- **SLIP 1]**

```
my_list=[1,2,3,4,5,6]
print("\nTask 1]")
print("Original List : ",my_list)
my_list.reverse()               #reverse kahi return nahi krt..update krto
print("Reversed List : ",my_list)

my_list2=[2,1,4,3,2,3,2,2,1]
new_list=[]
print("\nTask 2]")
print("Original : ",my_list2)
for i in my_list2:
  if i not in new_list:
    new_list.append(i)
print("New Unique List : ",new_list)

tup=(1,2,3,4,2,1,2,2)
print("\nTask 4]")
print("Tuple : ",tup)
print("Occurrences of 2 in Tuple : ",tup.count(2))

dict1={"Name":"Soumya","Age":19}
dict2={"Gender":"Female","Dept":"Comp RL"}
print("\nTask 5]")
print("Dictionary 1 : ",dict1)
print("Dictionary 2 : ",dict2)
dict1.update(dict2)               #update kahi return nahi krt existing madhe navin add krto fkta
print("After merging 2 dictionaries : ",dict1)

print("\nTask 6]")
d1={'b':3,'c':2,'a':1}
print("Original Dict : ",d1)
sorted_dict = dict(sorted(d1.items()))
print("Sorted Dictionary : ",sorted_dict)

s1={4,2,1}
s2={6,1,2}
print("\nTask 7 & 8]")
print("Set 1 : ",s1)
print("Set 2 : ",s2)

s3=s1 & s2
#OR s3=s1.intersection(s2)
print("Intersection of s1 and s2 : ",s3)

s4=s1 | s2
#OR s4=s1.union(s2)
print("Union of s1 and s2 : ",s4)
```

#**OUTPUT** :

```
Task 1]
Original List :  [1, 2, 3, 4, 5, 6]
Reversed List :  [6, 5, 4, 3, 2, 1]

Task 2]
Original :  [2, 1, 4, 3, 2, 3, 2, 2, 1]
New Unique List :  [2, 1, 4, 3]

Task 4]
```

Tuple : (1, 2, 3, 4, 2, 1, 2, 2)
Occurrences of 2 in Tuple : 4

Task 5]
Dictionary 1 : {'Name': 'Soumya', 'Age': 19}
Dictionary 2 : {'Gender': 'Female', 'Dept': 'Comp RL'}
After merging 2 dictionaries : {'Name': 'Soumya', 'Age': 19, 'Gender': 'Female', 'Dept': 'Comp RL'}

Task 6]
Original Dict : {'b': 3, 'c': 2, 'a': 1}
Sorted Dictionary : {'a': 1, 'b': 3, 'c': 2}

Task 7 & 8]
Set 1 : {1, 2, 4}
Set 2 : {1, 2, 6}
Intersection of s1 and s2 : {1, 2}
Union of s1 and s2 : {1, 2, 4, 6}

➢ SLIP 2]

```
import pandas as pd
df=pd.read_csv("D:\Devl\cbb.csv")
print("\nTask 1]")
print("1st 5 records in dataframe : ")
print(df.head())
print("last 5 records in dataframe : ")
print(df.tail())

print("\nTask 2]")
print("No. of Records in the dataset(Rows,Columns) : ",df.shape)

print("\nTask 3]")
print("\nNo. of Missing values in each column : ")
print(df.isnull().sum())

print("\nTask 4]")
print("Maximum value present in Each column : ", df.max(numeric_only=True))

print("\nTask 5]")
for col in df.columns:
  print(f"\nColumn : {col} ")
  print(df[col].unique())

print("\nTask 6]")
print("Number of Unique Values in each Column : ",df.nunique())

print("\nTask 7]")
for col in df.columns:
  print(f"\nColumn : {col} ")
  print(df[col].value_counts())
```

➢ SLIP 3]

```
import numpy as np

print("\nTask 1]")
arr1 = np.arange(10,49)
print("1D Array : ",arr1)

print("\nTask 2]")
arr2=arr1[:15].reshape((3,5))
print("3x5 Matrix : ",arr2)

print("\nTask 3]")
new_arr=arr1[arr1%3==0]
print("Numbers Divisible by 3 : ",new_arr)

print("\nTask 4]")
a1 = np.random.randint(1,50,(3,3))
a2=np.random.randint(1,50,(3,3))
print("Array 1 : ",a1)
print("Array 2 : ",a2)

add = a1+a2
print("\nAddition of Array 1 & Array 2 : ")
print(add)
```

```
sub = a1-a2
print("\nSubtraction of Array 1 & Array 2 : ")
print(sub)

mul = a1*a2
print("\nMultiplication of Array 1 & Array 2 : ")
print(mul)

div = a1/a2
print("\nDivision of Array 1 & Array 2 : ")
print(div)
```

