Qn 1.

Program:

```
def display():
    print("Pulusu Chakradhar")

def main():
    display()

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

Output:

```
Pulusu Chakradhar
```

Qn 2.

Program:

```
def display():
    print("1\ New Delhi is the capital of India\n2\ Tamilnadu is the land of yogic culture")

def main():
    display()

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

Output:

```
1\ New Delhi is the capital of India
2\ Tamilnadu is the land of yogic culture
```

Qn 3.

Program:

```
def pyramid(n):
    print()
    for i in range(0, n):
        for j in range(0, i+1):
            print("* ",end="")
        print("\r")

def main():
     n=int(input("Enter pyramid size:\n"))
     pyramid(n)

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

Output:

```
Enter pyramid size:
5

*
* *
* * *
* * * *
* * * * *
```

Qn 4.
Program:

```
def simple_interest(p, n, r):
    interest = (p * n * r)//100
    return interest

def main():
     p = int(input("Enter principal amount:\n"))
     n = 5
     r = 8
     print("Simple interest for given principal amount is ",simple_interest(p, n, r))

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

Output:

```
Enter principal amount:
89000
Simple interest for given principal amount is  35600
```

Qn 5.

Program:

```
def calculator(a, b):
    print("Addition:",a+b)
    print("Subtraction:",a-b)
    print("Multiplication:",a*b)
    print("Division:",a/b)
    print("Modulus:",a%b)

def main():
     a = int(input("Enter first number:\n"))
     b = int(input("Enter first number:\n"))
     calculator(a, b)


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

Output:

```
Enter first number:
78

Enter first number:
36
Addition: 114
Subtraction: 42
Multiplication: 2808
Division: 2.1666666666666665
Modulus: 6
```

Qn 6.

Program:

```
def grade(mark):
    if mark >= 90 and mark <= 100:
        return "S"
    if mark >= 80 and mark < 90:
        return "A"
    if mark >= 70 and mark < 80:
        return "B"
    if mark >= 60 and mark < 70:
        return "C"
    if mark >= 50 and mark < 60:
        return "D"
    if mark < 50:
        return "F"

def main():
     mark = int(input("Enter the mark:\n"))
     print("The Grade is ",grade(mark))


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

Output:

```
Enter the mark:
67
The Grade is  C
```

Qn 7.

Program:

```
def num_to_wrd(n):
    dict = {'1': 'ONE',
            '2': 'TWO',
            '3': 'THREE',
            '4': 'FOUR',
            '5': 'FIVE',
            '6': 'SIX',
            '7': 'SEVEN',
            '8': 'EIGHT',
            '9': 'NINE',
            '0': 'ZERO'
           }
    for i in n:
        print(dict.get(i), end=(" "))

def main():
     n = input("Enter a number:\n")
     num_to_wrd(n)


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

Output:

```
Enter a number:
5678
FIVE SIX SEVEN EIGHT
```

Qn 8.

Program:

```
def smallest(a,b,c):
    if a < b and a < c:
        return a
    elif b < a and b < c:
        return b
    elif c < a and c < b:
        return c
    else:
        return "Enter three unique numbers"
def largest(a,b,c):
    if a > b and a > c:
        return a
    elif b > a and b > c:
        return b
    elif c > a and c > b:
        return c
    else:
        return "Enter three unique numbers"

def main():
    print("Enter three numbers")
    a = input()
    b = input()
    c = input()
    print("The smallest of three numbers is ", smallest(a, b, c))
    print("The largest of three numbers is ", largest(a, b, c))


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

Output:

```
Enter three numbers

34

56

0
The smallest of three numbers is  0
The largest of three numbers is  56
```

Qn 9.

