

CYCLE - 1

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

```
def is_prime(num):
    if num <= 1:
        return False
    if num <= 3:
        return True
    if 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

print("*****")
print("Name: Ashish P S")
print("Roll No: SJC22MCA-2015")
print("Batch: 2022-24")
print("*****")
print()
start = int(input("Enter the start of the interval: "))
end = int(input("Enter the end of the interval: "))

print(f"Non-prime numbers in the interval [{start}, {end}]:")
for num in range(start, end + 1):
    if not is_prime(num):
        print(num, end=" ")
```

Run:  non_prime x

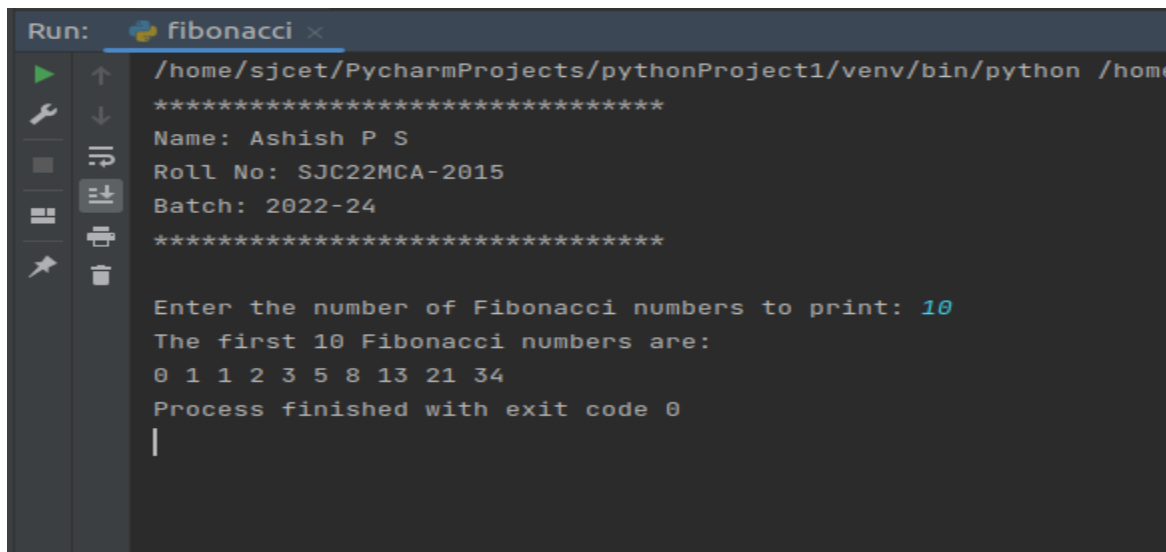
```
/home/sjcet/PycharmProjects/pythonProject1/venv/bin/python /home/sjcet/PycharmProjects/pythonProject1/non_prime.py  
*****  
Name: Ashish P S  
Roll No: SJC22MCA-2015  
Batch: 2022-24  
*****  
  
Enter the start of the interval: 1  
Enter the end of the interval: 50  
Non-prime numbers in the interval [1, 50]:  
1 4 6 8 9 10 12 14 15 16 18 20 21 22 24 25 26 27 28 30 32 33 34 35 36 38 39 40 42 44 45 46 48 49 50  
Process finished with exit code 0  
|
```

2. Program to print the first N Fibonacci numbers.

```
def print_fibonacci(n):
    a, b = 0, 1
    count = 0
    while count < n:
        print(a, end=" ")
        next_fib = a + b
        a = b
        b = next_fib
        count += 1

print("*****")
print("Name: Ashish P S")
print("Roll No: SJC22MCA-2015")
print("Batch: 2022-24")
print("*****")
print()
N = int(input("Enter the number of Fibonacci numbers to print: "))

if N <= 0:
    print("Please enter a positive integer.")
else:
    print(f"The first {N} Fibonacci numbers are:")
    print_fibonacci(N)
```



```
Run: fibonacci x
/home/sjcet/PycharmProjects/pythonProject1/venv/bin/python /home
*****
Name: Ashish P S
Roll No: SJC22MCA-2015
Batch: 2022-24
*****

Enter the number of Fibonacci numbers to print: 10
The first 10 Fibonacci numbers are:
0 1 1 2 3 5 8 13 21 34
Process finished with exit code 0
|
```

