**Practice Questions Answers**

**Name –** Aaron D’Costa

**Roll No. –** 22BRS1129

1. Write a Python program to create the multiplication table (from 1 to 10) of a number.

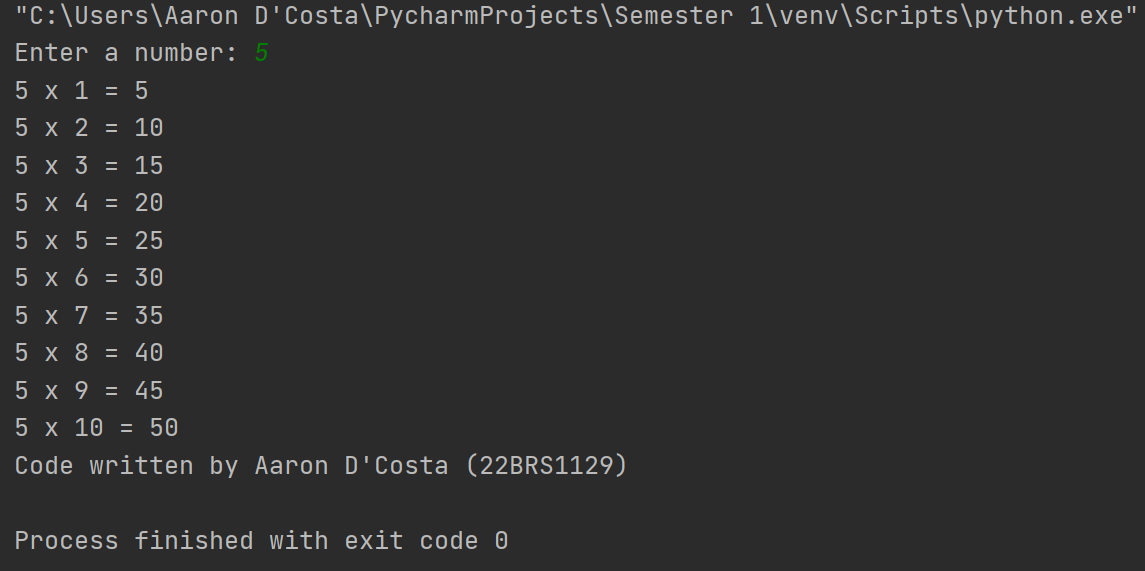
# get the number from the user

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

# create the multiplication table using a for loop

for i in range(1, 11):

print(f"{number} x {i} = {number \* i}")



1. Find the sum of series:
2. 1 + 1/2 + 1/3 + ….. + 1/N.

# get the value of N from the user

N = int(input("Enter the value of N: "))

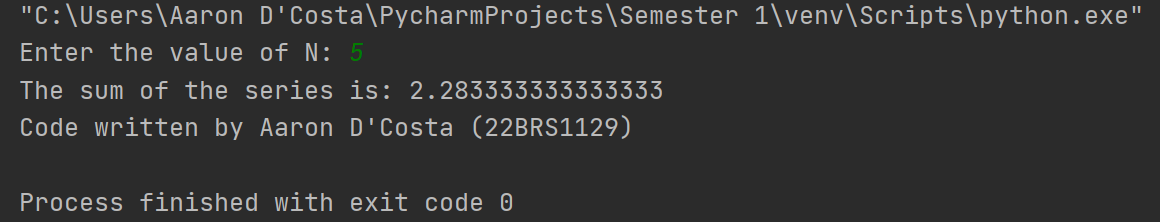
# calculate the sum of the series using a for loop

sum = 0

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

sum += 1/i

print(f"The sum of the series is: {sum}")



1. 1 + 2/4 + 3/9 + ....+ N/(N\*N)

# get the value of N from the user

N = int(input("Enter the value of N: "))