#OUTPUT :

```
Task 1]
1D Array :  [10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33
 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48]

Task 2]
3x5 Matrix :  [[10 11 12 13 14]
 [15 16 17 18 19]
 [20 21 22 23 24]]

Task 3]
Numbers Divisible by 3 :  [12 15 18 21 24 27 30 33 36 39 42 45 48]

Task 4]
Array 1 :  [[34 25 44]
 [41  2 21]
 [ 5 23  6]]
Array 2 :  [[11  2 45]
 [ 4 32 40]
 [49 44 17]]

Addition of Array 1 & Array 2 :
[[45 27 89]
 [45 34 61]
 [54 67 23]]

Subtraction of Array 1 & Array 2 :
[[ 23  23  -1]
 [ 37 -30 -19]
 [-44 -21 -11]]

Multiplication of Array 1 & Array 2 :
[[ 374   50 1980]
 [ 164   64  840]
 [ 245 1012  102]]

Division of Array 1 & Array 2 :
[[ 3.09090909 12.5         0.97777778]
 [10.25        0.0625      0.525     ]
 [ 0.10204082  0.52272727  0.35294118]]
```

- **SLIP 4]**

```
import numpy as np
arr1=np.linspace(1,100,15)
print("\nArray with 15 evenly spaced Numbers between 1 to 100 : \n",arr1)

arr2 = np.random.randint(1,50,(4,4))
arr2[arr2%2==0]=-1
print("\n4x4 Array (even numbers replaced with -1) : \n",arr2)

arr3=np.random.randint(1,100,(5,5))
print("\n5x5 array :")
print(arr3)
print("\n",arr3[1])
print(arr3[:,-1])
print(arr3[-1,:])

arr4 = np.arange(1,51)
print("\nARRAY 4 :",arr4,"\n")
print("\nOdd Numbers in ARRAY 4 : ")
for i in arr4:
if(i%2==1):
print(i,end=" ")

arr5=np.random.rand(3,3)
arr5_mean=arr5.mean()
print("\nARRAY 5:",arr5)
print("Mean of Array 5 : ",arr5_mean)
```

#OUTPUT:

```
Array with 15 evenly spaced Numbers between 1 to 100 :
[ 1. 8.07142857 15.14285714 22.21428571 29.28571429 36.35714286 43.42857143 50.5 57.57142857
64.64285714 71.71428571 78.78571429 85.85714286 92.92857143 100. ]
4x4 Array (even numbers replaced with -1) :
[[33 -1 -1 5] [11 25 -1 39] [31 29 5 17] [ 3 -1 -1 -1]] 5x5 array : [[66 87 75 27 40] [57 79 78 34 11] [66 70 60 54 25]
[ 8 76 98 89 88] [21 9 90 50 55]] [57 79 78 34 11] [40 11 25 88 55] [21 9 90 50 55]

ARRAY 4 : [ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37
38 39 40 41 42 43 44 45 46 47 48 49 50]
Odd Numbers in ARRAY 4 :
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49
ARRAY 5: [[0.30269301 0.79763771 0.99194758] [0.01559192 0.64737175 0.48336941] [0.61231809 0.40343741
0.53741102]]

Mean of Array 5 : 0.5324197672138146
```

- **SLIP 5** :

```
import numpy as np

arr1 = np.arange(1,17)
print("Array 1 before Reshaping : ")
print(arr1)
new_arr1 = arr1.reshape(4,4)
print("Array 1 After reshaping : ")
print(new_arr1)

arr2 = np.random.randint(10,100,(3,4))
print("Array 2 : ")
print(arr2)
arr2_trans = arr2.T
print("Transpose of Array 2 : ")
print(arr2_trans)
print("Shape of Array 2 : ",arr2_trans.shape)

arr3 = np.random.randint(1,20,10)
print("\nArray 3 : ",arr3)
arr3_sqr = np.square(arr3)
print("Squared Array 3 : ")
print(arr3_sqr)

arr4 = np.zeros((4,5))
print(arr4)

arr5 = np.arange(1,21)
arr5_new = arr5[arr5>5]
print("\nArray of elements >5")
print(arr5_new)
```

#**OUTPUT :**

```
PS C:\Users\Soumya> & C:/Users/Soumya/AppData/Local/Python/pythoncore-3.14-64/python.exe d:/Devl/slip5.py
Array 1 before Reshaping :
[ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16]
Array 1 After reshaping :
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]
 [13 14 15 16]]
Array 2 :
[[72 86 66 55]
 [24 37 94 95]
 [11 26 83 10]]
Transpose of Array 2 :
[[72 24 11]
 [86 37 26]
 [66 94 83]
 [55 95 10]]
Shape of Array 2 :  (4, 3)

Array 3 :  [18  2  7 18  4  4 17 15 16 19]
Squared Array 3 :
[324   4  49 324  16  16 289 225 256 361]

[[0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]]
```

Array of elements >5
[6 7 8 9 10 11 12 13 14 15 16 17 18 19 20]