SUMANTH S
AF0363570

PYTHON LAB: NUMPY FUNCTIONS

QUESTIONS:

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])

```
#import numpy library
import numpy as np

# Input temperature data
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])

# assigning threshold values
hot_day_threshold =35
cold_day_threshold = 5

# create a list
hot_days = []
cold_days = []

# traversing the array to compare the threshold values
for day, temperature in enumerate(temperatures, start =1):
  if temperature > hot_day_threshold:
     hot_days.append((day, temperature))
  elif temperature < cold_day_threshold:
     cold_days.append((day, temperature))

# print Hot days
print("Hot Days :")
print("Days"," ","Temperature (C)")
for day,temperature in hot_days:
  print(day,"    ",temperature)
print(" ")

# print Cold days
print("Cold Days:")
print("Days ","  ","Temperature (C)")
for day,temperature in cold_days:
  print(day,"     ",temperature)
```

Output:

```
Hot Days:
Days   Temperature (C)
3        36.8
6        38.7
10      37.2

Cold Days:
Days     Temperature (C)
11         4.0
14       -4.0
15       -12.0
```

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])

```
# given data to monthly sales
monthly_sales = np.array([120, 135, 148, 165, 180, 155, 168, 190, 205, 198, 210, 225])

# Reshape the array into a 4x3 matrix
quarterly_sales = monthly_sales.reshape(4, 3)

# Print the quarterly sales data
print("Quarterly Sales:")
for i, quarter in enumerate(quarterly_sales, start=1):
    print(f"Quarter {i} Sales (in thousands of Dollar) :")
    print(quarter)
    print(" ")
```

Output:

```
Quarterly Sales:
Quarter 1 Sales (in thousands of Dollar) :
[120 135 148]

Quarter 2 Sales (in thousands of Dollar) :
[165 180 155]

Quarter 3 Sales (in thousands of Dollar) :
```

```
[168 190 205]

Quarter 4 Sales (in thousands of Dollar) :
[198 210 225]
```

Suppose you have a dataset containing customer data, and you want to split this data into two groups: one group for customers who made a purchase in the last 30 days and another group for customers who haven't made a purchase in the last 30 days.
Input: customer_ids = np.array([101, 102, 103, 104, 105, 106, 107, 108, 109, 110]) last_purchase_days_ago = np.array([5, 15, 20, 25, 30, 35, 40, 45, 50, 55])

```
#assigning the data
customer_ids = np.array([101, 102, 103, 104, 105, 106, 107, 108, 109, 110])
last_purchase_days_ago = np.array([5, 15, 20, 25, 30, 35, 40, 45, 50, 60])

#creating a list
recent_customers = customer_ids[last_purchase_days_ago <= 30]
inactive_customers = customer_ids[last_purchase_days_ago > 30]

#print the output
print("Recent Customers:")
print(recent_customers)
print("\nInactive Customers:")
print(inactive_customers)
```

Output:

```
Recent Customers:
[101 102 103 104 105]

Inactive Customers:
[106 107 108 109 110]
```

Suppose you have two sets of employee data—one containing information about full-time employees and another containing information about part-time employees. You want to combine this data to create a comprehensive employee dataset for HR analysis.
Input: # Employee data for full-time employees
full_time_employees = np.array([[101, 'John Doe', 'Full-Time', 55000], [102, 'Jane Smith', 'Full-Time', 60000], [103, 'Mike Johnson', 'Full-Time', 52000]])
# Employee data for part-time employees
part_time_employees = np.array([[201, 'Alice Brown', 'Part-Time', 25000], [202, 'Bob Wilson', 'Part-Time', 28000], [203, 'Emily Davis', 'Part-Time', 22000]])

```
#importing numpy library
import numpy as np

# Employee data for full-time employees
full_time_employees = np.array([
    [101, 'John Doe', 'Full-Time', 55000],
    [102, 'Jane Smith', 'Full-Time', 60000],
    [103, 'Mike Johnson', 'Full-Time', 52000]
])

# Employee data for part-time employees
part_time_employees = np.array([
    [201, 'Alice Brown', 'Part-Time', 25000],
    [202, 'Bob Wilson', 'Part-Time', 28000],
    [203, 'Emily Davis', 'Part-Time', 22000]
])

# Combine the data for full-time and part-time employees
comprehensive_employee_data = np.concatenate((full_time_employees, part_time_employees), axis=0)

# Display the comprehensive employee dataset
print("Comprehensive Employee Dataset:")
print(comprehensive_employee_data)
```

Output:

```
Comprehensive Employee Dataset:
[['101' 'John Doe' 'Full-Time' '55000']
 ['102' 'Jane Smith' 'Full-Time' '60000']
 ['103' 'Mike Johnson' 'Full-Time' '52000']
 ['201' 'Alice Brown' 'Part-Time' '25000']
 ['202' 'Bob Wilson' 'Part-Time' '28000']
 ['203' 'Emily Davis' 'Part-Time' '22000']]
```