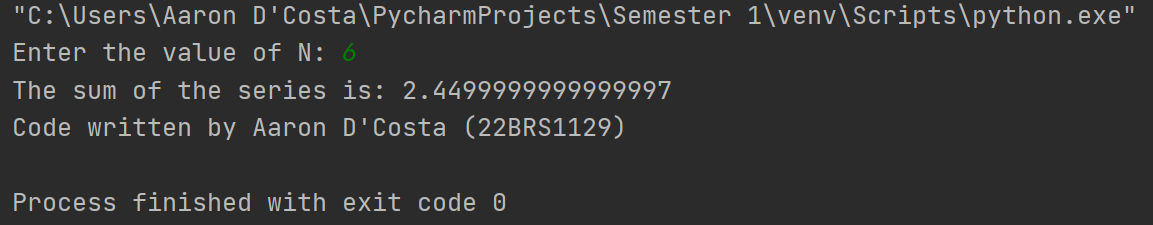
# calculate the sum of the series using a for loop

sum = 0

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

sum += i/(i\*i)

print(f"The sum of the series is: {sum}")



1. 1 + sqrt(2) + sqrt(3) + sqrt(4) + sqrt(N)

# get the value of N from the user

N = int(input("Enter the value of N: "))

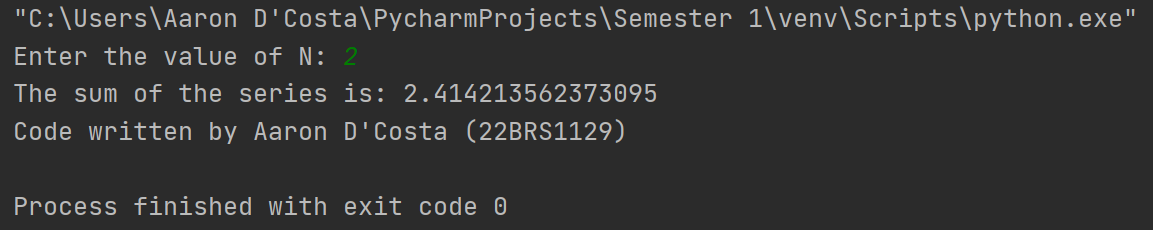
# calculate the sum of the series using a for loop

sum = 0

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

sum += i\*\*0.5

print(f"The sum of the series is: {sum}")



3. Write a Python program which iterates the integers from 1 to 50. For

multiples ofthree print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".

for i in range(1, 51):

if i % 3 == 0 and i % 5 == 0:

print("FizzBuzz")

elif i % 3 == 0:

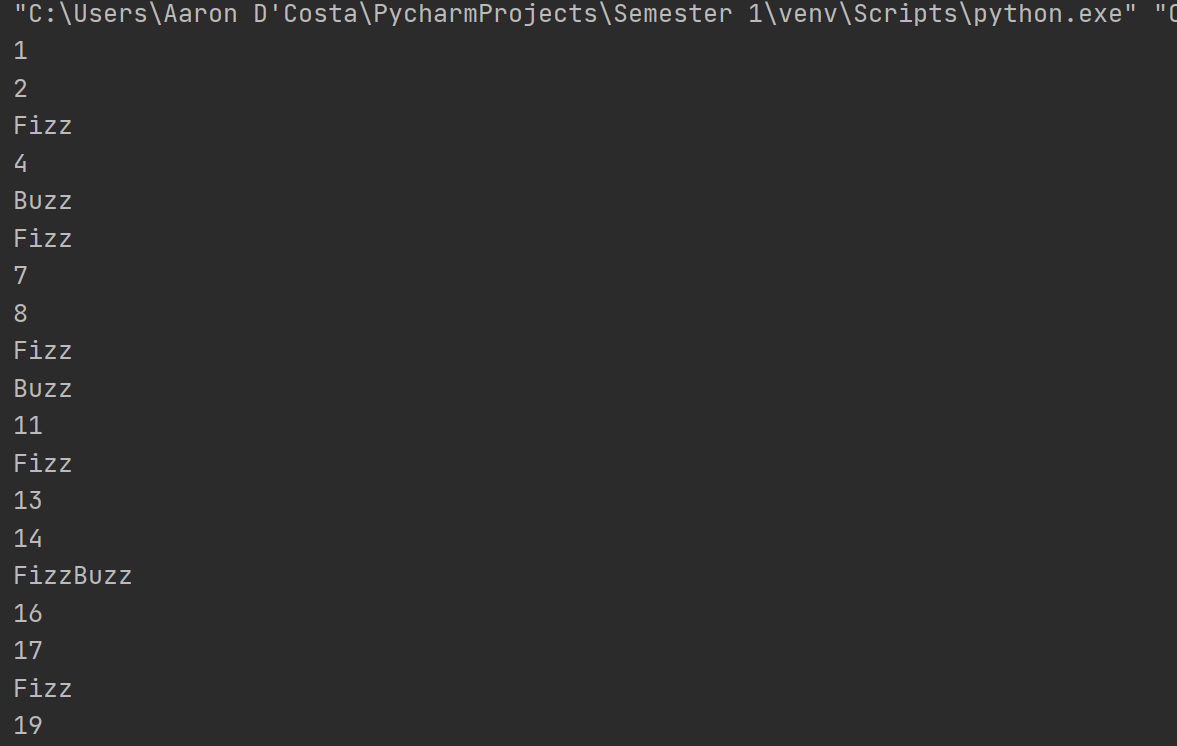
print("Fizz")

