PRACTICAL FILE

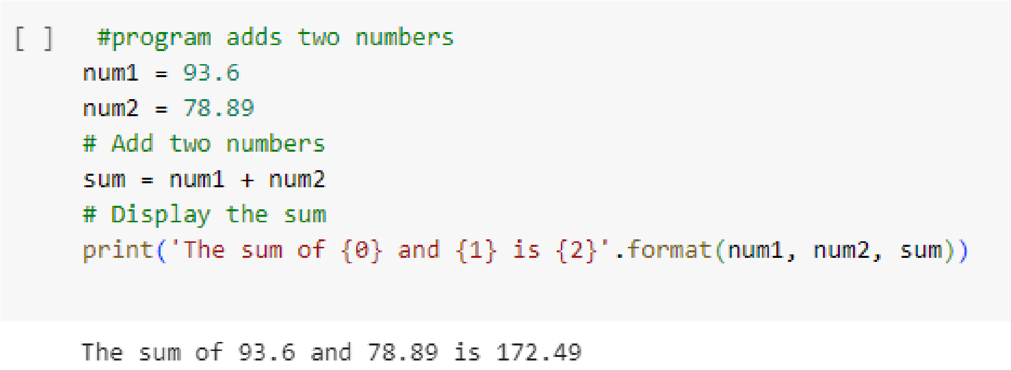
PROGRAMING IN PYTHON

SUBMITTED TO:	**SUBMITTED BY:**
DR. KAVITA MITTAL	**SURAJ DEV KANT**
(ASSOCIATE PROFESSOR)	**ENROLMENT NO:** **030122013**
	(BCA)2st SEMESTER

PROGRAMMING FOR PROBLEM SOLVING THROUGH PYTHON(BCA)

INDEX

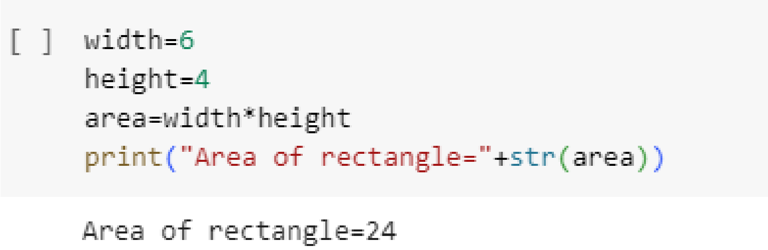
1. Write a Program to find sum of two numbers.



```
#program adds two numbers
num1 = 93.6
num2 = 78.89
# Add two numbers
sum = num1 + num2
# Display the sum
print('The sum of {0} and {1} is {2}'.format(num1, num2, sum))
```

```
The sum of 93.6 and 78.89 is 172.49
```

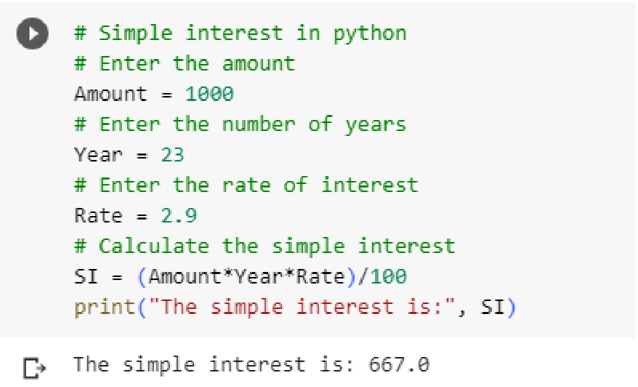
2. Write a Program to calculate area of rectangle.



```
width=6
height=4
area=width*height
print("Area of rectangle="+str(area))
```

```
Area of rectangle=24
```

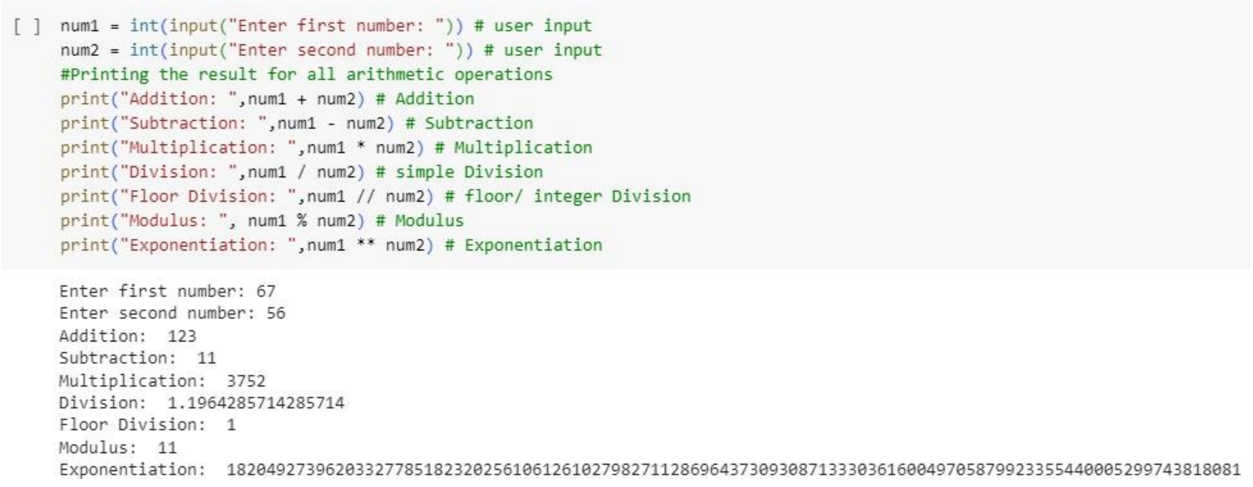
3. Write a Program to find the simple interest.



```
# Simple interest in python
# Enter the amount
Amount = 1000
# Enter the number of years
Year = 23
# Enter the rate of interest
Rate = 2.9
# Calculate the simple interest
SI = (Amount*Year*Rate)/100
print("The simple interest is:", SI)
```

```
The simple interest is: 667.0
```

