**SECTION A – Python**

# 1. Calculate sum of numbers until user enters zero

total = 0

while True:

num = int(input("Enter a number (0 to stop): "))

if num == 0:

break

total += num

print("Sum of numbers:", total)

# 2. Input two numbers and add them

a = int(input("Enter first number: "))

b = int(input("Enter second number: "))

print("Sum:", a + b)

# 3. Swap two variables

x = int(input("Enter first number: "))

y = int(input("Enter second number: "))

x, y = y, x

print("After swapping: x =", x, "y =", y)

# 4. Convert kilometers to miles

km = float(input("Enter distance in kilometers: "))

miles = km \* 0.621371

print("Distance in miles:", miles)

# 5. Convert Celsius to Fahrenheit

celsius = float(input("Enter temperature in Celsius: "))

fahrenheit = (celsius \* 9/5) + 32

print("Temperature in Fahrenheit:", fahrenheit)

# 6. Find area and circumference of a circle

radius = float(input("Enter radius of the circle: "))

area = 3.14159 \* radius \*\* 2

circumference = 2 \* 3.14159 \* radius

print("Area:", area)

print("Circumference:", circumference)

# 7. Calculate simple interest

principal = float(input("Enter principal amount: "))

rate = float(input("Enter rate of interest: "))

time = float(input("Enter time in years: "))

simple\_interest = (principal \* rate \* time) / 100

print("Simple Interest:", simple\_interest)

# 8. Input marks of 5 subjects and find total and percentage

marks = []

for i in range(1, 6):

mark = float(input(f"Enter marks for subject {i}: "))

marks.append(mark)

total = sum(marks)

percentage = total / 5

print("Total marks:", total)

print("Percentage:", percentage, "%")

# 9. Check if a number is odd or even

num = int(input("Enter a number: "))

if num % 2 == 0:

print(num, "is Even")

else:

print(num, "is Odd")

# 10. Check if a year is a leap year or not

year = int(input("Enter a year: "))

if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):

print(year, "is a Leap Year")

else:

print(year, "is not a Leap Year")

**SECTION B- Data Science:**

**Q1**

import matplotlib.pyplot as plt

items = ['Ordinary bread', 'Fruit Bread', 'Cakes and pastries', 'Biscuits', 'Others']

sales = [320, 80, 160, 120, 40]

plt.pie(sales, labels=items, autopct='%1.1f%%', startangle=140)

plt.title("Sales in a Baker's Shop")

plt.show()

**Q2**

import matplotlib.pyplot as plt

students = ['Ajay', 'Bali', 'Dipti', 'Faiyaz', 'Geetika', 'Hari']

marks = [450, 500, 300, 360, 400, 540]

plt.bar(students, marks, color='blue')

plt.title('Marks of Students')

plt.xlabel('Students')

plt.ylabel('Marks')

plt.show()

**Q3**

import matplotlib.pyplot as plt

categories = ['Beast Animals', 'Other Land Animals', 'Birds', 'Water Animals', 'Reptiles']

numbers = [150, 400, 225, 175, 50]

plt.pie(numbers, labels=categories, autopct='%1.1f%%', startangle=140)

plt.title('Animals in a Zoological Park')

plt.show()

**Q4**

import matplotlib.pyplot as plt

activities = ['School', 'Homework', 'Play', 'Sleep', 'Others']

hours = [7, 4, 2, 8, 3]

plt.pie(hours, labels=activities, autopct='%1.1f%%', startangle=140)

plt.title('Time Spent by a Student')

plt.show()

**Q5**

import matplotlib.pyplot as plt

religions = ['Hindu', 'Nepali', 'Islam', 'Christian']

workers = [450, 270, 255, 105]

plt.pie(workers, labels=religions, autopct='%1.1f%%', startangle=140)

plt.title('Workers by Religion')

plt.show()

**Q6**

import matplotlib.pyplot as plt

religions = ['Hindu', 'Nepali', 'Islam', 'Christian']

workers = [450, 270, 255, 105]

plt.pie(workers, labels=religions, autopct='%1.1f%%', startangle=90)

plt.title("Workers by Religion")

plt.show()

**SECTION –C -Computer Vision**

**Q1**

import cv2

# Load the image

image = cv2.imread('image.jpg')

# Display the image

cv2.imshow('Image', image)

cv2.waitKey(0)

cv2.destroyAllWindows()

**Q2**

import cv2

# Load the image

image = cv2.imread('image.jpg')

# Convert BGR to RGB

rgb\_image = cv2.cvtColor(image, cv2.COLOR\_BGR2RGB)

# Display the RGB image

cv2.imshow('RGB Image', rgb\_image)

cv2.waitKey(0)

cv2.destroyAllWindows()

**Q3**

import cv2

import numpy as np

# Load the image

image = cv2.imread('image.jpg', cv2.IMREAD\_GRAYSCALE)

# Find min and max pixel values

min\_val, max\_val, \_, \_ = cv2.minMaxLoc(image)

print(f"Minimum pixel value: {min\_val}, Maximum pixel value: {max\_val}")

**Q4**

import cv2

# Load the image

image = cv2.imread('image.jpg')

# Get the shape of the image

height, width, channels = image.shape

print(f"Height: {height}, Width: {width}, Channels: {channels}")