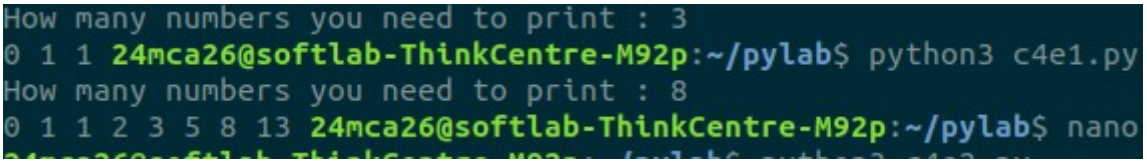
CYCLE 4

EXPERIMENT 1

SOURCE CODE

```
num = int(input("How many numbers you need to print : "))
def fib(n):
    if n <= 1:
        return n
    return fib(n-1) + fib(n-2)
for i in range(num):
    print(fib(i), end = " ")
```

OUTPUT



EXPERIMENT 2

SOURCECODE

```
def add(a, b):
    return a + b

def subtract(a, b):
    return a - b

def multiply(a, b):
    return a * b

def divide(a, b):
    if b == 0:
        return "Cannot divide by zero."
    return a / b

def display_menu():
    print("Select  operation: ")
    print("1. Addition")
    print("2. Subtraction")
    print("3. Multiplication")
    print("4. Division")
    print("5. Exit")

while True:
    display_menu()
    choice = input("Enter choice : ")
```

OUTPUT

```
Select  operation:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter choice : 1
Enter  first number: 20
Enter  second number: 30
ans = 50.0
Select  operation:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter choice : 2
Enter  first number: 30
Enter  second number: 10
ans = 20.0
Select  operation:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter choice : 3
```

```
Enter choice : 3
Enter  first number: 12
Enter  second number: 2
ans = 24.0
Select  operation:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter choice : 4
Enter  first number: 10
Enter  second number: 2
ans = 5.0
Select  operation:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter choice : 5
Bye ...
```

EXPERIMENT 3

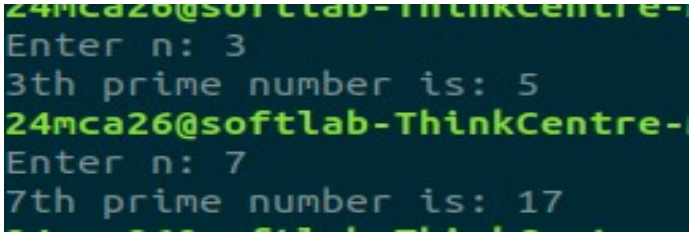
SOURCECODE

```
def is_prime(n):
    if n < 2:
        return False
    for i in range(2, n):
        if n % i == 0:
            return False
    return True

num = 2
count = 0
n = int(input("Enter n: "))

while True:
    if is_prime(num):
        count += 1
        if count == n:
            print(f"{n}th prime number is: {num}")
            break
    num += 1
```

OUTPUT



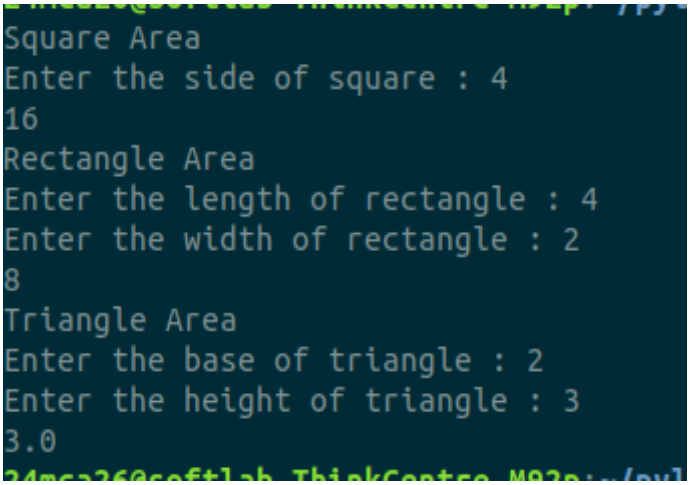
EXPERIMENT 4

SOURCECODE

```
square = lambda s: s * s
rectangle = lambda l, w: l * w
triangle = lambda b, h: 0.5 * b * h

print("Square Area")
side = int(input("Enter the side of square : "))
print(square(side))
print("Rectangle Area")
length = int(input("Enter the length of rectangle : "))
width = int(input("Enter the width of rectangle : "))
print(rectangle(length, width))
print("Triangle Area")
base = int(input("Enter the base of triangle : "))
height = int(input("Enter the height of triangle : "))
print(triangle(base, height))
```

OUTPUT

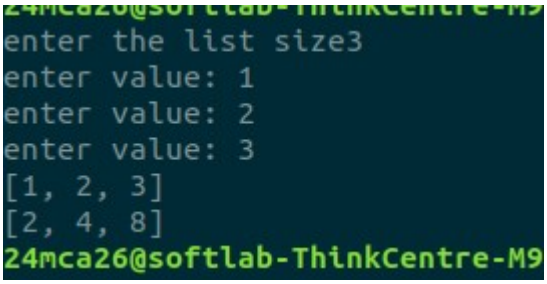


EXPERIMENT 5

SOURCECODE

```
lst=[]
size=int(input("enter the list size"))
for i in range(0,size):
        a=int(input("enter value: "))
        lst.append(a)
print(lst)
val=list(map(lambda x:2**x,lst))
print(val)
```