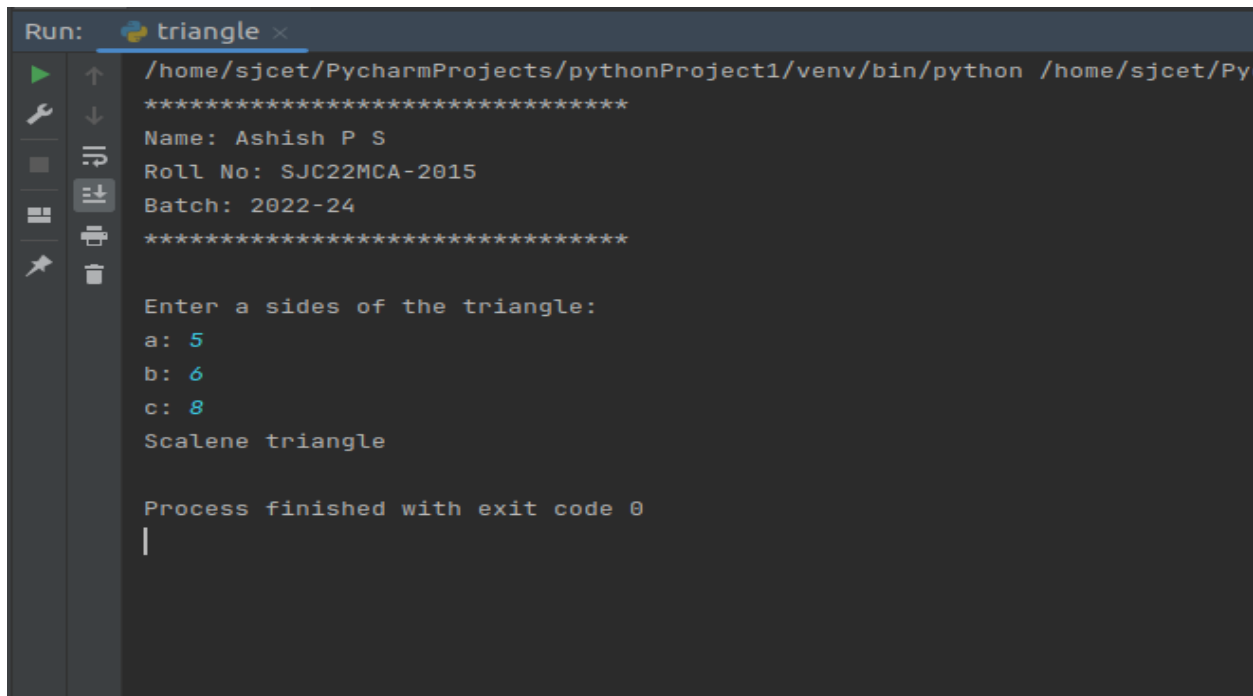
3. Given sides of a triangle, write a program to check whether given triangle is an isosceles, equilateral or scalene.

```
print("*****")
print("Name: Ashish P S")
print("Roll No: SJC22MCA-2015")
print("Batch: 2022-24")
print("*****")
print()
print("Enter a sides of the triangle: ")
a = int(input("a: "))
b = int(input("b: "))
c = int(input("c: "))

if a == b == c:
    print("Equilateral triangle")

elif a == b or b == c or a == c:
    print("Isosceles triangle")

else:
    print("Scalene triangle")
```



```
Run: triangle x
/home/sjcet/PycharmProjects/pythonProject1/venv/bin/python /home/sjcet/Py
*****
Name: Ashish P S
Roll No: SJC22MCA-2015
Batch: 2022-24
*****

Enter a sides of the triangle:
a: 5
b: 6
c: 8
Scalene triangle

Process finished with exit code 0
|
```

