DATA SCEINCE AND MACHINE LEARNING LAB

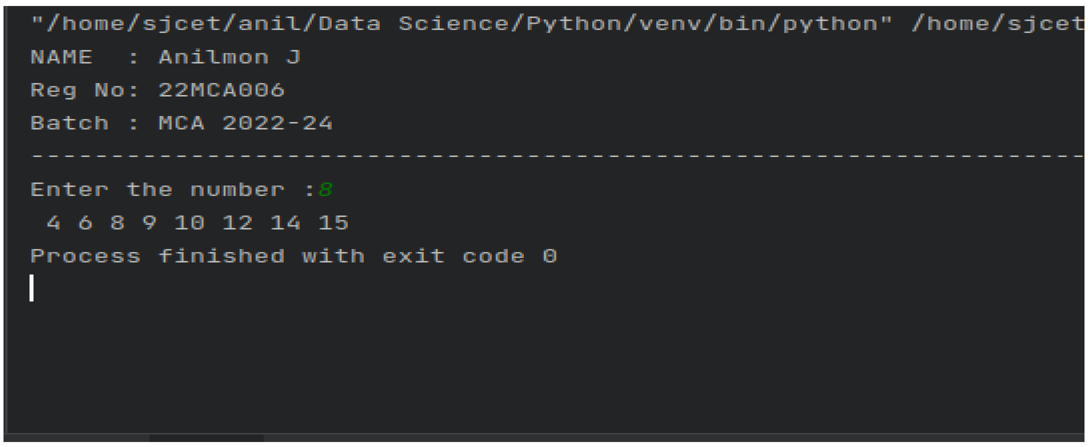
LAB CYCLE 1

1. Program to Print all non-Prime Numbers in an Interval.

Code:

```
def is_prime(num):
    if num <= 1:
        return False
    for i in range(2, num):
        if num % i == 0:
            return False
    return True
print("NAME  : Anilmon J")
print("Reg No: 22MCA006")
print("Batch : MCA 2022-24 ")
print("-------------------------------------------------------------------------------")

N = int(input("Enter the number :"))
current_num = 2
print( end=" ")
while N > 0:
    if not is_prime(current_num):
        print(current_num, end=" ")
        N -= 1
    current_num += 1
```

Output:



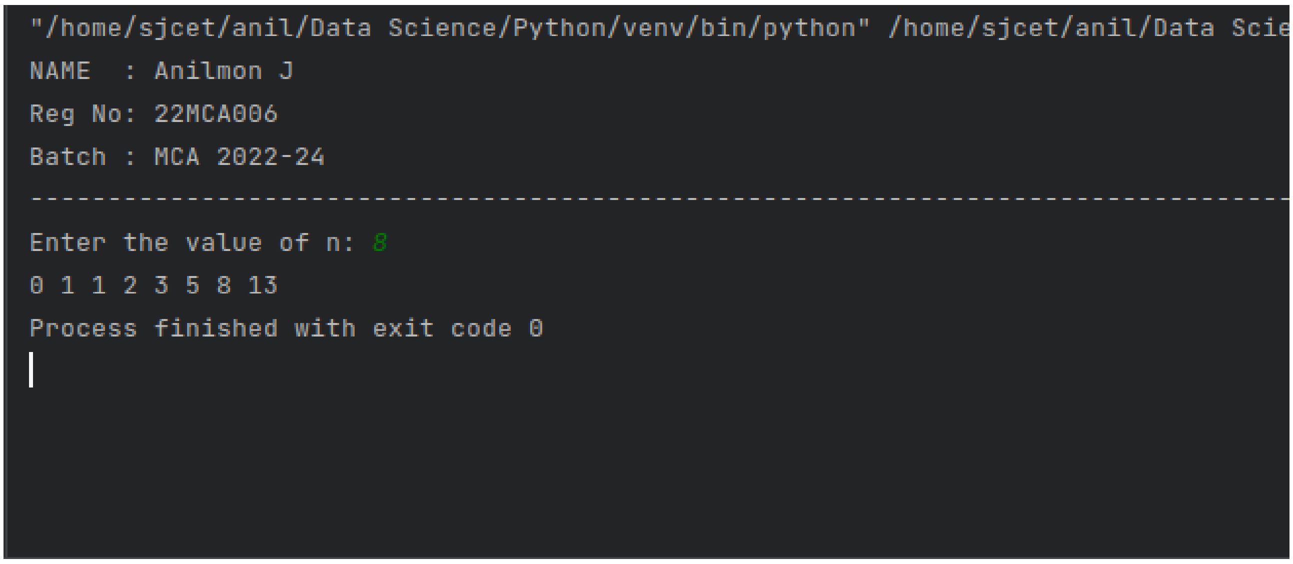
```
"/home/sjcet/anil/Data Science/Python/venv/bin/python" /home/sjcet
NAME   : Anilmon J
Reg No: 22MCA006
Batch : MCA 2022-24
-----------------------------------------------------------------------
Enter the number :8
 4 6 8 9 10 12 14 15
Process finished with exit code 0
```

2. Program to print the first N Fibonacci numbers.

Code:

```
print("NAME  : Anilmon J")
print("Reg No: 22MCA006")
print("Batch : MCA 2022-24 ")
print("-------------------------------------------------------------------------------")

num = int(input("Enter the value of n: "))
a= 0
b = 1
n = 0
count = 1

while(count <= num):
    print(n,end=" ")
    count += 1
    a = b
    b = n
    n = a + b
    t_number = a + b
```

Output:



