 원소의 중복 개수 배열을 출력하세요.

```
qlist = [4,7,7,1,9]
alist, count = np.unique(qlist, return_counts=True)
print (alist)
print (count)
```

```
[1 4 7 9]
[1 1 2 1]
```

Q. [1, 1, 2, 2, 2, 3]에서 0 ~ 5에 해당하는 정수의 개수 배열을 출력하세요.

```
qlist = [1, 1, 2, 2, 2, 3]
print (np.bincount(qlist, minlength=5))
```

```
[0 2 3 1 0]
```

Q(응용) 'ID', 'Prod', 'Price' 3개의 column과 10개의 row으로 구성되는 데이터셋을 아래를 참조하

- ID는 1 ~ 10으로 적용
- Prod는 상품코드로서 np.random을 이용하여 10보다 적은 정수를 랜덤하게 선택
- Price는 가격으로서 np.random을 이용하여 1000 보다 적은 양의 소수를 랜덤하게 선택
- 3개의 컬럼을 결합하는 방법은 concat() 함수를 사용

random하게 Youtube 영상 추출하기

```
import pandas as pd
import numpy as np
```

```
3
4 vid = np.arange(3000)+1
5 sr_vid = pd.Series(vid, name='ID')
6
7 vgen = np.random.randint(0, 2, size = 3000)
8 sr_vgen = pd.Series(vgen, name = 'Gender')
9
10 vkey = np.random.randint(26, size = 3000)
11 sr_vkey = pd.Series(vkey, name='Keyword')
12
13 vtype = np.random.randint(10, 30, size = 3000)
14 sr_vtype = pd.Series(vtype, name = 'Type')
15
16 vtime = np.random.randint(0,25, size = 3000)
17 sr_vtime = pd.Series(vtime, name = 'Clock')
18
19 vclick = np.random.randint(20000000, size = 3000)
20 sr_vclick = pd.Series(vclick, name = 'clicks')
21
22 vpercent = np.random.randint(100, size = 3000)
23 sr_vpercent = pd.Series(vpercent, name = 'View Time Percentage of Running Time')
24
25 vgoods = np.random.randint(vclick)
26 sr_vgoods = pd.Series(vgoods, name = "goods")
27
28 sr_vratio = pd.Series(vgoods/vclick*100, name = 'Ratio')
29
30 ds = pd.concat ([sr_vid, sr_vgen, sr_vkey, sr_vtype, sr_vtime, sr_vpercent, sr_v
31 print (ds)
```

```
          ID  Gender  Keyword  ...    clicks     goods      Ratio
0          1       1       25  ...   4227124   2903517  68.687765
1          2       0        5  ...     27530       886   3.218307
2          3       1       19  ...  10885127   9190656  84.433154
3          4       0       25  ...   5957058   2648401  44.458204
4          5       1       18  ...  10552923   9672759  91.659524
...      ...     ...      ...  ...       ...       ...        ...
2995    2996       0        9  ...  10457299   7108839  67.979686
2996    2997       0        9  ...  13819176   1058462   7.659371
2997    2998       0       16  ...   9262306   4692925  50.666918
2998    2999       1       17  ...  12626604   1547428  12.255298
2999    3000       0       11  ...  19211585  11096363  57.758707

[3000 rows x 9 columns]
```