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

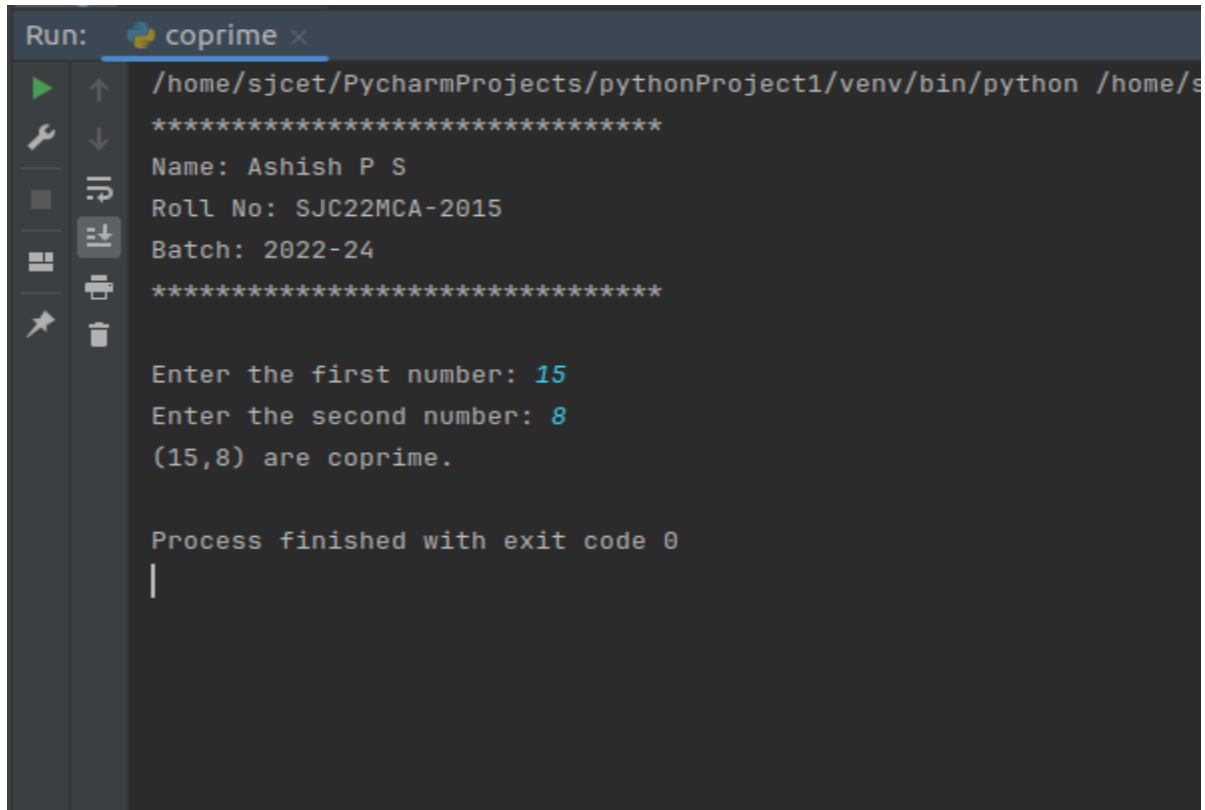
```
def calculate_gcd(a, b):
    while b:
        a, b = b, a % b
    return a
```

```
def are_coprime(a, b):
    gcd = calculate_gcd(a, b)
    return gcd == 1
```

```
print("*****")
print("Name: Ashish P S")
print("Roll No: SJC22MCA-2015")
print("Batch: 2022-24")
print("*****")
print()
num1 = int(input("Enter the first number: "))
num2 = int(input("Enter the second number: "))
```

```
if are_coprime(num1, num2):
    print(f"({num1},{num2}) are coprime.")
else:
```

```
print(f"({num1},{num2}) are not coprime.")
```



The screenshot shows a PyCharm Run window titled 'Run: coprime x'. The console output is as follows:

```
/home/sjcet/PycharmProjects/pythonProject1/venv/bin/python /home/s  
*****  
Name: Ashish P S  
Roll No: SJC22MCA-2015  
Batch: 2022-24  
*****  
  
Enter the first number: 15  
Enter the second number: 8  
(15,8) are coprime.  
  
Process finished with exit code 0  
|
```

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

```
import math
print("*****")
print("Name: Ashish P S")
print("Roll No: SJC22MCA-2015")
print("Batch: 2022-24")
print("*****")
print()
a = float(input("Enter value of a: "))
b = float(input("Enter value of b: "))
c = float(input("Enter value of c: "))
discr = b**2 - 4*a*c

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

```
Run: quad_eqn x
/home/sjcet/PycharmProjects/pythonProject1/venv/bin/pytho
*****
Name: Ashish P S
Roll No: SJC22MCA-2015
Batch: 2022-24
*****

