

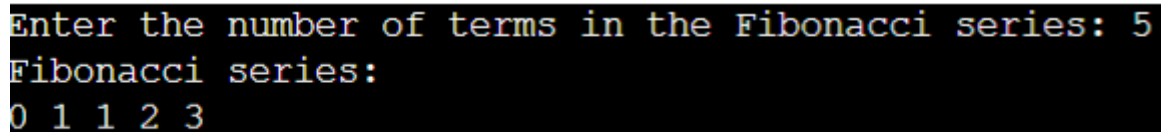
PROGRAM-1

AIM: Write a program to print the Fibonacci series using recursion.

SOURCE CODE:

```
def fibo(n,a=0,b=1):
    print(a, end=" ")
    if n <= 1:
        return
    fibo(n-1, b, a+b)
num_terms = int(input("Enter the number of terms in the Fibonacci series: "))
print("Fibonacci series:")
fibo(num_terms)
```

OUTPUT:

A screenshot of a terminal window with a black background and yellow text. It shows the execution of the program where the user enters '5' for the number of terms, and the output displays the first five terms of the Fibonacci series: 0, 1, 1, 2, 3.

```
Enter the number of terms in the Fibonacci series: 5
Fibonacci series:
0 1 1 2 3
```

PROGRAM-2

AIM: Write the to implement a menu-driven calculator. Use separate functions for the different operations.

SOURCE CODE:

```
def add(a,b):
    print(f"{a}+{b}={a+b}")
    return a+b
def sub(a,b):
    print(f"{a}-{b}={a-b}")
    return a-b
def mul(a,b):
    print(f"{a}*{b}={a*b}")
    return a*b
def div(a,b):
    print(f"{a}/{b}={a/b}")
    if b==0:
        print("!!! Division is not possible by zero !!!\n")
    else:
        return(f"{a}/{b} is {a/b}")
while(True):
    print("\n1.ADDITION\n2.SUBTRACTION\n3.MULTIPLICATION\n4.DIVISION\n5.EXIT")
    c=int(input("Enter your choice:"))
    if c==5:
        break
    n1=int(input("Enter a number1:"))
    n2=int(input("Enter a number2:"))
    if c==1:
        add(n1,n2)
    elif c==2:
        sub(n1,n2)
    elif c==3:
        mul(n1,n2)
    elif c==4:
        div(n1,n2)
    elif c!=1,2,3,4]:
        print("\nEnter a valid choice\n")
```

OUTPUT:

```
1.ADDITION
2.SUBTRACTION
3.MULTIPLICATION
4.DIVISION
5.EXIT
Enter your choice:1
Enter a number1:2
Enter a number2:3
2+3=5
```

```
1.ADDITION
2.SUBTRACTION
3.MULTIPLICATION
4.DIVISION
5.EXIT
Enter your choice:2
Enter a number1:3
Enter a number2:2
3-2=1
```

```
1.ADDITION
2.SUBTRACTION
3.MULTIPLICATION
4.DIVISION
5.EXIT
Enter your choice:3
Enter a number1:4
Enter a number2:5
4*5=20
```

```
1.ADDITION
2.SUBTRACTION
3.MULTIPLICATION
4.DIVISION
5.EXIT
Enter your choice:4
Enter a number1:20
Enter a number2:5
20/5=4.0
```

```
1.ADDITION
2.SUBTRACTION
3.MULTIPLICATION
4.DIVISION
5.EXIT
Enter your choice:5
```

PROGRAM-3

AIM: Write a program to print the nth prime number.

SOURCE CODE:

```
def isprime(num):
    if num <= 1:
        return False
    for i in range(2, int(num ** 0.5) + 1):
        if num % i == 0:
            return False
    return True
def nthprime(n):
    count, current = 0, 2
    while count < n:
        if isprime(current):
            count += 1
        if count < n:
            current += 1
    return current
n = int(input("Enter n: "))
print(f"The {n}th prime number is {nthprime(n)}.")
```

OUTPUT:

```
Enter n: 5
The 5th prime number is 11.
```

PROGRAM-4

AIM: Write lambda functions to find the area of square, rectangle and triangle.