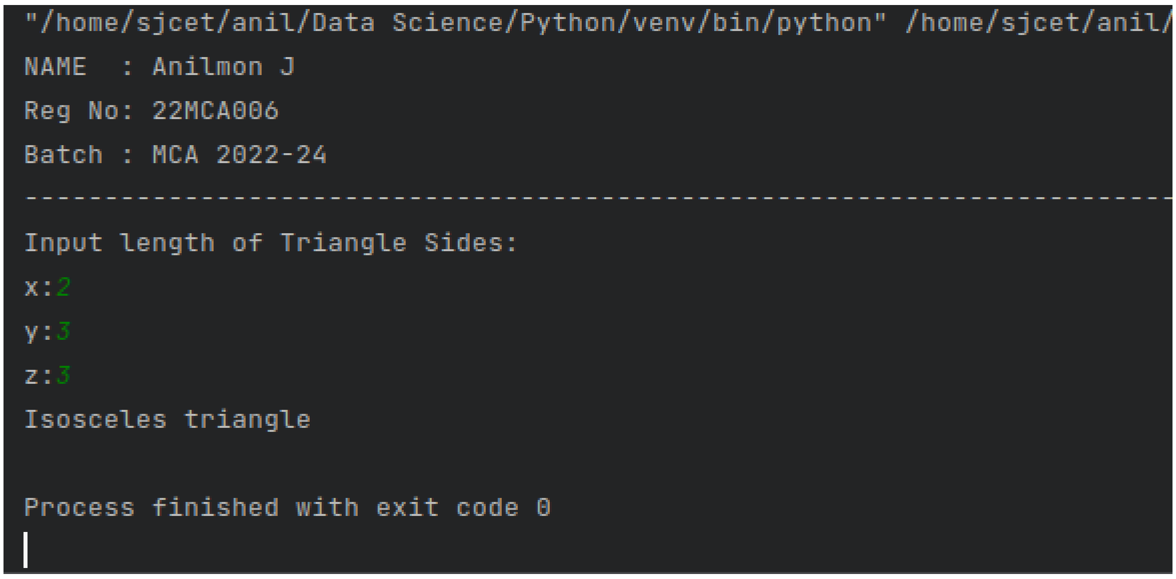
```
"/home/sjcet/anil/Data Science/Python/venv/bin/python" /home/sjcet/anil/Data Scie
NAME  : Anilmon J
Reg No: 22MCA006
Batch : MCA 2022-24
---------------------------------------------------------------------------------
Enter the value of n: 8
0 1 1 2 3 5 8 13
Process finished with exit code 0
```

3. Given sides of a triangle, write a program to check whether given triangle is an isosceles, equilateral or scalene.

Code:

```
print("NAME  : Anilmon J")
print("Reg No: 22MCA006")
print("Batch : MCA 2022-24 ")
print("-----------------------------------------------------------------------------")

print("Input length of Triangle Sides:")
x=int(input("x:"))
y=int(input("y:"))
z=int(input("z:"))
if( x == y == z):
    print("Equilateral triangle")
elif(x == y or y==z or z==x):
    print("Isosceles triangle")
else:
    print("Triangle is Scalene")
```

Output:



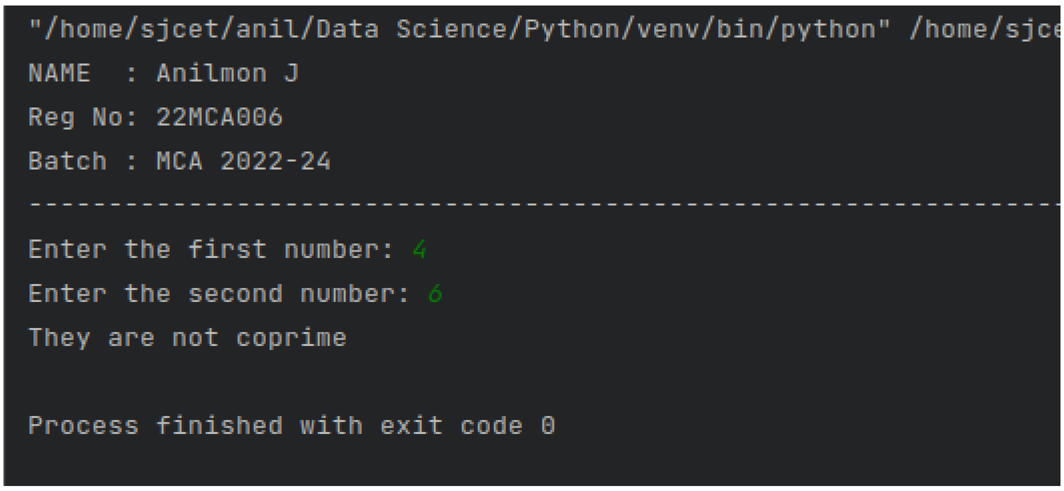
```
"/home/sjcet/anil/Data Science/Python/venv/bin/python" /home/sjcet/anil/
NAME  : Anilmon J
Reg No: 22MCA006
Batch : MCA 2022-24
---------------------------------------------------------------------------
Input length of Triangle Sides:
x:2
y:3
z:3
Isosceles triangle

Process finished with exit code 0
```

4. Program to check whether given pair of number is coprime.

Code:

```
import math
def are_coprime(a, b):
    gcd = math.gcd(a, b)
    return gcd == 1
print("NAME  : Anilmon J")
print("Reg No: 22MCA006")
print("Batch : MCA 2022-24 ")
print("--------------------------------------------------------------------------------")
a = int(input("Enter the first number: "))
b= int(input("Enter the second number: "))

if are_coprime(a,b):
    print(f"{a} and {b} are coprime")
else:
    print("They are not coprime")
```

Output:



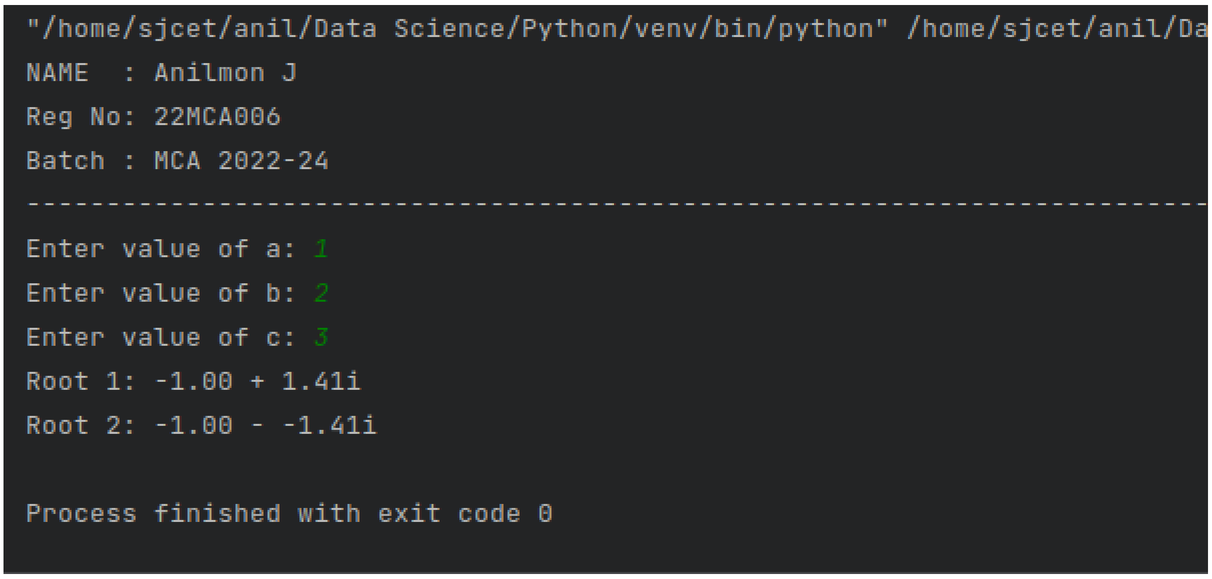
```
"/home/sjcet/anil/Data Science/Python/venv/bin/python" /home/sjce
NAME  : Anilmon J
Reg No: 22MCA006
Batch : MCA 2022-24
-----------------------------------------------------------------
Enter the first number: 4
Enter the second number: 6
They are not coprime

Process finished with exit code 0
```

