Lab Assignment 18

Student Name : Aryan Verma

Student Id: AF0417100

Topic: Numpy Functions

KEY POINTS OF NUMPY:

- **NUMPY** IS A POWERFUL PYTHON LIBRARY DESIGNED FOR NUMERICAL COMPUTING, PROVIDING ROBUST SUPPORT FOR LARGE, MULTI-DIMENSIONAL ARRAYS AND MATRICES.

- IT INCLUDES AN EXTENSIVE ARRAY OF MATHEMATICAL FUNCTIONS THAT FACILITATE EFFICIENT OPERATIONS ON DATA STRUCTURES.

- NUMPY ARRAYS ARE STORED IN CONTIGUOUS BLOCKS OF MEMORY, RESULTING IN FASTER PERFORMANCE COMPARED TO TRADITIONAL PYTHON LISTS.

- THE LIBRARY SUPPORTS **BROADCASTING**, ENABLING OPERATIONS BETWEEN ARRAYS OF DIFFERENT SHAPES WITHOUT THE NEED FOR EXPLICIT LOOPS.

- IT OFFERS **VECTORIZED OPERATIONS**, WHICH STREAMLINE ELEMENT-WISE CALCULATIONS BY ELIMINATING THE NEED FOR MANUAL LOOPING.

- NUMPY PROVIDES COMPREHENSIVE TOOLS FOR **LINEAR ALGEBRA**, **RANDOM NUMBER GENERATION**, AND VARIOUS MATRIX OPERATIONS.

- THE LIBRARY INTEGRATES SMOOTHLY WITH OTHER POPULAR LIBRARIES LIKE **PANDAS**, **MATPLOTLIB**, AND **SCIKIT-LEARN**, ENHANCING DATA ANALYSIS AND SCIENTIFIC COMPUTING CAPABILITIES.

- NUMPY ALLOWS FOR FLEXIBLE **SHAPE MANIPULATION** (E.G., RESHAPING, FLATTENING, TRANSPOSING) TO ACCOMMODATE DIVERSE DATA HANDLING REQUIREMENTS.

- IT IS ESSENTIAL FOR PERFORMING NUMERICAL TASKS IN FIELDS SUCH AS DATA SCIENCE, MACHINE LEARNING, AND ENGINEERING APPLICATIONS.

SOME FUNCTIONS USED IN NUMPY:

- `NUMPY.ARRAY()`: CREATES AN ARRAY FROM LISTS OR TUPLES, FORMING THE FOUNDATION FOR NUMERICAL OPERATIONS.

- `NUMPY.ZEROS()`: GENERATES AN ARRAY FILLED WITH ZEROS, USEFUL FOR INITIALIZING MATRICES AND DATA STRUCTURES.

- `NUMPY.ONES()`: PRODUCES AN ARRAY FILLED WITH ONES, COMMONLY USED IN MATHEMATICAL COMPUTATIONS AND ALGORITHMS.

- `NUMPY.ARANGE()`: CREATES AN ARRAY WITH A SPECIFIED RANGE OF VALUES, FACILITATING ITERATION IN LOOPS.

- `NUMPY.LINSPACE()`: GENERATES EVENLY SPACED VALUES OVER A SPECIFIED INTERVAL, BENEFICIAL FOR PLOTTING AND FUNCTION EVALUATIONS.

- **NUMPY.RESHAPE(): CHANGES THE SHAPE OF AN ARRAY WITHOUT MODIFYING ITS DATA, AIDING IN DATA MANIPULATION AND PREPARATION.**