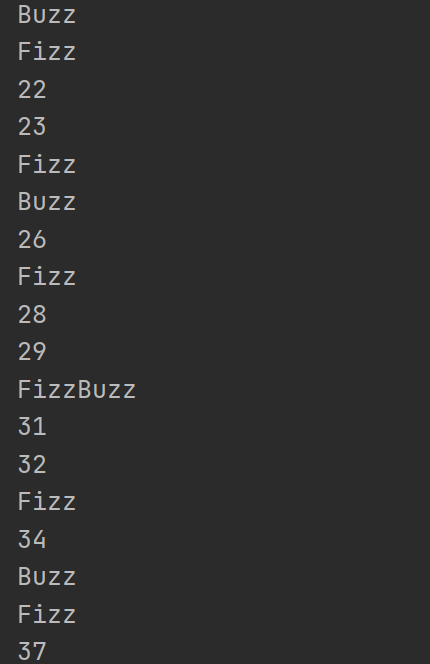
elif i % 5 == 0:

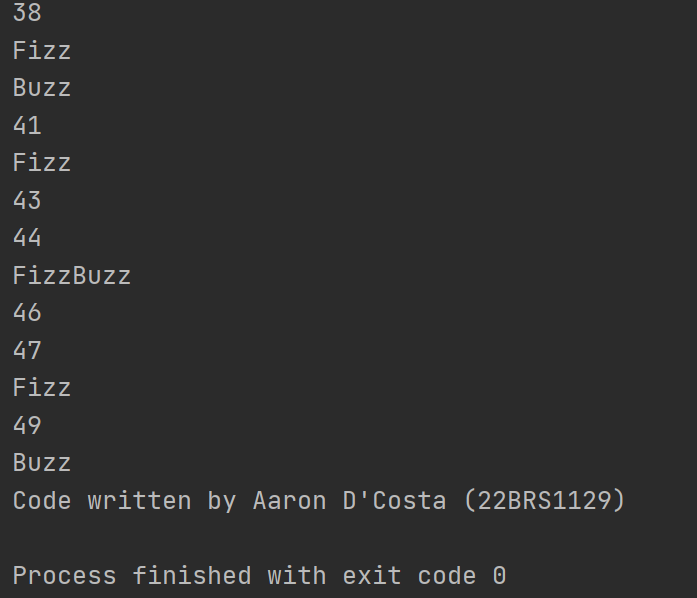
print("Buzz")

else:

print(i)







4. Write a Python program to find numbers between 1 and 400 (both included)

where each digit of a number is an even number. The numbers obtained should be stored in a list and displayed.

