

Software Engineering Laboratory | Lab Spring 2024

Assignment-4:NumPy and its Applications

Name-Sumit Kumar

Roll No-22CS30056

This document contains all the codes of Assignment-4:NumPy and its Applications

The codes for each of the questions work when there is a Book1.csv,Book2.csv,Book3.csv,a.png file are present else. It prints an error that these files are not present.

Question-1

```
import numpy as np
import os

#the file name is book1.csv
file1="book1.csv"

if os.path.isfile(file1):
    #extracting the 1-D array from the csv file
    arr_read_1 = np.loadtxt(file1,dtype=str)

    #reading the csv file and converting the string to integer
    arr1=arr_read_1[1:,1].astype(np.uint32)

    # print statement for question1
    #printing the max and min of the book-1.csv file
    # using the numpy max and min function
    print(f"The minimum of the {file1} 1-D array is {np.min(arr1)}")
    print(f"The maximum of the {file1} 1-D array is {np.max(arr1)}")

else:
    print(f"The file {file1} is not present or could not be read.")
```

Question-2

```
import numpy as np
import os

#the file name is book1.csv
file1="book1.csv"

if os.path.isfile(file1):
    #extracting the 1-D array from the csv file
    arr_read_1 = np.loadtxt(file1,dtype=str)

    #reading the csv file and converting the string to integer
    arr1=arr_read_1[1:,1].astype(np.uint32)

    # print statement for question2
    print("The sorted array is =",end=" ")
    #using the numpy sort function to sort the array
    print(np.sort(arr1))

else:
    print(f"The file {file1} is not present or could not be read.")
```

Question-3

```
import numpy as np
import os

#the file name is book1.csv
file1="book1.csv"

if os.path.isfile(file1):
    #extracting the 1-D array from the csv file
    arr_read_1 = np.loadtxt(file1,dtype=str)

    #reading the csv file and converting the string to integer
    arr1=arr_read_1[1:,1].astype(np.uint32)

    #print statement for question3
    print("The reverse sorted array is =",end=" ")
    #using the numpy sort function to sort the array in reverse order
    print(np.sort(arr1)[::-1])

else:
    print(f"The file {file1} is not present or could not be read.")
```

Question-4

```
import numpy as np
import os.path

# Initialize the array for the print statement of question 4
arr = []

# List of file names
file_names = ["book1.csv", "book2.csv", "book3.csv"]

# Iterate through each file
for file_name in file_names:
    if os.path.isfile(file_name):
        # Extracting the 1-D array from the csv file
        arr_read = np.loadtxt(file_name, dtype=str, skiprows=1)
        # Reading the csv file and converting the string to integer
        arr_data = arr_read[:, 1].astype(np.float32)
        # Calculate the mean and append to the array
        arr.append(np.mean(arr_data))
    else:
        print(f"File '{file_name}' is not present.")

# Print the mean of the 3 csv files in a list format
# if all files are present
if len(arr) == len(file_names):
    print("The mean of the 3 csv files is =", arr)
```

Question-5

```
import numpy as np
import cv2 as cv

# Question 5 solving
image_file="a.png"

#reading the image file
X1=cv.imread(image_file)
if X1 is None:
    print("The file is not present or could not be read.")
else:
    #converting the image to the color image
    cv.imshow('color image',X1)
    X=X1.astype(np.float64)

    # Waits for a keystroke
    cv.waitKey(0)
    # Destroys all the windows created
    cv.destroyAllWindows()
```

Question-6

```
import numpy as np
import cv2 as cv

image_file="a.png"

#reading the image file
X1=cv.imread(image_file)

if X1 is None:
    print("The file is not present or could not be read.")
else:
    X=X1.astype(np.float64)

    #Question 6 solving
    X=cv.cvtColor(X1,cv.COLOR_BGR2GRAY)
    #converting the image to the greyscale image
    cv.imshow('greyscale image',X)

    # Waits for a keystroke
    cv.waitKey(0)
    # Destroys all the windows created
    cv.destroyAllWindows()
```

Question-7

```
import numpy as np
import cv2 as cv
import copy

image_file="a.png"

#reading the image file
X1=cv.imread(image_file)

if X1 is None:
    print("The file is not present or could not be read.")

else:
    X=X1.astype(np.float64)

    #converting the image to the greyscale image
    X=cv.cvtColor(X1,cv.COLOR_BGR2GRAY)
    X=X.astype(int)

    #Question 7 solving
    # make Y transpose of x
    X_copy=copy.deepcopy(X)
    Y=np.transpose(X_copy)

    #multiply X and Y using the numpy library for the question in part 7
    Z=np.matmul(X,Y)
    print("The matrix multiplication using the numpy library is:")
    print(Z)
```

