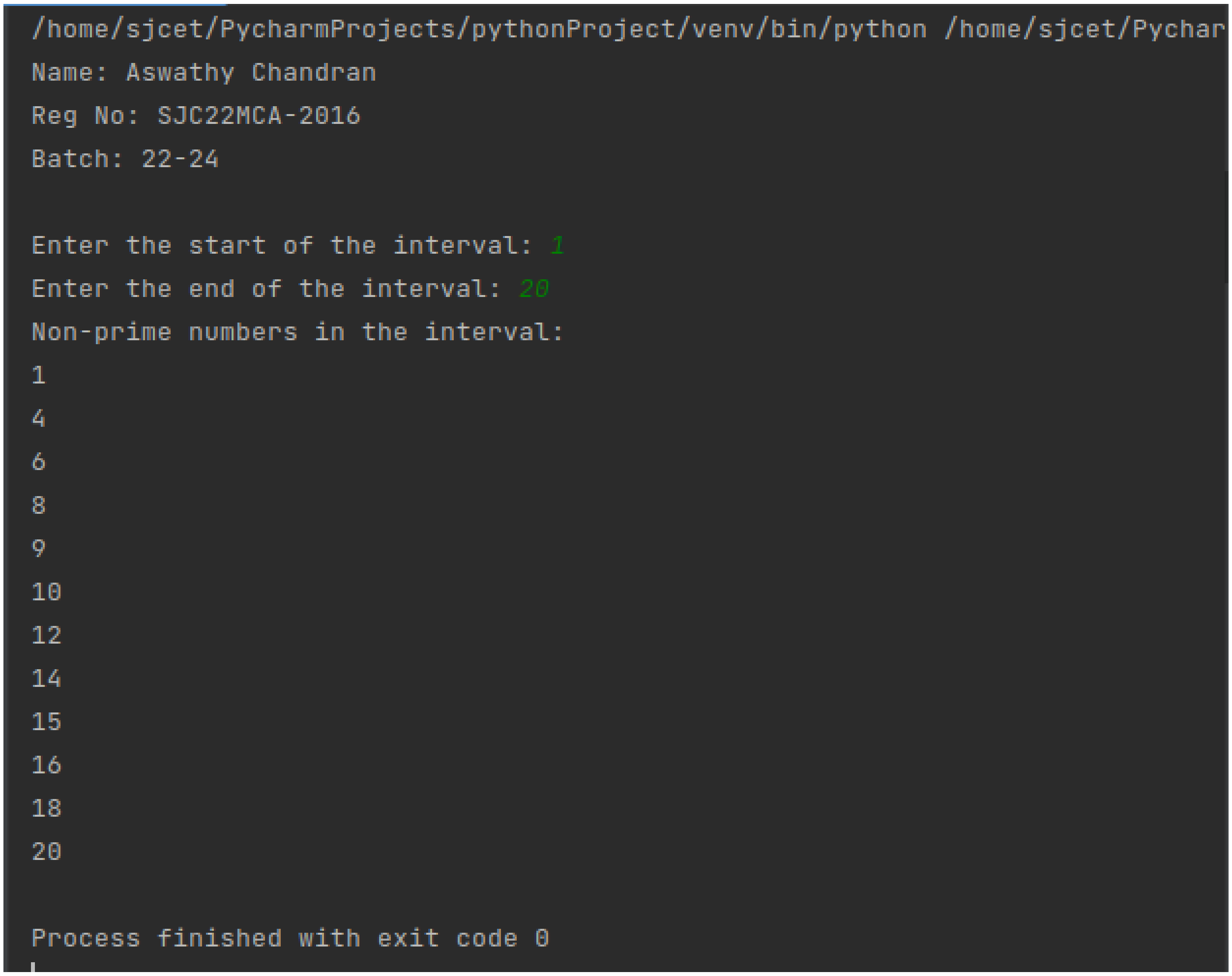
LAB CYCLE-1

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

Code

```
print("Name: Aswathy Chandran")
print("Reg No: SJC22MCA-2016")
print("Batch: 22-24")
print()
def is_prime(num):
    if num <= 1:

        return False
    elif num <= 3:
        return True
    elif num % 2 == 0 or num % 3 == 0:
        return False
    i = 5
    while i * i <= num:
        if num % i == 0 or num % (i + 2) == 0:
            return False
        i += 6
    return True


def print_non_primes_in_interval(start, end):
    for number in range(start, end + 1):
        if not is_prime(number):
            print(number)


if __name__ == "__main__":
    start = int(input("Enter the start of the interval: "))
    end = int(input("Enter the end of the interval: "))

    print("Non-prime numbers in the interval:")
    print_non_primes_in_interval(start, end)
```

Output



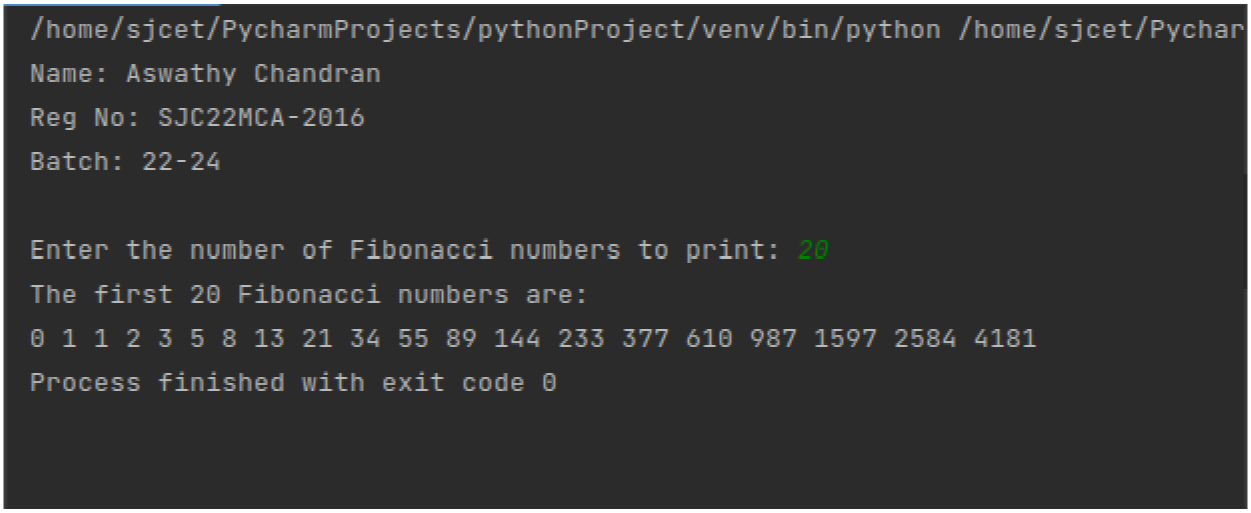
```
/home/sjcet/PycharmProjects/pythonProject/venv/bin/python /home/sjcet/Pychar
Name: Aswathy Chandran
Reg No: SJC22MCA-2016
Batch: 22-24

Enter the start of the interval: 1
Enter the end of the interval: 20
Non-prime numbers in the interval:
1
4
6
8
9
10
12
14
15
16
18
20

Process finished with exit code 0
```

