Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2023-24

(310908) Python Programming Laboratory

Class: FY-MCA Shift / Div : F2 / B Roll Number : 51124

Name: Sameer Kakade Assignment No:6 Date ofImplementation:21/10/2023

```
# 1. Program to Display Powers of 2 Using Anonymous Function


powers_of_2 = lambda n: [2**i for i in range(n+1)]


n = 5

result = powers_of_2(n)

print(result)


# Output:

# [1, 2, 4, 8, 16, 32]



# 2. Program to Find Numbers Divisible by Another Number


def find_divisible_numbers(numbers, divisor):

    divisible_numbers = list(filter(lambda x: x % divisor == 0, numbers))

    return divisible_numbers


numbers = [10, 20, 33, 46, 55, 60, 75]

divisor = 5
```

```
result = find_divisible_numbers(numbers, divisor)
print(result)

# Output:
# [10, 20, 55, 60]


# 3. Program to Convert Decimal to Binary, Octal, and Hexadecimal

decimal_number = 25

binary = bin(decimal_number)
octal = oct(decimal_number)
hexadecimal = hex(decimal_number)

print(f"Binary: {binary}")
print(f"Octal: {octal}")
print(f"Hexadecimal: {hexadecimal}")

# Output:
# Binary: 0b11001
# Octal: 0o31
# Hexadecimal: 0x19


# 4. Program to Find ASCII Value of Character
```

```
character = 'A'
ascii_value = ord(character)
print(f"ASCII value of {character} is {ascii_value}")

# Output:
# ASCII value of A is 65


# 5. Program to Find HCF or GCD

import math

num1 = 24
num2 = 36
hcf = math.gcd(num1, num2)
print(f"HCF of {num1} and {num2} is {hcf}")

# Output:
# HCF of 24 and 36 is 12


# 6. Program to Find LCM

def find_lcm(x, y):
    lcm = (x*y) // math.gcd(x, y)
    return lcm
```

```
num1 = 24
num2 = 36
lcm = find_lcm(num1, num2)
print(f"LCM of {num1} and {num2} is {lcm}")

# Output:
# LCM of 24 and 36 is 72



# 7. Program to Find the Factors of a Number

def find_factors(number):
    factors = [i for i in range(1, number+1) if number % i == 0]
    return factors

number = 36
result = find_factors(number)
print(f"The factors of {number} are {result}")

# Output:
# The factors of 36 are [1, 2, 3, 4, 6, 9, 12, 18, 36]



# 8. Program to Make a Simple Calculator

def add(x, y):
    return x + y
```

```
def subtract(x, y):
    return x - y


def multiply(x, y):
    return x * y


def divide(x, y):
    return x / y


# Example usage
a = 10
b = 5
print(f"Addition: {add(a, b)}")
print(f"Subtraction: {subtract(a, b)}")
print(f"Multiplication: {multiply(a, b)}")
print(f"Division: {divide(a, b)}")


# Output:
# Addition: 15
# Subtraction: 5
# Multiplication: 50
# Division: 2.0



# 9. Program to Shuffle Deck of Cards
```

```
import itertools
import random

suits = ['Hearts', 'Diamonds', 'Clubs', 'Spades']
ranks = ['2', '3', '4', '5', '6', '7', '8', '9', '10', 'Jack', 'Queen', 'King', 'Ace']

deck = list(itertools.product(ranks, suits))
random.shuffle(deck)
print(deck)

# Output:
# A shuffled deck of cards



# 10. Program to Display Calendar

import calendar

year = 2023
month = 10

cal = calendar.month(year, month)
print(cal)

# Output:
# Display of calendar for October 2023
# 11. Program to Display Fibonacci Sequence Using Recursion

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

num_terms = 10
fib_sequence = [fibonacci(i) for i in range(num_terms)]
print(fib_sequence)

# Output:
# [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]


# 12. Program to Find Sum of Natural Numbers Using Recursion

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

num_terms = 10
sum_natural = sum_of_natural_numbers(num_terms)
```

```
print(f"The sum of first {num_terms} natural numbers is {sum_natural}")


# Output:
# The sum of first 10 natural numbers is 55



# 13. Program to Find Factorial of Number Using Recursion


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


number = 5
fact = factorial(number)
print(f"The factorial of {number} is {fact}")


# Output:
# The factorial of 5 is 120



# 14. Program to Convert Decimal to Binary Using Recursion


def decimal_to_binary(n):
    if n > 1:
        decimal_to_binary(n // 2)
    print(n % 2, end='')


decimal_number = 25

print(f"The binary representation of {decimal_number} is ", end='')

decimal_to_binary(decimal_number)


# Output:

# The binary representation of 25 is 11001
```