5. Program to find the roots of a quadratic equation(rounded to 2 decimal places).

Code:

```
print("NAME  : Anilmon J")
print("Reg No: 22MCA006")
print("Batch : MCA 2022-24 ")
print("-----------------------------------------------------------------------------")
import math
a = float(input("Enter value of a: "))
b = float(input("Enter value of b: "))
c = float(input("Enter value of c: "))
discri = b**2 - 4*a*c

if discri > 0:
    root1 = (-b + math.sqrt(discri)) / (2*a)
    root2 = (-b - math.sqrt(discri)) / (2*a)
    print(f"Root 1: {round(root1, 2)}")
    print(f"Root 2: {round(root2, 2)}")
elif discri == 0:
    root = -b / (2*a)
    print(f"Root: {round(root, 2)}")
else:
    real_part = -b / (2*a)
    img_part = math.sqrt(-discri) / (2*a)
    root1 = complex(real_part, img_part)
    root2 = complex(real_part, -img_part)
    print(f"Root 1: {root1.real:.2f} + {root1.imag:.2f}i")
    print(f"Root 2: {root2.real:.2f} – {root2.imag:.2f}i")
```

Output:



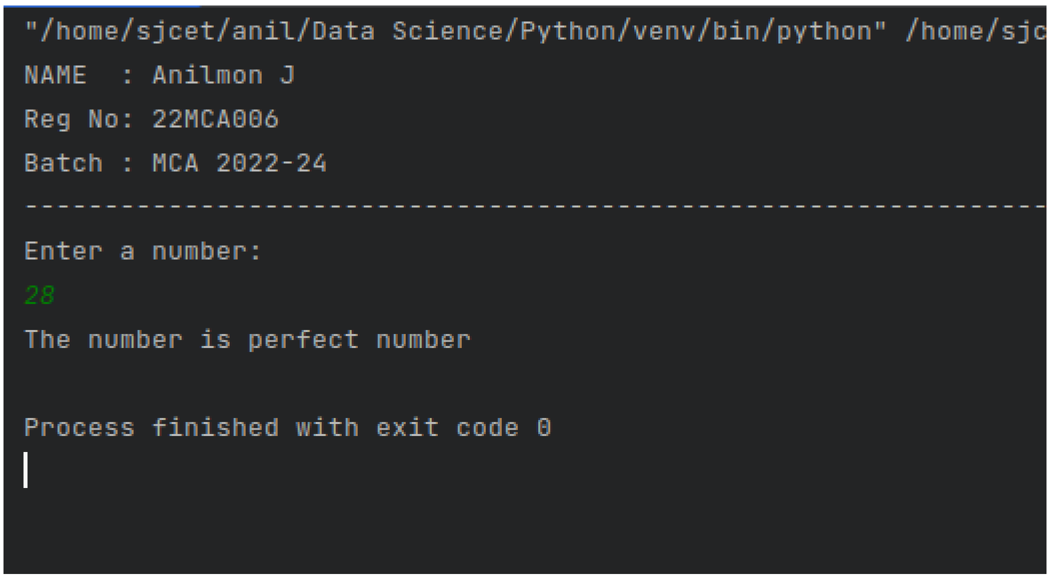
```
"/home/sjcet/anil/Data Science/Python/venv/bin/python" /home/sjcet/anil/Da
NAME  : Anilmon J
Reg No: 22MCA006
Batch : MCA 2022-24
---------------------------------------------------------------------------
Enter value of a: 1
Enter value of b: 2
Enter value of c: 3
Root 1: -1.00 + 1.41i
Root 2: -1.00 - -1.41i

Process finished with exit code 0
```

6. Program to check whether a given number is perfect number or not(sum of factors=number).

Code:

```
print("NAME  : Anilmon J")
print("Reg No: 22MCA006")
print("Batch : MCA 2022-24 ")
print("--------------------------------------------------------------------------------")

n=int(input("Enter a number:\n"))
sum=0
for i in range(1,n):
    if(n % i == 0):
        sum=sum+i
if( sum == n):
    print("The number is perfect number")
else:
    print("The entered number is not perfect")
```

Output:



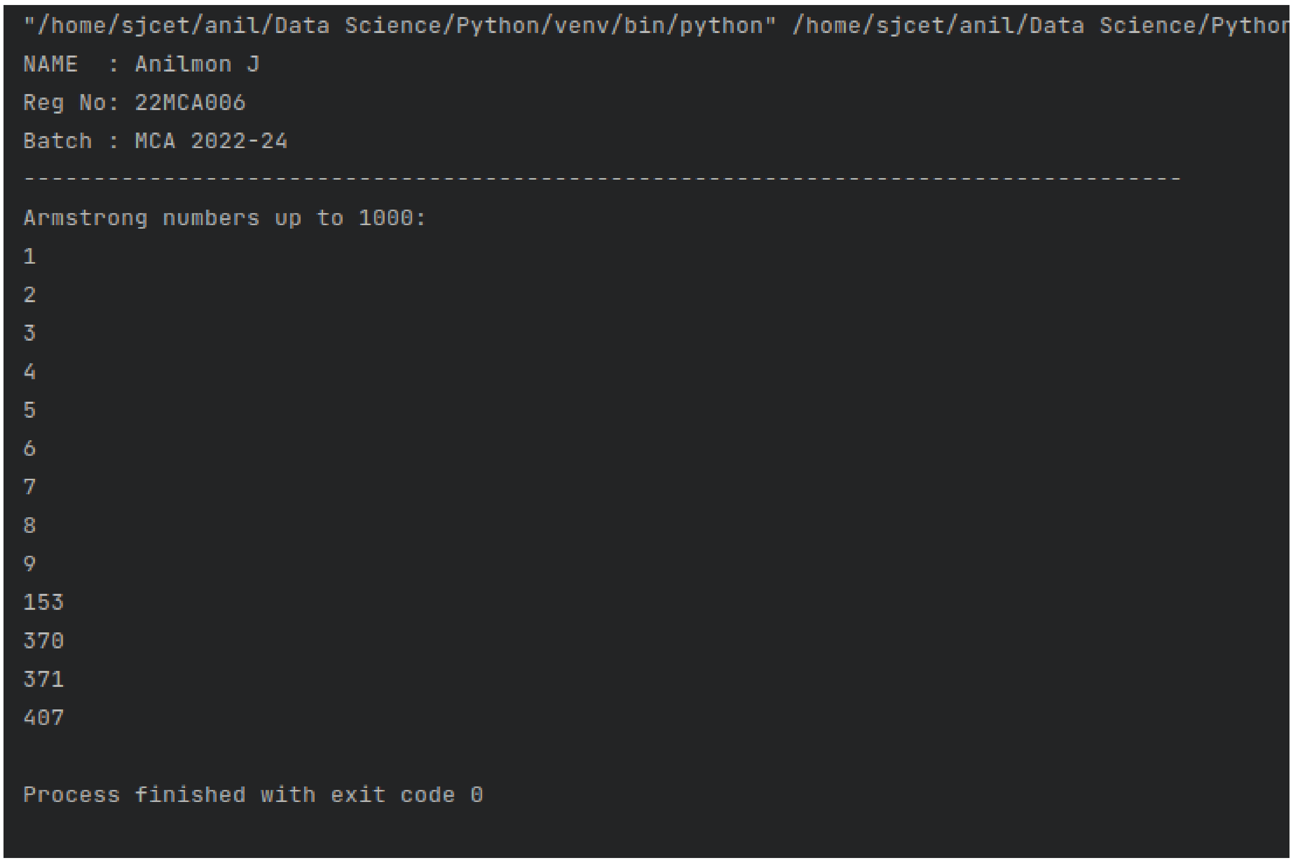
```
"/home/sjcet/anil/Data Science/Python/venv/bin/python" /home/sjc
NAME  : Anilmon J
Reg No: 22MCA006
Batch : MCA 2022-24
-----------------------------------------------------------------
Enter a number:
28
The number is perfect number

Process finished with exit code 0
```

7. Program to display amstrong numbers upto 1000.

Code:

```
def is_armstrong_number(num):
    num_str = str(num)
    num_digits = len(num_str)
    armstrong_sum = sum(int(digit) ** num_digits for digit in num_str)
    return armstrong_sum == num
print("NAME  : Anilmon J")
print("Reg No: 22MCA006")
print("Batch : MCA 2022-24 ")
print("--------------------------------------------------------------------------------")
print("Armstrong numbers up to 1000:")
for num in range(1, 1001):
    if is_armstrong_number(num):
        print(num)
```

Output:



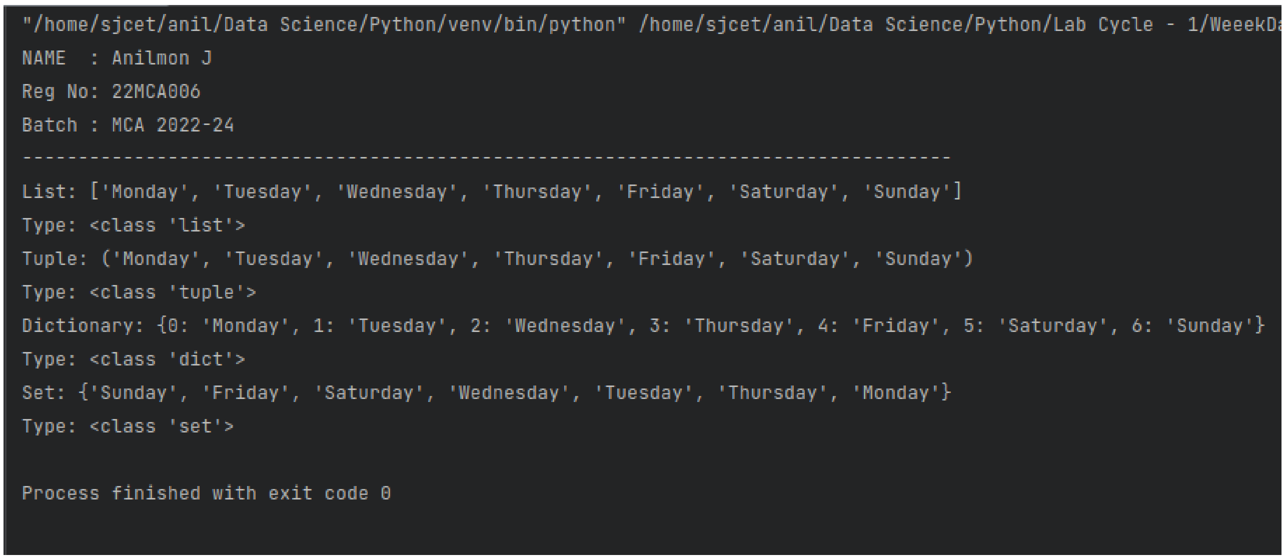
```
"/home/sjcet/anil/Data Science/Python/venv/bin/python" /home/sjcet/anil/Data Science/Pythor
NAME  : Anilmon J
Reg No: 22MCA006
Batch : MCA 2022-24
------------------------------------------------------------------------------------
Armstrong numbers up to 1000:
1
2
3
4
5
6
7
8
9
153
370
371
407

Process finished with exit code 0
```

8. Store and display the days of a week as a List, Tuple, Dictionary, Set. Also demonstrate different ways to store values in each of them. Display its type also.