Enter value of a: 4
Enter value of b: 5
Enter value of c: 6
Root 1: -0.62 + 1.05i
Root 2: -0.62 - -1.05i

Process finished with exit code 0
|
```


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

```
print("*****")
print("Name: Ashish P S")
print("Roll No: SJC22MCA-2015")
print("Batch: 2022-24")
print("*****")
print()
```

```
def is_perfect_number(num):
    if num <= 0:
        return False

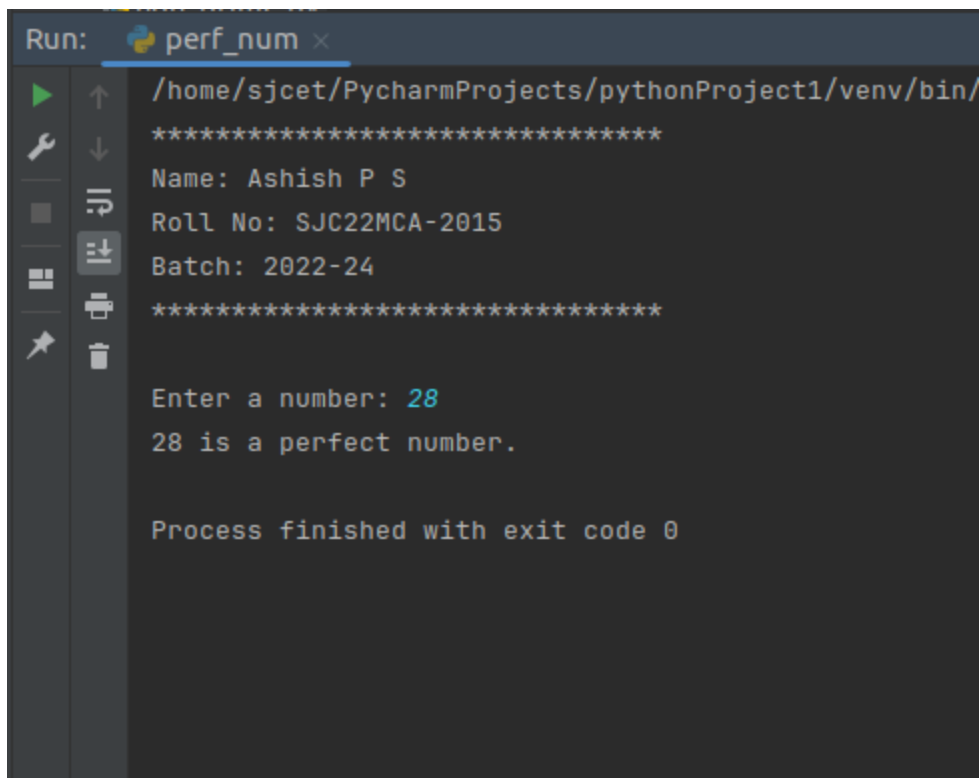
    factors_sum = 1

    for i in range(2, int(num ** 0.5) + 1):
        if num % i == 0:
            factors_sum += i
            if i != num // i:
                factors_sum += num // i

    return factors_sum == num

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

if is_perfect_number(num):
    print(num, "is a perfect number.")
else:
    print(num, "is not a perfect number.")
```



```
Run: perf_num x
/home/sjcet/PycharmProjects/pythonProject1/venv/bin/
*****
Name: Ashish P S
Roll No: SJC22MCA-2015
Batch: 2022-24
*****
Enter a number: 28
28 is a perfect number.

Process finished with exit code 0
```

7. Program to display armstrong numbers upto 1000

```
print("*****")
print("Name: Ashish P S")
print("Roll No: SJC22MCA-2015")
print("Batch: 2022-24")
print("*****")
print()
lower = 1
upper = 1000

for num in range(lower, upper + 1):

    order = len(str(num))

    sum = 0

    temp = num
    while temp > 0:
        digit = temp % 10
        sum += digit ** order
        temp //= 10

    if num == sum:

        print(num)
```

```
Run: armstrog x
/home/sjcet/PycharmProjects/pythonProject1/venv/bin/pyt
*****
Name: Ashish P S
Roll No: SJC22MCA-2015
Batch: 2022-24
*****
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.

```
print("*****")
print("Name: Ashish P S")
print("Roll No: SJC22MCA-2015")
print("Batch: 2022-24")
print("*****")
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))
```

```
Run: daysofweek x
/home/sjcet/PycharmProjects/pythonProject1/venv/bin/python /home/sjcet/PycharmProjects/pythonProject1/daysofweek.py
*****
Name: Ashish P S
Roll No: SJC22MCA-2015
Batch: 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: {'Tuesday', 'Wednesday', 'Friday', 'Monday', 'Thursday', 'Saturday', 'Sunday'}
Type: <class 'set'>

Process finished with exit code 0
```