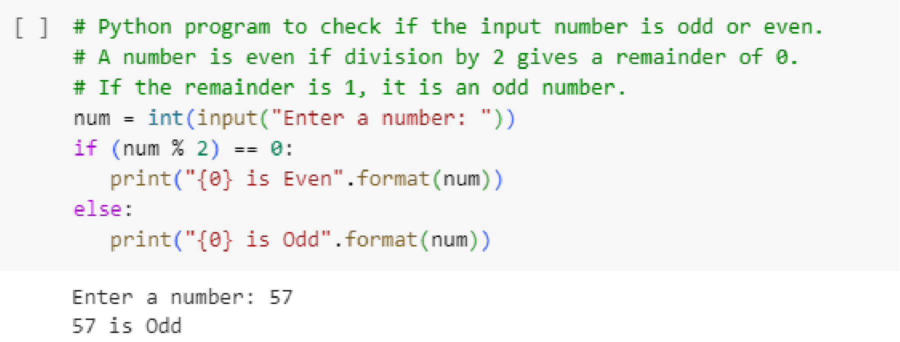
4. Write a Program to enter two numbers and print the arithmetic operations. like +, -,*, /, // and %.



```
[ ] num1 = int(input("Enter first number: ")) # user input
    num2 = int(input("Enter second number: ")) # user input
    #Printing the result for all arithmetic operations
    print("Addition: ",num1 + num2) # Addition
    print("Subtraction: ",num1 - num2) # Subtraction
    print("Multiplication: ",num1 * num2) # Multiplication
    print("Division: ",num1 / num2) # simple Division
    print("Floor Division: ",num1 // num2) # floor/ integer Division
    print("Modulus: ", num1 % num2) # Modulus
    print("Exponentiation: ",num1 ** num2) # Exponentiation
```

```
Enter first number: 67
Enter second number: 56
Addition:  123
Subtraction:  11
Multiplication:  3752
Division:  1.1964285714285714
Floor Division:  1
Modulus:  11
Exponentiation:  1820492739620332778518232025610612610279827112869643730930871333036160049705879923355440005299743818081
```

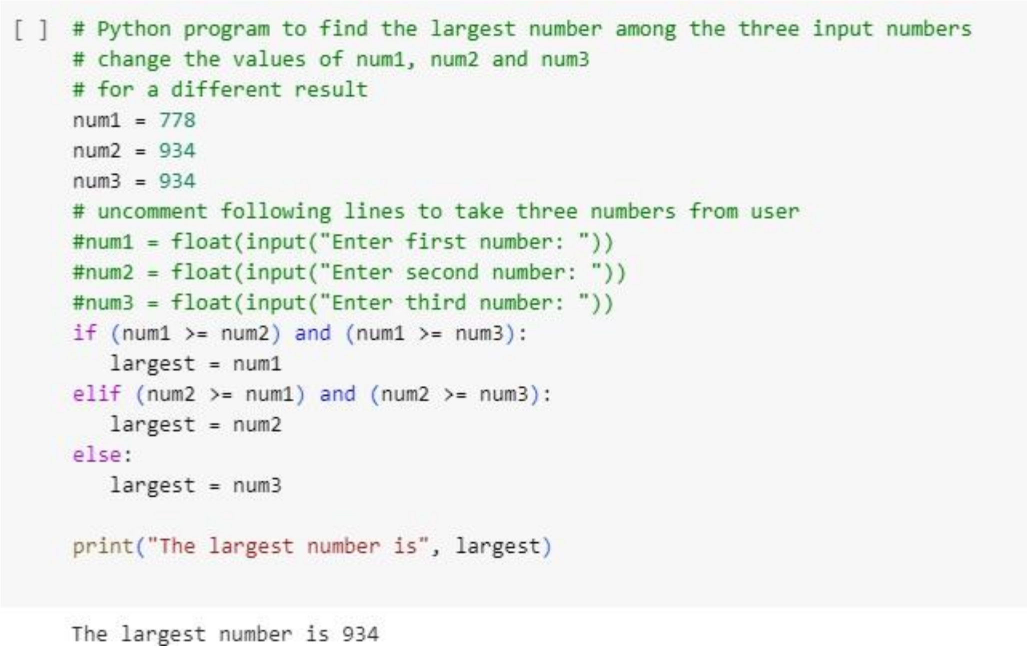
5. Write a program to find whether an inputted number is even or odd.



```
[ ] # Python program to check if the input number is odd or even.
    # A number is even if division by 2 gives a remainder of 0.
    # If the remainder is 1, it is an odd number.
    num = int(input("Enter a number: "))
    if (num % 2) == 0:
       print("{0} is Even".format(num))
    else:
       print("{0} is Odd".format(num))
```

```
Enter a number: 57
57 is Odd
```