Code:

```
print("NAME  : Anilmon J")
print("Reg No: 22MCA006")
print("Batch : MCA 2022-24 ")
print("---------------------------------------------------------------------------")
days_list = ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
"Saturday", "Sunday"]
print("List:", days_list)
print("Type:", type(days_list))

days_tuple = ("Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
"Saturday", "Sunday")
print("Tuple:", days_tuple)
print("Type:", type(days_tuple))

days_dict = {0: "Monday", 1: "Tuesday", 2: "Wednesday", 3: "Thursday",
4: "Friday", 5: "Saturday", 6: "Sunday"}
print("Dictionary:", days_dict)
print("Type:", type(days_dict))

days_set = {"Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
"Saturday", "Sunday"}
print("Set:", days_set)
print("Type:", type(days_set))
```

Output:



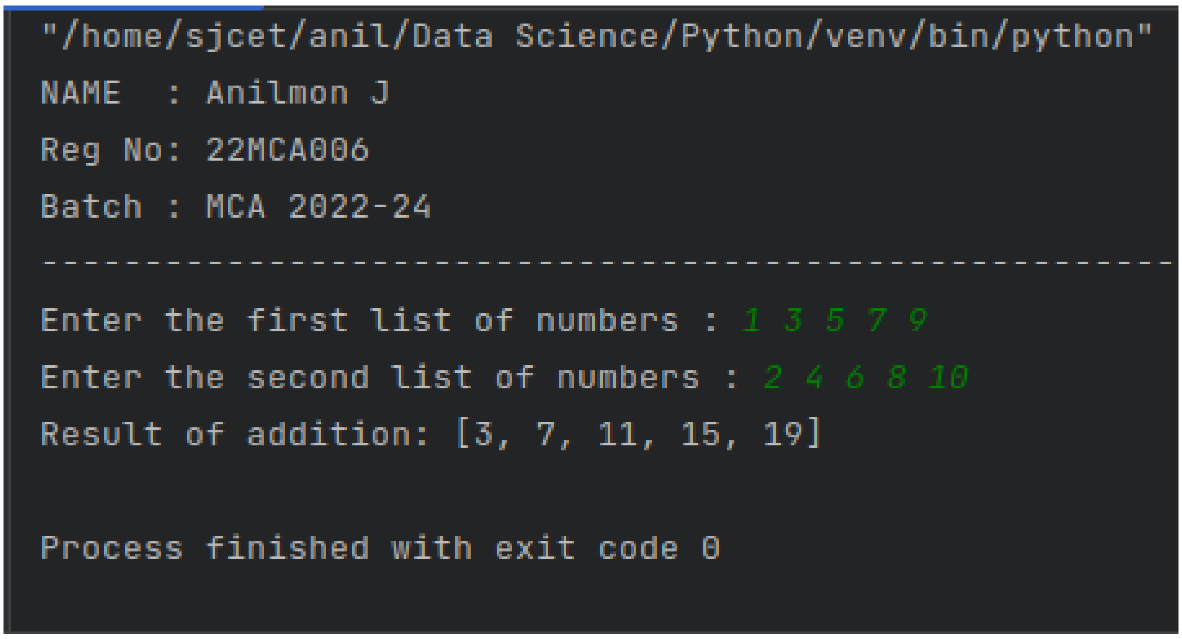
```
"/home/sjcet/anil/Data Science/Python/venv/bin/python" /home/sjcet/anil/Data Science/Python/Lab Cycle - 1/WeeekDa
NAME  : Anilmon J
Reg No: 22MCA006
Batch : MCA 2022-24
---------------------------------------------------------------------------------
List: ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
Type: <class 'list'>
Tuple: ('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday')
Type: <class 'tuple'>
Dictionary: {0: 'Monday', 1: 'Tuesday', 2: 'Wednesday', 3: 'Thursday', 4: 'Friday', 5: 'Saturday', 6: 'Sunday'}
Type: <class 'dict'>
Set: {'Sunday', 'Friday', 'Saturday', 'Wednesday', 'Tuesday', 'Thursday', 'Monday'}
Type: <class 'set'>

Process finished with exit code 0
```

9. Write a program to add elements of given 2 lists.

Code:

```
def add_lists(a,b):
    if len(a) != len(b):
        return None
    result = []
    for i in range(len(a)):
        result.append(a[i] + b[i])
    return result
try:
    print("NAME  : Anilmon J")
    print("Reg No: 22MCA006")
    print("Batch : MCA 2022-24 ")

print("-------------------------------------------------------------------------")
    a = input("Enter the first list of numbers : ").split()
    a = [int(x) for x in a]

    b = input("Enter the second list of numbers : ").split()
    b = [int(x) for x in b]

    result = add_lists(a, b)

    if result is None:
        print("The lists have different lengths and cannot be added.")
    else:
        print("Result of addition:", result)
except ValueError:
    print("Invalid input. Please enter valid numbers separated by spaces.")
```

Output:



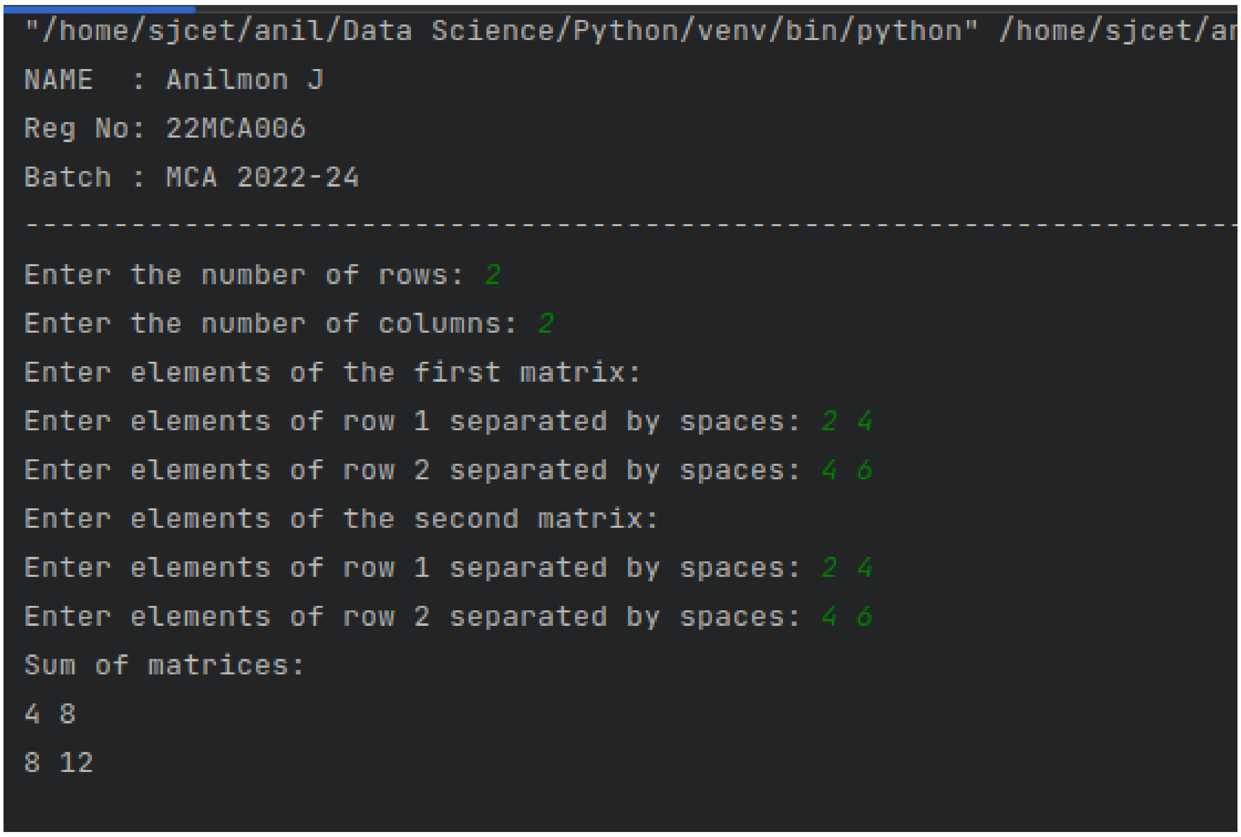
```
"/home/sjcet/anil/Data Science/Python/venv/bin/python"
NAME   : Anilmon J
Reg No: 22MCA006
Batch : MCA 2022-24
-------------------------------------------------------
Enter the first list of numbers : 1 3 5 7 9
Enter the second list of numbers : 2 4 6 8 10
Result of addition: [3, 7, 11, 15, 19]

Process finished with exit code 0
```