2. Program to print the first N Fibonacci numbers.

Code

```
print("Name: Aswathy Chandran")
print("Reg No: SJC22MCA-2016")
print("Batch: 22-24")
print()
def print_fibonacci_numbers(n):
    fib_sequence = []
    a, b = 0, 1
    for _ in range(n):
        fib_sequence.append(a)
        a, b = b, a + b
    return fib_sequence

if __name__ == "__main__":
    try:
        n = int(input("Enter the number of Fibonacci numbers to
print: "))
        if n <= 0:
            print("Please enter a positive integer.")
        else:
            fibonacci_sequence = print_fibonacci_numbers(n)
            print(f"The first {n} Fibonacci numbers are:")
            for number in fibonacci_sequence:
                print(number, end=" ")
    except ValueError:
        print("Invalid input. Please enter a positive integer.")
```

Output



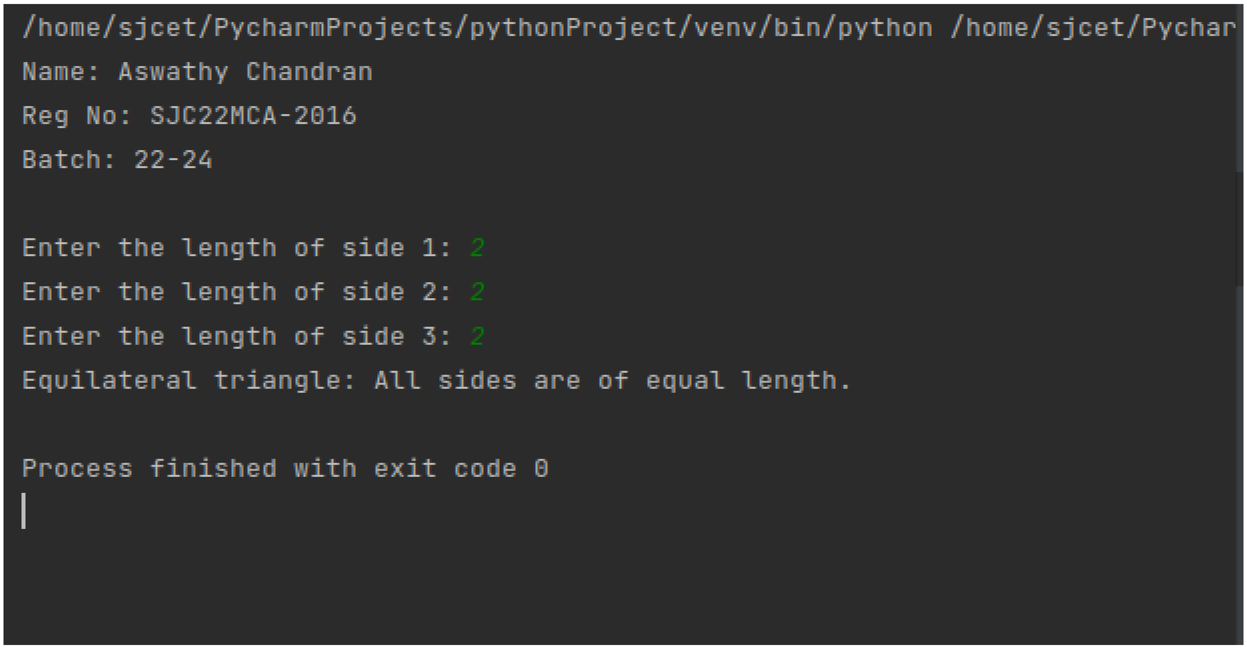
```
/home/sjcet/PycharmProjects/pythonProject/venv/bin/python /home/sjcet/Pychar
Name: Aswathy Chandran
Reg No: SJC22MCA-2016
Batch: 22-24

Enter the number of Fibonacci numbers to print: 20
The first 20 Fibonacci numbers are:
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181
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: Aswathy Chandran")
print("Reg No: SJC22MCA-2016")
print("Batch: 22-24")
print()
def classify_triangle(a, b, c):
    if a <= 0 or b <= 0 or c <= 0:
        return "Invalid input: All sides of the triangle must be
greater than zero."
    elif a + b <= c or a + c <= b or b + c <= a:
        return "Not a triangle: The sum of the lengths of any
two sides must be greater than the length of the third side."
    elif a == b == c:
        return "Equilateral triangle: All sides are of equal
length."
    elif a == b or b == c or a == c:
        return "Isosceles triangle: Two sides are of equal
length."
    else:
        return "Scalene triangle: All sides have different
lengths."

# Input side lengths of the triangle
side1 = float(input("Enter the length of side 1: "))
side2 = float(input("Enter the length of side 2: "))
side3 = float(input("Enter the length of side 3: "))

result = classify_triangle(side1, side2, side3)
print(result)
```