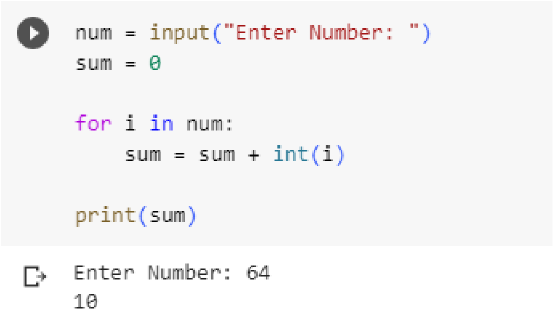
6. Write a program to find the largest of three numbers.



```
[ ] # Python program to find the largest number among the three input numbers
    # change the values of num1, num2 and num3
    # for a different result
    num1 = 778
    num2 = 934
    num3 = 934
    # uncomment following lines to take three numbers from user
    #num1 = float(input("Enter first number: "))
    #num2 = float(input("Enter second number: "))
    #num3 = float(input("Enter third number: "))
    if (num1 >= num2) and (num1 >= num3):
       largest = num1
    elif (num2 >= num1) and (num2 >= num3):
       largest = num2
    else:
       largest = num3

    print("The largest number is", largest)
```

```
The largest number is 934
```

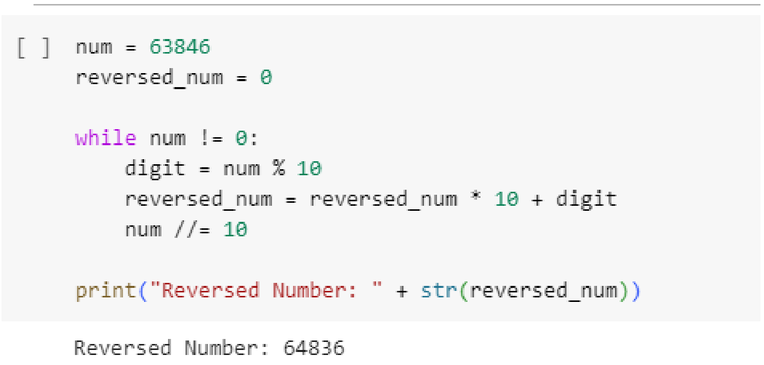
7. Write a Program to find sum of digits of a number.



```
num = input("Enter Number: ")
sum = 0

for i in num:
    sum = sum + int(i)

print(sum)
```

```
Enter Number: 64
10
```

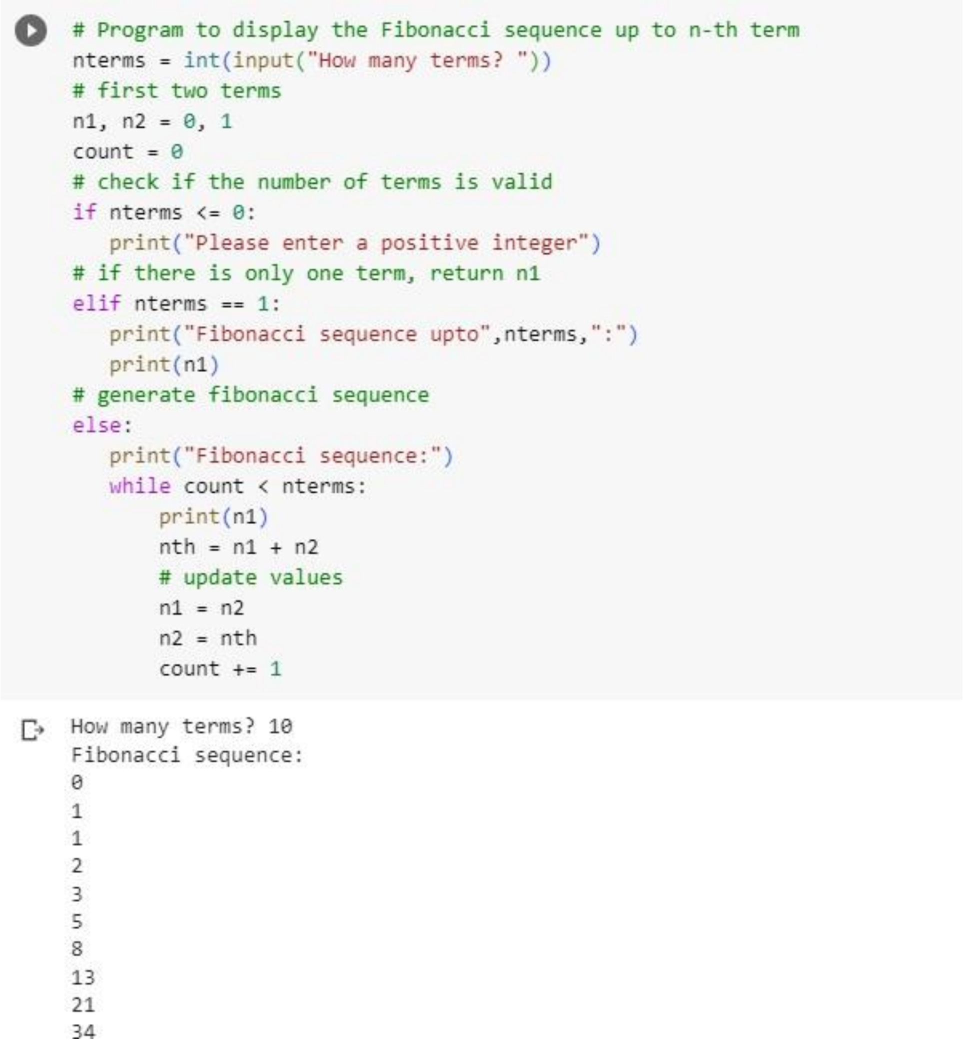
8. Write a Program to find reverse of the entered number.



```
[ ] num = 63846
    reversed_num = 0

    while num != 0:
        digit = num % 10
        reversed_num = reversed_num * 10 + digit
        num //= 10

    print("Reversed Number: " + str(reversed_num))
```

```
Reversed Number: 64836
```

