UIET CSJM UNIVERSITY KANPUR

DIGITAL IMAGE PROCESSING

DIT-S401

LAB FILE

Bachelor of Technology

Information Technology

Submitted By:

Manshi Sharma
CSJMA20001390155
(2k20 batch)

Submitted To:

Mr. Hemant Kumar

INDEX

lab-1-1

November 1, 2023

LAB 1

Write a small program to print your CV.

[ ]:
```
def print_cv():
    # Personal Information
    print("Manshi Sharma")
    print("Bachelors of Technology")
    print("Information Technology")
    print("University Institute of Engineering and Technology,")
    print("CSJMU, KANPUR")
    print("Education")
    print("+91-9045148449")
    print("manshisharma1372@gmail@
    gmail.com")
    print("GitHub Profile")
    print("LinkedIn Profile")

    #  Education
    print("Education")
    print("· University Institute of Engineering and Technology, CSJMU, KANPUR")
    print("   Bachelor of Technology (Information Technology)")

    # Experience
    print("Experience")
    print("2020-2024")
    print("CGPA: 9.22")
```

```
    print("• Learn and Build(Intern)")
    print("    Full Stack Developer")
    print("    May 2022 - Jul 2022")
    print("    Remote- Demonstrated high autonomy and proficiency by␣
↪single-handedly conceiving, designing, and implementing three")
    print("    distinct full-stack web applications using Python Django. These␣
↪included a comprehensive To-Do List, an intuitive")
    print("    Comment Management system, and a highly functional Admission␣
↪Portal.- Exhibited self-reliance in navigating the entire project lifecycle,␣
↪from concept to deployment. Effectively harnessed")
    print("    Python and Django, optimizing database operations and resolving␣
↪coding challenges independently, resulting in")
    print("    seamless functionality and exceptional user experiences.-␣
↪Successfully met project milestones and maintained high-quality standards as␣
↪the sole contributor. These solo")
    print("    endeavors underscored not only technical expertise but also the␣
↪ability to work independently and deliver fully")
    print("    functional web applications.")
    print("• SmartInternz(Intern)")
    print("    Android Developer")
    print("    Aug 2022 - Sep 2022")
    print("    Remote- Underwent intensive training via the Google Developer␣
↪Platform, mastering Android app development using")
    print("    cutting-edge technologies like Android Studio and Kotlin. Acquired␣
↪a strong foundation in mobile app development, ensuring a solid␣
↪understanding of the Android ecosystem- Actively engaged in hands-on␣
↪learning, applying newfound knowledge to develop a series of practical␣
↪projects")
    print("    throughout the training program. These included various minor␣
↪projects that facilitated a gradual and deeper")
    print("    understanding of Android development concepts, tools, and best␣
↪practices.- Demonstrated proficiency by independently conceiving, designing,␣
↪and developing a fully functional Shopping")
    print("    Grocery app. Leveraged Android Studio and Kotlin to create an␣
↪intuitive and user-friendly application, showcasing")
    print("    strong problem-solving skills and the ability to deliver a␣
↪polished final product.")

    # Personal Projects
    print("Personal Projects")
    print("• Student Admission Application")
    print("    Designed and built a comprehensive student admission portal from␣
↪the ground up.")
    print("    May 2022 - Jul 2022")
```

```
    print("    Tools & technologies used: Python, Django, Bootstrap- Designed and␣
↪implemented a full-stack student admission portal with Python Django,␣
↪incorporating a database")
    print("    and CRUD functionality to streamline data management and enhance␣
↪user interaction.")
    print("• Employee Management System")
    print("    A Full Stack Application")
    print("    Nov 2022")
    print("    Tools & technologies used: Python, Tkinter library, mySql-␣
↪Designed and developed an efficient employee management system using Python,␣
↪tkinter for the user interface,")
    print("    and MySQL for the database, optimizing workforce organization and␣
↪data management.)")

    # Technical Skills and Interests
    print("Technical Skills and  Interests")
    print("Languages: C++, Python, Java, Java Script, CSS")
    print("Developer Tools: Visual Code, pip(Python), git")
    print("Frameworks: Django, Bootstrap")
    print("Databases: MySql, SqLite")
    print("Soft Skills: Problem-Solving, Data Structures, and Algorithms,␣
↪Critical Thinking, Creativity, Leadership")
    print("Areas of Interest: Professional Development, Health Wellness")

# Call the function to print the CV
print_cv()
```

Write a program that takes the month (1,2,3……12) as input. Print whether the season is summer, winter, spring, or autumn depending upon the input method.

