**#Program 8: Image Segmentation using K-means**

import numpy as np

import matplotlib.pyplot as plt

import cv2

# Read in the image

image = cv2.imread('E:/flower.jpg')

# Change color to RGB (from BGR)

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

plt.imshow(image)

# Reshaping the image into a 2D array of pixels and 3 color values (RGB)

pixel\_vals = image.reshape((-1,3))

# Convert to float type

pixel\_vals = np.float32(pixel\_vals)

#the below line of code defines the criteria for the algorithm to stop running,

#which will happen is 100 iterations are run or the epsilon (which is the required accuracy)

#becomes 85%

criteria = (cv2.TERM\_CRITERIA\_EPS + cv2.TERM\_CRITERIA\_MAX\_ITER, 100, 0.85)

# then perform k-means clustering wit h number of clusters defined as 3

#also random centres are initally chosed for k-means clustering

k = 3

retval, labels, centers = cv2.kmeans(pixel\_vals, k, None, criteria, 10, cv2.KMEANS\_RANDOM\_CENTERS)

# convert data into 8-bit values

centers = np.uint8(centers)

segmented\_data = centers[labels.flatten()]

# reshape data into the original image dimensions

segmented\_image = segmented\_data.reshape((image.shape))

numpy\_horizontal\_concat = np.concatenate((image, segmented\_image), axis=1)

plt.imshow(numpy\_horizontal\_concat)