9. Write a Program to enter the number of terms and to print the Fibonacci Series.



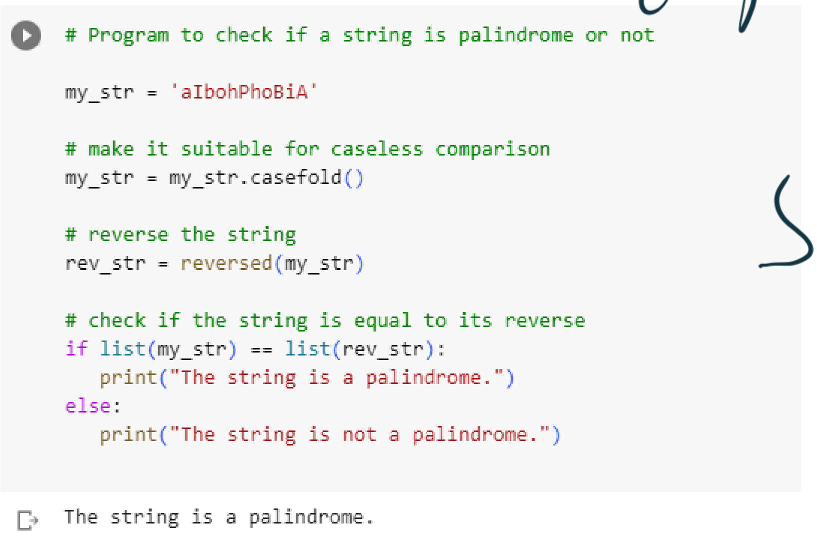
```
# Program to display the Fibonacci sequence up to n-th term
nterms = int(input("How many terms? "))
# first two terms
n1, n2 = 0, 1
count = 0
# check if the number of terms is valid
if nterms <= 0:
   print("Please enter a positive integer")
# if there is only one term, return n1
elif nterms == 1:
   print("Fibonacci sequence upto",nterms,":")
   print(n1)
# generate fibonacci sequence
else:
   print("Fibonacci sequence:")
   while count < nterms:
       print(n1)
       nth = n1 + n2
       # update values
       n1 = n2
       n2 = nth
       count += 1
```

```
How many terms? 10
Fibonacci sequence:
0
1
1
2
3
5
8
13
21
34
```

10. Write a Program to enter the string and to check if it's palindrome or not using loop.

def fib(n)

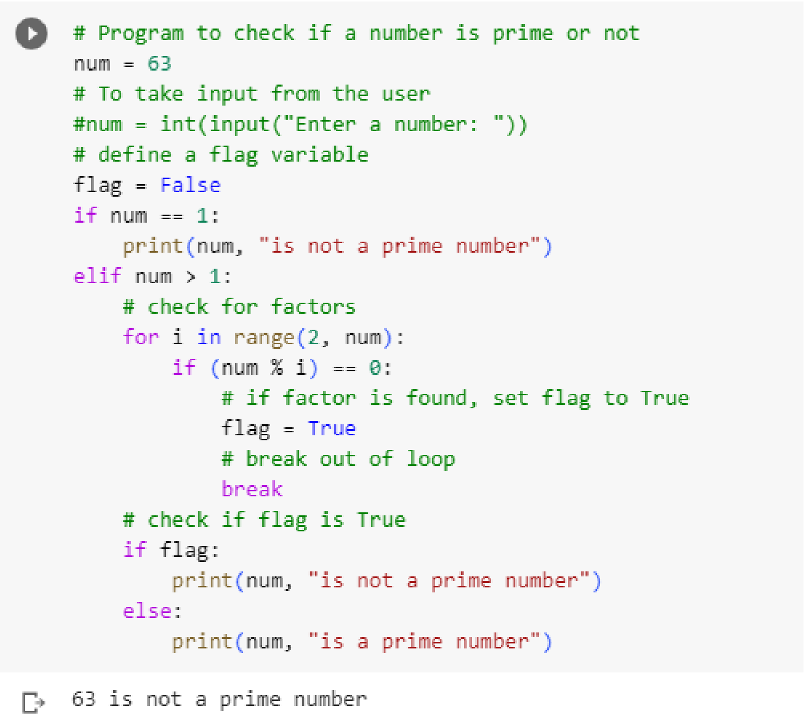
str n = str(



```
# Program to check if a string is palindrome or not

my_str = 'aIbohPhoBiA'

# make it suitable for caseless comparison
my_str = my_str.casefold()

# reverse the string
rev_str = reversed(my_str)

# check if the string is equal to its reverse
if list(my_str) == list(rev_str):
   print("The string is a palindrome.")
else:
   print("The string is not a palindrome.")
```

```
The string is a palindrome.
```

