**List Tuples Dic Answers**

**Name – Aaron D’Costa**

**Roll No. – 22BRS1129**

1. Find the transpose of a given matrix using list comprehension.

# the matrix to transpose

matrix = [[1, 2, 3],

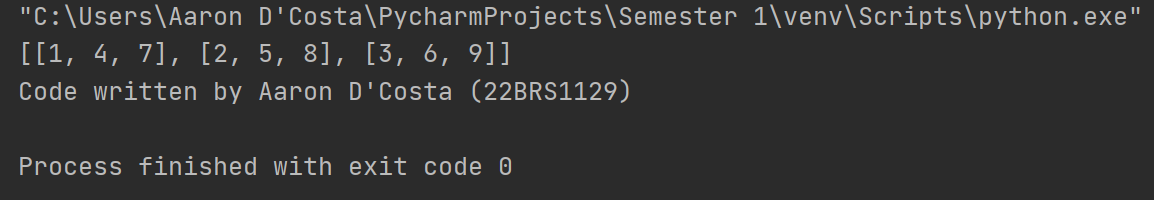
[4, 5, 6],

[7, 8, 9]]

# find the transpose of the matrix

transpose = [[row[i] for row in matrix] for i in range(len(matrix[0]))]

print(transpose)



1. Write a Python program to find the repeated items of a tuple.

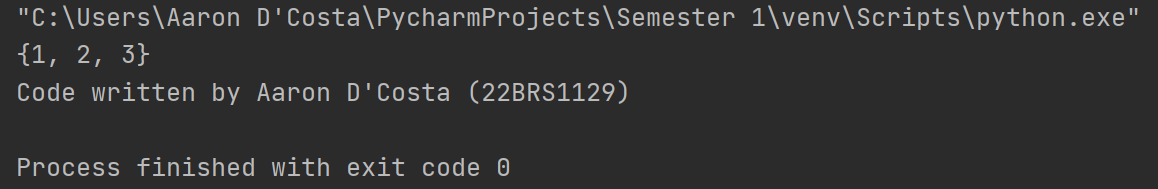
# the tuple to find the repeated items of

tup = (1, 2, 3, 4, 5, 1, 2, 3, 6)

# find the repeated items

repeated\_items = {item for item in tup if tup.count(item) > 1}

print(repeated\_items)



1. Given a Python list. Turn every item of a list into its square input:

aList = [1, 2, 3, 4, 5, 6, 7]

output: [1, 4, 9, 16, 25, 36, 49]

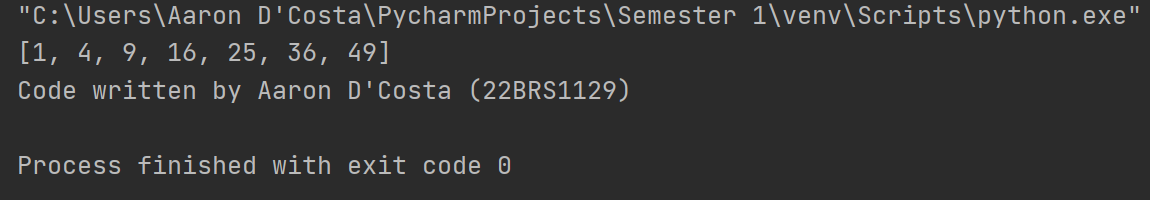
# the list to square the elements of

a\_list = [1, 2, 3, 4, 5, 6, 7]

# create a new list with the squared elements

squared\_list = [item\*\*2 for item in a\_list]

print(squared\_list)



1. Write a program that reads a string and prints the letters in decreasing order of frequency.

DOUBT

1. Write a program to perform row wise sum and column wise sum of a matrix and store the results in two separate matrices namely row\_sum and column\_sum.

# the matrix to perform row-wise and column-wise sum on

matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

# create an empty matrix to store the row-wise sums

row\_sum = []

# compute the row-wise sums and store them in row\_sum

for row in matrix:

row\_sum.append(sum(row))

# create an empty matrix to store the column-wise sums

column\_sum = []