<u>Output</u>



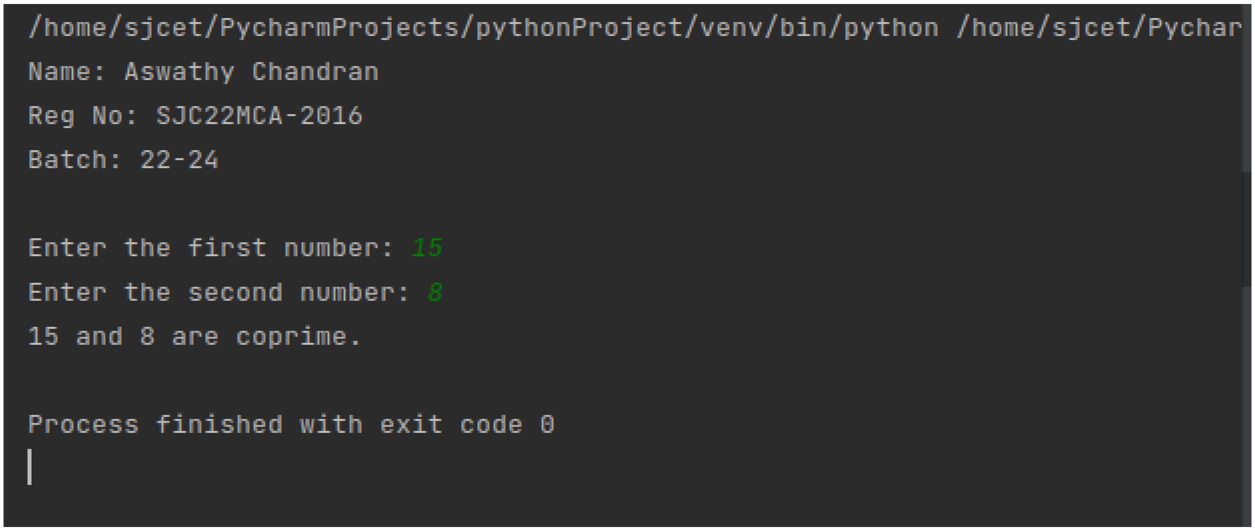
```
/home/sjcet/PycharmProjects/pythonProject/venv/bin/python /home/sjcet/Pychar
Name: Aswathy Chandran
Reg No: SJC22MCA-2016
Batch: 22-24

Enter the length of side 1: 2
Enter the length of side 2: 2
Enter the length of side 3: 2
Equilateral triangle: All sides are of equal length.

Process finished with exit code 0
```

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

Code

```
print("Name: Aswathy Chandran")
print("Reg No: SJC22MCA-2016")
print("Batch: 22-24")
print()
import math

def are_coprime(a, b):
    gcd = math.gcd(a, b)
    return gcd == 1

# Input two numbers
num1 = int(input("Enter the first number: "))
num2 = int(input("Enter the second number: "))

if are_coprime(num1, num2):
    print(f"{num1} and {num2} are coprime.")
else:
    print(f"{num1} and {num2} are not coprime.")
```

Output



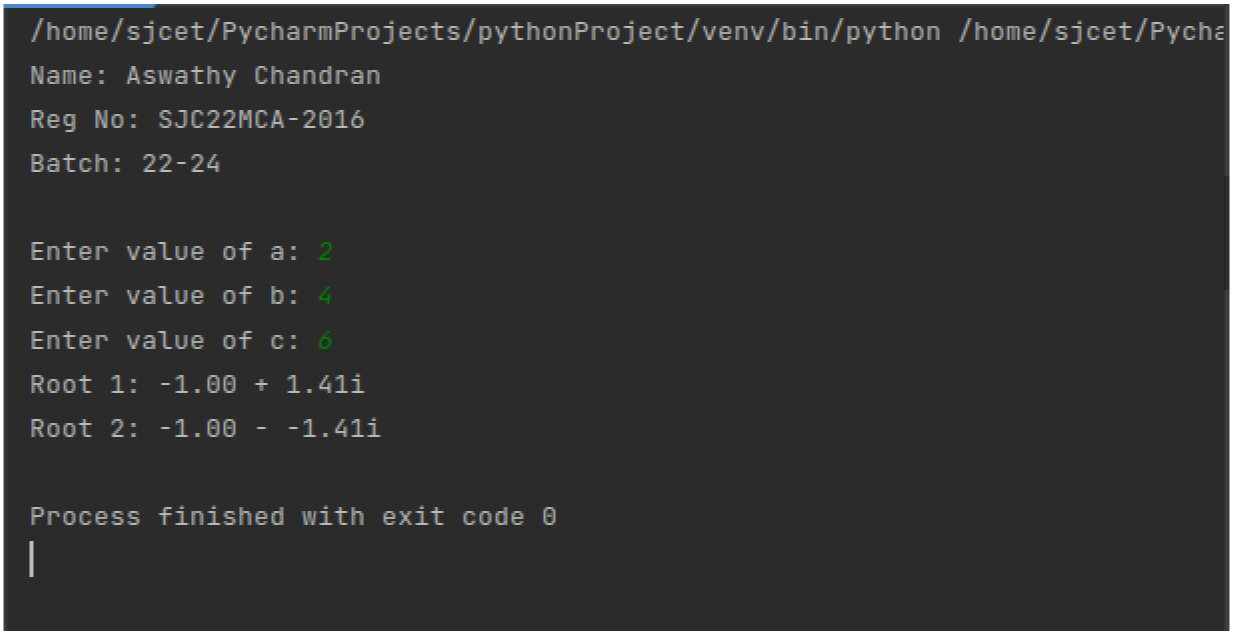
```
/home/sjcet/PycharmProjects/pythonProject/venv/bin/python /home/sjcet/Pychar
Name: Aswathy Chandran
Reg No: SJC22MCA-2016
Batch: 22-24

Enter the first number: 15
Enter the second number: 8
15 and 8 are coprime.

Process finished with exit code 0
```

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

Code

```
import math
print("Name: Aswathy Chandran")
print("Reg No: SJC22MCA-2016")
print("Batch: 22-24")
print()

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



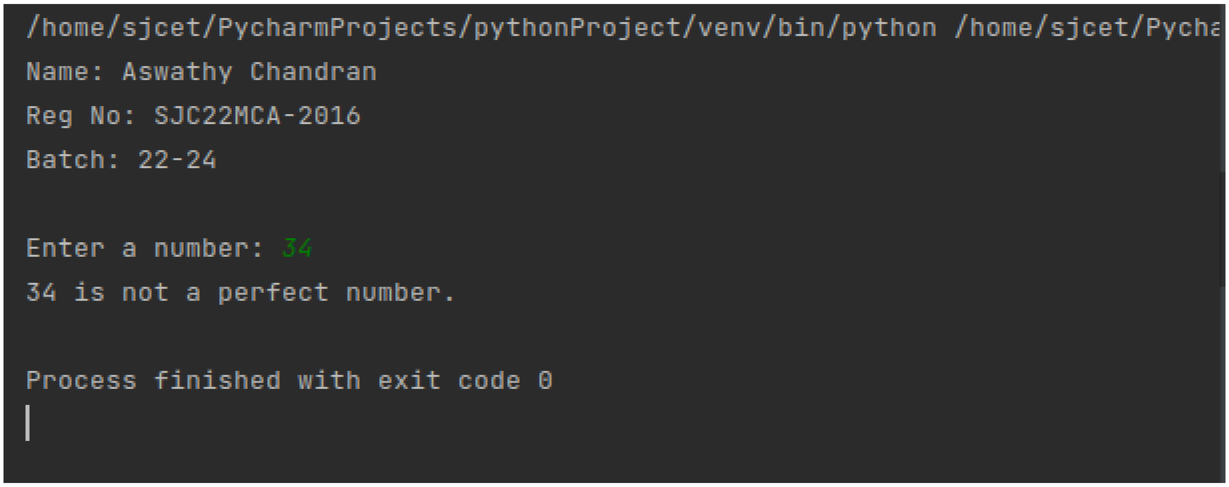
```
/home/sjcet/PycharmProjects/pythonProject/venv/bin/python /home/sjcet/Pycha
Name: Aswathy Chandran
Reg No: SJC22MCA-2016
Batch: 22-24