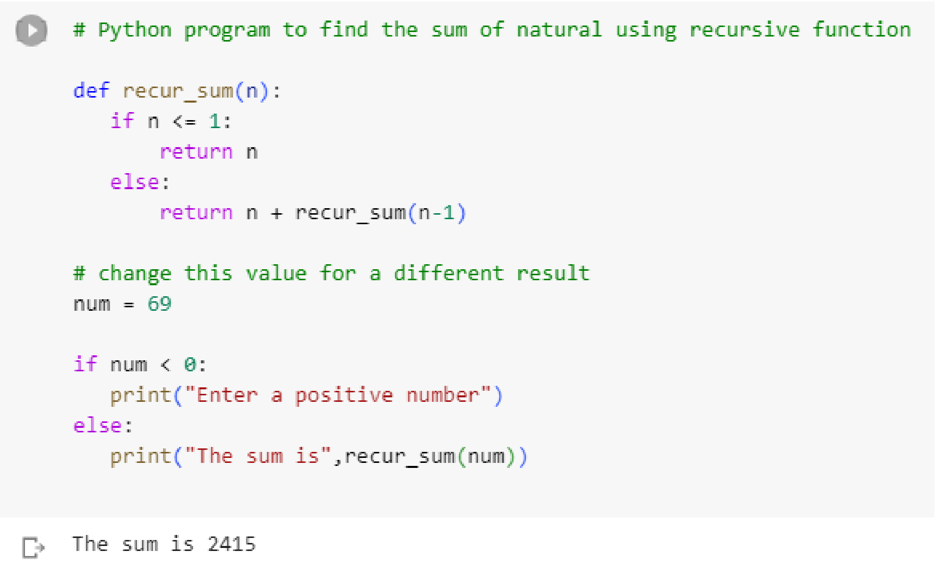
11. Write a Program to check if a number is prime or not.



```
# Program to check if a number is prime or not
num = 63
# To take input from the user
#num = int(input("Enter a number: "))
# define a flag variable
flag = False
if num == 1:
    print(num, "is not a prime number")
elif num > 1:
    # check for factors
    for i in range(2, num):
        if (num % i) == 0:
            # if factor is found, set flag to True
            flag = True
            # break out of loop
            break
    # check if flag is True
    if flag:
        print(num, "is not a prime number")
    else:
        print(num, "is a prime number")
```

```
63 is not a prime number
```

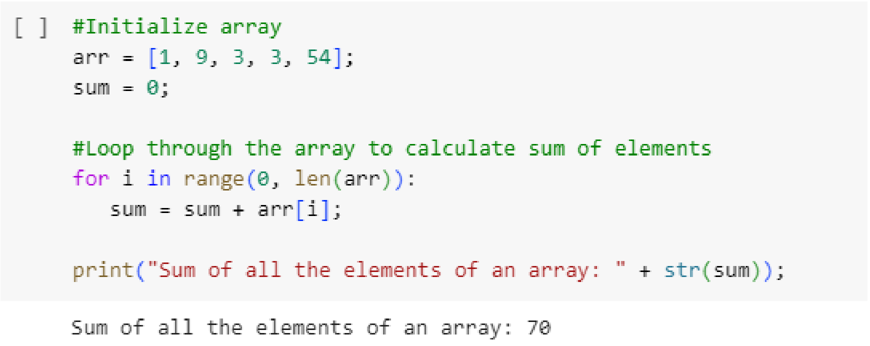
12. Write a program to find sum of n numbers using recursion.



```
# Python program to find the sum of natural using recursive function

def recur_sum(n):
   if n <= 1:
       return n
   else:
       return n + recur_sum(n-1)

# change this value for a different result
num = 69

if num < 0:
   print("Enter a positive number")
else:
   print("The sum is",recur_sum(num))
```

```
The sum is 2415
```

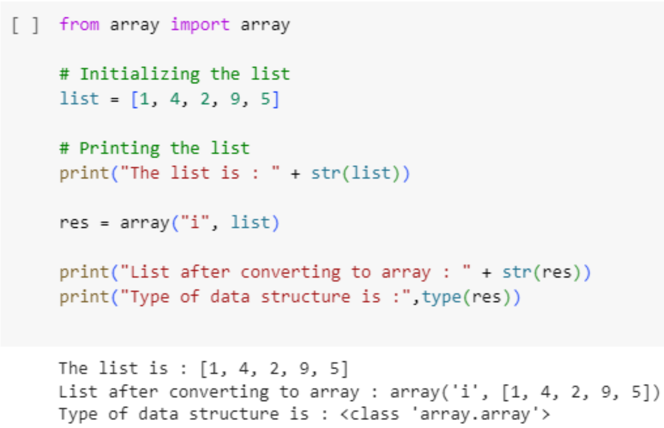
13. Write a program to find sum of array elements.



```
#Initialize array
arr = [1, 9, 3, 3, 54];
sum = 0;

#Loop through the array to calculate sum of elements
for i in range(0, len(arr)):
   sum = sum + arr[i];

print("Sum of all the elements of an array: " + str(sum));
```

```
Sum of all the elements of an array: 70
```