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

```
print("*****")
print("Name: Ashish P S")
print("Roll No: SJC22MCA-2015")
print("Batch: 2022-24")
print("*****")
print()
```

```
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
```

```
list1_str = input("Enter elements of the first list separated by spaces: ")
list2_str = input("Enter elements of the second list separated by spaces: ")

list1 = [int(x) for x in list1_str.split()]
list2 = [int(x) for x in list2_str.split()]

if len(list1) != len(list2):
    print("Lists are of different lengths!!")
else:
    result_list = add_lists(list1, list2)
    if result_list is not None:
        print("Resultant Sum:")
        print(result_list)
```

```
Run: list_add x
/home/sjcet/PycharmProjects/pythonProject1/venv/bin/python /home/sjc
*****
Name: Ashish P S
Roll No: SJC22MCA-2015
Batch: 2022-24
*****

Enter elements of the first list separated by spaces: 45 65 15
Enter elements of the second list separated by spaces: 25 80 12
Resultant Sum:
[70, 145, 27]

Process finished with exit code 0
|
```

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


```
print("*****")
print("Name: Ashish P S")
print("Roll No: SJC22MCA-2015")
print("Batch: 2022-24")
print("*****")
print()
```

```
def add_matrices(mat1, mat2):
    rows = len(mat1)
    cols = len(mat1[0])

    result = [[0 for _ in range(cols)] for _ in range(rows)]

    for i in range(rows):
        for j in range(cols):
            result[i][j] = mat1[i][j] + mat2[i][j]

    return result
```

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

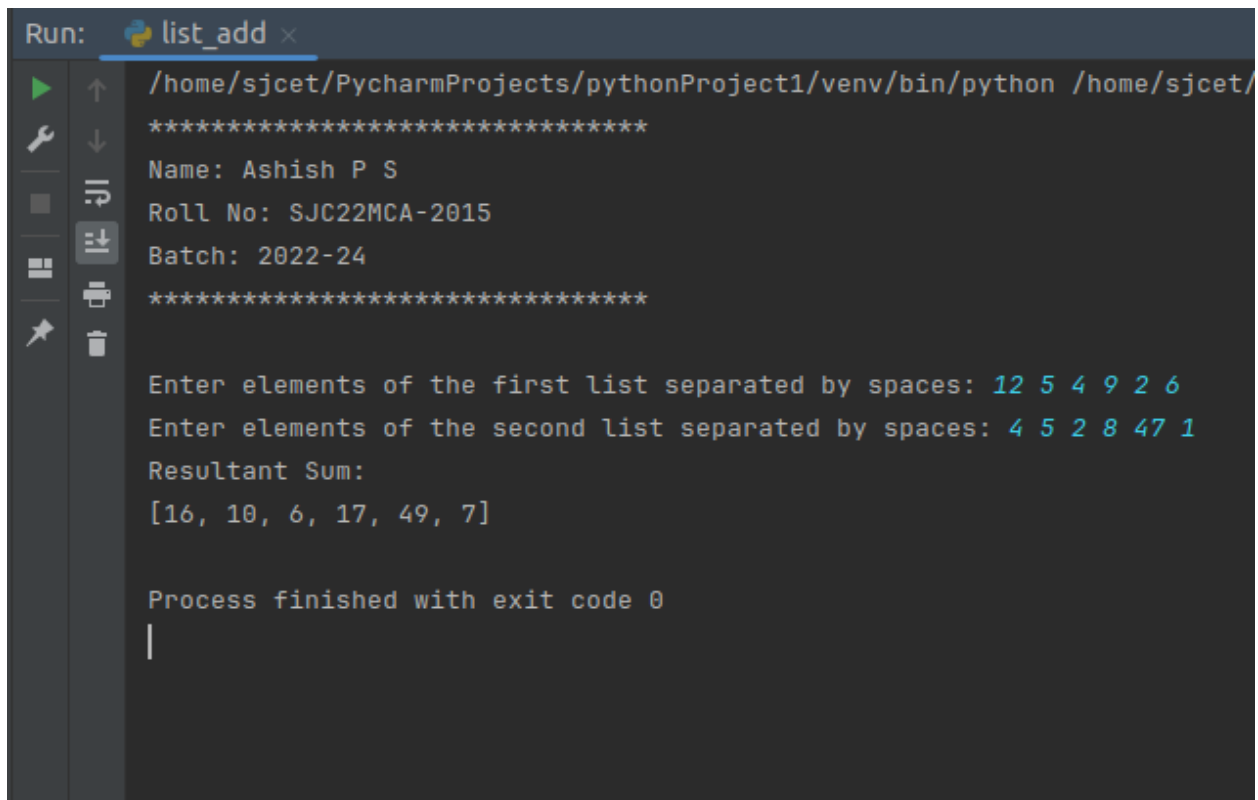
```
matrix1 = []
matrix2 = []
```

```
print("Enter elements of the first matrix:")
for i in range(rows):
    row = [int(x) for x in input().split()]
    matrix1.append(row)
```

```
print("Enter elements of the second matrix:")
for i in range(rows):
    row = [int(x) for x in input().split()]
    matrix2.append(row)
```

```
if len(matrix1) != len(matrix2) or len(matrix1[0]) != len(matrix2[0]):
    print("Matrices have different dimensions. Cannot perform addition.")
else:
    result_matrix = add_matrices(matrix1, matrix2)
    print("Sum of the two matrices:")
```

```
for row in result_matrix:  
    print(" ".join(map(str, row)))
```



```
Run: list_add x  
/home/sjcet/PycharmProjects/pythonProject1/venv/bin/python /home/sjcet/  
*****  
Name: Ashish P S  
Roll No: SJC22MCA-2015  
Batch: 2022-24  
*****  
  
Enter elements of the first list separated by spaces: 12 5 4 9 2 6  
Enter elements of the second list separated by spaces: 4 5 2 8 47 1  
Resultant Sum:  
[16, 10, 6, 17, 49, 7]  
  
Process finished with exit code 0  
|
```

