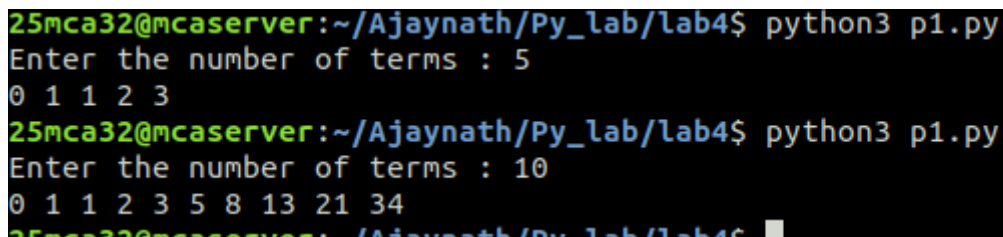


1. Write a program to print the Fibonacci series using recursion.

```
def fibonacci(n):
    if n==1:
        return 0
    elif n==2:
        return 1
    else:
        return fibonacci(n-1)+fibonacci(n-2)
```

```
n=int(input("Enter the number of terms : "))
for i in range(1,n+1):
    print(fibonacci(i),end=" ")
print()
```



```
25mca32@mcaserver:~/Ajaynath/Py_lab/lab4$ python3 p1.py
Enter the number of terms : 5
0 1 1 2 3
25mca32@mcaserver:~/Ajaynath/Py_lab/lab4$ python3 p1.py
Enter the number of terms : 10
0 1 1 2 3 5 8 13 21 34
```

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

```
def add(a,b):
    r=a+b
    print("Sum is ",r)
def sub(a,b):
    r=a-b
    print("difference is ",r)
def div(a,b):
    r=a/b
    print("quotient is ",r)
def mult(a,b):
    r=a*b
    print("product is ",r)
def rem(a,b):
    r=a%b
    print("remainder is ",r)
i=1
while(i):
    print("1.Add")
    print("2.SUB")
    print("3.DIV")
    print("4.MULTIPLICATION")
    print("5.REMAINDER")
    print("6.EXIT")
    n=int(input("Enter the choice : "))
    if n==6:
```

```

        i=0
        continue
a=int(input("Enter the first number : "))
b=int(input("Enter the second number : "))
if n==1:
    add(a,b)
elif n==2:
    sub(a,b)
elif n==3:
    div(a,b)
elif n==4:
    mult(a,b)
elif n==5:
    rem(a,b)
else:
    print("Invalid choice")

```

```

25mca32@mcaserver:~/Ajaynath/Py_lab/Lab4$ python3 p2.py
1.Add
2.SUB
3.DIV
4.MULTIPLICATION
5.REMAINDER
6.EXIT
Enter the choice : 1
Enter the first number : 1
Enter the second number : 2
Sum is 3
1.Add
2.SUB
3.DIV
4.MULTIPLICATION
5.REMAINDER
6.EXIT
Enter the choice : 2
Enter the first number : 5
Enter the second number : 4
difference is 1
1.Add
2.SUB
3.DIV
4.MULTIPLICATION
5.REMAINDER
6.EXIT
Enter the choice : 3
Enter the first number : 15
Enter the second number : 3
quotient is 5.0
1.Add
2.SUB
3.DIV
4.MULTIPLICATION
5.REMAINDER
6.EXIT
Enter the choice : 4
Enter the first number : 5
Enter the second number : 4
product is 20
1.Add
2.SUB
3.DIV
4.MULTIPLICATION
5.REMAINDER
6.EXIT
Enter the choice : 5
Enter the first number : 28
Enter the second number : 5
remainder is 3
1.Add
2.SUB
3.DIV
4.MULTIPLICATION
5.REMAINDER
6.EXIT
Enter the choice : 6
25mca32@mcaserver:~/Ajaynath/Py_lab/Lab4$ 

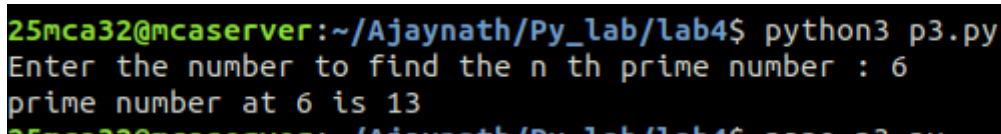
```

3. Write a program to print the n th prime number.

[Use function to check whether a number is prime or not]

```
def isprime(n):
    limit=int(n**(1/2))
    if n==2:
        return 1
    for i in range(2,limit+1):
        if n%i==0:
            return 0
    else:
        return 1

num=int(input("Enter the number to find the n th prime number : "))
i=2
j=0
while(1):
    if isprime(i):
        j+=1
        if j==num:
            print(f"prime number at {num} is {i}")
            break
    i+=1
```



```
25mca32@mcaserver:~/Ajaynath/Py_lab/lab4$ python3 p3.py
Enter the number to find the n th prime number : 6
prime number at 6 is 13
25mca32@mcaserver:~/Ajaynath/Py_lab/lab4$
```