Program:

```
def swap(a,b):
    temp = a
    a = b
    b = temp
    return a,b

def main():
    print("Enter two numbers")
    a = input()
    b = input()
    print("Before swap a={} b={}".format(a,b))
    a,b = swap(a,b)
    print("After swap a={} b={}".format(a, b))


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

Output:

```
Enter two numbers

78

90
Before swap a=78 b=90
After swap a=90 b=78
```

Qn 10.
Program:

```
def second_large(List):
    largest = max(List)
    smallest = min(List)
    trace = 0
    for i in List:
        if i > smallest and i < largest:
            if i > trace:
                trace = i
    return trace

def main():
    List = []
    n = int(input("Enter the list size\n"))
    print("Enter the numbers")
    for i in range(n):
        List.append(int(input()))
    print("The second largest element is ",second_large(List))


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

Output:

```
Enter the list size
5
Enter the numbers

1

2

3

4

5
The second largest element is  4
```

Qn 11.

Program:

```
def swap(a,b):
    a = a + b
    b = a - b
    a = a - b
    return a,b

def main():
    print("Enter two numbers")
    a = int(input())
    b = int(input())
    print("Before swap a={} b={}".format(a,b))
    a,b = swap(a,b)
    print("After swap a={} b={}".format(a, b))


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

Output:

```
Enter two numbers

78

90
Before swap a=78 b=90
After swap a=90 b=78
```

Qn 12.

Program:

```
def DecimalToBinary(num):
    if num > 1:
        DecimalToBinary(num // 2)
    print(num % 2, end = '')

def main():
    n = int(input("Enter a decimal number\n"))
    print("The binary representation of given number is")
    DecimalToBinary(n)


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

Output:

```
Enter a decimal number
54
The binary representation of given number is
110110
```

Qn 13.

Program:

```
def odd_even(n):
     if n % 2 == 0:
         return True
     else:
         return False

def main():
    n = int(input("Enter a number\n"))
    if odd_even(n):
        print("The number is Even")
    else:
        print("The number is Odd")

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

Output:

```
Enter a number
6
The number is Even

Enter a number
9
The number is Odd
```

Qn 14.

Program:

```
def pyramid(n):
    print()
    for i in range(0, n):
        for j in range(1, i+2):
            print(j,end="")
        print("\r")

def main():
     n=int(input("Enter pyramid size:\n"))
     pyramid(n)

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

Output:

```
Enter pyramid size:
5

1
12
123
1234
12345
```

Qn 15.

Program:

```
def sum_digits(n):
    total = 0
    for i in n:
        total = total + int(i)
    return total

def main():
     n=input("Enter a number:\n")
     print("The sum of digits in given number ",sum_digits(n))

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

Output:

```
Enter a number:
666
The sum of digits in given number  18
```

Qn 16.

Program:

```
def sum_digits(n):
    total = 0
    for i in str(n):
        total = total + int(i)
    return total

def main():
     n=input("Enter a number:\n")
     total = sum_digits(n)
     while total >= 10:
         total = sum_digits(total)

     print("The sum of digits in given number ",total)

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

Output:

```
Enter a number:
1999
The sum of digits in given number  1
```

Qn 17.

Program:

```
def reverse(number):
   length=len(number)
   reverse=number[length::-1]
   return reverse



def main():
    number = (input("Enter a number\n"))
    if number == reverse(number):
        print("The given number is a palindrome")
    else:
        print("The given number is not a palindrome")

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

Output:

```
Enter a number
4554
The given number is a palindrome

Enter a number
1234
The given number is not a palindrome
```

Qn 18.

Program:

```
def fact(number):
   f = 1
   for i in range(1, number + 1):
       f = f * i
   return f



def main():
    number = int(input("Enter a number\n"))
    print("The factorial of the given number is ",fact(number))

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

Output:

```
Enter a number
5
The factorial of the given number is  120
```

Qn 19.