Enter value of a: 2
Enter value of b: 4
Enter value of c: 6
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 factor =number)

Code

```
def is_perfect_number(num):
    if num <= 0:
        return False
    sum_of_divisors = 0
    for i in range(1, num):
        if num % i == 0:
            sum_of_divisors += i
    return sum_of_divisors == num
try:
    print("Name: Aswathy Chandran")
    print("Reg No: SJC22MCA-2016")
    print("Batch: 22-24")
    print()
    num = int(input("Enter a number: "))
    if is_perfect_number(num):
        print(f"{num} is a perfect number.")
    else:
        print(f"{num} is not a perfect number.")
except ValueError:
    print("Invalid input. Please enter a valid number.")
```

Output



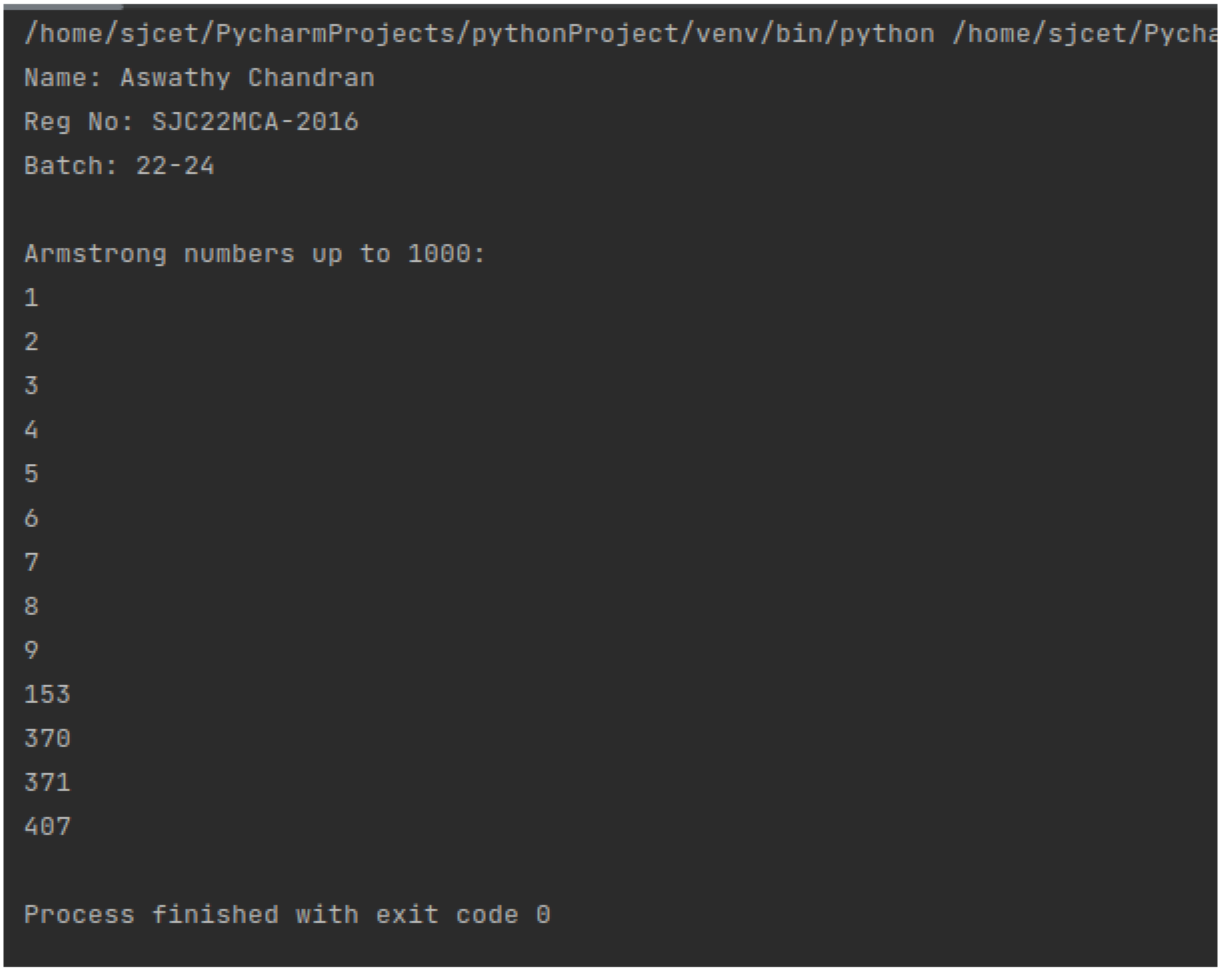
```
/home/sjcet/PycharmProjects/pythonProject/venv/bin/python /home/sjcet/Pycha
Name: Aswathy Chandran
Reg No: SJC22MCA-2016
Batch: 22-24

Enter a number: 34
34 is not a 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: Aswathy Chandran")
print("Reg No: SJC22MCA-2016")
print("Batch: 22-24")
print()
print("Armstrong numbers up to 1000:")
for num in range(1, 1001):
    if is_armstrong_number(num):
        print(num)
```

