2) AIM: Demonstrating creation of functions, passing parameters and return values

a) Defined a function F as Fn= Fn-1+Fn-2 . Write a python program which accepts a value for N (where N>0) as input and pass this value to the function. Display suitable error message if the condition for input value is not allowed.

```
def recur_fibo(n):
    if n <= 1:
        return n
    else:
        return(recur_fibo(n-1) + recur_fibo(n-2))
nterms = int(input("Enter the number"))
# check if the number of terms is valid
if nterms <= 0:
    print("Please enter a positive integer")
else:
    print("Fibonacci sequence:")
    for i in range(nterms):
        print(recur_fibo(i))
```

OUTPUT:

CASE1:

```
Enter the number5
Fibonacci sequence:
0
1
1
2
3
```

CASE 2:

```
Enter the number-5
Please enter a positive integer
```

b) Develop a python program to convert binary to decimal, octal to hexadecimal using functions.

```
def binary_to_decimal(binary):
   decimal = 0
   power = 0
   while binary != 0:
      decimal += (binary % 10) * (2 ** power)
      binary //= 10
      power += 1
   return decimal

def octal_to_hexadecimal(octal):
   decimal = 0
   power = 0
   while octal != 0:
      decimal += (octal % 10) * (8 ** power)
      octal //= 10
      power += 1

   hexadecimal = ""
   while decimal != 0:
      remainder = decimal % 16
      if remainder < 10:
         hexadecimal = str(remainder) + hexadecimal
      else:
         hexadecimal = chr(ord('A') + remainder - 10) + hexadecimal
      decimal //= 16

   return hexadecimal

binary_number = input("Enter a binary number: ")
decimal_number = binary_to_decimal(int(binary_number))
print("Decimal equivalent:", decimal_number)

octal_number = input("Enter an octal number: ")
hexadecimal_number = octal_to_hexadecimal(int(octal_number))
print("Hexadecimal equivalent:", hexadecimal_number)
```

Output:

```
Enter a binary number: 101

Decimal equivalent: 5

Enter an octal number: 755

Hexadecimal equivalent: 1ED
```