Program:

```
def digits(number):
   length=len(number)
   return length

def main():
    number = (input("Enter a number\n"))
    print("The number of digits in the given number is ",digits(number))

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

Output:

```
Enter a number
3221
The number of digits in the given number is  4
```

Qn 20.

Program:

```
def reverse(number):
   length=len(number)
   reverse=number[length::-1]
   return reverse



def main():
    number = (input("Enter a number\n"))
    print("The reverse of the given number is ",reverse(number))

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

Output:

```
Enter a number
345
The reverse of the given number is  543
```

Qn 21.

Program:

```
def check_narcis(number):
    numbers = str(number)
    power = len(numbers)
    total = 0
    for n in numbers:
        total += pow(int(n), power)
    if total == number:
        return True
    return False


def main():
    number = int(input("Enter a number\n"))
    if check_narcis(number):
        print("The given number is a Narcissistic number")
    else:
        print("The given number is not a Narcissistic number")

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

Output:

```
Enter a number
153
The given number is a Narcissistic number

Enter a number
67
The given number is not a Narcissistic number
```

Qn 22.

Program:

```
def prime(num):
    isprime = True
    if num > 1:
        for i in range(2,num):
            if (num % i) == 0:
                isprime = False
                break
        return isprime
    else:
       return isprime

def main():
    number = int(input("Enter a number\n"))
    if prime(number):
        print("The given number is a Prime number")
    else:
        print("The given number is not a Prime number")

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

Output:

```
Enter a number
3
The given number is a Prime number

Enter a number
6
The given number is not a Prime number
```

Qn 23.

Program:

```
def prime(num):
    isprime = True
    if num > 1:
        for i in range(2,num):
            if (num % i) == 0:
                isprime = False
                break
        return isprime
    else:
       return isprime

def main():
    start = int(input("Enter the start number range\n"))
    end = int(input("Enter the end number range\n"))
    print("The prime numbers within this range")
    for i in range(start, end + 1):
        if prime(i) == True:
            print(i)

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

Output:

```
Enter the start number range
100
Enter the end number range
200
The prime numbers within this range
101
103
107
109
113
127
131
137
139
149
151
157
163
167
173
179
181
191
193
197
199
```

Qn 24.

Program:

```
def check_narcis(number):
    numbers = str(number)
    power = len(numbers)
    total = 0
    for n in numbers:
        total += pow(int(n), power)
    if total == number:
        return True
    return False


def main():
    start = int(input("Enter the start number range\n"))
    end = int(input("Enter the end number range\n"))
    print("The Narcissistic numbers within this range")
    for i in range(start, end + 1):
        if check_narcis(i) == True:
            print(i)


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

Output:

```
Enter the start number range
100

Enter the end number range
1000
The Narcissistic numbers within this range
153
370
371
407
```

Qn 25.

Program:

```
def checkAdamNumber(number) :
    reverse=number[::-1]
    number_sqr=int(number)**2
    reverse_sqr=int(reverse )**2
    if int(number_sqr )==int(str(reverse_sqr)[::-1]):
        print(number+" is a adam number")
    else:
        print(number+" is not a adam number")

def main():
    number = input("Enter a number \n")
    checkAdamNumber(number)

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

Output:

```
Enter a number
7
7 is not a adam number

Enter a number
11
11 is a adam number
```

Qn 26.
Program:

```
def numberofDigits(n):
    cnt = 0
    while n > 0:
        cnt += 1
        n //= 10
    return cnt

def cal(num):
    digit = numberofDigits(num)
    powTen = pow(10, digit - 1)

    for i in range(digit):

        firstDigit = num // powTen

        left = (num * 10 + firstDigit -
               (firstDigit * powTen * 10))
        print(left, end = " ")

        num = left

def main():
    n = int(input("Enter the number\n"))
    cal(n)