Output



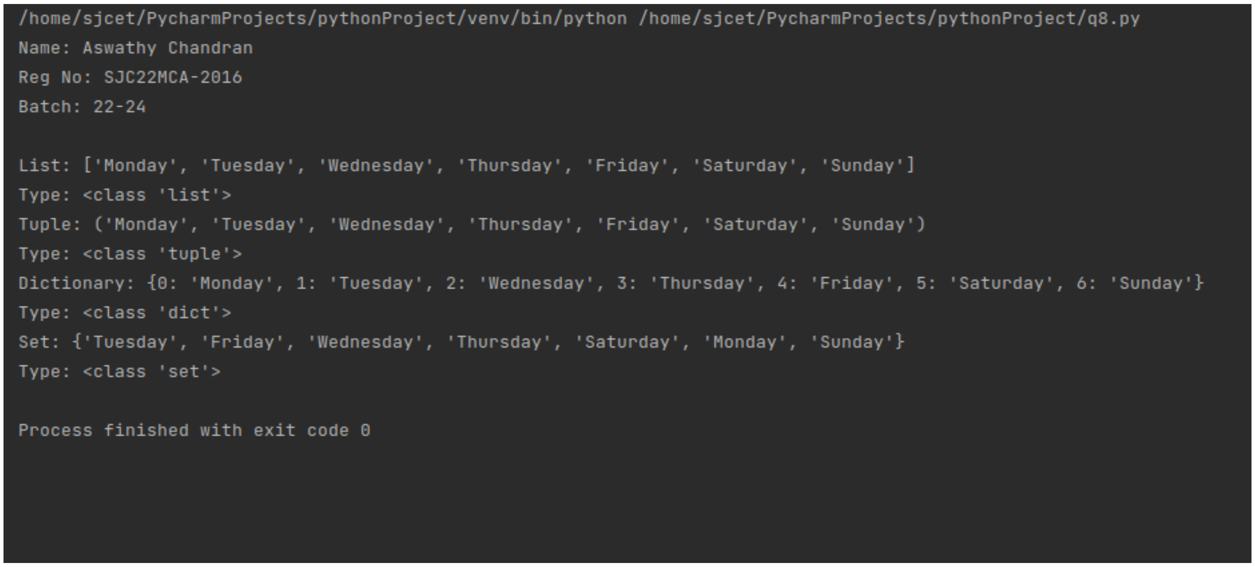
```
/home/sjcet/PycharmProjects/pythonProject/venv/bin/python /home/sjcet/Pycha
Name: Aswathy Chandran
Reg No: SJC22MCA-2016
Batch: 22-24

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: Aswathy Chandran")
print("Reg No: SJC22MCA-2016")
print("Batch: 22-24")
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



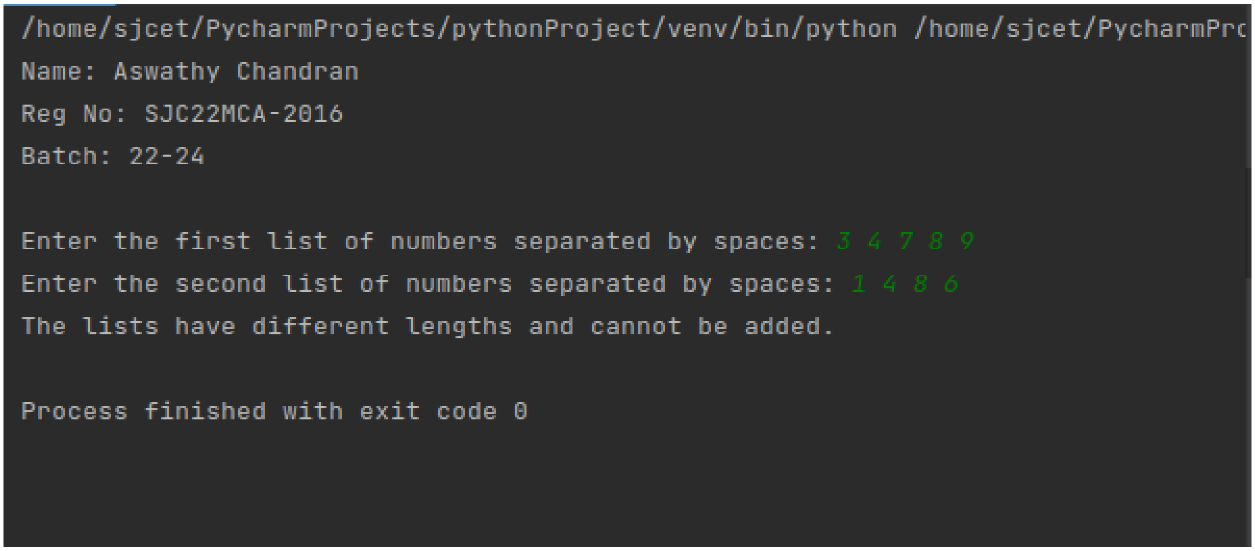
```
/home/sjcet/PycharmProjects/pythonProject/venv/bin/python /home/sjcet/PycharmProjects/pythonProject/q8.py
Name: Aswathy Chandran
Reg No: SJC22MCA-2016
Batch: 22-24

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: {'Tuesday', 'Friday', 'Wednesday', 'Thursday', 'Saturday', 'Monday', 'Sunday'}
Type: <class 'set'>

Process finished with exit code 0
```

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

Code

```
def add_lists(list1, list2):

    if len(list1) != len(list2):
        return None

    result = []

    for i in range(len(list1)):
        result.append(list1[i] + list2[i])

    return result

try:
    print("Name: Aswathy Chandran")
    print("Reg No: SJC22MCA-2016")
    print("Batch: 22-24")
    print()
    list1 = input("Enter the first list of numbers separated by
spaces: ").split()
    list1 = [int(x) for x in list1]

    list2 = input("Enter the second list of numbers separated by
spaces: ").split()
    list2 = [int(x) for x in list2]

    result = add_lists(list1, list2)