10. Write a program to find the sum of 2 matrices using nested List.

Code:

```
def add_matrices(matrix1, matrix2):

    if len(matrix1) != len(matrix2) or len(matrix1[0]) != len(matrix2[0]):
        return None


    result = [[0 for _ in range(len(matrix1[0]))] for _ in range(len(matrix1))]

    for i in range(len(matrix1)):
        for j in range(len(matrix1[0])):
            result[i][j] = matrix1[i][j] + matrix2[i][j]

    return result

try:
    print("NAME  : Anilmon J")
    print("Reg No: 22MCA006")
    print("Batch : MCA 2022-24 ")

print("-----------------------------------------------------------------------------")
    rows = int(input("Enter the number of rows: "))
    cols = int(input("Enter the number of columns: "))

    print("Enter elements of the first matrix:")
    matrix1 = []
    for i in range(rows):
        row = input(f"Enter elements of row {i + 1} separated by spaces:
").split()
        matrix1.append([int(x) for x in row])

    print("Enter elements of the second matrix:")
    matrix2 = []
    for i in range(rows):
        row = input(f"Enter elements of row {i + 1} separated by spaces:
").split()
        matrix2.append([int(x) for x in row])


    result = add_matrices(matrix1, matrix2)

    if result is None:
        print("Matrix dimensions are not compatible for addition.")
    else:
        print("Sum of matrices:")
        for row in result:
            print(" ".join(map(str, row)))
except ValueError:
    print("Invalid input. Please enter valid numbers.")
```

Output:



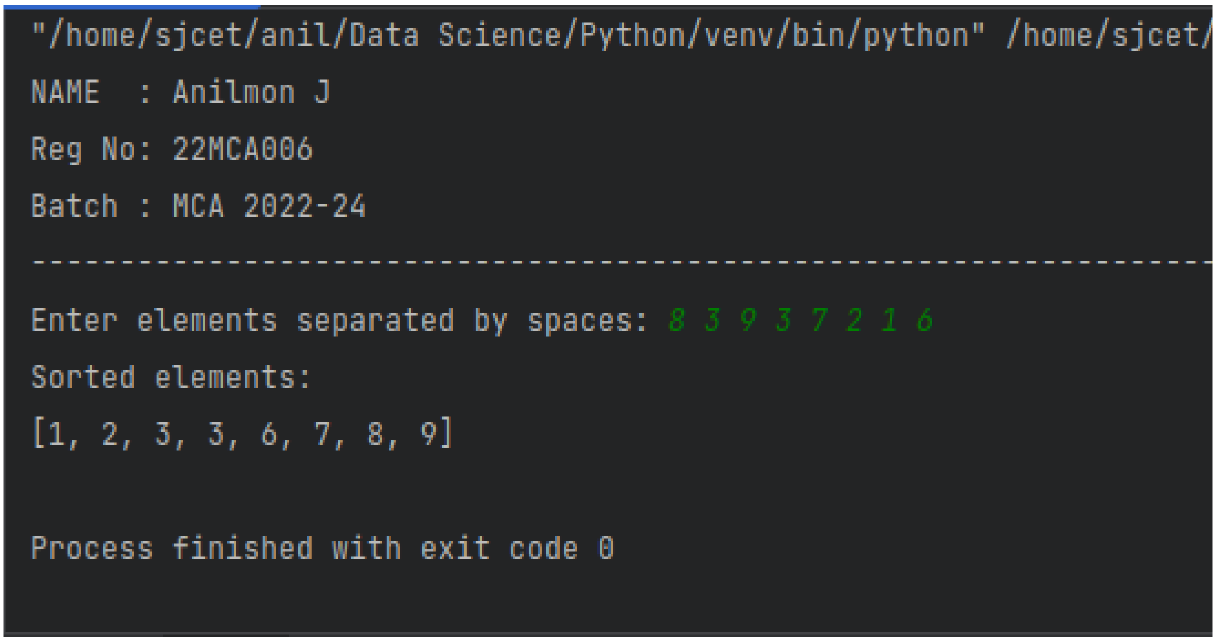
```
"/home/sjcet/anil/Data Science/Python/venv/bin/python" /home/sjcet/ar
NAME  : Anilmon J
Reg No: 22MCA006
Batch : MCA 2022-24
----------------------------------------------------------------------
Enter the number of rows: 2
Enter the number of columns: 2
Enter elements of the first matrix:
Enter elements of row 1 separated by spaces: 2 4
Enter elements of row 2 separated by spaces: 4 6
Enter elements of the second matrix:
Enter elements of row 1 separated by spaces: 2 4
Enter elements of row 2 separated by spaces: 4 6
Sum of matrices:
4 8
8 12
```

11. Write a program to perform bubble sort on a given set of elements.

Code:

```
def bubble_sort(arr):
    n = len(arr)

    for i in range(n):

        swapped = False

        for j in range(0, n - i - 1):

            if arr[j] > arr[j + 1]:
                arr[j], arr[j + 1] = arr[j + 1], arr[j]
                swapped = True

        if not swapped:
            break

try:
    print("NAME  : Anilmon J")
    print("Reg No: 22MCA006")
    print("Batch : MCA 2022-24 ")

print("-----------------------------------------------------------------------------")
    elements = input("Enter elements separated by spaces: ").split()
    elements = [int(x) for x in elements]

    bubble_sort(elements)

    print("Sorted elements:")
    print(elements)
except ValueError:
    print("Invalid input. Please enter valid numbers separated by spaces.")
```