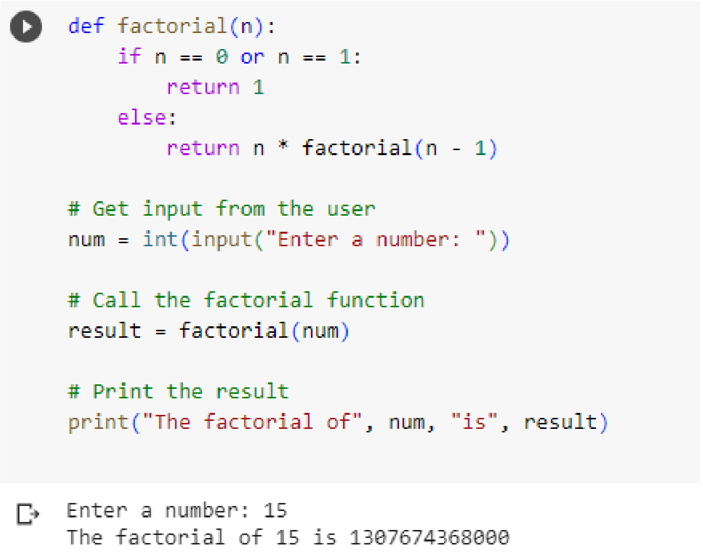
14. Write a program to convert list into array.



```
[ ] from array import array

    # Initializing the list
    list = [1, 4, 2, 9, 5]

    # Printing the list
    print("The list is : " + str(list))

    res = array("i", list)

    print("List after converting to array : " + str(res))
    print("Type of data structure is :",type(res))
```

```
    The list is : [1, 4, 2, 9, 5]
    List after converting to array : array('i', [1, 4, 2, 9, 5])
    Type of data structure is : <class 'array.array'>
```

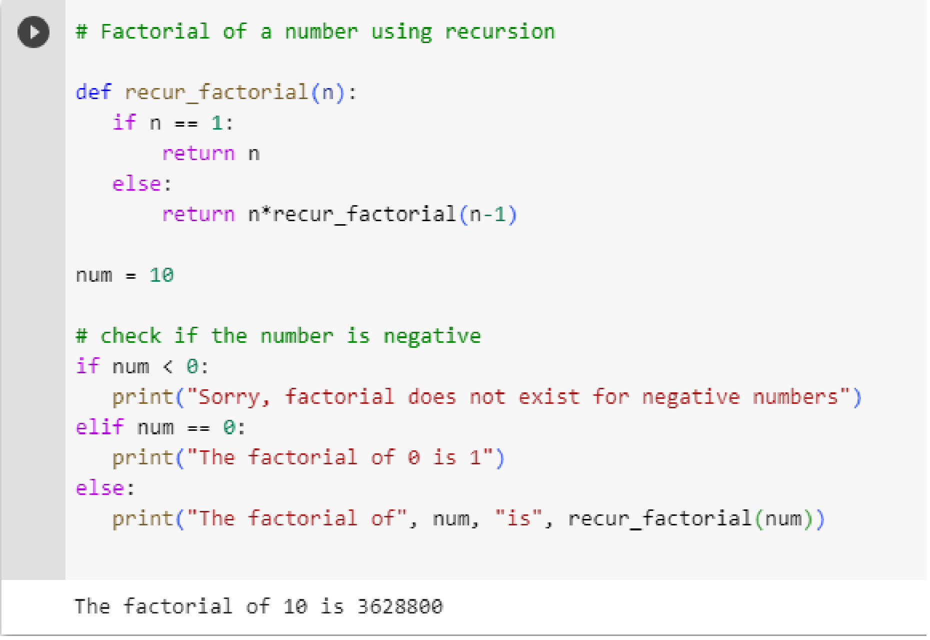
15. Write a program find the factorial of the number using function.



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

# Get input from the user
num = int(input("Enter a number: "))

# Call the factorial function
result = factorial(num)

# Print the result
print("The factorial of", num, "is", result)
```

```
Enter a number: 15
The factorial of 15 is 1307674368000
```

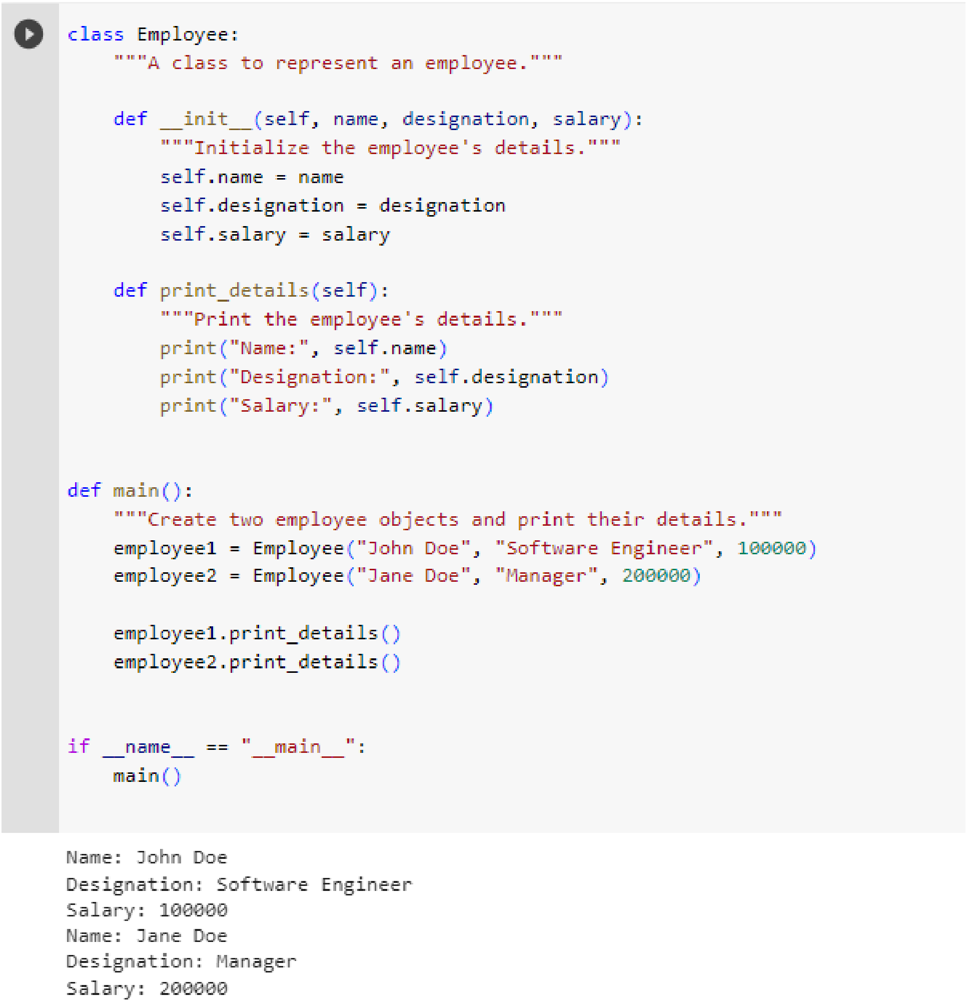
16. Write a program to find the factorial of the number using recursion.