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

```
print("*****")
print("Name: Ashish P S")
print("Roll No: SJC22MCA-2015")
print("Batch: 2022-24")
print("*****")
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=" ")
```

Run:

bub_sort ×



C:\Users\HP\PycharmProjects\tutorialProject\venv\Scripts\

Name: Ashish P S

Roll No: SJC22MCA-2015

Batch: 2022-24

Enter elements separated by spaces: 25 84 54 72 12

Sorted array:

12 25 54 72 84

Process finished with exit code 0

|

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

```
print("*****")
print("Name: Ashish P S")
print("Roll No: SJC22MCA-2015")
print("Batch: 2022-24")
print("*****")
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}")
```

Run:  vowel_count x



C:\Users\HP\PycharmProjects\tutorialProject\venv\Scripts\

Name: Ashish P S

Roll No: SJC22MCA-2015

Batch: 2022-24

Enter a string: *The Bear don't scare me at all*

Vowel counts in the string:

a: 4

e: 4

i: 0

o: 1

u: 0

A: 0

E: 0

I: 0

O: 0

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: $256 > 2+5+6=13$
 $256-13=243$
 $243-9=232$**

```
print("*****")
print("Name: Ashish P S")
print("Roll No: SJC22MCA-2015")
print("Batch: 2022-24")
print("*****")
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)
```

```
Run: serial_operation x
C:\Users\HP\PycharmProjects\tutorialProject\venv\Scripts\python
*****
Name: Ashish P S
Roll No: SJC22MCA-2015
Batch: 2022-24
*****

Enter a positive number: 36
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

```
print("*****")
print("Name: Ashish P S")
print("Roll No: SJC22MCA-2015")
print("Batch: 2022-24")
print("*****")
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.")
```

```
Run: mobnum ×
C:\Users\HP\PycharmProjects\tutorialProject\venv\Scripts\
*****
Name: Ashish P S
Roll No: SJC22MCA-2015
Batch: 2022-24
*****

Enter a 10-digit mobile number: 8547162150
Digits absent in the mobile number: 3, 9

Process finished with exit code 0
|
```