- **NUMPY.TRANSPOSE(): SWITCHES THE AXES OF AN ARRAY, FREQUENTLY EMPLOYED IN LINEAR ALGEBRA OPERATIONS.**

- **NUMPY.SPLIT(): DIVIDES AN ARRAY INTO MULTIPLE SUB-ARRAYS, ENABLING ORGANIZED DATA HANDLING AND ANALYSIS.**

- **NUMPY.SUM(): COMPUTES THE SUM OF ARRAY ELEMENTS, COMMONLY USED IN AGGREGATING DATA FOR ANALYSIS.**

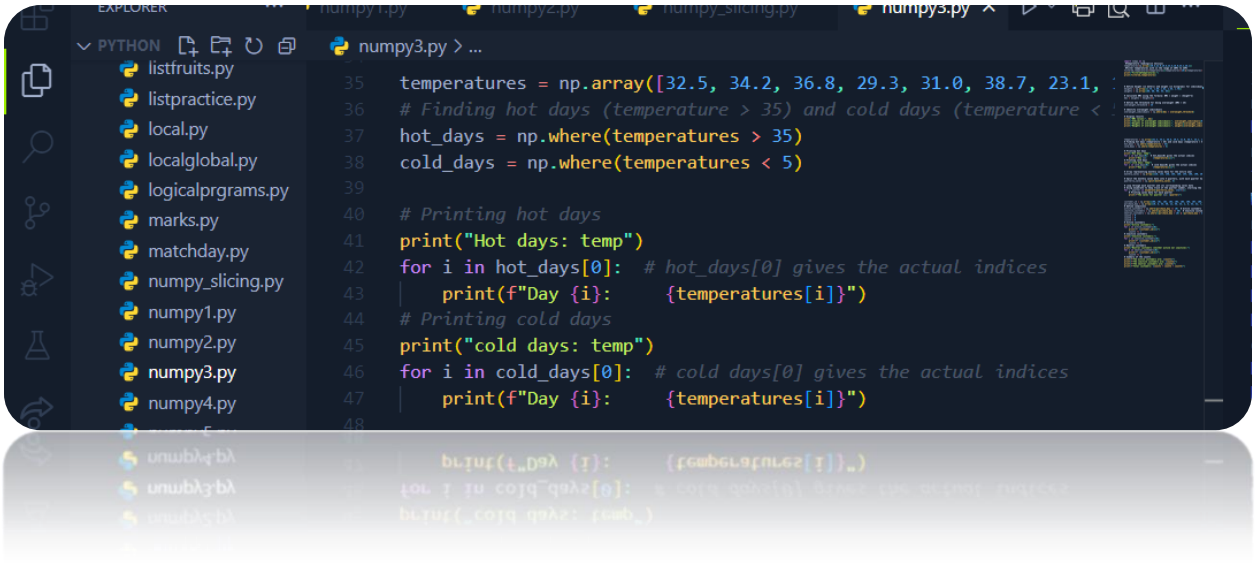
- **NUMPY.MEAN(): CALCULATES THE AVERAGE VALUE OF AN ARRAY, CRUCIAL FOR STATISTICAL ANALYSIS AND DATA INTERPRETATION.**

1. Suppose you have a dataset containing daily temperature readings for a city, and you want to identify days with extreme temperature conditions. Find days where the temperature either exceeded 35 degrees Celsius (hot day) or dropped below 5 degrees Celsius (cold day).

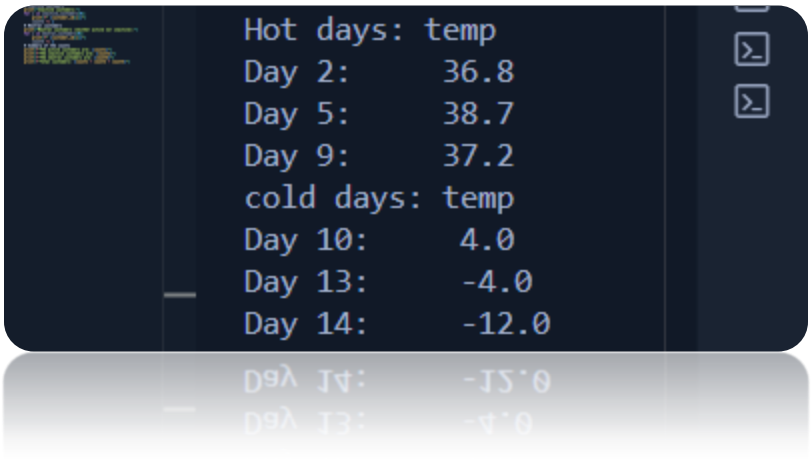
Input:

temperatures = np.array([32.5, 34.2, 36.8, 29.3, 31.0, 38.7, 23.1, 18.5, 22.8, 37.2,4,25,12,-4,-12])

Ans:



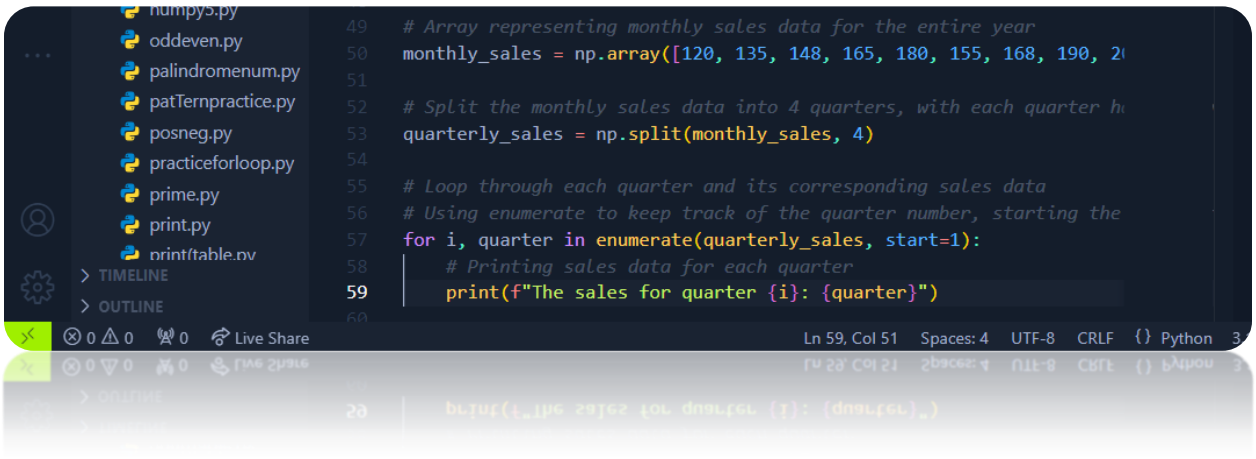
Output:



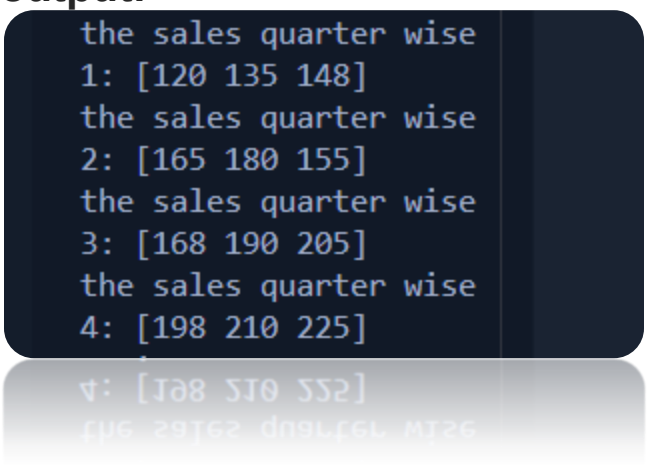
2. Suppose you have a dataset containing monthly sales data for a company, and you want to split this data into quarterly reports for analysis and reporting purposes.

Input: monthly_sales = np.array([120, 135, 148, 165, 180, 155, 168, 190, 205, 198, 210, 225])

Ans:

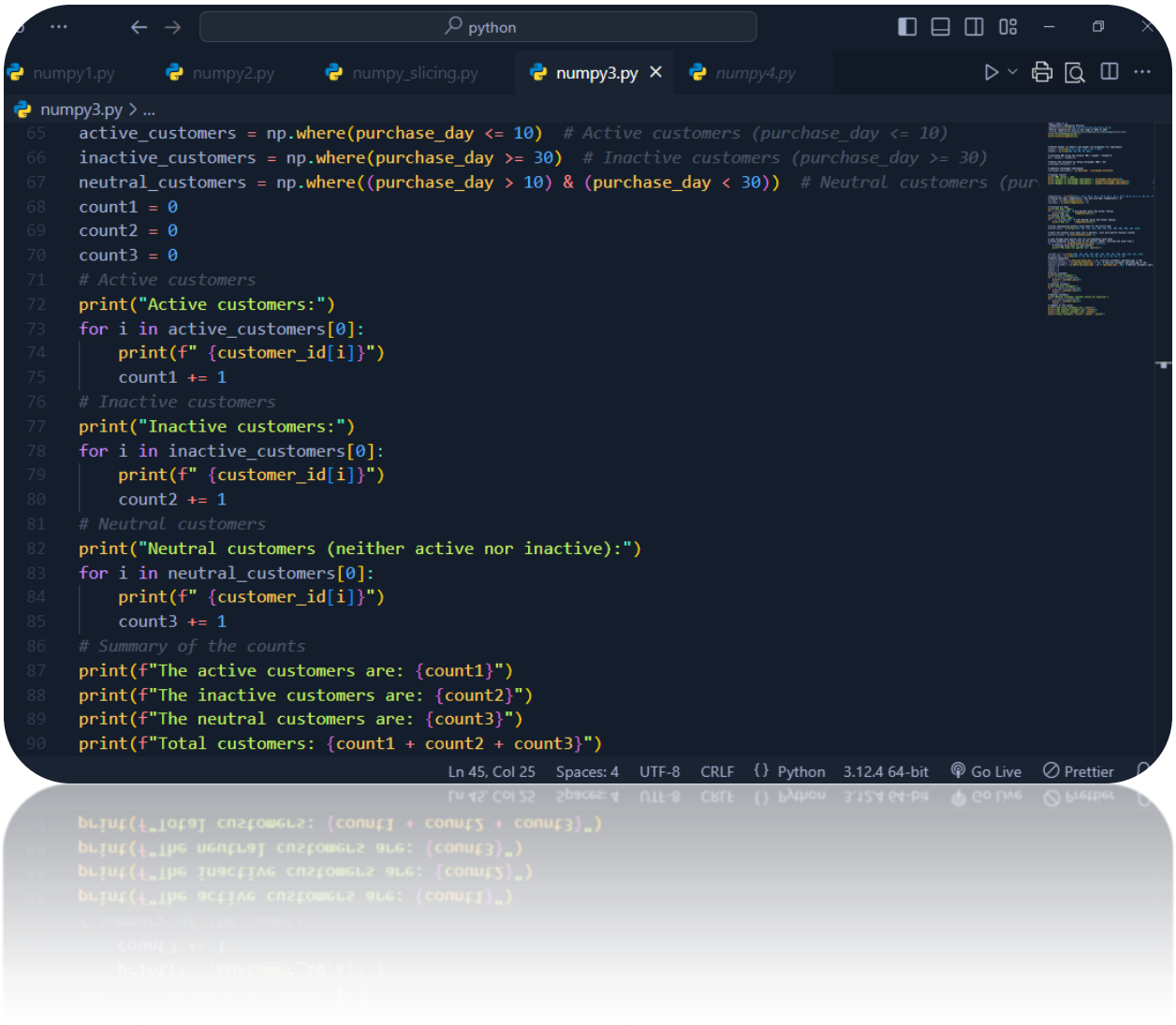


output:

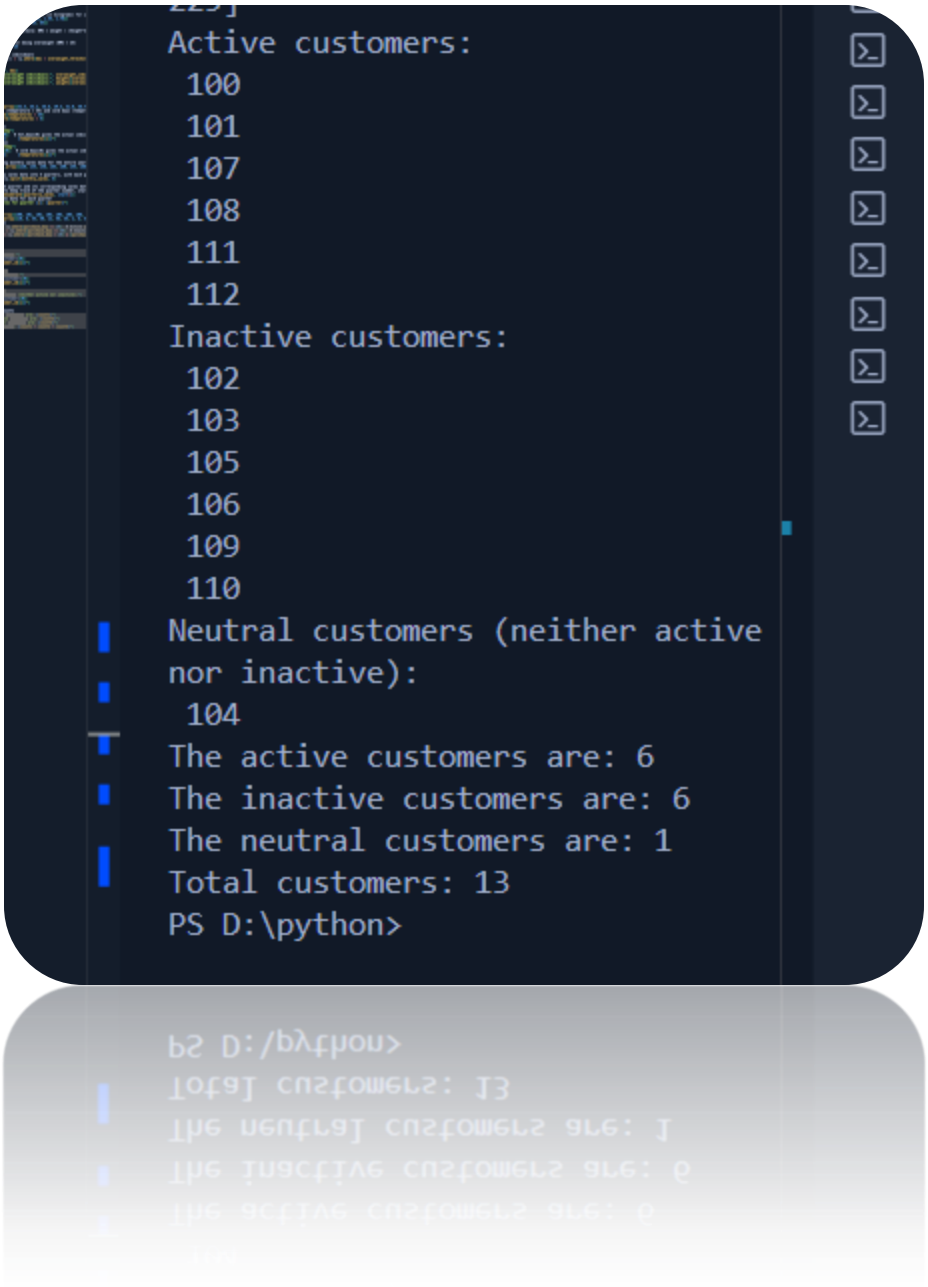


Sample program

Ans:


python
numpy1.py
numpy2.py
numpy_slicing.py
numpy3.py
numpy4.py
numpy3.py > ...
active_customers = np.where(purchase_day <= 10) # Active customers (purchase_day <= 10)
inactive_customers = np.where(purchase_day >= 30) # Inactive customers (purchase_day >= 30)
neutral_customers = np.where((purchase_day > 10) & (purchase_day < 30)) # Neutral customers (pur
count1 = 0
count2 = 0
count3 = 0
# Active customers
print("Active customers:")
for i in active_customers[0]:
print(f" {customer_id[i]}")
count1 += 1
# Inactive customers
print("Inactive customers:")
for i in inactive_customers[0]:
print(f" {customer_id[i]}")
count2 += 1
# Neutral customers
print("Neutral customers (neither active nor inactive):")
for i in neutral_customers[0]:
print(f" {customer_id[i]}")
count3 += 1
# Summary of the counts
print(f"The active customers are: {count1}")
print(f"The inactive customers are: {count2}")
print(f"The neutral customers are: {count3}")
print(f"Total customers: {count1 + count2 + count3}")
Ln 45, Col 25
Spaces: 4
UTF-8
CRLF
{} Python
3.12.4 64-bit
Go Live
Prettier

output:


Active customers:
100
101
107
108
111
112
Inactive customers:
102
103
105
106
109
110
Neutral customers (neither active nor inactive):
104
The active customers are: 6
The inactive customers are: 6
The neutral customers are: 1
Total customers: 13
PS D:\python>