Artificial Intelligence (Code 417)

Class X – Record Writing

Task A

1) Write a Python program to calculate net run rate (NRR) of a cricket tournament.

Net Run Rate (NRR) is calculated as:

$$\text{NRR} = \left(\frac{\text{Total Runs Scored}}{\text{Total Overs Faced}}\right) - \left(\frac{\text{Total Runs Conceded}}{\text{Total Overs Bowled}}\right)$$

```
# Input: Total runs scored, total overs faced, total runs conceded, total overs bowled
runs_scored = float(input("Enter total runs scored by the team: "))
overs_faced = float(input("Enter total overs faced by the team: "))
runs_conceded = float(input("Enter total runs conceded by the team: "))
overs_bowled = float(input("Enter total overs bowled by the team: "))

# Calculate run rate for and against the team
run_rate_scored = runs_scored / overs_faced
run_rate_conceded = runs_conceded / overs_bowled

# Calculate the Net Run Rate (NRR)
net_run_rate = run_rate_scored - run_rate_conceded

# Output NRR
print("Net Run Rate (NRR): ",net_run_rate)
```

```
Enter total runs scored by the team: 300
Enter total overs faced by the team: 50
Enter total runs conceded by the team: 280
Enter total overs bowled by the team: 50
Net Run Rate (NRR):  0.40000000000000036
```

2) Write a Python program to check whether a given input is uppercase or lowercase or a digit

```
#take user input for a single character
val = input("Enter a character:")
#check the input and print whether it is a digit or uppercase or lowercase
if val.isupper():
  print(val,"is an Upper case input")
elif val.islower():
  print(val,"is lower case input")
elif val.isdigit():
  print(val,"is a digit")
else:
  print(val,"is a special character")
```

```
Enter a character:a
a is lower case input
```

3) Write a Python program to find a minimum number in a user created list. Hint: Use min ( ) function, if required.

```
#create an empty list
mylist=[]
#take size input from user
size=int(input("Enter the size of the list:"))
#run a loop to collect the list items and append it to the list
for i in range(size):
  item=eval(input("Enter list item:"))
  mylist.append(item)
#display the final list created
print(mylist)
#check for the min value and print it
minimum = min(mylist)
print("The minimum number in the above list is",minimum)
```

```
Enter the size of the list:5
Enter list item:-3
Enter list item:2.5
Enter list item:5
Enter list item:0
Enter list item:-8
[-3, 2.5, 5, 0, -8]
The minimum number in the above list is -8
```

Note: Python's eval function interprets the character string and determines its type automatically. So if you enter the number 25, the eval function converts it to an integer, assigns it to the variable age and gives age the type int .

4) Demonstrate the following list functions: Append, pop, remove and delete.

```
#create a list
mylist = [1,2,3,4,5]
#demo of append
mylist.append("I am from append")
print("Updated list after append",mylist)
#demo of pop
mylist.pop()
print("Updated list after pop",mylist)
#demo of remove
mylist.remove(1)
print("Updated list after remove",mylist)
#demo of delete
print("Deleting the mylist")
del(mylist)
#checking whether the list is successfuly deleted or not
print(mylist)
```

```
Updated list after append [1, 2, 3, 4, 5, 'I am from append']
Updated list after pop [1, 2, 3, 4, 5]
Updated list after remove [2, 3, 4, 5]
Deleting the mylist
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-3894fa099806> in <cell line: 16>()
     14 del(mylist)
     15 #checking whether the list is successfuly deleted or not
---> 16 print(mylist)

NameError: name 'mylist' is not defined
```

5)Write a Python program to check whether the user input is Armstrong number or not.

An Armstrong number is a special kind of number in math. It's a number that equals the sum of its digits, each raised to a power. For example, if you have a number like 153, it's an Armstrong number because 1^3 + 5^3 + 3^3 equals 153.

```
#take 3 digit user input
num = int(input("Enter a 3 didgit number:"))
#set accumulator to 0
sum = 0
#create a temperory variable
temp = num
#run loop to find armstrong
while temp>0:
  digit = temp%10
  sum = sum+(digit**3)
  temp = temp//10
#evaluate input as armstrong or not
if num == sum:
  print(num,"is an Armstrong number")
else:
  print(num,"is not an Armstrong number")
```

```
Enter a 3 didgit number:153
153 is an Armstrong number
```

6)Write a Python program to generate the following pattern:

1

2 3

4 5 6

7 8 9 10

```
#initialise
num = 1

for i in range(1, 5):  # The outer loop controls the number of rows
    for j in range(i):  # The inner loop controls the number of elements in each row
        print(num, end=" ")
        num += 1
    print()  # Moves to the next line after each row
```

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

7)Write a Python program to create a user defined list of N numbers. Swap a number with its successor if it is divisible by 5.