```
# Factorial of a number using recursion

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

num = 10

# check if the number is negative
if num < 0:
   print("Sorry, factorial does not exist for negative numbers")
elif num == 0:
   print("The factorial of 0 is 1")
else:
   print("The factorial of", num, "is", recur_factorial(num))
```

```
The factorial of 10 is 3628800
```

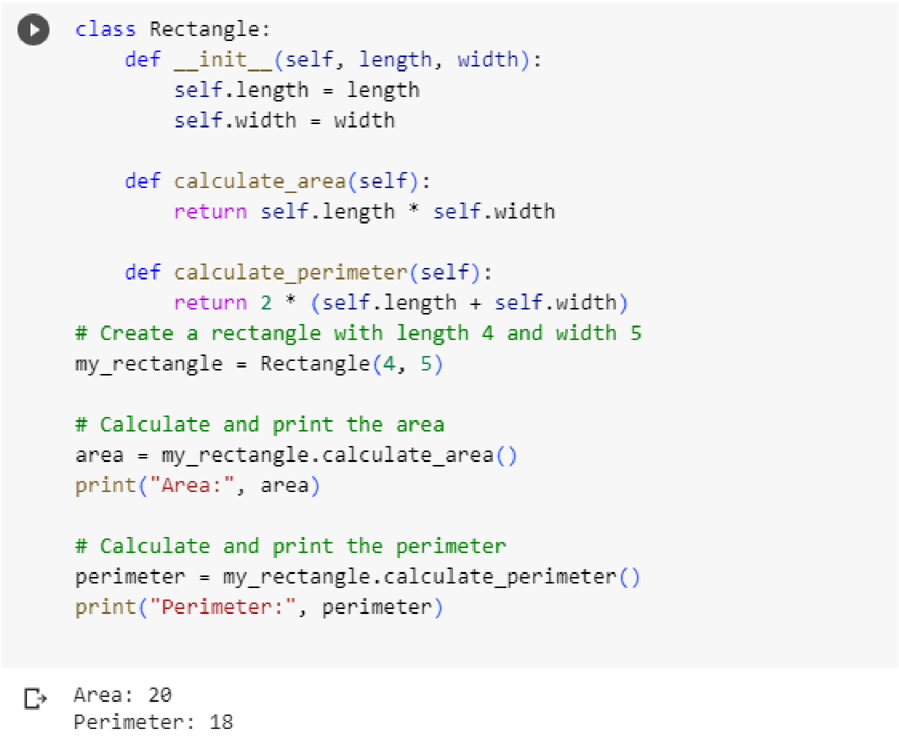
17.Write a Program to store Employee Details using class.



```
class Employee:
    """A class to represent an employee."""

    def __init__(self, name, designation, salary):
        """Initialize the employee's details."""
        self.name = name
        self.designation = designation
        self.salary = salary

    def print_details(self):
        """Print the employee's details."""
        print("Name:", self.name)
        print("Designation:", self.designation)
        print("Salary:", self.salary)


def main():
    """Create two employee objects and print their details."""
    employee1 = Employee("John Doe", "Software Engineer", 100000)
    employee2 = Employee("Jane Doe", "Manager", 200000)

    employee1.print_details()
    employee2.print_details()


if __name__ == "__main__":
    main()
```

```
Name: John Doe
Designation: Software Engineer
Salary: 100000
Name: Jane Doe
Designation: Manager
Salary: 200000
```

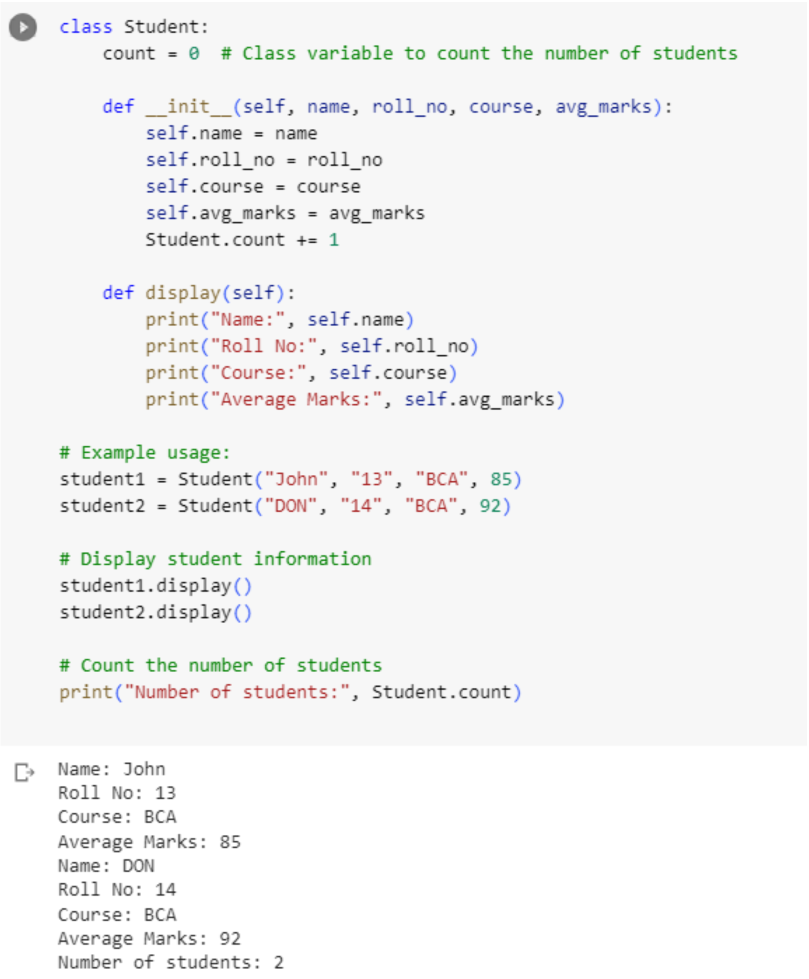
18. Write a program to define a class rectangle and calculate area and perimeter of rectangle.



