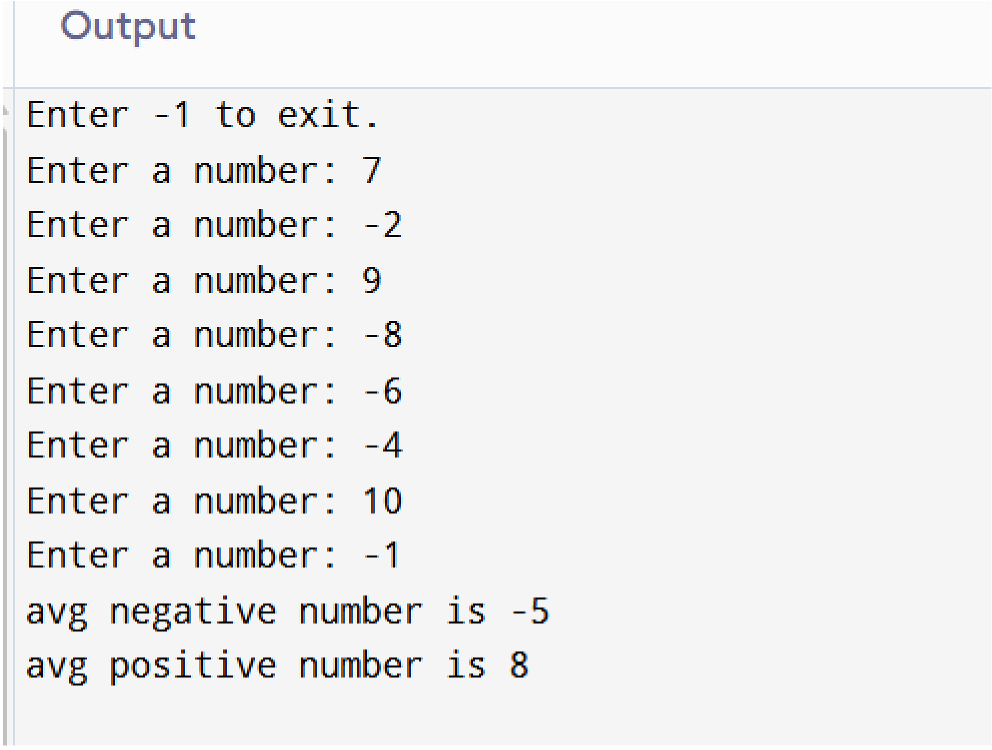
LEVEL WISE QUESTIONS(Sample)

ASSIGNMENT-1

1)Read the number until -1 is encounter. find the avg of positive numbers and negative numbers entered by user

PROGRAM:-

```
# Initialize variables
positive_sum = 0
positive_count = 0
negative_sum = 0
negative_count = 0

print("Enter -1 to exit.")
while True:
    num = int(input("Enter a number: "))

    if num == -1:
        break
    elif num > 0:
        positive_sum += num
        positive_count += 1
    elif num < 0:
        negative_sum += num
        negative_count += 1

# Calculate and display averages
if negative_count > 0:
    avg_negative = negative_sum // negative_count
    print(f"avg negative number is {avg_negative}")
else:
    print("No negative numbers entered.")

if positive_count > 0:
    avg_positive = positive_sum // positive_count
    print(f"avg positive number is {avg_positive}")
else:
    print("No positive numbers entered.")
```



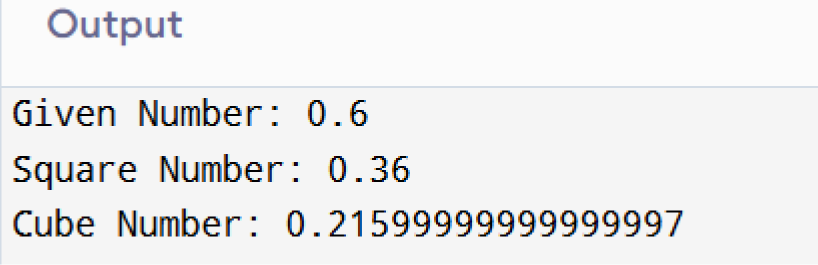
```
Enter -1 to exit.
Enter a number: 7
Enter a number: -2
Enter a number: 9
Enter a number: -8
Enter a number: -6
Enter a number: -4
Enter a number: 10
Enter a number: -1
avg negative number is -5
avg positive number is 8
```

2) Write a python program to find the square, cube of the given decimal number.

