SIMATS ENGINEERING

SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES

CHENNAI-602105

CSA0839 – PYTHON PROGRAMMING FOR CYBER SECURITY

Submitted By :S. Jaswanth Venkata Rahul

1)

numbers = []

while True:

num = int(input("Enter a number (-1 to exit): "))

if num == -1:

break

numbers.append(num)

positive\_numbers = [n for n in numbers if n > 0]

negative\_numbers = [n for n in numbers if n < 0]

avg\_positive = sum(positive\_numbers) / len(positive\_numbers) if positive\_numbers else 0

avg\_negative = sum(negative\_numbers) / len(negative\_numbers) if negative\_numbers else 0

print(f"Avg negative number is {avg\_negative}, avg positive number is {avg\_positive}")

A computer screen with numbers and letters

AI-generated content may be incorrect.

2)

# Function to calculate square and cube of a decimal number

def calculate\_square\_and\_cube(number):

square = number \*\* 2

cube = number \*\* 3

return square, cube

# Input

given\_number = 0.6

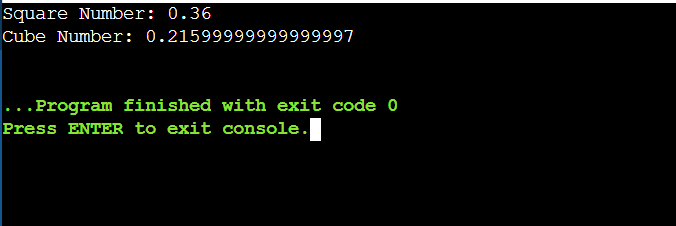
# Calculation

square\_number, cube\_number = calculate\_square\_and\_cube(given\_number)

# Output

print(f"Square Number: {square\_number}")

print(f"Cube Number: {cube\_number}")



3)

# Pattern Printing Program

char = input("Enter the Character to be printed: ")

rows = int(input("Number of rows: "))

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

print((char + ' ') \* i)

A screen shot of a computer

AI-generated content may be incorrect.

4)

A = 7

B = 5

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

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

A screen shot of a computer

AI-generated content may be incorrect.

5)

# Leap Year Checker

def is\_leap\_year(year):

if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):

return "Leap Year"

else:

return "Not a Leap Year"

# Sample Input

year = 2000

print(is\_leap\_year(year))

A black screen with green text

AI-generated content may be incorrect.

6)

# Duplicate Array Finder

def find\_duplicates(array):

return list(set(array))

# Sample Input

array = {1, 2, 3, 4, 1}

# Output

duplicate\_array = find\_duplicates(array)

print("duplicate array =", duplicate\_array)

A computer screen with green text

AI-generated content may be incorrect.

7)

# Check whether the number is positive or negative

number = 23

if number > 0:

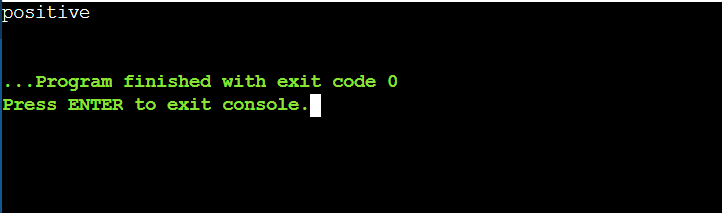
print("positive")

elif number < 0:

print("negative")

else:

print("zero")



8)

# Average of Mean, Median, and Mode

import statistics

def calculate\_average(data):

mean = statistics.mean(data)

median = statistics.median(data)

mode = statistics.mode(data)

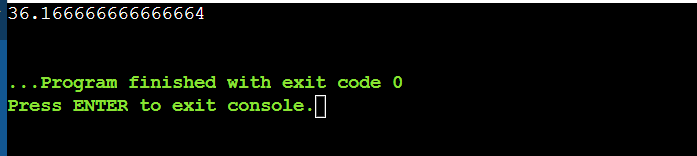
average = (mean + median + mode) / 3

return average

data = [12, 45, 83, 52]

average\_result = calculate\_average(data)

print(average\_result)



9)

# Store the arrays in non-increasing order

def sort\_non\_increasing(arr):

return sorted(arr, reverse=True)

# Sample Input

input\_array = [1, 8, 3, 4, 0]

# Output

output\_array = sort\_non\_increasing(input\_array)

print(output\_array)

A screen shot of a computer

AI-generated content may be incorrect.

10)

# Intersecting elements in two tuples

def intersect\_tuples(tuple1, tuple2):

return tuple(set(tuple1) & set(tuple2))

# Sample Input

tuple1 = (2, 3, 4, 5)

tuple2 = (3, 4, 8, 6)

# Output

result = intersect\_tuples(tuple1, tuple2)

print(result)