Output:



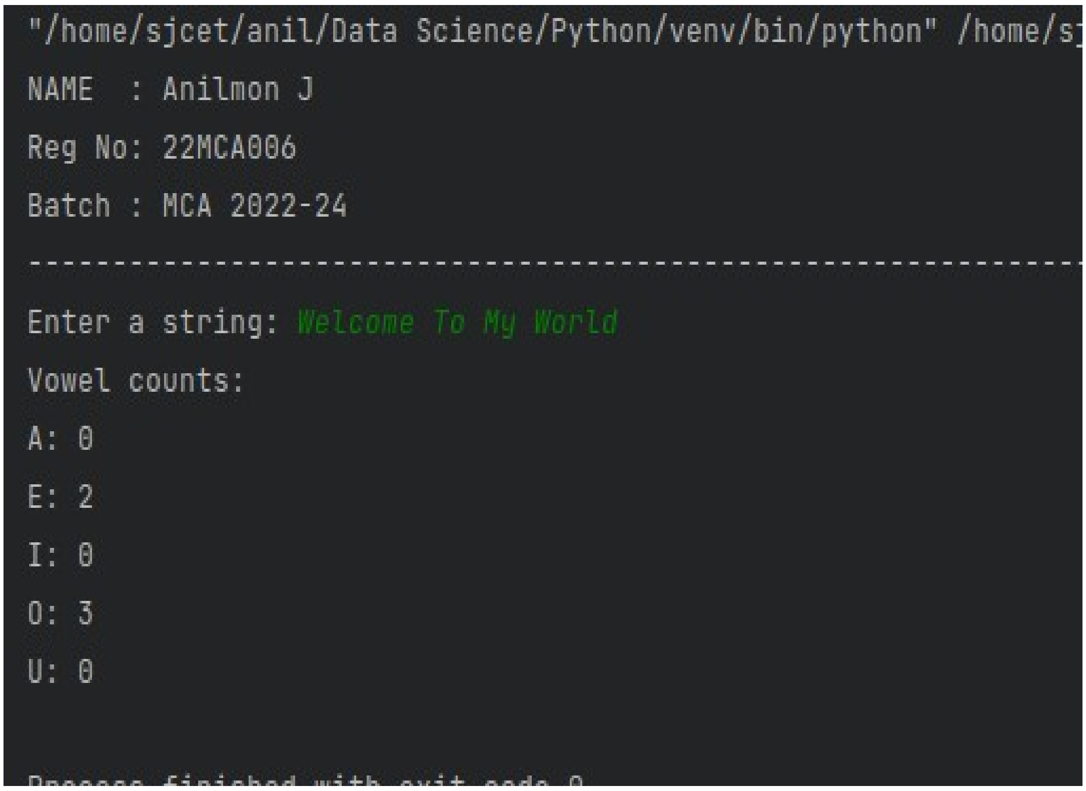
```
"/home/sjcet/anil/Data Science/Python/venv/bin/python" /home/sjcet/
NAME  : Anilmon J
Reg No: 22MCA006
Batch : MCA 2022-24
-------------------------------------------------------------------
Enter elements separated by spaces: 8 3 9 3 7 2 1 6
Sorted elements:
[1, 2, 3, 3, 6, 7, 8, 9]

Process finished with exit code 0
```

12. Program to find the count of each vowel in a string(use dictionary)

Code:

```
def count_vowels(string):

    vowel_counts = {'A': 0, 'E': 0, 'I': 0, 'O': 0, 'U': 0}

    string = string.upper()

    for char in string:

        if char in vowel_counts:

            vowel_counts[char] += 1

    return vowel_counts

try:
    print("NAME  : Anilmon J")
    print("Reg No: 22MCA006")
    print("Batch : MCA 2022-24 ")

print("--------------------------------------------------------------------------------")

    input_string = input("Enter a string: ")

    vowel_counts = count_vowels(input_string)

    print("Vowel counts:")
    for vowel, count in vowel_counts.items():
        print(f"{vowel}: {count}")
except ValueError:
    print("Invalid input. Please enter a valid string.")
```

Output:



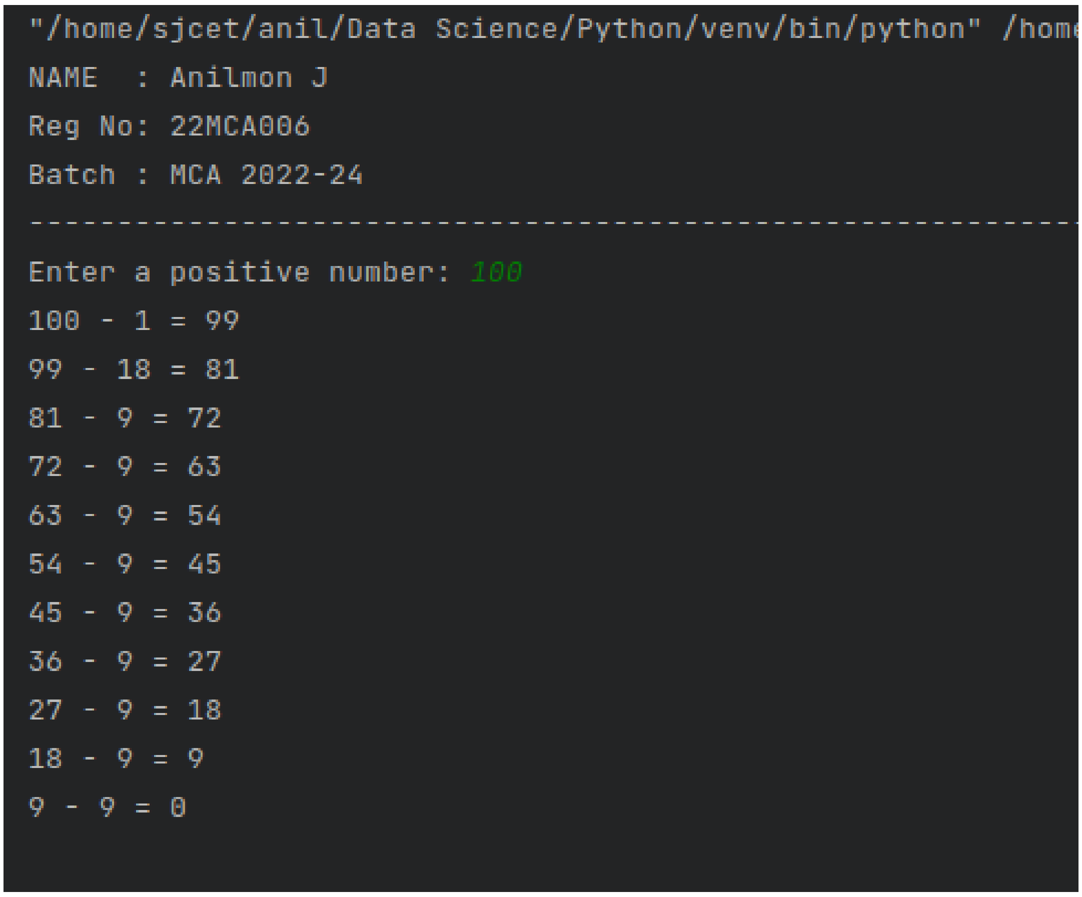
```
"/home/sjcet/anil/Data Science/Python/venv/bin/python" /home/sj
NAME  : Anilmon J
Reg No: 22MCA006
Batch : MCA 2022-24
------------------------------------------------------------
Enter a string: Welcome To My World
Vowel counts:
A: 0
E: 2
I: 0
O: 3
U: 0

Process finished with exit code 0
```