PROGRAM:-

```
# Read input from user
num = float(input("Given Number: "))

# Calculate square and cube
square = num ** 2
cube = num ** 3

# Display results
```

```
print(f"Square Number: {square}")

print(f"Cube Number: {cube}")
```

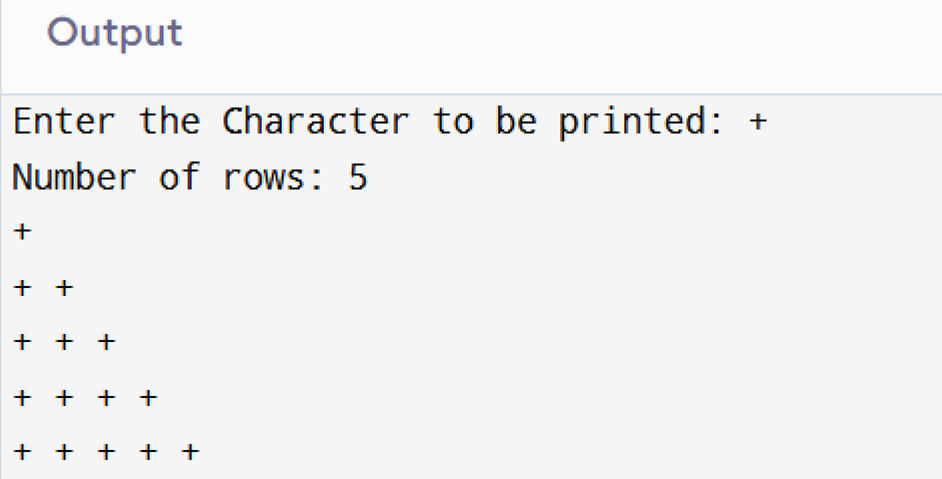


Output

```
Given Number: 0.6
Square Number: 0.36
Cube Number: 0.21599999999999997
```

3) Write a python program to print the following pattern.

```
# Read input from user

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

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


# Print the pattern

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

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



Output

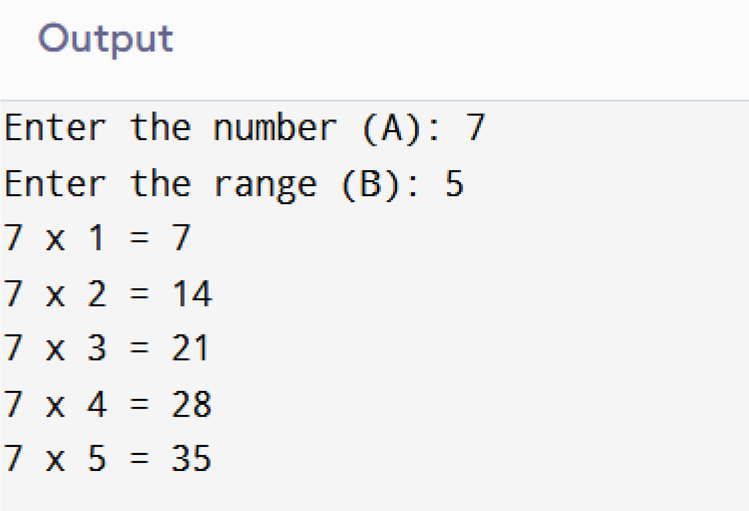
```
Enter the Character to be printed: +
Number of rows: 5
+
+ +
+ + +
+ + + +
+ + + + +
```

4)Python Program to Display the Multiplication Table

PROGRAM:-

```
# Read input values

A = int(input("Enter the number (A): "))

B = int(input("Enter the range (B): "))


# Display the multiplication table
```

```
for i in range(1, B + 1):
    print(f"{A} x {i} = {A * i}")
```



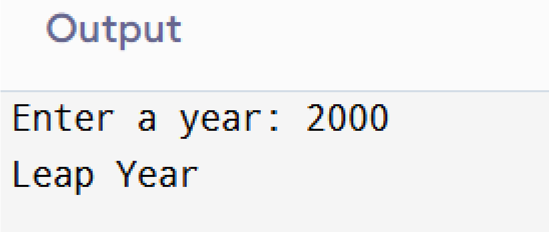
Output

```
Enter the number (A): 7
Enter the range (B): 5
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
```

5)Write a program to find whether it is leap year or not?

PROGRAM:-

```
# Read input year from user
year = int(input("Enter a year: "))