Example:

Mylist = [1,2,3,4,5,6]

Output: [1,2,3,4,6,5]

```
# Input the list size and elements
N = int(input("Enter the number of elements in the list: "))
mylist = []
for i in range(N):
    num = int(input(f"Enter element {i+1}: "))
    mylist.append(num)

# Swapping numbers divisible by 5 with their successors
for i in range(len(mylist) - 1):
    if mylist[i] % 5 == 0:
        # Swap the current element with the next one
        mylist[i], mylist[i + 1] = mylist[i + 1], mylist[i]

# Output the modified list
print("Output:", mylist)
```

```
Enter the number of elements in the list: 5
Enter element 1: 2
Enter element 2: 3
Enter element 3: 5
Enter element 4: 4
Enter element 5: 1
Output: [2, 3, 4, 1, 5]
```

8) Write a python program to count of frequency of every element in a list

Example: Mylist = [1,1,3,4,4]

Output:

Frequency of each element:

1: 2 times

3: 1 times

4: 2 times

```
# Input the list size and elements
N = int(input("Enter the number of elements in the list: "))
mylist = []
for i in range(N):
    num = int(input(f"Enter element {i+1}: "))
    mylist.append(num)

# List to store elements that have been counted
counted = []

print("Frequency of each element:")
```

```
# Count frequency manually
for i in mylist:
    if i not in counted:  # Check if the element has already been counted
        count = mylist.count(i)
        print(f"{i}: {count} times")
        counted.append(i)  # Add element to counted list to avoid recounting
```

```
Enter the number of elements in the list: 5
Enter element 1: 1
Enter element 2: 1
Enter element 3: 2
Enter element 4: 3
Enter element 5: 3
Frequency of each element:
1: 2 times
2: 1 times
3: 2 times
```

9) Write a Python program to check whether an element exists within a tuple.

```
# Input the tuple
mytuple = (1, 2, 3, 4, 5)  # You can define your tuple here

# Input the element to check
element = int(input("Enter the element to check: "))

# Check if the element exists in the tuple
if element in mytuple:
    print(f"{element} exists in the tuple.")
else:
    print(f"{element} does not exist in the tuple.")
```

```
Enter the element to check: 3
3 exists in the tuple.
```

10)Write a Python program to demonstrate tuple slicing.

```
# Define a tuple
mytuple = (10, 20, 30, 40, 50, 60, 70, 80)

# Demonstrate tuple slicing
print("Original Tuple:", mytuple)

# Slice the tuple from index 1 to 4 (excluding index 4)
slice1 = mytuple[1:4]
print("Sliced Tuple (index 1 to 3):", slice1)

# Slice the tuple from the beginning to index 3 (excluding index 3)
slice2 = mytuple[:3]
print("Sliced Tuple (beginning to index 2):", slice2)

# Slice the tuple from index 3 to the end
slice3 = mytuple[3:]
print("Sliced Tuple (from index 3 to end):", slice3)

# Slice the entire tuple (creates a copy)
slice4 = mytuple[:]
print("Sliced Tuple (entire tuple):", slice4)

# Slice the tuple with a step of 2
slice5 = mytuple[::2]
print("Sliced Tuple (with step of 2):", slice5)

# Slice the tuple in reverse order
slice6 = mytuple[::-1]
print("Sliced Tuple (reversed):", slice6)
```

```
Original Tuple: (10, 20, 30, 40, 50, 60, 70, 80)
Sliced Tuple (index 1 to 3): (20, 30, 40)
Sliced Tuple (beginning to index 2): (10, 20, 30)
Sliced Tuple (from index 3 to end): (40, 50, 60, 70, 80)
Sliced Tuple (entire tuple): (10, 20, 30, 40, 50, 60, 70, 80)
```

```
Sliced Tuple (with step of 2): (10, 30, 50, 70)
Sliced Tuple (reversed): (80, 70, 60, 50, 40, 30, 20, 10)
```

Task B

1) Write a Python program to calculate the mean, median and mode on the list scores = [10, 20, 10, 30, 10] using NumPy.

```
import numpy as np

# Define the list of scores
scores = [10, 20, 10, 30, 10]

# Calculate the mean
mean = np.mean(scores)
print("Mean:", mean)

# Calculate the median
median = np.median(scores)
print("Median:", median)

# Calculate the mode (using NumPy's unique and bincount)
values, counts = np.unique(scores, return_counts=True)
mode = values[np.argmax(counts)]
print("Mode:", mode)
```

```
Mean: 16.0
Median: 10.0
Mode: 10
```

