NAME-DEBAGNI BHATTACHARJEE.

ROLL NO-04

ENROLMENT NO-12023006015004. SEC-A.

SEMESTER -3rd

DEPARTMENT -COMPUTER APPLICATION (MCA)

1)To find the sum of square root of any three numbers.

```
def sum_of_three_numbers(a, b, c):
    return a + b + c
num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))
num3 = float(input("Enter the third number: "))
sum_result = sum_of_three_numbers(num1, num2, num3)
print("The sum of {}, {}, and {} is: {}".format(num1, num2, num3, sum_result))
```

Output-

```
Enter the first number: 4
```

Enter the second number: 9
Enter the third number: 16
The sum of square root is :7.0

2)To solve the quadratic equation.

```
Import cmath
def solve_quadratic(a, b, c):
    d = (b**2) – (4*a*c)
    sol1 = (-b – cmath.sqrt(d)) / (2 * a)
    sol2 = (-b + cmath.sqrt(d)) / (2 * a)
    return sol1, sol2
a = float(input(“Enter the coefficient a: “))
b= float(input(“Enter the coefficient b: “))
c= float(input(“Enter the coefficient c: “))
solution1, solution2 = solve_quadratic(a, b, c)
print(“The solutions to the equation are: {} and
{}”.format(solution1, solution2))
```

Output –

Enter the coefficient a: 1
Enter the coefficient b: -3
Enter the coefficient c: 2
The solutions to the equation are: (1+0j) and
(2+0j)

3)Find GCD of two numbers.

```
def gcd(a, b):
    while b:
        a,b = b, a % b
    return a
num1 = int(input("Enter the first number: "))
num2 = int(input("Enter the second number: "))
gcd_result = gcd(num1, num2)
print("The GCD of {} and {} is: {}".format(num1,
num2, gcd_result))
```

Output –

```
Enter the first number: 48
Enter the second number: 18
The GCD of 48 and 18 is: 6
```

4)Compute a)5 to the power of 8 b)square root of 400 c)exponent of 5 d)Logarithm of 625 base

```
Import math
power_result = 5 ** 8
sqrt_result = math.sqrt(400)
```

```
exp_result = math.exp(5)
log_result = math.log(625, 5)
print("5 to the power of 8 is:", power_result)
print("Square root of 400 is:", sqrt_result)
print("Exponent of 5 is:", exp_result)
print("Logarithm of 625 with base 5 is:", log_result)
```

Output-

```
5 to the power of 8 is: 390625
Square root of 400 is: 20.0
Exponent of 5 is: 148.4131591025766
Logarithm of 625 with base 5 is: 4.0
```

5)Compute a)sin of 60 degree b)cos of pi
c)sin(0.866025403784438 6) d)tan of 90 degree

```
Import math
def degrees_to_radians(degrees):
   return degrees * (math.pi / 180)
sin_60_degrees = math.sin(degrees_to_radians(60))
```

```
cos_pi = math.cos(math.pi)

sin_value = math.sin(0.8660254037844386)

try:

   tan_90_degrees =
math.tan(degrees_to_radians(90))

except Exception as e:

   tan_90_degrees = str(e)

print("Sin of 60 degrees is:", sin_60_degrees)

print("Cos of pi is:", cos_pi)

print("Sin of 0.8660254037844386 is:", sin_value)

print("Tan of 90 degrees is:", tan_90_degrees)
```

Output –

```
Sin of 60 degrees is: 0.8660254037844386

Cos of pi is: -1.0

Sin of 0.8660254037844386 is: 0.7617599814162892

Tan of 90 degrees is: 1.6331239353195370e+16
```

6)Define a sum function with two parameters and call the function

```
def sum(a, b):
    return a + b
result = sum(5, 8)
print("The sum of 5 and 8 is:", result)
```