[ ]:
```
# Input the month as an integer (1 to 12)
month = int(input("Enter the month (1-12): "))

# Define lists for each season
winter_months = [12, 1, 2]
spring_months = [3, 4, 5]
summer_months = [6, 7, 8]
autumn_months = [9, 10, 11]

# Determine the season using if-else statements
if month in winter_months:
    season = "Winter"
elif month in spring_months:
    season = "Spring"
elif month  in summer_months:
    season = "Summer"
elif month in autumn_months:
```

```
    season = "Autumn"
else:
    season = "Invalid month"

# Print the determined season
print(f"The season for month {month} is {season}.")
```

1 To determine whether a year is a leap year or not.

[ ]:
```
# Input a year as an integer
year = int(input("Enter a year: "))

# Check if it's a leap year
if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
    print(f"{year} is a leap year.")
else:
    print(f"{year} is not a leap year.")
```

write a program that takes a sentence as input. Compute the frequency of each words and print them.

[ ]:
```
# Input a sentence
sentence = input("Enter a sentence: ")

# Remove punctuation and convert to lowercase
sentence = sentence.lower()
for char in '.,?!':
    sentence = sentence.replace(char, '')

# Split the sentence into words
words = sentence.split()

# Create a dictionary to store word frequencies
word_freq = {}

# Calculate word frequencies
for word in words:
    if word in word_freq:
        word_freq[word] += 1
    else:
        word_freq[word] = 1

# Print the word frequencies
print("Word Frequencies:")
for word, freq in word_freq.items():
    print(f"{word}: {freq}")
```

lab-2

November 1, 2023

1.Write a program to generate a dictionary that contains (i,sqrt(i)),where i is an integer between 1 and n.n is a number input by the user.

[ ]:
```
import math

n = int(input("Enter a value for n: "))

result_dict = {i: math.sqrt(i) for i in range(1, n + 1)}

print(result_dict)
```

2.Write a simple calculator program using functions add,sub,mul,div.The program should accept two numbers and operator and call the corresponding function to perform the operation.

[ ]:
```
def add(x, y):
    return x + y

def sub(x, y):
    return x - y

def mul(x, y):
    return x * y

def div(x, y):
    if y == 0:
        return "Division by zero is not allowed"
    return x / y

num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))
operator = input("Enter an operator (+, -, *, /): ")


if operator == '+':
    result = add(num1, num2)
elif operator == '-':
    result = sub(num1, num2)
elif operator == '*':
```

```
    result = mul(num1, num2)
elif operator == '/':
    result = div(num1, num2)
else:
    result = "Invalid operator"


print(f"Result: {result}")
```

3.Write a function that generates list with values that are square of number between 1 and 20.

[ ]:
```
def generate_squares():
    squares = [i ** 2 for i in range(1, 21)]
    return squares

squares_list = generate_squares()

print(squares_list)
```

4.Define a class names Shape with static method printType.Define methods draw() and area().Now define two class Rectangle and Triangle.Rectangle has two attributes length and width.the Triangle class has attributes a,b and c.Override the two methods of shape class.demonstrate the functionality of class by creating its objects.

[ ]:
```
class Shape:
    @staticmethod
    def printType():
        print("This is a generic shape.")

    def draw(self):
        pass

    def area(self):
        pass

class Rectangle(Shape):
    def __init__(self, length, width):
        self.length = length
        self.width = width

    def draw(self):
        print(f"Drawing a rectangle with length {self.length} and width {self.
↪width}")

    def area(self):
        return self.length * self.width
```

```
class Triangle(Shape):
    def __init__(self, a, b, c):
        self.a = a
        self.b = b
        self.c = c

    def draw(self):
        print(f"Drawing a triangle with sides a={self.a}, b={self.b}, and
 ↪c={self.c}")

    def area(self):
        s = (self.a + self.b + self.c) / 2
        return (s * (s - self.a) * (s - self.b) * (s - self.c)) ** 0.5

length = float(input("Enter the length of the rectangle: "))
width = float(input("Enter the width of the rectangle: "))

a = float(input("Enter the length of side "a" of the triangle: "))
b = float(input("Enter the length of side "b" of the triangle: "))
c = float(input("Enter the length of side "c" of the triangle: "))


Shape.printType()

rectangle = Rectangle(length, width)
rectangle.draw()
print(f"Area of the rectangle: {rectangle.area()}")

triangle = Triangle(a, b, c)
triangle.draw()
print(f"Area of the triangle: {triangle.area()}")
```

5.Using recursion,write a program to calculate the reverse of a string.

[ ]:
```
def reverse_string(input_string):

    if len(input_string) <= 1:
        return input_string

    return reverse_string(input_string[1:]) + input_string[0]

input_str = input("Enter a string: ")

reversed_str = reverse_string(input_str)

print("Reversed string:", reversed_str)
```

lab-3

November 1, 2023

NUMPY LIBRARY

Import numpy as np.

[ ]:
```
import numpy as np
```

Create an array of shape(2,3,4) of zeros.

[ ]:
```
zeros_array = np.zeros((2, 3, 4))

print(zeros_array)
```

Create an array of shape(2,3,4) of ones.

[ ]:
```
ones_array = np.ones((2, 3, 4))

print(ones_array)
```

Create an array with values 0 to 999 using np.arrange function.