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

Output:

```
Enter the number
3214
2143 1432 4321 3214
```

Qn 27.
Program:

```
def numberofDigits(n):
    cnt = 0
    while n > 0:
        cnt += 1
        n //= 10
    return cnt

def prime(num):
    isprime = True
    if num > 1:
        for i in range(2,num):
            if (num % i) == 0:
                isprime = False
                break
        return isprime
    else:
       return isprime

def cal(num):
    iscircprime = True
    digit = numberofDigits(num)
    powTen = pow(10, digit - 1)

    for i in range(digit):

        firstDigit = num // powTen

        left = (num * 10 + firstDigit -
               (firstDigit * powTen * 10))
        if prime(left) == False:
            iscircprime = False
            break

        num = left
    return iscircprime

def main():
    n = int(input("Enter the number\n"))
    if cal(n) == True:
        print("The number is a circular prime")
    else:
        print("The number is not a circular prime")

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

Output:

```
Enter the number
828
The number is not a circular prime

Enter the number
3779
The number is a circular prime
```

Qn 28.

Program:

```
def numberofDigits(n):
    cnt = 0
    while n > 0:
        cnt += 1
        n //= 10
    return cnt

def prime(num):
    isprime = True
    if num > 1:
        for i in range(2,num):
            if (num % i) == 0:
                isprime = False
                break
        return isprime
    else:
       return isprime

def cal(num):
    iscircprime = True
    digit = numberofDigits(num)
    powTen = pow(10, digit - 1)

    for i in range(digit):

        firstDigit = num // powTen

        left = (num * 10 + firstDigit -
               (firstDigit * powTen * 10))
        if prime(left) == False:
            iscircprime = False
            break

        num = left
    return iscircprime

def main():
    start = int(input("Enter the start number range\n"))
    end = int(input("Enter the end number range\n"))
    print("The circular prime number within this range")
    for i in range(start, end + 1):
        if cal(i) == True:
            print(i)

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

Output:

```
Enter the start number range
1000

Enter the end number range
50000
The circular prime number within this range
1193
1931
3119
3779
7793
7937
9311
9377
11939
19391
19937
37199
39119
```

Qn 29.

Program:

```
def compute_lcm(x, y):

  if x > y:
     greater = x
  else:
     greater = y

  while(True):
     if((greater % x == 0) and (greater % y == 0)):
        lcm = greater
        break
     greater += 1

  return lcm

def main():
   num1 = int(input("Enter the first number\n"))
   num2 = int(input("Enter the second number\n"))
   print("The LCM of two numbers is", compute_lcm(num1, num2))



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

Output:

```
Enter the first number
6

Enter the second number
8
The LCM of two numbers is 24
```

Qn 30.

Program:

```
def gcd(a,b):
    if(b==0):
        return a
    else:
        return gcd(b,a%b)

def main():
    a=int(input("Enter first number:\n"))
    b=int(input("Enter second number:\n"))
    print("GCD is: ", gcd(a,b))

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

Output:

```
Enter first number:
21

Enter second number:
49
GCD is:  7
```

Qn 31.

Program:

```
def sum_N(n):
    total = (n * (n + 1))//2
    return total

def main():
    n=int(input("Enter the sequence size:\n"))
    print("Sum of the series is: ", sum_N(n))

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

Output:

```
Enter the sequence size:
100
Sum of the series is:  5050
```

Qn 32.

Program:

```
def mul_table(n):
    for i in range(10):
        print("{} X {} = {}".format(n,i,(n*i)))

def main():
    n=int(input("Enter a number:\n"))
    print("Multiplication table")
    mul_table(n)

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

Output:

```
Enter a number:
13
Multiplication table
13 X 0 = 0
13 X 1 = 13
13 X 2 = 26
13 X 3 = 39
13 X 4 = 52
13 X 5 = 65
13 X 6 = 78
13 X 7 = 91
13 X 8 = 104
13 X 9 = 117
```

Qn 33.