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

```
sarea=lambda s:s*s
rarea=lambda l,b:l*b
tarea=lambda b,h:0.5*b*h

s=float(input("Enter the side of square : "))
rlen=float(input("Enter the rectangle length : "))
rbreadth=float(input("Enter the rectangle width : "))
tbreadth=float(input("Enter the triangle breadth : "))
theight=float(input("Enter the triangle height : "))

print(f"Area of square is {sarea(s)}")
print(f"Area of rectangle is {rarea(rlen,rbreadth)}")
print(f"Area of triangle is {tarea(tbreadth,theight)}")
```

```

25mca32@mcaserver:~/Ajaynath/Py_lab/lab4$ python3 p4.py
Enter the side of square : 5
Enter the rectangle length : 2
Enter the rectangle width : 4
Enter the triangle breadth : 2
Enter the triangle height : 6
Area of square is 25.0
Area of rectangle is 8.0
Area of triangle is 6.0

```

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

```

n=int(input("Enter the limit :"))
squares=list(map(lambda x:2**x,range(n)))
print(squares)

```

```

25mca32@mcaserver:~/Ajaynath/Py_lab/lab4$ python3 p5.py
Enter the limit :5
[1, 2, 4, 8, 16]

```

6. Write a program to display multiples of 3 using anonymous function. (Hint use filter and lambda function)

```

n=int(input("Enter the range : "))
multi_of_three=list(filter(lambda x:x%3==0,range(1,n+1)))
print(multi_of_three)

```

```

25mca32@mcaserver:~/Ajaynath/Py_lab/lab4$ python3 p6.py
Enter the range : 25
[3, 6, 9, 12, 15, 18, 21, 24]

```

7. 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].

```

def fact(n):
    if n==1:
        return 1
    else:
        return n*fact(n-1)

n=int(input("(1/1!)+(4/2!)+(27/3!)+...\nEnter the limit : "))
list=[((i**i)/fact(i)) for i in range(1,n+1)]
print(sum(list))

```

```

25mca32@mcaserver:~/Ajaynath/Py_lab/lab4$ nano p7.py
25mca32@mcaserver:~/Ajaynath/Py_lab/lab4$ python3 p7.py
(1/1!)+(4/2!)+(27/3!)+...
Enter the limit : 5
44.20833333333333

```

8. 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

```

def compare(str1,str2,n):
    if str1[0:n]==str2[0:n]:
        return True
    else:
        return False

```

```

str1=input("Enter the string 1 : ")
str2=input("Enter the string 2 : ")
num=int(input("Enter the number : "))
print(compare(str1,str2,num))

```

```

25mca32@mcaserver:~/Ajaynath/Py_lab/lab4$ python3 p8.py
Enter the string 1 : Ajay
Enter the string 2 : kiran
Enter the number : 3
False
25mca32@mcaserver:~/Ajaynath/Py_lab/lab4$ python3 p8.py
Enter the string 1 : Ajay
Enter the string 2 : Ajaynath
Enter the number : 4
True

```

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

```

def add_num(*num):
    """
    This function accepts any number of integers and return its sum

    parameters:
    *num :variable length integer arguments

    returns:
    int : sum of all integers passed

    """

```

```
return sum(num)
```

```
print(add_num.__doc__)
print(f"add_num(1,2,3) = {add_num(1,2,3)}")
print(f"add_num(2,4) = {add_num(2,4)}")
print(f"add_num(1,3,5,7,9) = {add_num(1,3,5,7,9)}")
print(f"add_num(1,2,3,4,5,6,7,8,9) = {add_num(1,2,3,4,5,6,7,8,9)}")
```

```
25mca32@mcaserver:~/Ajaynath/Py_lab/lab4$ python3 p9.py

This function accepts any number of integers and return its sum

parameters:
*num :variable length integer arguments

returns:
int : sum of all integers passed

add_num(1,2,3) = 6
add_num(2,4) = 6
add_num(1,3,5,7,9) = 25
add_num(1,2,3,4,5,6,7,8,9) = 45
```

10. 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: $c(n, r) = n!/(r! * (n - r)!)$

```
def fact(n):
    if n==1:
        return 1
    else:
        return n*fact(n-1)
def p(n,r):
    return fact(n)/fact(n-r)

def c(n,r):
    return p(n,r)/fact(r)

print("Enter the details to find permutation nPr :")
n=int(input("n: "))
r=int(input("r: "))
print(f"{n}P{r}={p(n,r)}")
print("Enter the details to find Combination nCr :")
n=int(input("n: "))
r=int(input("r: "))
print(f"{n}C{r}={c(n,r)}")
```

```
25mca32@mcaserver:~/Ajaynath/Py_lab/lab4$ python3 p10.py
Enter the details to find permutation nPr :
n: 5
r: 3
5p3=60.0
Enter the details to find Combination nCr :
n: 5
r: 3
5C3=10.0
```