Main Flow Services And Technologies
PYTHON PROGRAMMING INTERNSHIP
Task - 1

1. The sum of two numbers

```
n1 = int(input("Enter a number:")) #getting input from user
n2 = int(input("Enter a number:"))
sum = n1+n2 #sum the two integer using + operator
print(sum)
```

2. Odd or Even

```
num = int(input("Enter a number:")) #integer input from user
if num%2==0:
#mod operator used if remainder is equal to 0 or not condition
        print("Even")
else:
        print("Odd")
```

3. Factorial Calculation

```
import math #math library is imported
nnn = int(input("Enter a number:")) #input from user
print("n!n!n! = ", math.factorial(nnn))
#factorial operation pre defined in math is used
```

4. Fibonacci Sequence

```
def fibonacci_sequence(n):#developer defined function
    if n < 0: #condition for invalid inputs from user
        return "Invalid input."
    if n == 0: #as we know fib have seed values 0 and 1
        return print(0)
    elif n == 1:
        return print(0,1)
    print(0, 1, end=" ")
    n1, n2 = 0, 1
    for i in range(1, n): #concept for n value greater than 1
        n3 = n1 + n2
        print(n3, end=" ")
```

```
        n1 = n2
        n2 = n3
n = int(input("Enter the number: ")) #input to the function
fibonacci_sequence(n) #function call
```

5. Reverse a String

```
str_or = input("Enter the string:") #input string from user
print("Reversed String:", str_or[::-1])
#-1 gets the last value and hence str is reversed
```

6. Palindrome Check

```
def palindrome(string): #function for palindrome
        string = string.replace(" ", "").lower()
 # in ASCII 'A' is different from 'a' so converted into lower case
        return string == string[::-1] # concept from reverse string
strp = input("Enter string:") # input from user
print(palindrome(strp)) #function call
```

7. Leap Year Check

```
def is_leapyr(year): #function definition to find leap year
        return (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0)
year = int(input("Enter a year:")) #input year from user
print(is_leapyr(year))#function call
```

8. Armstrong Number

```
def is_amstrong(num): #function defined for armstrong
        str_num = str(num) #integer is converted to string to get each digits
        no_digit = len(str_num) #length of the sting
        armstrong_sum = sum(int(digit) ** no_digit for digit in str_num)
        #each digit is powered with the length of number i.e. number of digits
        #the digits are defined in the str_num
        return armstrong_sum == num
num = int(input("Enter the number:"))
print(is_amstrong(num))
```

9. Custom Encryption-Decryption System

```
def encrypt(text, shift): #Encryption function
    encrypted_text = ""
    for char in text:
         #chr() converts number from ASCII table to character
      encrypted_text += chr((ord(char) + shift) % 128)
#ord() gives number from the ASCII table
```

```
    return encrypted_text
def decrypt(text, shift): #Decryption function
    decrypted_text = ""
    for char in text:
        decrypted_text += chr((ord(char) - shift) % 128)
    return decrypted_text
print("Custom Encryption-Decryption System")
msg = input("Enter the message: ").strip()
#user input message to be encrypt and removes the unwanted white
spaces
shift = int(input("Enter the shift number: ")) #user input shift number
encrypted_msg = encrypt(msg, shift) #variable stores encrypt function call
print("Encrypted Message:", encrypted_msg)
decrypted_msg = decrypt(encrypted_msg, shift)
#variable stores decrypt function call
print("Decrypted Message:", decrypted_msg)
```