Output –

```
The sum of 5 and 8 is: 13
```

7)WAP to reverse a given string.

```
def reverse_string(s):
    return s[::-1]
input_string = input("Enter a string: ")
reversed_string = reverse_string(input_string)
print("The reversed string is:", reversed_string)
```

Output –

```
Enter a string: hello
The reversed string is: olleh
```

8) Write a function to calculate the power of a number using recursion

```
def power(base, exponent):
    if exponent == 0:
        return 1
    elif exponent > 0:
        return base * power(base, exponent – 1)
    else:
        return 1 / power(base, -exponent)
base = 2
exponent = 3
result = power(base, exponent)
print(f”{base} to the power of {exponent} is {result}”)
```

Output –

2 to the power of 3 is 8

9)Convert Decimal number to Binary

```
def decimal_to_binary(n):
   if n == 0:
      return “0”
   else:
      return decimal_to_binary(n // 2) + str(n % 2)
decimal_number = 10
binary_result =
decimal_to_binary(decimal_number)
binary_result = binary_result.lstrip(‘0’)
print(f”The binary representation of
{decimal_number} is {binary_result}”)
```

Output –

The binary representation of 10 is 1010

10)Write a program in Python to check if a number is Krishnamurthy number.

```
      import math
```

```
def is_krishnamurthy_number(n):
    digits = str(n)
    sum_of_factorials =
sum(math.factorial(int(digit)) for digit in digits)
    return sum_of_factorials == n
number = 145
if is_krishnamurthy_number(number):
    print(f"{number} is a Krishnamurthy
number")
else:
    print(f"{number} is not a Krishnamurthy
number")
```

Output –

145 is a Krishnamurthy number

11) Write a program in Python to find the sum of digits of a number

```
def sum_of_digits(n):
    sum_digits = 0
    while n > 0:
        sum_digits += n % 10
        n//= 10
    return sum_digits
```

```
number = 12345
result = sum_of_digits(number)
print(f"The sum of the digits of {number} is {result}")
```

Output –

The sum of the digits of 12345 is 15

12) Write a program in Python that prompts the user to input a number and prints its multiplication table.

```
def print_multiplication_table(number):
    for I in range(1, 11):
        print(f"{number} x {i} = {number * i}")
number = int(input("Enter a number: "))
print_multiplication_table(number)
```

Output –

Enter a number: 5

5 x 1 = 5

5 x 2 = 10

5 x 3 = 15

5 x 4 = 20

5 x 5 = 25

5 x 6 = 30

5 x 7 = 35

5 x 8 = 40

5 x 9 = 45

5 x 10 = 50

13) Write a Python program to print the first 6 terms of a geometric sequence starting with 2 and having a common ratio of 3.

```
def geometric_sequence(start, ratio, terms):
    sequence = []
    current_term = start
    for _ in range(terms):
        sequence.append(current_term)
        current_term *= ratio
  return sequence
start = 2
ratio = 3
terms = 6
sequence = geometric_sequence(start, ratio, terms)
print(“The first 6 terms of the geometric sequence are:", sequence)
```

Output –

The first 6 terms of the geometric sequence are:
[2, 6, 18, 54, 162, 486]

14)Print the series upto N terms:
1,2,6,24,120,720 ...

```
def factorial(n):
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial(n – 1)
def print_factorial_series(N):
    series = []
    for I in range(1, N + 1):
        series.append(factorial(i))
    return series
n = int(input(“Enter the number of terms: “))
series = print_factorial_series(N)
print(f”The first {N} terms of the series are:
{series}”)
```

Output –

Enter the number of terms: 6
The first 6 terms of the series are: [1, 2, 6, 24, 120, 720]

15)Write a Python program that prompts the user to enter a base number and an exponent, and then calculates the power of the base to the exponent. The program should not use the exponentiation operator (**) or the math.pow() function.

```
def calculate_power(base, exponent):
    result = 1
    for _ in range(exponent):
        result *= base
    return result
base = int(input("Enter the base number: "))
exponent = int(input("Enter the exponent: "))
result = calculate_power(base, exponent)
print(f"{base} to the power of {exponent} is {result}")
```