Question-8

```
import numpy as np
import cv2 as cv
import copy
import time

image_file="a.png"

#reading the image file
X1=cv.imread(image_file)

if X1 is None:
    print("The file is not present or could not be read.")
else:
    X=X1.astype(np.float64)

    #converting the image to the greyscale image
    X=cv.cvtColor(X1,cv.COLOR_BGR2GRAY)
    X=X.astype(int)
    #Question 7 solving
    # make Y transpose of x
    X_copy=copy.deepcopy(X)
    Y=np.transpose(X_copy)

    #multiply X and Y using the numpy library for the question in part 7
    time_1_strt=time.time()
    print("Matrix multiplication by the Numpy Library is going on....")
    Z=np.matmul(X,Y)
    # print(Z)
    print("Matrix multiplication by the Numpy Library is done")
    time_1_end=time.time()

    #Question 8 for checking the time taken by
    # the numpy library and the python program
    time_2_strt = time.time()
    # x_less=np.divide(X,100)
    # y_less=np.divide(Y,100)
    x_less=X
```



```

y_less=Y
Z = []
# Check if the matrices can be multiplied
if len(X[0]) != len(Y):
    raise ValueError("The matrix multiplication is not possible")
else:
    Z = np.zeros((len(x_less), len(x_less)), dtype=np.int64)

# gh=1
# Perform matrix multiplication
print("Matrix multiplication by the Python Program is going on....")
for i in range(len(x_less)):
    # print(gh)
    # gh+=1
    # continue
    for j in range(len(y_less[0])):
        for k in range(len(y_less)):
            Z[i][j] += x_less[i][k] * y_less[k][j]

# X=np.multiply(x_less,100)
# Y=np.multiply(y_less,100)
# print(Z)

print("Matrix multiplication by the Python Program is done")
time_2_end = time.time()

#Printing the time taken by the numpy
# library and the python program
print(f"The time taken by the Numpy Library
is{time_1_end-time_1_strt}")
print(f"The time taken by the Python Program
is{time_2_end-time_2_strt}")

```

Question-9

```
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt

image_file="a.png"

#reading the image file
X1=cv.imread(image_file)

if X1 is None:
    print("The file is not present or could not be read.")

else:
    X=X1.astype(np.float64)

    #converting the image to the greyscale image
    X=cv.cvtColor(X1,cv.COLOR_BGR2GRAY)

    # Question 9 solving
    #plot the histogram by using the matplotlib library
    plt.hist(X.ravel(),256,[0,256])
    plt.show()
```

Question-10

```
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt

image_file="a.png"

#reading the image file
X1=cv.imread(image_file)

if X1 is None:
    print("The file is not present or could not be read.")

else:
    X=X1.astype(np.float64)

    #converting the image to the greyscale image
    X=cv.cvtColor(X1,cv.COLOR_BGR2GRAY)

    #Question 10 solving
    X[40:71,100:201]=0
    cv.imshow('greyscale image rectangular box',X)
    # Waits for a keystroke
    cv.waitKey(0)
    # Destroys all the windows created
    cv.destroyAllWindows()
```

Question-11

```
import numpy as np
import cv2 as cv
import copy

image_file="a.png"

#reading the image file
X1=cv.imread(image_file)

if X1 is None:
    print("The file is not present or could not be read.")

else:
    X=X1.astype(np.float64)

    #converting the image to the greyscale image
    X=cv.cvtColor(X1,cv.COLOR_BGR2GRAY)

    #Question 11 solving
    Z50=copy.deepcopy(X)
    Z70=copy.deepcopy(X)
    Z100=copy.deepcopy(X)
    Z150=copy.deepcopy(X)

    #Thresholding the image of Z50,Z70,Z100,Z150
    print("Thresholding the image of Z50,Z70,Z100,Z150....")
    Z50[Z50>50]=1
    Z50[Z50<=50]=0

    Z70[Z70>70]=1
    Z70[Z70<=70]=0

    Z100[Z100>100]=1
    Z100[Z100<=100]=0

    Z150[Z150>150]=1
    Z150[Z150<=150]=0

    print("Thresholding the image of Z50,Z70,Z100,Z150 is done")
```

Question-12

```
import numpy as np
import cv2 as cv

image_file="a.png"

#reading the image file
X1=cv.imread(image_file)

if X1 is None:
    print("The file is not present or could not be read.")
else:
    X=X1.astype(np.float64)

    #converting the image to the greyscale image
    X=cv.cvtColor(X1,cv.COLOR_BGR2GRAY)

    #Question 12 solving
    #filter creation
    filter_1=np.array([[ -1, -1, -1],[0,0,0],[1,1,1]])
    #showing the filtered image
    filtered_image=cv.filter2D(X1,-1,filter_1)
    cv.imshow('filtered image',filtered_image)
    # Waits for a keystroke
    cv.waitKey(0)
    # Destroys all the windows created
    cv.destroyAllWindows()
```