    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



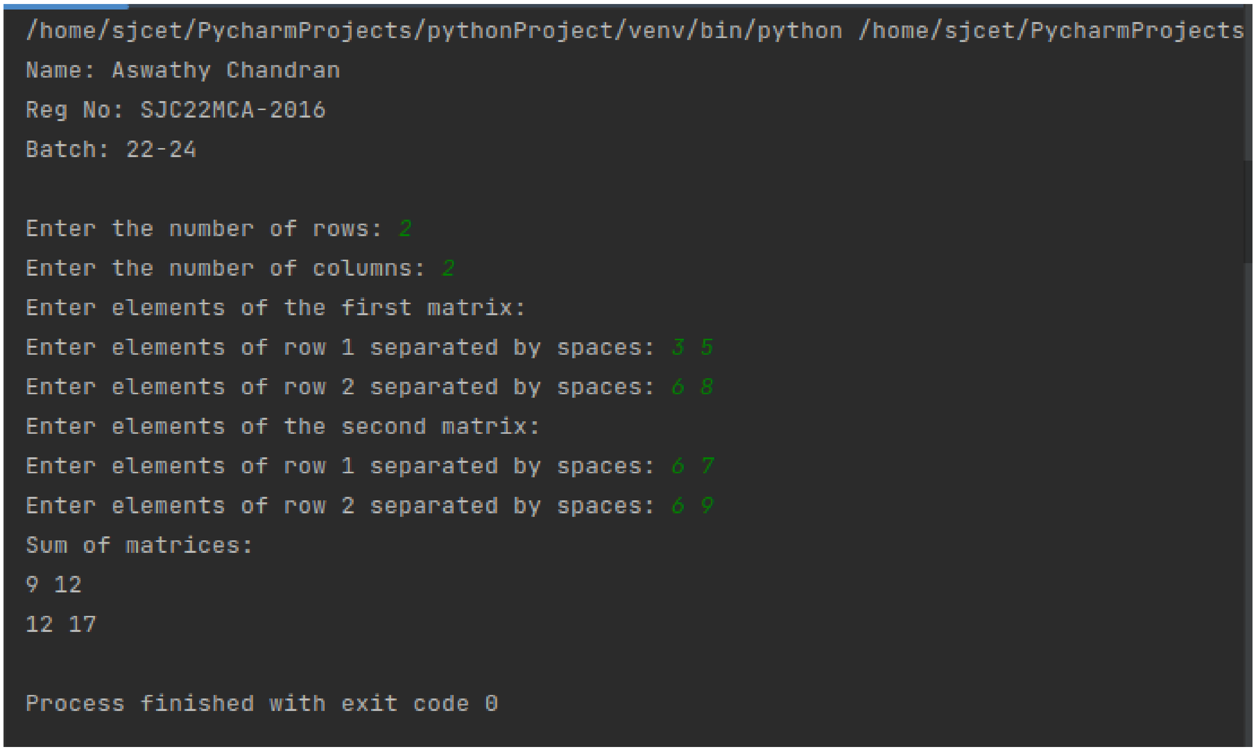
```
/home/sjcet/PycharmProjects/pythonProject/venv/bin/python /home/sjcet/PycharmPro
Name: Aswathy Chandran
Reg No: SJC22MCA-2016
Batch: 22-24

Enter the first list of numbers separated by spaces: 3 4 7 8 9
Enter the second list of numbers separated by spaces: 1 4 8 6
The lists have different lengths and cannot be added.

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: Aswathy Chandran")
    print("Reg No: SJC22MCA-2016")
    print("Batch: 22-24")
    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



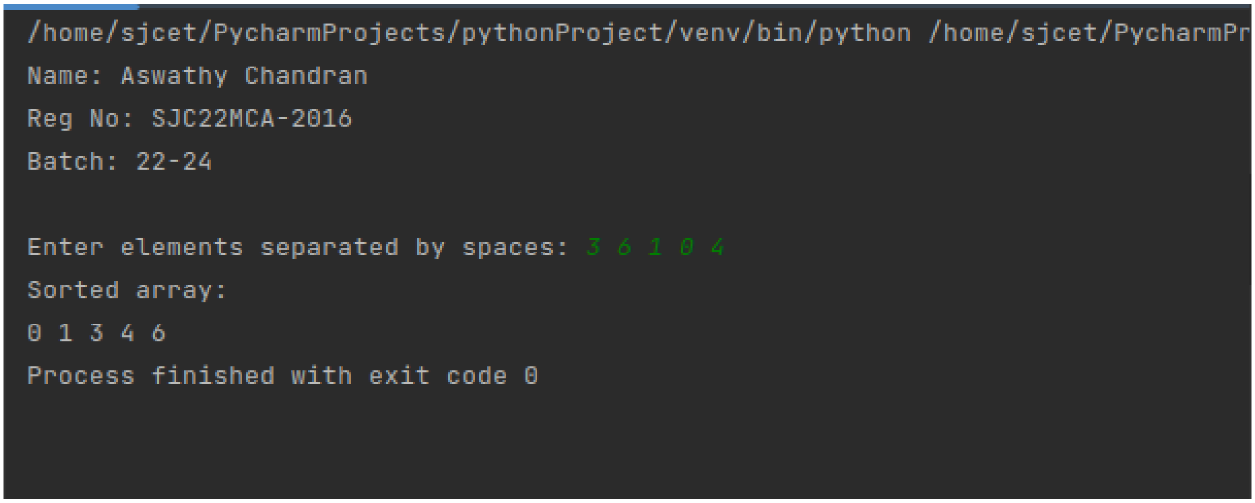
```
/home/sjcet/PycharmProjects/pythonProject/venv/bin/python /home/sjcet/PycharmProjects
Name: Aswathy Chandran
Reg No: SJC22MCA-2016
Batch: 22-24

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: 3 5
Enter elements of row 2 separated by spaces: 6 8
Enter elements of the second matrix:
Enter elements of row 1 separated by spaces: 6 7
Enter elements of row 2 separated by spaces: 6 9
Sum of matrices:
9 12
12 17

Process finished with exit code 0
```

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

Code

```
print("Name: Aswathy Chandran")
print("Reg No: SJC22MCA-2016")
print("Batch: 22-24")
print()
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



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

bubble_sort(elements)