Output –

```
Enter the base number: 2
Enter the exponent: 3
2 to the power of 3 is 8
```

Miscellaneous

16)Write a Python program to check whether a number is divisible by 5 or not.

```
number = int(input("Enter a number: "))
if number % 5 == 0:
    print(f"{number} is divisible by 5")
else:
    print(f"{number} is not divisible by 5")
```

Output –

```
Enter a number: 25
25 is divisible by 5
```

17)Write a Python program to check whether a number is Buzz or not.

```
def is_buzz_number(num):
    if num % 7 == 0 or num % 10 == 7:
        return True
    return False
test_numbers = [7, 70, 27, 26, 49, 123]
```

```
    if is_buzz_number(number):
        print(f"{number} is a Buzz number.")
    else:
        print(f"{number} is not a Buzz number.")
```

Output –

```
7 is a Buzz number.
70 is a Buzz number.
27 is a Buzz number.
26 is not a Buzz number.
49 is a Buzz number.
123 is not a Buzz number.
```

18)Write a Python program to calculate factorial of 12.

```
import math
def factorial_of_number(n):
    return math.factorial(n)
number = 12
```

```
result = factorial_of_number(number)
print(f"The factorial of {number} is {result}.")
```

Output-

The factorial of 12 is 479001600.

19)Write a Python program to calculate the sum of natural numbers up to a certain range.

```
def sum_of_natural_numbers_loop(n):
    total = 0
    for I in range(1, n + 1):
        total += i
    return total
def sum_of_natural_numbers_formula(n):
    return n * (n + 1) // 2
range_limit = 10
sum_loop =
sum_of_natural_numbers_loop(range_limit)
```

sum_formula = sum_of_natural_numbers_formula(range_limit)

print(f"Sum of natural numbers up to {range_limit} (using loop): {sum_loop}")

print(f"Sum of natural numbers up to {range_limit} (using formula): {sum_formula}")

Output –

Sum of natural numbers up to 10 (using loop): 55

Sum of natural numbers up to 10 (using formula): 55

20)Admission to a professional course is subject to the following conditions:

(a) marks in Mathematics >= 60 (b) marks in Physics >=50

(c) marks in Chemistry >=40 (d) Total in all 3 subjects >=200

(Or)

Total in Maths & Physics>=150

Given the marks in the 3 subjects of n (user input) students, write a program to process

the applications to list the eligible candidates.

```
def is_eligible(maths, physics, chemistry):
   If (maths >= 60 and physics >= 50 and chemistry >= 40 and
      (maths + physics + chemistry) >= 200):
      Return True
   Elif (maths + physics) >= 150:
      Return True
   Return False
def process_applications(students):
   eligible_candidates = []
   for I, (maths, physics, chemistry) in enumerate(students):
      if is_eligible(maths, physics, chemistry):
         eligible_candidates.append(I + 1)  # Store student number (1-based index)
   return eligible_candidates
```

```
n= int(input("Enter the number of students: "))
students_marks = []
for I in range(n):
    print(f"Enter marks for student {I + 1}:")
    maths = int(input("Mathematics: "))
    physics = int(input("Physics: "))
    chemistry = int(input("Chemistry: "))
    students_marks.append((maths, physics, chemistry))
eligible = process_applications(students_marks)
if eligible:
    print("Eligible candidates are:", ', '.join(map(str, eligible)))
else:
    print("No eligible candidates.")
```

Output –

```
Enter the number of students: 3
Enter marks for student 1:
```

Mathematics: 65

Physics: 55

Chemistry: 45

Enter marks for student 2:

Mathematics: 70

Physics: 60

Chemistry: 50

Enter marks for student 3:

Mathematics: 50

Physics: 45

Chemistry: 55

Eligible candidates are: 1, 2

21)Write a Python program to print all multiple of 10 between a given interval.

```
def print_multiples_of_ten(start, end):
    if start > end:
```

```
    print("Invalid interval. Start should be less than or equal to end.")
    return
  print(f"Multiples of 10 between {start} and {end}:")
  for num in range(start, end + 1):
    if num % 10 == 0:
      print(num, end=" ")
  print()
start = int(input("Enter the start of the interval: "))
end = int(input("Enter the end of the interval: "))
print_multiples_of_ten(start, end)
```

Output –

```
Enter the start of the interval: 5
Enter the end of the interval: 50
Multiples of 10 between 5 and 50:
10 20 30 40 50
```

22) Write a Python program to find median of a set of numbers.

```
def find_median(numbers):
    numbers.sort()
    n= len(numbers)
    if n % 2 == 0:
        median = (numbers[n//2 – 1] + numbers[n//2]) / 2
    else:
        median = numbers[n//2]
    return median
numbers = list(map(float, input(“Enter the numbers separated by spaces: “).split()))
median = find_median(numbers)
print(f”The median of the given numbers is {median}”)
```

Output –

```
Enter the numbers separated by spaces: 10 2 38 23 38 23 21
```

23) Write a program to compute the value of Euler's number that is used as the base of

natural logarithms. Use the following formula.

e= 1+ 1/1! +1 /2! + 1/3+..... 1/n!

```
import math
def compute_euler_number(n):
    e = 1
    for I in range(1, n + 1):
        e += 1 / math.factorial(i)
    return e
n = int(input("Enter the value of n: "))
euler_number = compute_euler_number(n)
print(f"The computed value of Euler's number € using {n} terms is: {euler_number}")
```

Output –

Enter the value of n: 10

The computed value of Euler's number (e) using 10 terms is: 2.7182818011463845

24)Write a Python program to generate all combination of 1, 2, or 3 using loop.

```
def generate_combinations():
    digits = [1, 2, 3]
    combinations = []
    for d1 in digits:
        combinations.append((d1,))
    for d1 in digits:
        for d2 in digits:
            combinations.append((d1, d2))
    for d1 in digits:
        for d2 in digits:
            for d3 in digits:
                combinations.append((d1, d2, d3))
    return combinations
```

```
all_combinations = generate_combinations()
print("All combinations of 1, 2, or 3:")
for combo in all_combinations:
    print(combo)
```

Output –

```
All combinations of 1, 2, or 3:
(1,)
(2,)
(3,)
(1, 1)
(1, 2)
(1, 3)
(2, 1)
(2, 2)
(2, 3)
(3, 1)
(3, 2)
(3, 3)
```

(1, 1, 1)

(1, 1, 2)

(1, 1, 3)

(1, 2, 1)

(1, 2, 2)

(1, 2, 3)

(1, 3, 1)

(1, 3, 2)

(1, 3, 3)

(2, 1, 1)

(2, 1, 2)

(2, 1, 3)

(2, 2, 1)

(2, 2, 2)

(2, 2, 3)

(2, 3, 1)

(2, 3, 2)

(2, 3, 3)

(3, 1, 1)

(3, 1, 2)

(3, 1, 3)

(3, 2, 1)

(3, 2, 2)

(3, 2, 3)

(3, 3, 1)

(3, 3, 2)

(3, 3, 3)

25)Write a Java program to read two integer values m and n and to decide and print whether m is multiple of n.

```
def is_multiple(m, n):
  if n == 0:
      return “Division by zero is not allowed.”
   elif m % n == 0:
      return f”{m} is a multiple of {n}.”
```

```
    else:
        return f"{m} is not a multiple of {n}."
m = int(input("Enter the value of m: "))
n= int(input("Enter the value of n: "))
result = is_multiple(m, n)
print(result)
```

Output –

```
Enter the value of m: 20
Enter the value of n: 5
20 is a multiple of 5.
```