Program:

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



def main():
    fib_range = int(input("Enter the range for Fibonacci Numbers\n"))
    if fib_range <= 0:
        print("Please enter a positive integer")
    else:
        print("Fibonacci sequence:")
        for i in range(fib_range):
            print(fibonacci(i))

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

Output:

```
Enter the range for Fibonacci Numbers
6
Fibonacci sequence:
0
1
1
2
3
5
```

Qn 34.
Program:

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


def main():
    fib_range = int(input("Enter the range for Fibonacci Numbers\n"))
    if fib_range <= 0:
        print("Please enter a positive integer")
    else:
        print("Sum of Fibonacci sequence:")
        total = 0
        for i in range(fib_range):
            total = total + fibonacci(i)
        print(total)

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

Output:

```
Enter the range for Fibonacci Numbers
6
Sum of Fibonacci sequence:
12
```

Qn 35.

Program:

```
def reverse(string):
   stringlength=len(string)
   reversedString=string[stringlength::-1]
   print(reversedString)



def main():
    string = (input("Enter a string value\n"))
    print("The reversed string is ")
    reverse(string)

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

Output:

```
Enter a string value
kare university
The reversed string is
ytisrevinu erak
```

Qn 36.
Program:

```
def reverse(string):
   stringlength=len(string)
   reversedString=string[stringlength::-1]
   return reversedString


def main():
    string = (input("Enter a string value\n"))
    if string == reverse(string):
        print("The given string is a palindrome")
    else:
        print("The given string is not a palindrome")

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

Output:

```
Enter a string value
apple
The given string is not a palindrome

Enter a string value
bob
The given string is a palindrome
```

Qn 37.

Program:

```
def vowels(string):
   vow_list=['a','e','i','o','u']
   counter = 0
   for i in string:
       if i in vow_list:
           counter = counter + 1
   return counter



def main():
    string = (input("Enter a string value\n"))
    print("The number of vowels in given string is ", vowels(string))

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

Output:

```
Enter a string value
apple
The number of vowels in given string is  2

Enter a string value
rhythm
The number of vowels in given string is  0
```

Qn 38.

Program:

```
def check_substring(string, substring):
    if substring in string:
        return True
    else:
        return False


def main():
    string = (input("Enter a string value\n"))
    substring = (input("Enter a sub-string to find\n"))
    if check_substring(string, substring):
        print("The given sub-string is present")
    else:
        print("The given sub-string is not present")

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

Output:

```
Enter a string value
Andhra Pradesh

Enter a sub-string to find
desh
The given sub-string is present

Enter a string value
Apple

Enter a sub-string to find
po
The given sub-string is not present
```

Qn 39.

Program:

```
def check_substring(string):
    count = string.split(" ")
    return len(count)



def main():
    string = (input("Enter a string value\n"))
    print("The number of sub-strings present in given string is ",check_substring(string))

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

Output:

```
Enter a string value
Bull taming is a sport in tamilnadu
The number of sub-strings present in given string is  7
```

Qn 40.

Program:

```
def check_substring(string):
    counter = 0
    for i in string:
        if i.isdigit():
            counter = counter + 1
    return counter


def main():
    string = (input("Enter a sentence\n"))
    print("The count of Numerical Characters in given sentence is ",check_substring(string))


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

Output:

```
Enter a sentence
On1 tw3 th4e5 fo89
The count of Numerical Characters in given sentence is  6
```

Qn 41.

Program:

```
def check_substring(string):
    count = string.split(" ")
    for i in reversed(count):
        print(i, end=(" "))



def main():
    string = (input("Enter a sentence\n"))
    print("The given sentence in reverse order")
    check_substring(string)

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

Output:

```
Enter a sentence
Bull taming is a sport in tamilnadu
The given sentence in reverse order
tamilnadu in sport a is taming Bull
```

Qn 42.

Program:

```
def check_substring(string1, string2):
    if string1.lower() == string2.lower():
        return True
    else:
        return False


def main():
    string1 = (input("Enter first string\n"))
    string2 = (input("Enter second string\n"))
    if check_substring(string1,string2):
        print("The two strings are same")
    else:
        print("The two strings are not same")