[ ]:
```
result_array = np.arange(1000)

print(result_array)
```

Create an array from the list[2,3.2,5.5,-6.4,-2.2,2.4] and assign it to the variable 'a'

[ ]:
```
a = np.array([2, 3.2, 5.5, -6.4, -2.2, 2.4])

print(a)
```

Do you know what a[1] will equal?Print it to see.

[ ]:
```
print(a[1])
```

Try printing a[1:4] to see what that equals.

[ ]:
```
print(a[1:4])
```

Create a 2-D array from the following list and assign it to the variable "a":[[2,3.2,5.5,-6.4,2.2,2.4],[1,22,4,0.1,5.3,-9],[3,1,2.1,21,1.1,-2]]

[ ]:
```
a = np.array([[2, 3.2, 5.5, -6.4, 2.2, 2.4],
              [1, 22, 4, 0.1, 5.3, -9],
              [3, 1, 2.1, 21, 1.1, -2]])

print(a)
```

Can you guess what the following slices are equal to?Print them to check your understanding.a[:,3]a[1:4,0:4]a[1:,2]

[ ]:
```
print("a[:,3]:")
print(a[:,3])

print("\na[1:4,0:4]:")
print(a[1:4,0:4])

print("\na[1:,2]:")
print(a[1:,2])
```

Create a 2-D array of shape(2,4) containing two lists(range(4),range(10,14))and assign it to the variable “arr”.Print the size of the array .Print the maximum and minimum of the array.

[ ]:
```
arr = np.array([list(range(4)), list(range(10, 14))])
print(arr)
print("Size of the array:", arr.shape)

print("Maximum value:", arr.max())
print("Minimum value:", arr.min())
```

Continue to use the array “arr” as defined above.Print the array re-shaped to (2,2,2).Print the array transposed.Print the array flattened to a single dimension.Print the array converted to floats.

[ ]:
```
arr = np.array([list(range(4)), list(range(10, 14))])

reshaped_arr = arr.reshape((2, 2, 2))

transposed_arr = arr.T

flattened_arr = arr.flatten()

float_arr = arr.astype(float)

print("Reshaped array to (2, 2, 2):\n", reshaped_arr)

print("\nTransposed array:\n", transposed_arr)

print("\nFlattened array:\n", flattened_arr)

print("\nArray converted to floats:\n", float_arr)
```

Create an array counting from 1 to 20 inclusive.

[ ]:
```
counting_array = np.arange(1, 21)

print(counting_array)
```

The array of multiples of 3 greater than 0 and less than 30.

[ ]:
```
multiples_of_3 = np.arange(3, 30, 3)

print(multiples_of_3)
```

The array of 8 equally spaced floats x where 0<=x<=1

[ ]:
```
equally_spaced_floats = np.linspace(0, 1, 8)

print(equally_spaced_floats)
```

Use np.arrange and reshape to create the array A=[[1 2 3 4][5 6 7 8]]

[ ]:
```
array_1d = np.arange(1, 9)

A = array_1d.reshape((2, 4))

print(A)
```

use np.array to create array B=[1 2]

[ ]:
```
B = np.array([1, 2])

print(B)
```

Use broadcasting to add B to A to create the final array A+B

[ ]:
```
B = B.reshape(2, 1)

result = A + B

print(result)
```

lab-4

November 1, 2023

[ ]:
```
import cv2
import matplotlib.pyplot as plt
import numpy as np
```

[ ]:
```
img = cv2.imread("mountain.jpg")
```

[ ]:
```
img
```

[ ]:
```
plt.imshow(img)
```

[ ]:
```
img.shape
```

[ ]:
```
img.shape[0]
```

[ ]:
```
img.shape[1]
```

[ ]:
```
img2 = cv2.resize(img,(300,300),2)
```

[ ]:
```
img2.shape
```

[ ]:
```
plt.imsave("NewImage.jpg",img2)
```

[ ]:
```
plt.imshow(img2)
```

[ ]:
```
img = cv2.imread("mountain.jpg",4)
plt.imshow(img)
```

[ ]:
```
import cv2
from google.colab.patches import cv2_imshow

image = cv2.imread('mountain.jpg')
cv2_imshow(image)
cv2.waitKey(0)
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
cv2_imshow( gray_image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

lab-5

November 1, 2023

DRAWING THE OLYMPIC LOGO FROM SCRATCH

[ ]:
```
import cv2
import numpy as np
from google.colab.patches import cv2_imshow

canvas = np.ones((800, 800, 3), dtype=np.uint8) * 255

colors = [ (255, 0, 0),(0, 0, 0),(0, 0, 255), (0, 255, 255),  (0, 255, 0),]

ring_positions = [(200, 200), (400, 200), (600, 200), (300, 350), (500, 350)]
ring_radius = 100

for color, position in zip(colors, ring_positions):
    cv2.circle(canvas, position, ring_radius, color, 10)

cv2_imshow( canvas)
cv2.waitKey(0)
cv2.destroyAllWindows()
```