SOURCE CODE:

```
sq=lambda side:side*side
rec=lambda l,b:l*b
tri=lambda ba,he:0.5*ba*he
print("\nAREA OF SQUARE")
side=float(input("Enter the side of the Square: "))
print(f"Area is {sq(side)}")
print("\nAREA OF RECTANGLE")
l=float(input("Enter the Length: "))
b=float(input("Enter the Breadth: "))
print(f"Area is {rec(l,b)}")
print("\nAREA OF TRIANGLE")
ba=float(input("Enter the base of the triangle: "))
he=float(input("Enter the height of th triangle: "))
print(f"Area is {tri(ba,he)}")
```

OUTPUT:

```
AREA OF SQUARE
Enter the side of the Square: 4
Area is 16.0

AREA OF RECTANGLE
Enter the Length: 5
Enter the Breadth: 4
Area is 20.0

AREA OF TRIANGLE
Enter the base of the triangle: 10
Enter the height of th triangle: 5
Area is 25.0
```

PROGRAM-5

AIM: Write a program to display powers of 2 using anonymous function. [Hint use map and lambda function]

SOURCE CODE:

```
num=list(map(int, input("Enter a list of numbers separated by spaces: ").split()))
power=list(map(lambda x:2**x,num))
print(f"Powers of 2 for the numbers {num} are {power}")
```

OUTPUT:

```
Enter a list of numbers separated by spaces: 1 2 3 4
Powers of 2 for the numbers [1, 2, 3, 4] are [2, 4, 8, 16]
```

PROGRAM-6

AIM: Write a program to display multiples of 3 using anonymous function. [Hint use filter and lambda function)

SOURCE CODE:

```
n=list(map(int,input("Enter the numbers seperated by spaces: ").split()))
mul=list(filter(lambda x:x%3==0,n))
print(f"Multiples of 3 = {mul}")
```

OUTPUT:

```
Enter the numbers seperated by spaces: 1 3 6 9
Multiples of 3 = [3, 6, 9]
```


PROGRAM-7

AIM: Write a program to sum the series $1/1! + 4/2! + 27/3! + \dots + \text{nth term}$. [Hint Use a function to find the factorial of a number].

SOURCE CODE:

```
def fact(x):
    f=1
    for i in range(1,x+1):
        f=f*i
    return f
n=int(input("Enter the number of terms: "))
terms=list(map(lambda x:(x**x)/fact(x),range(1,n+1)))
sum=0
for s in terms:
    sum+=s
print(f"sum of terms = {sum}")
```

OUTPUT:

```
Enter the number of terms: 3
sum of terms = 7.5
```

PROGRAM-8

AIM: Write a function called compare which takes two strings S1 and S2 and an integer n as arguments. The function should return True if the first n characters of both the strings are the same else the function should return False..

SOURCE CODE:

```
def compare(s1,s2,n):
    if n<=0:
        print("Number must be positive!!")
        return s1[:n]==s2[:n]
string1=input("Enter the First String: ")
string2=input("Enter the second String: ")
n=int(input("Enter the no:of Characters: "))
print(f"Equivalence = {compare(string1,string2,n)}")
```

OUTPUT:

```
Enter the First String: hello
Enter the second String: hewlo
Enter the no:of Characters: 3
Equivalence = False
```

```
Enter the First String: hello
Enter the second String: helwo
Enter the no:of Characters: 3
Equivalence = True
```

PROGRAM-9

AIM: Write a program to add variable length integer arguments passed to the function. [Also demo the use of docstrings]

SOURCE CODE:

```
def add_num(*args):  
    """  
    Sum of Integers  
    """  
    return sum(args)  
list1=(map(int,input("Enter the numbers seperated by space: ").split()))  
print("sum = ",add_num(*list1))
```

OUTPUT:

```
Enter the numbers seperated by space: 1 2 3  
sum = 6
```

PROGRAM-10

AIM: Write a program using functions to implement these formulae for permutations and combinations.

The Number of permutations of n objects taken r at a time: $p(n, r) = n!/(n - r)!$.

The Number of combinations of n objects taken r at a time is: $c(n, r) = n!/(r! (n - r)!)$

SOURCE CODE:

```
import math
def permutation(n,r):
    return math.factorial(n)//math.factorial(n-r)
def combination(n,r):
    return math.factorial(n)/(math.factorial(n-r)*math.factorial(r))
n=int(input("Enter the value for n: "))
r=int(input("Enter the value for r: "))
print(f"Permutations are {permutation(n,r)}")
print(f"Combinations are {combination(n,r)}")
```

OUTPUT:

```
Enter the value for n: 4
Enter the value for r: 2
Permutations are 12
Combinations are 6
```