# initialize an empty list to store even digits numbers

even\_digits\_numbers = []

# loop through all numbers from 1 to 400

for i in range(1, 401):

# set the flag to True initially, indicating that the number has only even digits

has\_even\_digits = True

# store the current number in a separate variable to modify it without changing the original value

n = i

# keep dividing the number by 10 and checking the remainder until it becomes 0

while n > 0:

# if the remainder is not 0, it means that the number has an odd digit

if n % 2 != 0:

# set the flag to False and break the loop

has\_even\_digits = False

break

# divide the number by 10 to remove the rightmost digit

n //= 10

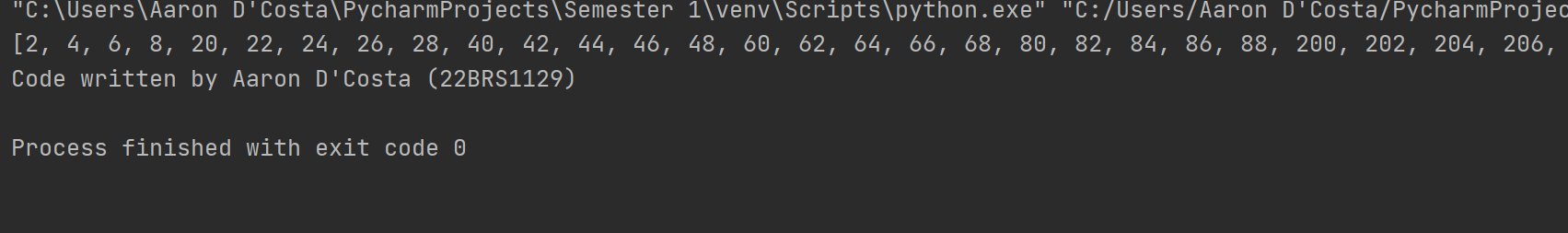
# if the number has only even digits, add it to the list

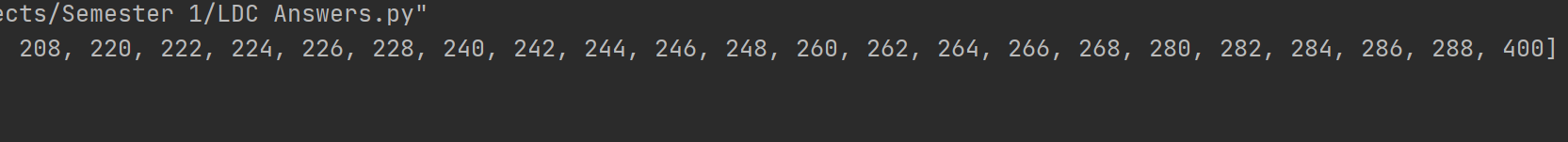
if has\_even\_digits:

even\_digits\_numbers.append(i)

# print the list of even digits numbers

print(even\_digits\_numbers)





5. Given a two Python list. Iterate both lists simultaneously such that list1

should display item in original order and list2 in reverse order

list1 = [10, 20, 30, 40]

list2 = [100, 200, 300, 400]

Expected output

10 400

20 300

30 200

40 100

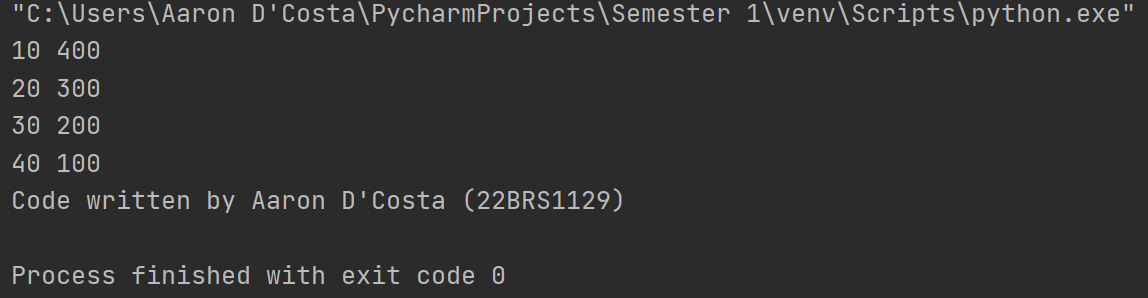
list1 = [10, 20, 30, 40]

list2 = [100, 200, 300, 400]

# Use the zip function to iterate through both lists simultaneously

for x, y in zip(list1, reversed(list2)):

print(x, y)



6. Get first, second best scores from the list.

List may contain duplicates.