Note: Unfortunately, NumPy doesn't have a built-in function to directly calculate the mode

2) Consider the following data of a clothes store and plot the data on the line chart and customize the chart as you wish:

Month	Jeans	T-Shirts	Shirts
March	1500	4400	6500
April	3500	4500	5000
May	6500	5500	5800
June	6700	6000	6300
July	6000	5600	6200
August	6800	6300	4500

```
import matplotlib.pyplot as plt

# Data for the clothes store sales
months = ["March", "April", "May", "June", "July", "August"]
jeans = [1500, 3500, 6500, 6700, 6000, 6800]
tshirts = [4400, 4500, 5500, 6000, 5600, 6300]
shirts = [6500, 5000, 5800, 6300, 6200, 4500]

# Plot the data
plt.figure(figsize=(10, 6))

plt.plot(months, jeans, marker='o', label="Jeans", color='blue', linewidth=2)
plt.plot(months, tshirts, marker='o', label="T-Shirts", color='green', linewidth=2)
plt.plot(months, shirts, marker='o', label="Shirts", color='red', linewidth=2)

# Customizing the chart
plt.title("Clothes Store Sales Over Months", fontsize=16, fontweight='bold')
plt.xlabel("Month", fontsize=14)
plt.ylabel("Sales", fontsize=14)
plt.grid(True, linestyle='--', alpha=0.6)
plt.legend(loc='upper right', fontsize=12)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)

# Display the chart
```

```
plt.tight_layout()
plt.show()
```

3) Write a Python program to display a scatter chart for the following points (2,5), (9,10), (8,3), (5,7), and (6,18).

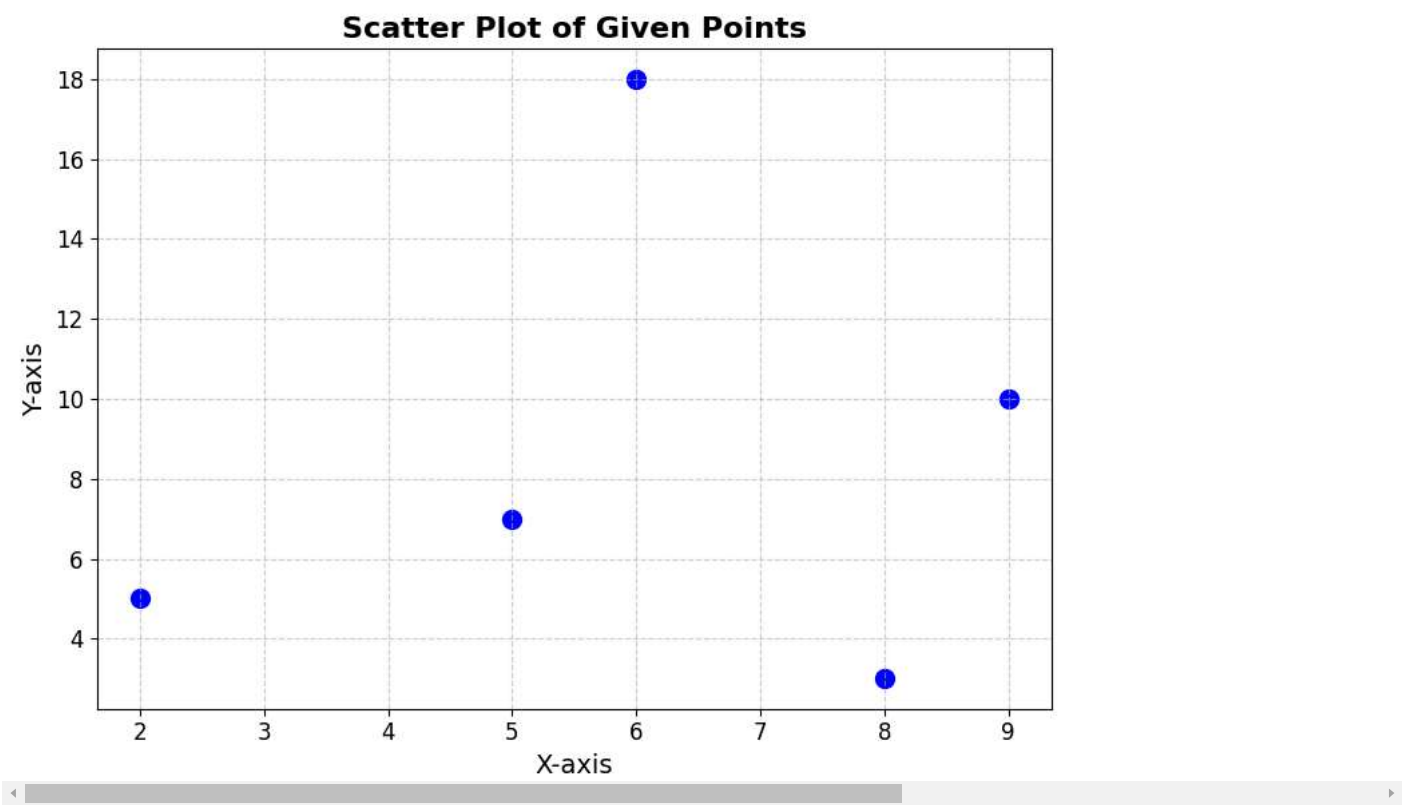
```
import matplotlib.pyplot as plt

# Data points
x = [2, 9, 8, 5, 6]
y = [5, 10, 3, 7, 18]

# Plot the scatter chart
plt.figure(figsize=(8, 6))
plt.scatter(x, y, color='blue', s=100, marker='o')

# Customizing the chart
plt.title("Scatter Plot of Given Points", fontsize=16, fontweight='bold')
plt.xlabel("X-axis", fontsize=14)
plt.ylabel("Y-axis", fontsize=14)
plt.grid(True, linestyle='--', alpha=0.6)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)

# Display the chart
plt.tight_layout()
plt.show()
```