13. Write a Python program that accept a positive number and subtract from this number the sum of its digits and so on. Continues this operation until the number is positive(eg: 256-&gt;2+5+6=13
256-13=243
243-9=232……..

Code:

```
def sum_of_digits(n):

    digit_sum = 0

    while n > 0:
        digit_sum += n % 10
        n //= 10

    return digit_sum

try:
    print("NAME  : Anilmon J")
    print("Reg No: 22MCA006")
    print("Batch : MCA 2022-24 ")

print("-----------------------------------------------------------------------------")
    num = int(input("Enter a positive number: "))

    if num <= 0:
        print("Please enter a positive number.")
    else:
        while num > 0:
            digit_sum = sum_of_digits(num)
            print(f"{num} - {digit_sum} = {num - digit_sum}")
            num -= digit_sum
except ValueError:
    print("Invalid input. Please enter a valid positive number.")
```

Output:



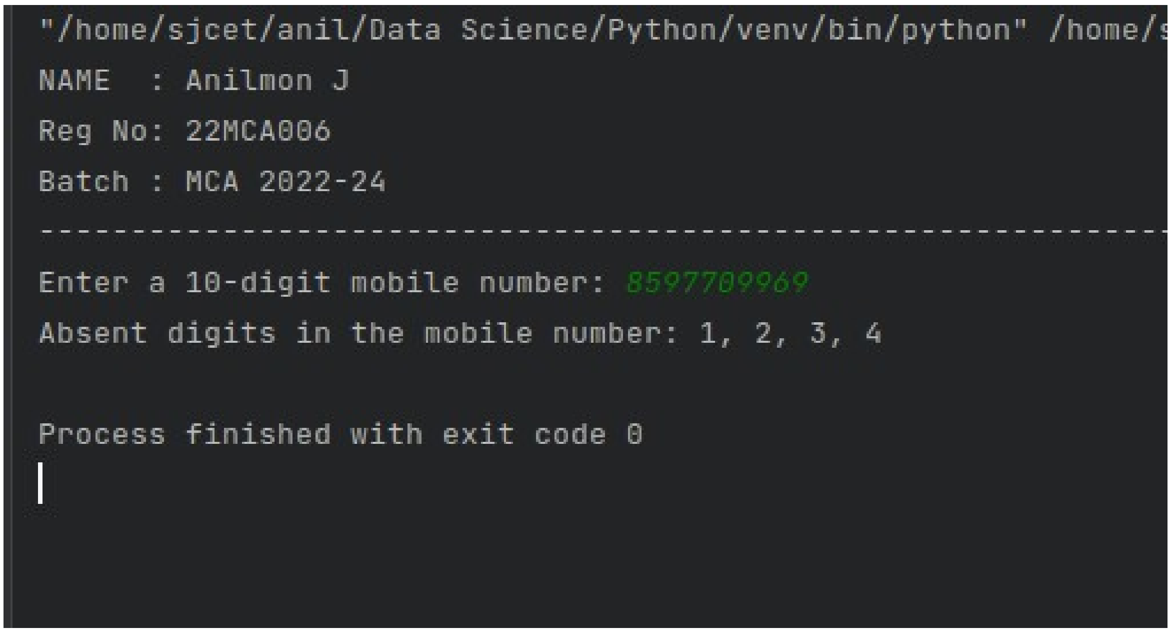
```
"/home/sjcet/anil/Data Science/Python/venv/bin/python" /home
NAME  : Anilmon J
Reg No: 22MCA006
Batch : MCA 2022-24
------------------------------------------------------------
Enter a positive number: 100
100 - 1 = 99
99 - 18 = 81
81 - 9 = 72
72 - 9 = 63
63 - 9 = 54
54 - 9 = 45
45 - 9 = 36
36 - 9 = 27
27 - 9 = 18
18 - 9 = 9
9 - 9 = 0
```

14. Write a Python program that accepts a 10 digit mobile number, and find the digits which are absent in a given mobile number.

Code:

```
def find_absent_digits(mobile_number):

    all_digits = set("0123456789")

    mobile_digits = set(mobile_number)

    absent_digits = all_digits - mobile_digits

    return sorted(list(absent_digits))

try:
    print("NAME  : Anilmon J")
    print("Reg No: 22MCA006")
    print("Batch : MCA 2022-24 ")

print("-----------------------------------------------------------------------------")
    mobile_number = input("Enter a 10-digit mobile number: ")

    if len(mobile_number) == 10 and mobile_number.isdigit():
        absent_digits = find_absent_digits(mobile_number)

        if absent_digits:
            print("Absent digits in the mobile number:", ', '.join(absent_digits))
        else:
            print("The mobile number contains all digits from 0 to 9.")
    else:
        print("Invalid input. Please enter a valid 10-digit mobile number.")
except ValueError:
    print("Invalid input. Please enter a valid 10-digit mobile number.")
```

Output:



```
"/home/sjcet/anil/Data Science/Python/venv/bin/python" /home/s
NAME  : Anilmon J
Reg No: 22MCA006
Batch : MCA 2022-24
---------------------------------------------------------------
Enter a 10-digit mobile number: 8597709969
Absent digits in the mobile number: 1, 2, 3, 4

Process finished with exit code 0
```