Ex: [86,86,85,85,85,83,23,45,84,1,2,0] => should get 86, 85

scores = [86,86,85,85,85,83,23,45,84,1,2,0]

# Sort the list in descending order

sorted\_scores = sorted(scores, reverse=True)

# Initialize first\_best and second\_best

first\_best = sorted\_scores[0]

second\_best = sorted\_scores[1]

# Iterate over the sorted list, starting from the third element

for score in sorted\_scores[2:]:

if score == first\_best:

# If the current element is equal to first\_best, increment the count variable

count += 1

elif score != first\_best and score != second\_best:

# If the current element is different from first\_best and second\_best, set second\_best to the current element and break out of the loop

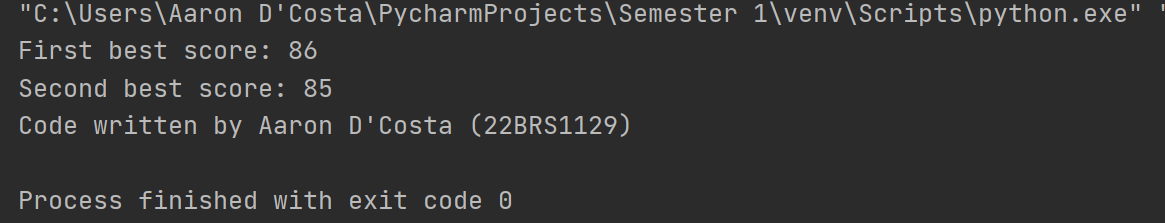
second\_best = score

break

# Print the values of first\_best and second\_best

print(f"First best score: {first\_best}")

print(f"Second best score: {second\_best}")



7. Have a list of number of days in a month and another list of months. Traverse

through both the lists appropriately.

Write program to display number of days in a month when the user enters the

month.

L1=[Jan,Feb,March...]

L2=[31,28,31..]

Input: Dec

Output:31

# Lists of months and number of days in each month

months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']

days = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]

# Get user input for month

user\_input = input("Enter a month: ")

# Iterate through the months list

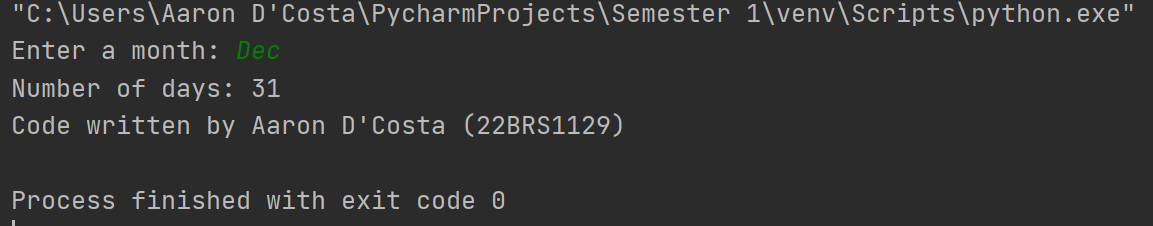
for i in range(len(months)):

# Check if the user input matches the current month

if months[i] == user\_input:

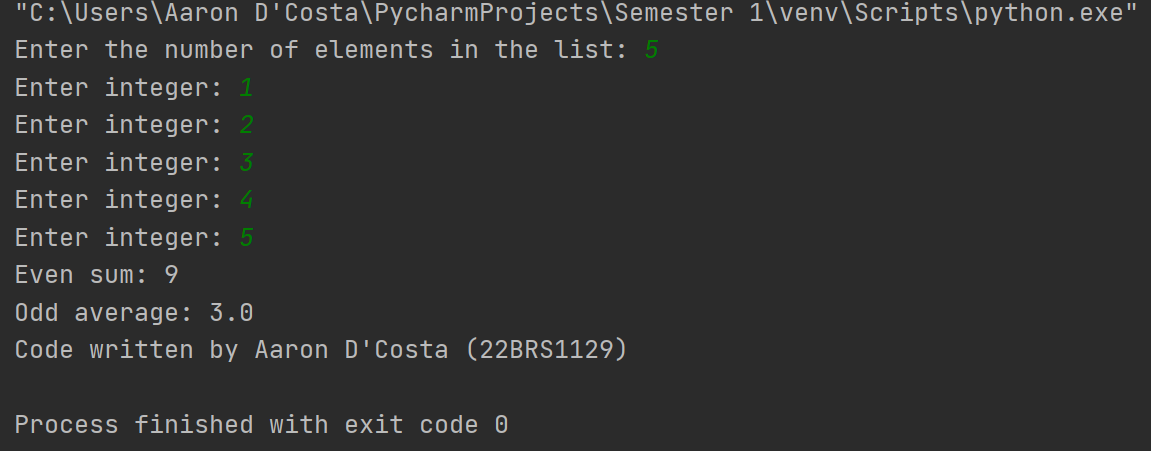
# Print the number of days for the month

print("Number of days:", days[i])



8. Get a list of intergers from the user. Find the sum of all the elements in the even position of the list and store it in a variable called "EvenSum". Find the average of all the elements in the odd position of the list and store it in another variable called "OddAverage". Display both the values.

# Get list of integers from the user  
lst = []  
n = int(input("Enter the number of elements in the list: "))  
for i in range(n):  
 m = int(input("Enter integer: "))  
 lst.append(m)  
  
# Initialize variables for sum and average  
even\_sum = 0  
odd\_count = 0  
odd\_sum = 0  
  
# Iterate through the elements in the list  
for i in range(len(lst)):  
 # If the element is in an even position, add it to the even sum  
 if i % 2 == 0:  
 even\_sum += lst[i]  
 # If the element is in an odd position, add it to the odd sum and increment the count  
 else:  
 odd\_sum += lst[i]  
 odd\_count += 1  
  
# Calculate the average of the elements in the odd positions  
odd\_average = odd\_sum / odd\_count  
  
# Print the even sum and odd average  
print("Even sum:", even\_sum)  
print("Odd average:", odd\_average)



9. Get a list of float values from the user and convert the elements to integer.

Remove the duplicate values in the resultant list as well.

Note: Do not use separate list.

Store the result in the same list.

Input: [2.3, 25.9, 456.01, 31.1, 25.8, 31.8]

Output: [2,26,456,31,32]

# Get the list of float values from the user

float\_list = [float(x) for x in input().split()]

# Convert the float values to integers and store them in the same list

# using list comprehension

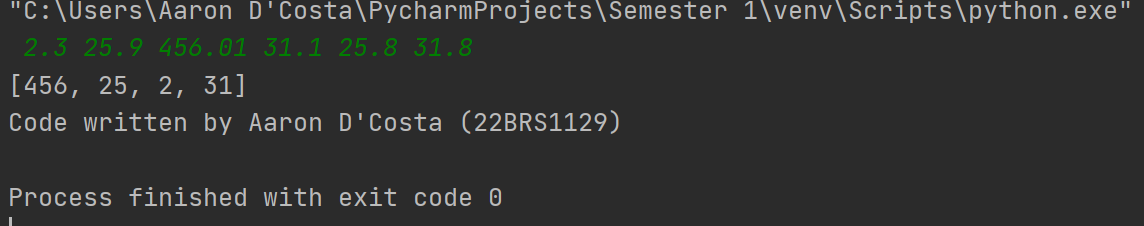
float\_list = [int(x) for x in float\_list]

# Remove duplicates using a set

float\_list = list(set(float\_list))

# Display the resulting list

print(float\_list)



10. Get a list of numbers from the user and sort the list in descending order of

the last digit of each of the numbers in the list.

Do not use separate list. Do the sorting operation in the same list

Input: [19,21,38,47,39,56,12,54,75,93]

Output:[19,38,47,56,75,54,93,12,21]

Note: 39 is deleted as there is a number existing in the resultant list which

ends in the number 9.

DOUBT