# Check for leap year
if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
    print("Leap Year")
else:
    print("Not a Leap Year")
```



Output

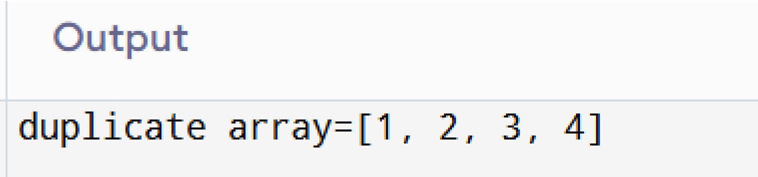
```
Enter a year: 2000
Leap Year
```

6)Write a program to find out the duplicate array

PROGRAM:-

```
# Sample input array
array = [1, 2, 3, 4, 1]
```

```
# Remove duplicates by converting to a set, then back to list
unique_array = list(set(array))

# Optional: sort the result to maintain consistent order
unique_array.sort()

print(f"duplicate array={unique_array}")
```



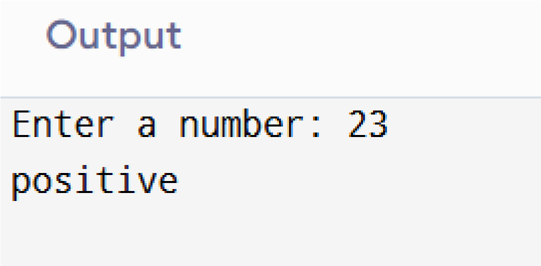
Output

```
duplicate array=[1, 2, 3, 4]
```

7)Check whether the number is positive or negative

PROGRAM:-

```
# Read input from user
num = float(input("Enter a number: "))

# Check the sign
if num > 0:
   print("positive")
elif num < 0:
   print("negative")
else:
   print("zero")
```



Output

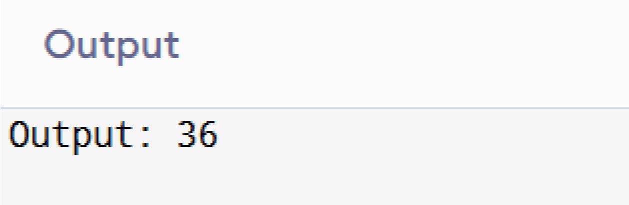
```
Enter a number: 23
positive
```

8)Write a python program to find the average of mean median mode

import statistics

PROGRAM:-

```
# Sample input
numbers = [12, 45, 83, 52]

# Calculate mean, median, and mode
mean = statistics.mean(numbers)
median = statistics.median(numbers)

# Mode requires at least one repeating element; handle exception if not found
try:
    mode = statistics.mode(numbers)
except statistics.StatisticsError:
    mode = mean  # or set to 0, or handle as needed

# Calculate average of mean, median, and mode
average = (mean + median + mode) / 3

# Print result
print("Output:", round(average))
```



Output

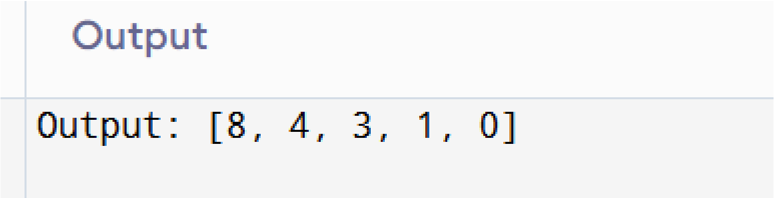
```
Output: 36
```

9)Write a python program to store the arrays in non-increasing order

PROGRAM:-

```
# Sample input
array = [1, 8, 3, 4, 0]
```

```
# Sort the array in non-increasing order
array.sort(reverse=True)

# Print the result
print("Output:", array)
```

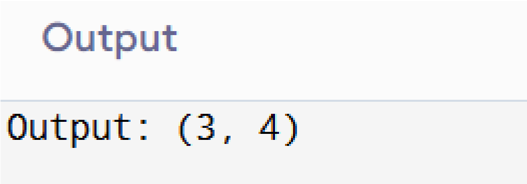


```
Output: [8, 4, 3, 1, 0]
```

10)Write a Python Program to Intersecting an elements

PROGRAM:-

```
# Sample input
tuple1 = (2, 3, 4, 5)
tuple2 = (3, 4, 8, 6)

# Find intersection using set
intersection = tuple(set(tuple1) & set(tuple2))

# Sort the result if needed
intersection = tuple(sorted(intersection))

# Print result
print("Output:", intersection)
```



```
Output: (3, 4)
```