print("Sorted array:")
for element in elements:
    print(element, end=" ")
```

Output



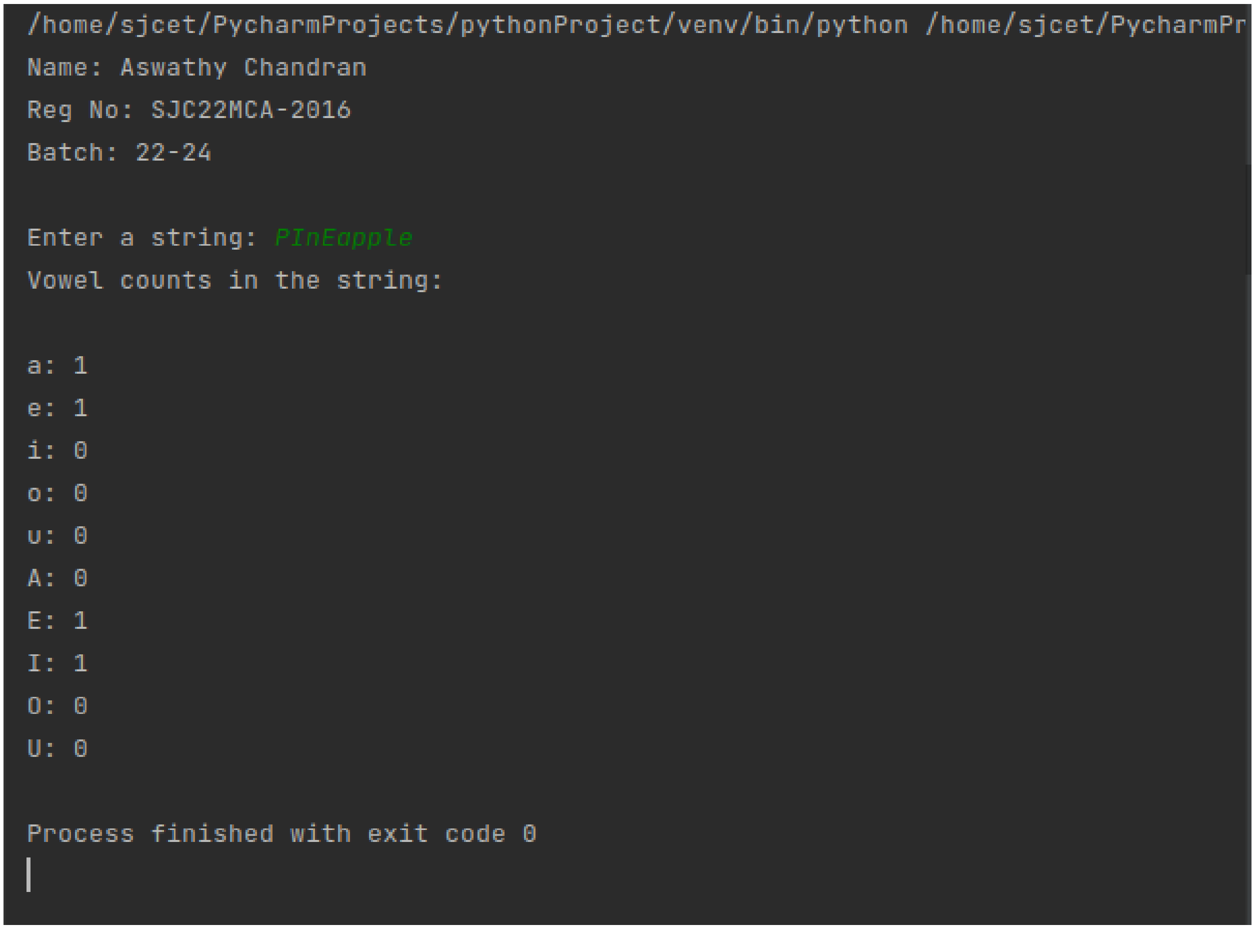
```
/home/sjcet/PycharmProjects/pythonProject/venv/bin/python /home/sjcet/PycharmPr
Name: Aswathy Chandran
Reg No: SJC22MCA-2016
Batch: 22-24

Enter elements separated by spaces: 3 6 1 0 4
Sorted array:
0 1 3 4 6
Process finished with exit code 0
```

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

Code

```
print("Name: Aswathy Chandran")
print("Reg No: SJC22MCA-2016")
print("Batch: 22-24")
print()
def count_vowels(input_string):
    vowel_counts = {'a': 0, 'e': 0, 'i': 0, 'o': 0, 'u': 0,
                    'A': 0, 'E': 0, 'I': 0, 'O': 0, 'U': 0}

    for char in input_string:

        if char in vowel_counts:
            vowel_counts[char] += 1

    return vowel_counts


input_string = input("Enter a string: ")

vowel_counts = count_vowels(input_string)

print("Vowel counts in the string:")
print()
for vowel, count in vowel_counts.items():
    print(f"{vowel}: {count}")
```

Output



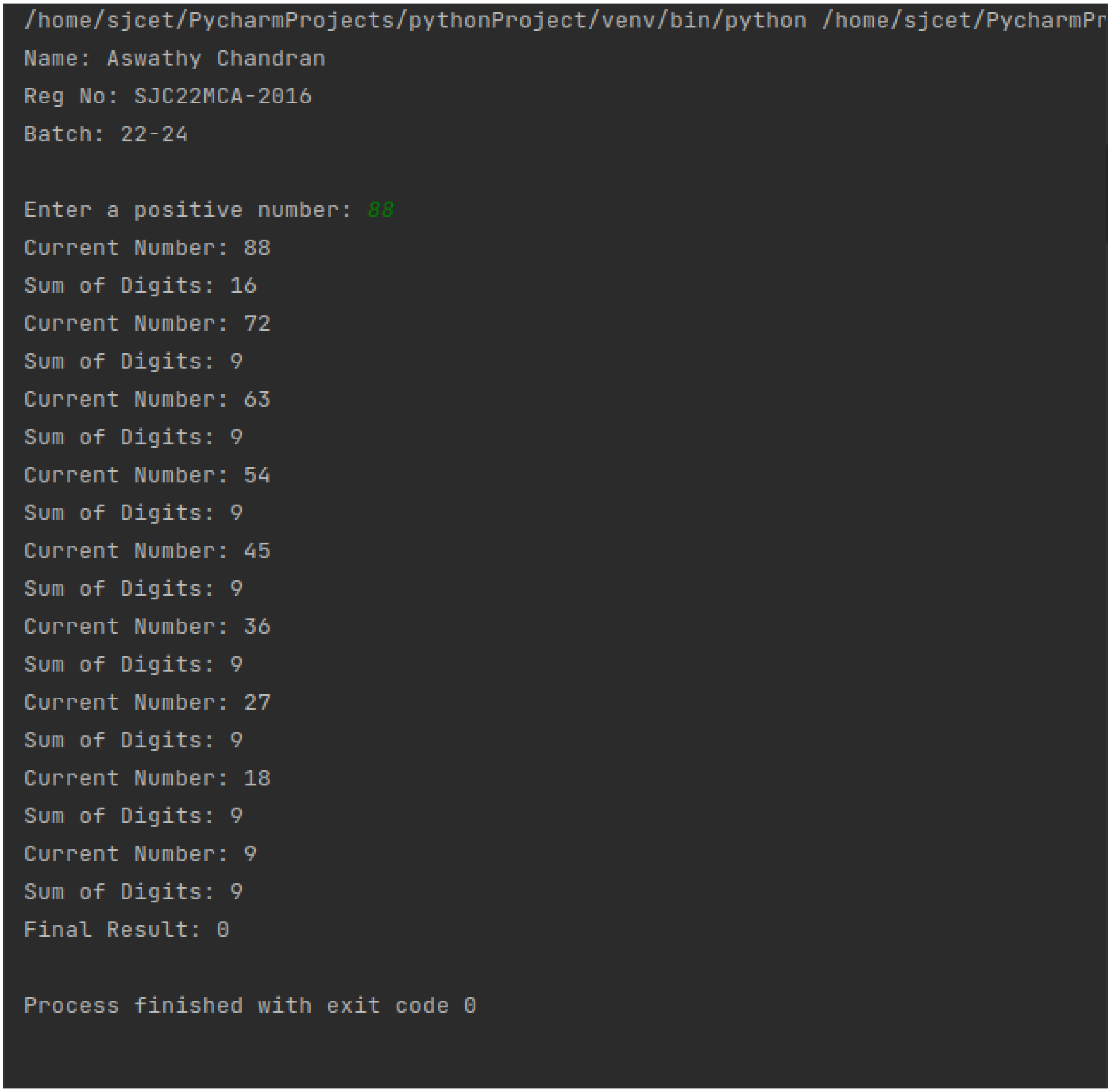
```
/home/sjcet/PycharmProjects/pythonProject/venv/bin/python /home/sjcet/PycharmPr
Name: Aswathy Chandran
Reg No: SJC22MCA-2016
Batch: 22-24