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

Output:

```
Enter first string
ABCD

Enter second string
abcd
The two strings are same

Enter first string
abcd

Enter second string
EFGH
The two strings are not same
```

Qn 43.
Program:

```
def check_substring(sentence, string1, string2):
    if string1 in sentence:
        sentence = sentence.replace(string1, string2)
        return sentence
    else:
        return "Could not find the string to replace in given sentence"


def main():
    sentence  = (input("Enter a sentence\n"))
    string1 = (input("Enter the string to find\n"))
    string2 = (input("Enter the string to replace with\n"))
    print(check_substring(sentence, string1, string2))



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

Output:

```
Enter a sentence
My state is Tamil Nadu

Enter the string to find
Tamil Nadu

Enter the string to replace with
Andhra Pradesh
My state is Andhra Pradesh
```

Qn 44.

Program:

```
def bubbleSort(arr):
    n = len(arr)

    for i in range(n-1):

        for j in range(0, n-i-1):

            if arr[j] > arr[j+1] :
                arr[j], arr[j+1] = arr[j+1], arr[j]

    return arr


def main():
    List = []
    n = int(input("Enter the list size\n"))
    print("Enter the numbers")
    for i in range(n):
        List.append(int(input()))
    print("Sorted list ",bubbleSort(List))


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

Output:

```
Enter the list size
5
Enter the numbers

89

65

10

24

200
Sorted list  [10, 24, 65, 89, 200]
```

Qn 45.

Program:

```
def selectionSort(A):
    for i in range(len(A)):
        min_idx = i
        for j in range(i+1, len(A)):
            if A[min_idx] > A[j]:
                min_idx = j
                A[i], A[min_idx] = A[min_idx], A[i]

    return A


def main():
    List = []
    n = int(input("Enter the list size\n"))
    print("Enter the numbers")
    for i in range(n):
        List.append(int(input()))
    print("Sorted list ",selectionSort(List))

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

Output:

```
Enter the list size
5
Enter the numbers

200

344

12

9

566
Sorted list  [9, 12, 200, 344, 566]
```

Qn 46.
Program:

```
def insertionSort(arr):
    for i in range(1, len(arr)):

        key = arr[i]
        j = i-1
        while j >=0 and key < arr[j] :
                arr[j+1] = arr[j]
                j -= 1
                arr[j+1] = key
    return arr

def main():
    List = []
    n = int(input("Enter the list size\n"))
    print("Enter the numbers")
    for i in range(n):
        List.append(int(input()))
    print("Sorted list ",insertionSort(List))

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

Output:

```
Enter the list size
5
Enter the numbers

666

777

0

12

90
Sorted list  [0, 12, 90, 666, 777]
```

Qn 47.

Program:

```
def add_matrix(M1, M2):
    result = [[M1[i][j] + M2[i][j]  for j in range(len(M1[0]))] for i in range(len(M1))]
    print(result)

def main():
    M1 = []
    M2 = []
    r = int(input("Enter the row size\n"))
    c = int(input("Enter the column size\n"))
    print("Enter first matrix elements")
    for i in range(r):
        List = []
        for j in range(c):
            List.append(int(input()))
        M1.append(List)
    print("Enter second matrix elements")
    for i in range(r):
        List = []
        for j in range(c):
            List.append(int(input()))
        M2.append(List)
    print("Matrix Addition")
    add_matrix(M1, M2)

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

Output:

```
Enter the row size
2
Enter the column size
2
Enter first matrix elements
1
2
3
4
Enter second matrix elements
4
3
2
1
Matrix Addition
[[5, 5], [5, 5]]
```

Qn 48.
Program:

```
def mul_matrix(X, Y):
    result = [[sum(a*b for a,b in zip(X_row,Y_col)) for Y_col in zip(*Y)] for X_row in X]
    print(result)

def main():
    M1 = []
    M2 = []
    r = int(input("Enter the row size\n"))
    c = int(input("Enter the column size\n"))
    print("Enter first matrix elements")
    for i in range(r):
        List = []
        for j in range(c):
            List.append(int(input()))
        M1.append(List)
    print("Enter second matrix elements")
    for i in range(r):
        List = []
        for j in range(c):
            List.append(int(input()))
        M2.append(List)
    print("Matrix Multiplication")
    mul_matrix(M1, M2)

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

Output:

```
Enter the row size
2
Enter the column size
2
Enter first matrix elements
1
2
3
4
Enter second matrix elements
4
3
2
1
Matrix Multiplication
[[8, 5], [20, 13]]
```

Qn 49.