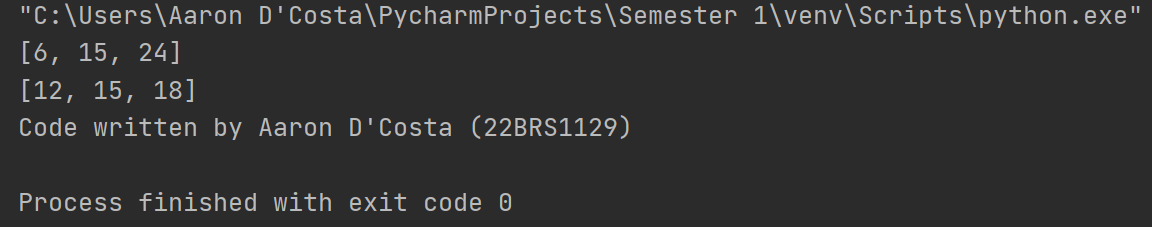
# compute the column-wise sums and store them in column\_sum

for i in range(len(matrix[0])):

column\_sum.append(sum([row[i] for row in matrix]))

print(row\_sum)

print(column\_sum)



1. Write a program to arrange all the elements in the matrix in descending order.

# the matrix to sort

matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

# create a flat list of all the elements in the matrix

elements = [element for row in matrix for element in row]

# sort the elements in descending order

elements.sort(reverse=True)

# create a new matrix with the same dimensions as the original matrix

sorted\_matrix = [[0 for \_ in range(len(matrix[0]))] for \_ in range(len(matrix))]

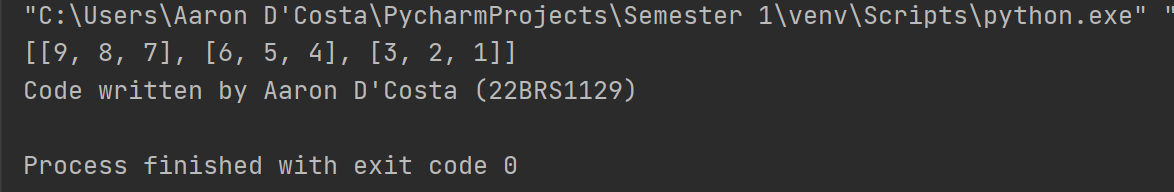
# populate the sorted\_matrix with the sorted elements

for i in range(len(matrix)):

for j in range(len(matrix[0])):

sorted\_matrix[i][j] = elements.pop(0)

print(sorted\_matrix)



1. Write a program to check whether two matrices are identical.

# the first matrix to compare

matrix1 = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

# the second matrix to compare

matrix2 = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

# flag to store whether the matrices are identical

identical = True

# compare each element of the matrices

for i in range(len(matrix1)):

for j in range(len(matrix1[0])):

if matrix1[i][j] != matrix2[i][j]:

identical = False

break

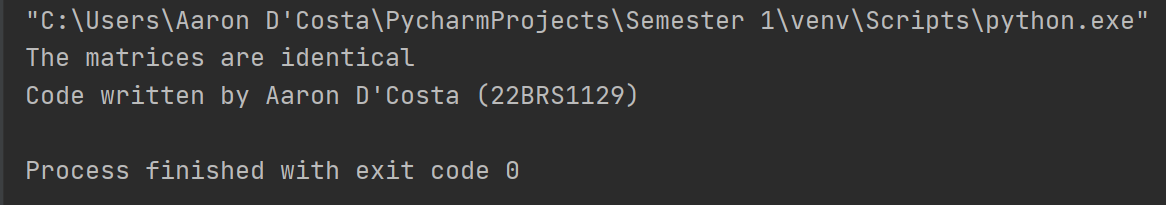
# print the result

if identical:

print("The matrices are identical")

else:

print("The matrices are not identical")



1. Write a program to get a sentence as input from the user. Using dictionary draw the histogram of characters and histogram of words in the given sentence.

# get the sentence from the user

sentence = input("Enter a sentence: ")

# create a dictionary to store the histogram of characters

char\_histogram = {}

# create a dictionary to store the histogram of words

word\_histogram = {}

# split the sentence into words

words = sentence.split()

# populate the histogram of characters

for char in sentence:

if char in char\_histogram:

char\_histogram[char] += 1

else:

char\_histogram[char] = 1

# populate the histogram of words

for word in words:

if word in word\_histogram:

word\_histogram[word] += 1

else:

word\_histogram[word] = 1

# print the histogram of characters

print("Histogram of characters:")

for char, count in char\_histogram.items():

