Assignment 7

Name: Yash Shinde

PRN: 1062231623

Div: A

---

1. Extract all odd numbers from an array of 1-10:

```python
import numpy as np

arr = np.arange(1, 11)
odd_numbers = arr[arr % 2 != 0]
print(odd_numbers)
```

Output:

```
[1 3 5 7 9]
```

2. Replace all odd numbers in an array of 1-10 with -1:

```python
arr = np.arange(1, 11)
arr[arr % 2 != 0] = -1
print(arr)
```

Output:

```
[-1  2 -1  4 -1  6 -1  8 -1 10]
```

3. Replace all even numbers in a 1D array with their negative:

```python
arr = np.arange(1, 11)
arr[arr % 2 == 0] = -arr[arr % 2 == 0]
print(arr)
```

Output:

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

4. Reverse a 1D array (first element becomes the last):

```python
arr = np.arange(1, 11)
reversed_arr = arr[::-1]
print(reversed_arr)
```

Output:

```
```

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

```

5. Reshape a 1D array to a 2D array with 5 rows and 2 columns:

```python
arr = np.arange(1, 11)
reshaped_arr = arr.reshape(5, 2)
print(reshaped_arr)
```

Output:

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

6. Count the number of occurrences of a specific value in an array:

```python
arr = np.array([1, 2, 3, 4, 1, 1, 2])
count = np.count_nonzero(arr == 1)
print(count)
```
```

Output:

```
3
```

7. Create a DataFrame from a dictionary of lists:

```python
import pandas as pd

data = {'X': [1, 2, 3, 4], 'Y': [5, 6, 7, 8]}
df = pd.DataFrame(data)
print(df)
```

Output:

```
  X  Y
0  1  5
1  2  6
2  3  7
3  4  8
```

8. Select the first 3 rows of a DataFrame:

```python
data = {'X': [1, 2, 3, 4], 'Y': [5, 6, 7, 8]}
```

```
df = pd.DataFrame(data)
first_three_rows = df.head(3)
print(first_three_rows)
```

Output:

```
  X  Y
0  1  5
1  2  6
2  3  7
```

9. Sort a DataFrame by a column:

```python
data = {'X': [4, 3, 2, 1], 'Y': [8, 7, 6, 5]}
df = pd.DataFrame(data)
sorted_df = df.sort_values(by='X')
print(sorted_df)
```

Output:

```
  X  Y
3  1  5
2  2  6
```

1 3 7

0 4 8

```

10. Replace missing values in a DataFrame:

```python
data = {'X': [1, 2, None, 4], 'Y': [5, None, 7, 8]}
df = pd.DataFrame(data)
df_filled = df.fillna(0)
print(df_filled)
```

Output:

```
   X   Y
0  1.0  5.0
1  2.0  0.0
2  0.0  7.0
3  4.0  8.0
```

11. Scatter plot using random distributions for ball sizes:

```python
import matplotlib.pyplot as plt
import numpy as np
```
```

```
# Generating random data
x = np.random.rand(50)
y = np.random.rand(50)
sizes = 1000 * np.random.rand(50)

# Scatter plot
plt.scatter(x, y, s=sizes, alpha=0.5, c='blue')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Scatter plot with different ball sizes')
plt.show()
```

Output:

A scatter plot is displayed with randomly distributed points of varying sizes.