Program:

```
import numpy as np
def matrix(X, r, c):
    result_r = [sum(X[i]) for i in range(r)]
    result_c = [sum(X[:,i]) for i in range(c)]
    print("Matrix Row Summation")
    print(result_r)
    print("Matrix Column Summation")
    print(result_c)


def main():
    M = []
    r = int(input("Enter the row size\n"))
    c = int(input("Enter the column size\n"))
    print("Enter first matrix elements")
    for i in range(r):
        List = []
        for j in range(c):
            List.append(int(input()))
        M.append(List)
    matrix(np.asarray(M), r, c)

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

Output:

```
Enter the row size
2

Enter the column size
2
Enter first matrix elements

1

2

3

4
Matrix Row Summation
[3, 7]
Matrix Column Summation
[4, 6]
```

Qn 50.

Program:

```
def main():
    class_dict = {}
    for i in range(1,6):
        student_dict = {}
        print("Enter details of Student ",i)
        name = input("Enter name\n")
        cgpa = input("Enter cgpa\n")
        age = input("Enter age\n")
        student_dict['Name'] = name
        student_dict['CGPA'] = cgpa
        student_dict['Age'] = age
        class_dict[str(i)] = student_dict
        list_stud = []
    for i in class_dict:
        if float(class_dict[i]['CGPA']) < 8.0:
            list_stud.append(class_dict[i]['Name'])
    if len(list_stud) > 0:
        print("Students with CGPA less than 8 ", list_stud)
    else:
        print("There are no students with less than 8 CGPA")

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

Output:

```
Enter details of Student  1
Enter name
AAA
Enter cgpa
9
Enter age
22
Enter details of Student  2
Enter name
BBB
Enter cgpa
7
Enter age
23
Enter details of Student  3
Enter name
CCC
Enter cgpa
8
Enter age
24
Enter details of Student  4
Enter name
```

```
DDD
Enter cgpa
9
Enter age
21
Enter details of Student  5
Enter name
EEE
Enter cgpa
5
Enter age
24
Students with CGPA less than 8  ['BBB', 'EEE']
```

Qn 51.

Program:

```
def list_manipulate(List):
    total = 0
    for i in range(len(List)):
        if i % 2 == 0:
            if List[i].isdigit():
                total = total + int(List[i])
    return total

def main():
    List = []
    n = int(input("Enter the list size\n"))
    print("Enter the numbers")
    for i in range(n):
        List.append(input())
    print("The sum of even indices of the list is  ",list_manipulate(List))

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

Output:

```
Enter the list size
5
Enter the numbers
1
2
3
4
5
The sum of even indices of the list is   9

Enter the list size
5
Enter the numbers
1
2
small
big
10
The sum of even indices of the list is   11
```

Qn 52.

Program:

```
def list_manipulate(List):
    pos = []
    for i in List:
        if i < 0:
            pos.append(i*-1)
        else:
            pos.append(i)
    return pos

def main():
    List = []
    n = int(input("Enter the list size\n"))
    print("Enter the numbers")
    for i in range(n):
        List.append(int(input()))
    print("The actual list is  ",List)
    print("The positive list is  ",list_manipulate(List))

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

Output:

```
Enter the list size
3
Enter the numbers

-1

2

-9
The actual list is   [-1, 2, -9]
The positive list is   [1, 2, 9]
```

Qn 53.