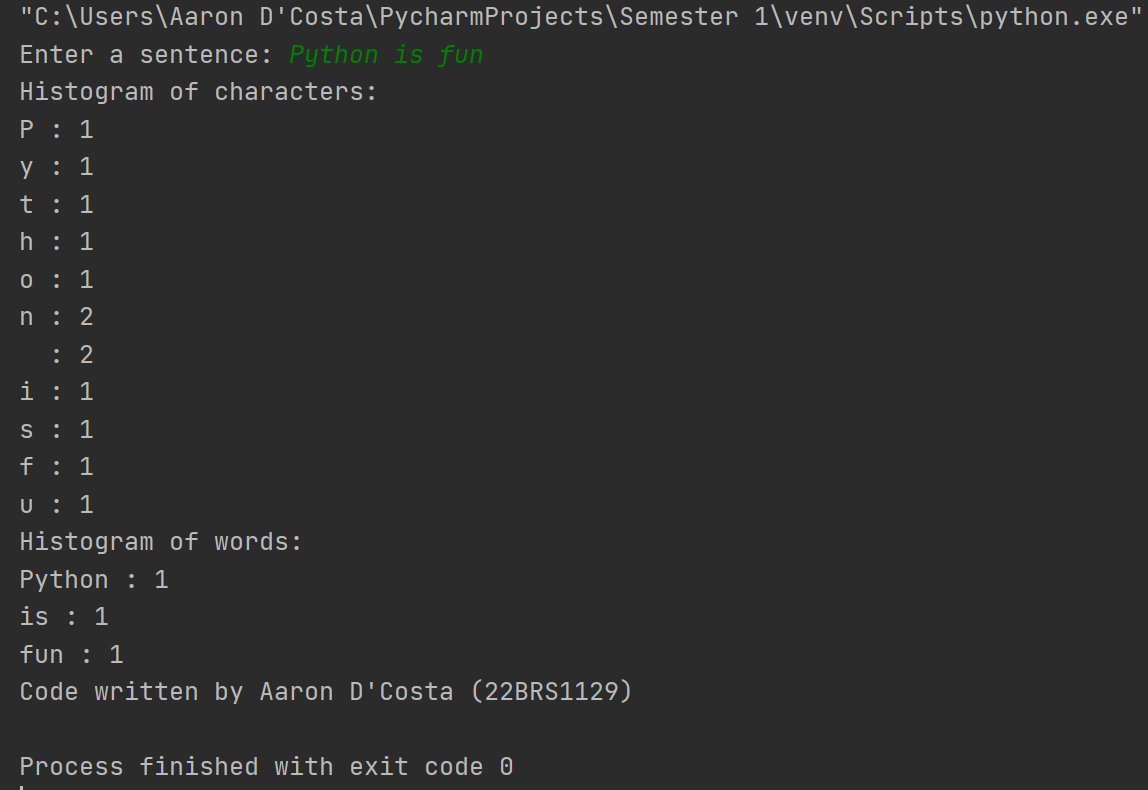
print(char, ":", count)

# print the histogram of words

print("Histogram of words:")

for word, count in word\_histogram.items():

print(word, ":", count)



1. Write a Python script to generate and print a dictionary that contains a number (between 1 and n) in the form (x, x\*x).

Sample Dictionary ( n = 5) :

Expected Output: {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}

# get the value of n from the user

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

# create an empty dictionary

dictionary = {}

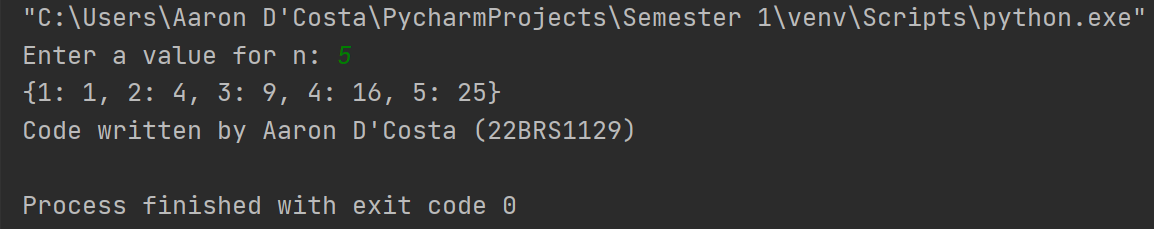
# populate the dictionary with values of x and x\*x

for i in range(1, n+1):

dictionary[i] = i\*i

# print the dictionary

print(dictionary)



10)Create a dictionary with the names as keys and marks as values by user input. Write a Python program to sum all the marks in a dictionary and display it.

# create an empty dictionary

student\_marks = {}

# get the number of students from the user

num\_students = int(input("Enter the number of students: "))

# get the names and marks of the students

for i in range(num\_students):

name = input("Enter the name of the student: ")

marks = int(input("Enter the marks of the student: "))

student\_marks[name] = marks

# sum the marks in the dictionary

total\_marks = sum(student\_marks.values())

# print the total marks

print("Total marks:", total\_marks)