Enter a string: PInEapple
Vowel counts in the string:

a: 1
e: 1
i: 0
o: 0
u: 0
A: 0
E: 1
I: 1
O: 0
U: 0

Process finished with exit code 0
```

13. Write a Python program that accepts 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->2+5+6=13,256-13=243,243-9=232)

Code

```
print("Name: Aswathy Chandran")
print("Reg No: SJC22MCA-2016")
print("Batch: 22-24")
print()
def sum_of_digits(number):

    digit_sum = 0
    while number > 0:
        digit_sum += number % 10
        number //= 10
    return digit_sum

def subtract_until_non_positive(number):
    while number > 0:
        print("Current Number:", number)
        digit_sum = sum_of_digits(number)
        print("Sum of Digits:", digit_sum)
        number -= digit_sum
    print("Final Result:", number)

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

if num <= 0:
    print("Please enter a positive number.")
else:
    subtract_until_non_positive(num)
```

Output



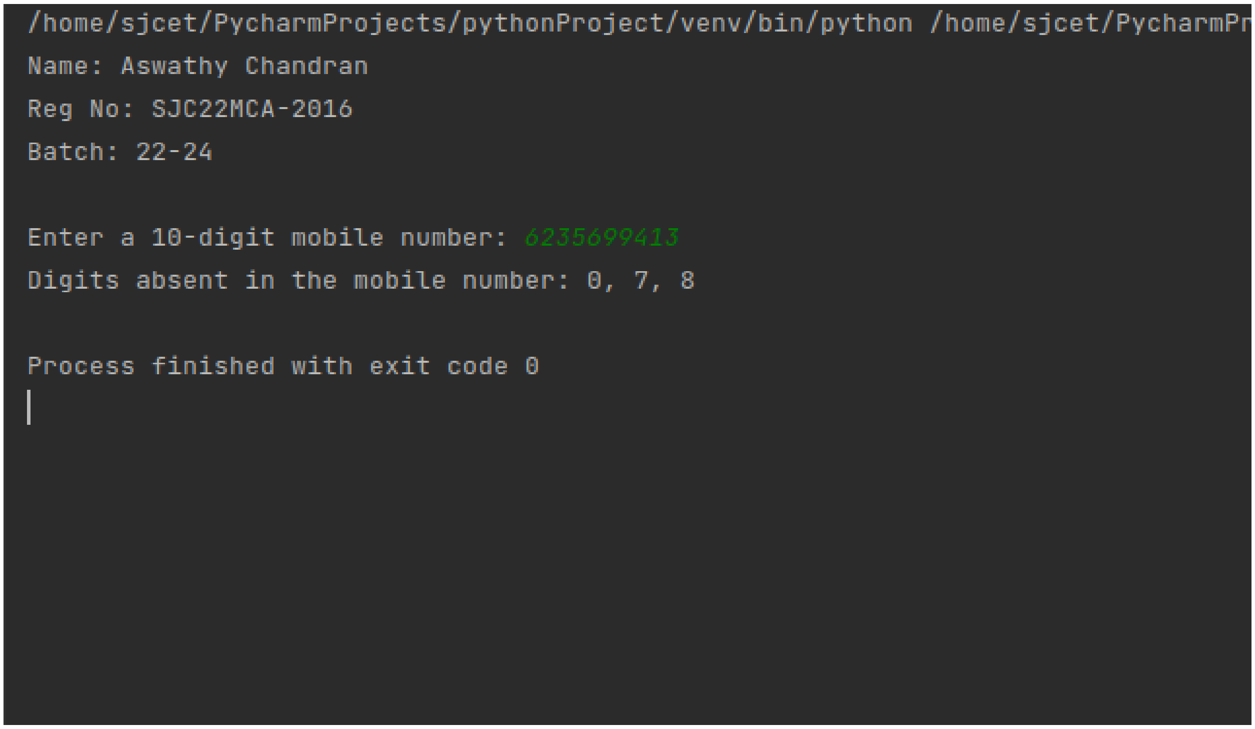
```
/home/sjcet/PycharmProjects/pythonProject/venv/bin/python /home/sjcet/PycharmPr
Name: Aswathy Chandran
Reg No: SJC22MCA-2016
Batch: 22-24

Enter a positive number: 88
Current Number: 88
Sum of Digits: 16
Current Number: 72
Sum of Digits: 9
Current Number: 63
Sum of Digits: 9
Current Number: 54
Sum of Digits: 9
Current Number: 45
Sum of Digits: 9
Current Number: 36
Sum of Digits: 9
Current Number: 27
Sum of Digits: 9
Current Number: 18
Sum of Digits: 9
Current Number: 9
Sum of Digits: 9
Final Result: 0

Process finished with exit code 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

```
print("Name: Aswathy Chandran")
print("Reg No: SJC22MCA-2016")
print("Batch: 22-24")
print()
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))


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("Digits absent in the mobile number:", ", 
".join(absent_digits))
    else:
        print("All digits are present in the mobile number.")
else:
    print("Invalid input. Please enter a valid 10-digit mobile 
number.")
```

Output



```
/home/sjcet/PycharmProjects/pythonProject/venv/bin/python /home/sjcet/PycharmPr
Name: Aswathy Chandran
Reg No: SJC22MCA-2016
Batch: 22-24

Enter a 10-digit mobile number: 6235699413
Digits absent in the mobile number: 0, 7, 8

Process finished with exit code 0
```