```
class Rectangle:
    def __init__(self, length, width):
        self.length = length
        self.width = width

    def calculate_area(self):
        return self.length * self.width

    def calculate_perimeter(self):
        return 2 * (self.length + self.width)
# Create a rectangle with length 4 and width 5
my_rectangle = Rectangle(4, 5)

# Calculate and print the area
area = my_rectangle.calculate_area()
print("Area:", area)

# Calculate and print the perimeter
perimeter = my_rectangle.calculate_perimeter()
print("Perimeter:", perimeter)
```

```
Area: 20
Perimeter: 18
```

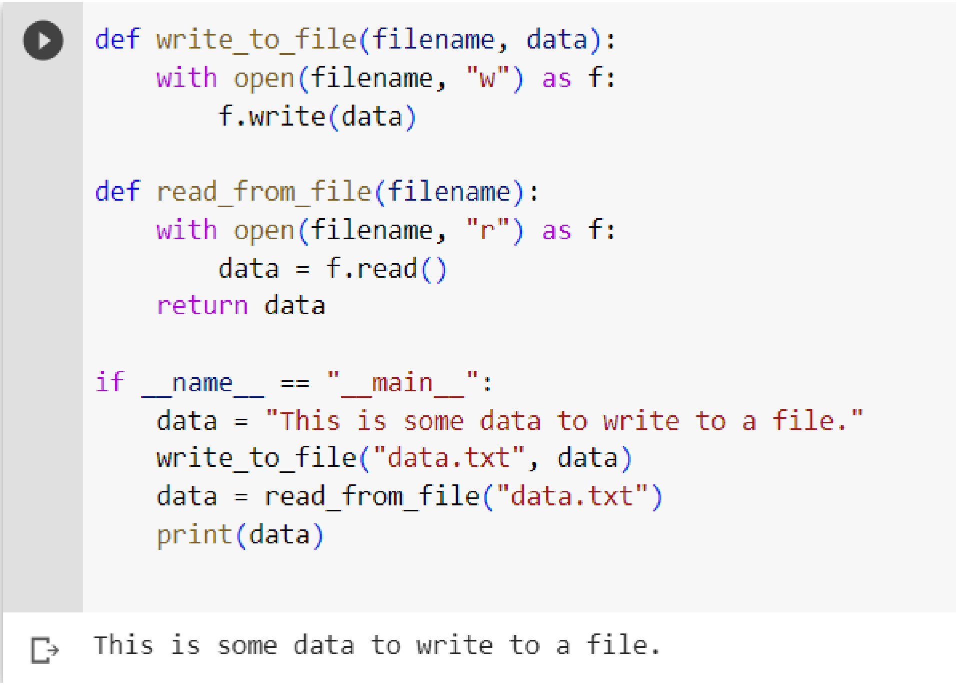
19. Write a program to create a class student and display the of student that include name roll no, course, average marks also count the number of students.



```
class Student:
    count = 0  # Class variable to count the number of students

    def __init__(self, name, roll_no, course, avg_marks):
        self.name = name
        self.roll_no = roll_no
        self.course = course
        self.avg_marks = avg_marks
        Student.count += 1

    def display(self):
        print("Name:", self.name)
        print("Roll No:", self.roll_no)
        print("Course:", self.course)
        print("Average Marks:", self.avg_marks)

# Example usage:
student1 = Student("John", "13", "BCA", 85)
student2 = Student("DON", "14", "BCA", 92)

# Display student information
student1.display()
student2.display()

# Count the number of students
print("Number of students:", Student.count)
```

```
Name: John
Roll No: 13
Course: BCA
Average Marks: 85
Name: DON
Roll No: 14
Course: BCA
Average Marks: 92
Number of students: 2
```

20.Write a Program to perform reading and writing operation in a text file.



```
def write_to_file(filename, data):
    with open(filename, "w") as f:
        f.write(data)

def read_from_file(filename):
    with open(filename, "r") as f:
        data = f.read()
    return data

if __name__ == "__main__":
    data = "This is some data to write to a file."
    write_to_file("data.txt", data)
    data = read_from_file("data.txt")
    print(data)
```

```
This is some data to write to a file.
```