4) Create a CSV file with following columns: Sl no, Song name, duration, movie. Write a Python code to read the CSV file saved in your system and display the first 5 rows.

Steps:

1. Open Excel and create the table.
2. Save it as a CSV file. Sample shown below.

3. Upload this CSV file to Google Colab.
4. Copy paste the path link to the code.

```
import pandas as pd

# Replace 'your_file.csv' with the path to your CSV file
file_path = '/content/Songs list.csv'

# Read the CSV file
data = pd.read_csv(file_path)

print("Displaying the CSV file")

# Display the all content of the CSV file
print(data)

print()

print("Displaying only first 5 rows")

# Display the first 5 rows
print(data.head())
```

```
Displaying the CSV file
    Sl No                    Song name Duration          Movie/Album
```

```
0      1                     Ghungroo     5:02                  War
1      2                 Chak De India     6:11       Chak De! India
2      3               Dynamite by BTS     3:19                   Be
3      4             Replay by Shinee      3:35     The SHINee World
4      5  Sorry, Sorry by Super Junior     3:52         Sorry, Sorry
5      6   Blank Space by Taylor Swift     3:51                 1989
6      7                   Life is God     3:57        High off life
7      8             Watermelon Sugar      2:54         Harry Styles
8      9                       Jai Ho      3:01  Slumdog Millionaire
9     10               Ghanana Ghanana     6:10                Lagaan

Displaying only first 5 rows
   Sl No                     Song name Duration       Movie/Album
0      1                     Ghungroo     5:02               War
1      2                 Chak De India     6:11    Chak De! India
2      3               Dynamite by BTS     3:19                Be
3      4             Replay by Shinee      3:35  The SHINee World
4      5  Sorry, Sorry by Super Junior     3:52      Sorry, Sorry
```

5) Write a Python program to read an image and display it as grayscale using Python.

```
import cv2
import matplotlib.pyplot as plt

# Read the image
image = cv2.imread('/content/tom and jerry.jpg')

# Check if the image was loaded successfully
if image is None:
    print("Error: Could not load image.")
else:
    # Convert the image to grayscale
    gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # Set up the figure
    plt.figure(figsize=(12, 6))

    # Display the original image
    plt.subplot(1, 2, 1)  # 1 row, 2 columns, first subplot
    plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))  # Convert BGR to RGB for proper display
    plt.title("Original Image", fontsize=16)
    plt.axis('off')  # Hide axes

    # Display the grayscale image
    plt.subplot(1, 2, 2)  # 1 row, 2 columns, second subplot
    plt.imshow(gray_image, cmap='gray')
    plt.title("Grayscale Image", fontsize=16)
    plt.axis('off')  # Hide axes

    # Show the plot
    plt.tight_layout()
    plt.show()
```