

# rk21mpa02

April 14, 2023

**0.1 Q1:** You have an array of shape (5, 5). Using NumPy, create a new array that contains the diagonal elements of the original array.

```
[1]: import numpy as np
a = np.arange(25).reshape(5,5)
print(a)
print(a.diagonal())
```

```
[[ 0  1  2  3  4]
 [ 5  6  7  8  9]
 [10 11 12 13 14]
 [15 16 17 18 19]
 [20 21 22 23 24]]
[ 0  6 12 18 24]
```

**0.2 Q2:** You have two arrays of shape (3, 3) and (3, 1). Using NumPy, perform matrix multiplication of these arrays.

```
[2]: import numpy as np
a = np.arange(9).reshape(3,3)
b = np.arange(3).reshape(3,1)
print(a)
print(b)
print(np.dot(a,b))
```

```
[[0 1 2]
 [3 4 5]
 [6 7 8]]
[[0]
 [1]
 [2]]
[[ 5]
 [14]
 [23]]
```

**0.3 Q3:** You are given an array of integers. Using NumPy, create a new array that contains only the unique elements of the original array.

```
[3]: import numpy as np
a = np.array([1,2,3,4,5,6,7,8,9,10,1,2,3,4,5,6,7,8,9,10])
print(np.unique(a))
```

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

**0.4 Q4:** You have two arrays of shape (3, 3) and (3, 4). Using NumPy, concatenate these arrays along the first axis.

```
[5]: import numpy as np
a = np.arange(9).reshape(3,3)
b = np.arange(12).reshape(3,4)
print(np.concatenate((a,b),axis=1))
```

```
[[ 0  1  2  0  1  2  3]
 [ 3  4  5  4  5  6  7]
 [ 6  7  8  8  9 10 11]]
```

**0.5 Q5:** You have an array of shape (2, 3, 4). Using NumPy, reshape it into an array of shape (2, 4, 3).

```
[6]: import numpy as np
a = np.arange(24).reshape(2,3,4)
print(a)
print(a.reshape(2,4,3))
```

```
[[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]]
```

```
[[12 13 14 15]
 [16 17 18 19]
 [20 21 22 23]]]
```

```
[[[ 0  1  2]
 [ 3  4  5]
 [ 6  7  8]
 [ 9 10 11]]
```

```
[[12 13 14]
 [15 16 17]
 [18 19 20]
 [21 22 23]]]
```

**0.6 Q6:** You have an array of shape (4, 4). Using NumPy, split it into two equal parts horizontally.

```
[7]: import numpy as np
a = np.arange(16).reshape(4,4)
print(a)
print(np.hsplit(a,2))
```

```
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]
 [12 13 14 15]]
[array([[ 0,  1],
        [ 4,  5],
        [ 8,  9],
        [12, 13]]), array([[ 2,  3],
        [ 6,  7],
        [10, 11],
        [14, 15]])]
```

**0.7 Q8:** You have two arrays of shape (2, 2) and (2, 2). Using NumPy, calculate the Kronecker product of these arrays.

```
[13]: import numpy as np
a = np.arange(4).reshape(2,2)
b = np.arange(4,8).reshape(2,2)
print(np.kron(a,b))
```

```
[[ 0  0  4  5]
 [ 0  0  6  7]
 [ 8 10 12 15]
 [12 14 18 21]]
```

**0.8 Q9:** You are given an array of numbers. Using NumPy, calculate the cumulative sum and cumulative product of this array.

```
[14]: import numpy as np
a = np.arange(1,11)
print(a)
print(np.cumsum(a))
print(np.cumprod(a))
```

```
[ 1  2  3  4  5  6  7  8  9 10]
[ 1  3  6 10 15 21 28 36 45 55]
[      1      2      6     24    120    720   5040  40320 362880]
3628800]
```

**0.9 Q10:** You have a data frame containing the names, ages and salaries of employees. Using Pandas, create a new data frame that contains only the names and salaries of employees who are older than 30 years.

```
[16]: import pandas as pd
data = {'Name': ['Garvit', 'Anshuman', 'Rahul', 'Kapil'], 'Age': [28,34,29,42],
        ↪ 'Salary': [9000, 10000, 8000, 12000]}
df = pd.DataFrame(data)
print(df)
print(df[df.Age > 30][['Name', 'Salary']])
```

	Name	Age	Salary
0	Garvit	28	9000
1	Anshuman	34	10000
2	Rahul	29	8000
3	Kapil	42	12000

  

	Name	Salary
1	Anshuman	10000
3	Kapil	12000

**0.10 Q11:** You have two data frames containing the names, ages and genders of students from two different classes. Using Pandas, merge these data frames on the basis of the names of the students and add a new column that contains the average age of the students from both classes.

```
[17]: import pandas as pd
df1 = pd.DataFrame({'Name': ['Jas', 'Milkha', 'Sita'], 'Age': [20, 21, 19],
        ↪ 'Gender': ['Male', 'Male', 'Female']})
df2 = pd.DataFrame({'Name': ['Jas', 'Sita', 'Dailjeet'], 'Age': [22, 18, 20],
        ↪ 'Gender': ['Male', 'Female', 'Male']})
merged_df = pd.merge(df1, df2, on='Name')
merged_df['Average Age'] = (merged_df['Age_x'] + merged_df['Age_y']) / 2
print(merged_df)
```

	Name	Age_x	Gender_x	Age_y	Gender_y	Average Age
0	Jas	20	Male	22	Male	21.0
1	Sita	19	Female	18	Female	18.5

**0.11 Q12:** You have a data frame containing the names and grades of students. Using Pandas, group the data frame by grades and calculate the mean, median and standard deviation of the grades for each group.

```
[19]: import pandas as pd
df = pd.DataFrame({'Name': ['Jas', 'Milkha', 'Sita', 'Dailjeet', 'Rahul',
        ↪ 'Kapil'],
        'Grade': [8, 9, 7, 8, 9, 7]})
print(df.groupby('Grade').agg({'Grade': ['mean', 'median', 'std']}))
```

Grade			
	mean	median	std
Grade			
7	7.0	7.0	0.0
8	8.0	8.0	0.0
9	9.0	9.0	0.0