Program:

```
def prime(num):
    isprime = True
    if num > 1:
        for i in range(2,num):
            if (num % i) == 0:
                isprime = False
                break
        return isprime
    else:
       return isprime

def main():
    List =[]
    start = int(input("Enter the start number range\n"))
    end = int(input("Enter the end number range\n"))
    for i in range(start, end + 1):
        if prime(i) == True:
            List.append(i)
    print("The prime numbers within this range",List)

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

Output:

```
Enter the start number range
10

Enter the end number range
20
The prime numbers within this range [11, 13, 17, 19]
```

Qn 54.

Program:

```
def most_frequent(List):
    counter = 1
    dup_num = []

    for i in List:
        curr_frequency = List.count(i)
        if(curr_frequency > counter):
            counter = curr_frequency
            dup_num.append(i)
    if len(dup_num) > 0:
        print("Duplicates found in the list", dup_num)
    else:
        print("No duplicates found in the list")

def main():
    List = []
    n = int(input("Enter the list size\n"))
    print("Enter the numbers")
    for i in range(n):
        List.append(int(input()))
    most_frequent(List)

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

Output:

```
Enter the list size
4
Enter the numbers
1
2
3
4
No duplicates found in the list

Enter the list size
4
Enter the numbers
1
2
3
2
Duplicates found in the list [2]
```

Qn 55.

Program:

```
def prime(num):
    isprime = True
    if num > 1:
        for i in range(2,num):
            if (num % i) == 0:
                isprime = False
                break
        return isprime
    else:
       return isprime


def main():
    List = []
    count = 0
    n = int(input("Enter the list size\n"))
    print("Enter the numbers")
    for i in range(n):
        List.append(int(input()))
    for i in List:
        if prime(i):
            count = count + 1
    print("There are {} prime numbers in given list".format(count))


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

Output:

```
Enter the list size
5
Enter the numbers

1

2

3

4

5
There are 4 prime numbers in given list
```

Qn 56.

Program:

```
def check_substring(string1, string2):
    sentence = string1.replace(string2, "")
    return sentence


def main():
    string1 = (input("Enter the first string\n"))
    string2 = (input("Enter the second string\n"))
    print("After removing:\n",check_substring(string1, string2))



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

Output:

```
Enter the first string
My state is Andhra Pradesh

Enter the second string
My state is
After removing:
  Andhra Pradesh
```

Qn 57.

Program:

```
def most_frequent(List):
    counter = 0
    num = List[0]

    for i in List:
        curr_frequency = List.count(i)
        if(curr_frequency> counter):
            counter = curr_frequency
            num = i
    return num

def main():
    List = []
    n = int(input("Enter the list size\n"))
    print("Enter the numbers")
    for i in range(n):
        List.append(int(input()))
    print("The maximum repeating element is ",most_frequent(List))

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

Output:

```
Enter the list size
5
Enter the numbers

2

5

6

5

7
The maximum repeating element is  5
```

Qn 58.

Program:

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

def main():
    num = int(input("Enter a number\n"))
    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))

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

Output:

```
Enter a number
5
The factorial of 5 is 120
```

Qn 59.

Program:

```
def gcd(a,b):
    if(b==0):
        return a
    else:
        return gcd(b,a%b)

def main():
    a=int(input("Enter first number:\n"))
    b=int(input("Enter second number:\n"))
    print("GCD is: ", gcd(a,b))

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

Output:

```
Enter first number:
36

Enter second number:
90
GCD is:  18
```

Qn 60.

Program:

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



def main():
    fib_range = int(input("Enter the range for Fibonacci Numbers\n"))
    if fib_range <= 0:
        print("Please enter a positive integer")
    else:
        print("Fibonacci sequence:")
        for i in range(fib_range):
            print(fibonacci(i))

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

Output:

```
Enter the range for Fibonacci Numbers
6
Fibonacci sequence:
0
1
1
2
3
5
```