OUTPUT



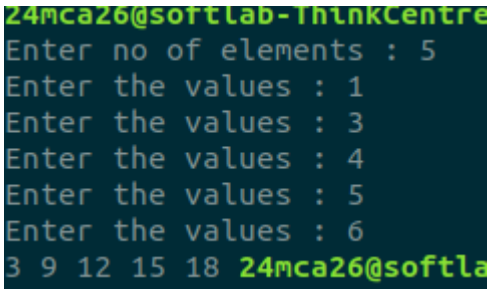
EXPERIMENT 6

SOURCECODE

```
lst = []
size = int(input("Enter no of elements : "))
for i in range(0, size):
    a = int(input("Enter the values : "))
    lst.append(a)
```

```
ans = lambda x : 3 * x
for i in range(0, size):
        print(ans(lst[i]), end = " ")
```

OUTPUT



EXPERIMENT 7

SOURCECODE

```
def fact(n):
    if n == 0 or n == 1:
        return 1
    else:
        return n * fact(n - 1)

def sum(n):
    final_sum = 0
    equation_parts = []

    for i in range(1, n + 1):
        numerator = i ** i
        denominator = fact(i)
        single_term = numerator / denominator
        final_sum = single_term + final_sum
        equation_parts.append(f"{numerator}/{i}!")
    equation_str = " + ".join(equation_parts)
    print(f"{equation_str} = {final_sum}")

n = int(input("Enter the number of terms : "))
sum(n)
```

OUTPUT

```
24mca26@softlab-ThinkCentre-M92p:~/pylab$ python3 c4e7.py
Enter the number of terms : 3
1/1! + 4/2! + 27/3! = 7.5
24mca26@softlab-ThinkCentre-M92p:~/pylab$ python3 c4e7.py
Enter the number of terms : 7
1/1! + 4/2! + 27/3! + 256/4! + 3125/5! + 46656/6! + 823543/7! = 272.40972222222223
24mca26@softlab-ThinkCentre-M92p:~/pylab$
```

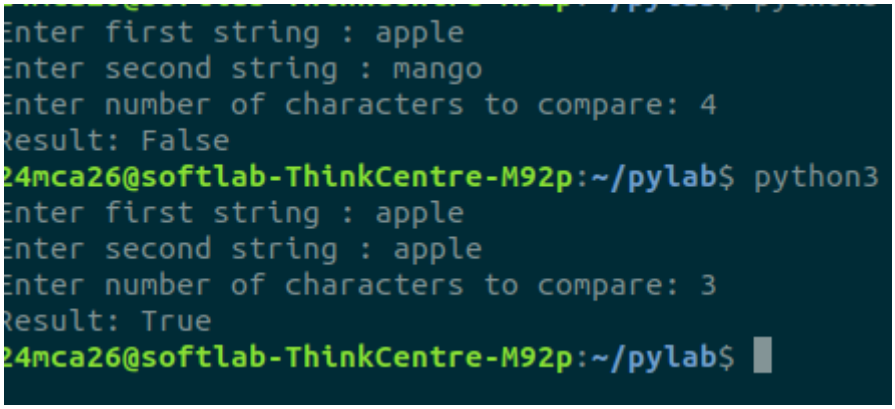
EXPERIMENT 8

SOURCECODE

```
def compare(S1, S2, n):
    if len(S1) < n or len(S2) < n:
        return False
    return S1[:n] == S2[:n]

S1 = input("Enter first string : ")
S2 = input("Enter second string : ")
n = int(input("Enter number of characters to compare: "))
print(f"Result: {compare(S1, S2, n)}")
```

OUTPUT



EXPERIMENT 9

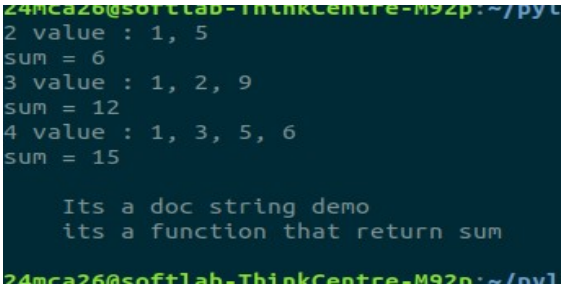
SOURCECODE

```
def sum_number(*args):
    """
    Its a doc string demo
    its a function that return sum
    """
    sum = 0;
    for num in args:
        sum = sum + num
    return sum
print("2 value : 1, 5")
print(f'sum = {sum_number(1, 5)}')
print("3 value : 1, 2, 9")
```

```
print(f'sum = {sum_number(1, 2, 9)}')
print("4 value : 1, 3, 5, 6")
print(f'sum = {sum_number(1, 3, 5, 6)}')

print(sum_number.__doc__)
```

OUTPUT



EXPERIMENT 10

SOURCECODE

```
def factorial(n):
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial(n - 1)

def permutation(n, r):
    if n < r:
        return "Error: n >= r "
    return factorial(n) // factorial(n - r)

def combination(n, r):
    if n < r:
        return "Error n >= r "
    return factorial(n) // (factorial(r) * factorial(n - r))

n = int(input("Enter the value n : "))
r = int(input("Enter the value r : "))


print(f'Permutation : {permutation(n, r)}')
print(f'Combination : {combination(n, r)}')
```

OUTPUT