Lab 3

# Program 1 :

## Program Solution:

start = 1500

end = 2700

result = []

for num in range(start, end + 1):

if num % 7 == 0 and num % 5 == 0:

result.append(num)

print(result)

# Program 2 :

## Program Solution:

# Function to convert Celsius to Fahrenheit

def celsius\_to\_fahrenheit(celsius):

fahrenheit = (celsius \* 9/5) + 32

return fahrenheit

# Function to convert Fahrenheit to Celsius

def fahrenheit\_to\_celsius(fahrenheit):

celsius = (fahrenheit - 32) \* 5/9

return celsius

celsius\_temp = 60

fahrenheit\_temp = 45

converted\_to\_fahrenheit = celsius\_to\_fahrenheit(celsius\_temp)

converted\_to\_celsius = fahrenheit\_to\_celsius(fahrenheit\_temp)

print(f"{celsius\_temp}°C is {converted\_to\_fahrenheit}°F")

print(f"{fahrenheit\_temp}°F is {converted\_to\_celsius}°C")

# Program 3 :

## Program Solution :

import random

# Generate a random number between 1 and 9

random\_number = random.randint(1, 9)

guess = 0

while guess != random\_number:

guess = int(input("Guess a number between 1 and 9: "))

# Check if the guess is correct

if guess == random\_number:

print("Tukka lag gya tera")

else:

print("Dubara Try Mar")

# 

# 

# Program 4 :

## Program Solution :

# Number of rows in the pattern

n = 5

# Construct the pattern

for i in range(1, n + 1):

print("\* " \* i)

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

print("\* " \* i)

# Program 5:

## Program Solution :

## 

str = input("Enter the word : ")

reverse = ""

count = len(str)

while(count > 0):

reverse = reverse + str[count - 1]

count = count - 1

print(reverse)

# Program 6 :

## Program Solution :

# Define the series of numbers

numbers = (1, 2, 3, 4, 5, 6, 7, 8, 9)

# Initialize counters

even\_count = 0

odd\_count = 0

# Loop through the numbers

for num in numbers:

# Check if the number is even or odd

if num % 2 == 0:

even\_count += 1

else:

odd\_count += 1

# Print the counts

print(f"Number of even numbers: {even\_count}")

print(f"Number of odd numbers: {odd\_count}")

# Program 7 :

## Program Solution :

# Define the sample list

datalist = [1452, 11.23, 1+2j, True, 'Pokemon', (0, -1), [5, 12],

{"class": 'V', "section": 'A'}]

# Loop through the list and print each item and its type

for item in datalist:

print(f"{item}: {type(item)}")

# Program 8 :

## Program Solution :

# Loop through numbers from 0 to 6

for num in range(7):

# Skip 3 and 6

if num == 3 or num == 6:

continue

print(num)

# Program 9 :

## Program Solution :

a, b = 0, 1

# Print Fibonacci numbers until the value exceeds 50

while a <= 50:

print(a, end=' ')

# Calculate the next Fibonacci number

a, b = b, a + b

# Program :

## Program Solution:

# Loop through numbers from 1 to 50

for num in range(1, 51):

# Check for multiples of both three and five

if num % 3 == 0 and num % 5 == 0:

print("FizzBuzz")

# Check for multiples of three

elif num % 3 == 0:

print("Fizz")

# Check for multiples of five

elif num % 5 == 0:

print("Buzz")

# Print the number itself

else:

print(num)

# Program 10 :

## Program Solution :

m = int(input("Enter the number of rows: "))

n = int(input("Enter the number of columns: "))

array = []

for i in range(m):

row = []

for j in range(n):

row.append(i \* j)

array.append(row)

# Print the array

print(array)

# Program 11 :

## Program Solution :

lines = []

while True:

line = input("Enter a line (leave blank to end): ")

if line == "":

break

lines.append(line.lower())

# Print the lines

for line in lines:

print(line)

# Program 12 :

## Program Solution :

# Prompt the user to enter a sequence of binary numbers

binary\_numbers = input("Enter a sequence of comma separated binary numbers: ").split(',')

# Initialize an empty list to store the results

result = []

# Loop through the binary numbers

for binary in binary\_numbers:

# Convert the binary number to an integer

number = int(binary, 2)

# Check if the number is divisible by 5

if number % 5 == 0:

result.append(binary)

# Print the result

print(','.join(result))

# Program 13 :

## Program Solution :

# Prompt the user to enter a string

input\_string = input("Enter a string: ")

letter\_count = 0

digit\_count = 0

for char in input\_string:

# Check if the character is a letter

if char.isalpha():

letter\_count += 1

# Check if the character is a digit

elif char.isdigit():

digit\_count += 1

# Print the counts

print(f"Letters: {letter\_count}")

print(f"Digits: {digit\_count}")

# Program 14 :

## Program Solution :

import re

# Prompt the user to enter a password

password = input("Enter a password: ")

# Define the criteria for a valid password

criteria = [

lambda s: any(x.islower() for x in s), # at least 1 lowercase letter

lambda s: any(x.isupper() for x in s), # at least 1 uppercase letter

lambda s: any(x.isdigit() for x in s), # at least 1 digit

lambda s: any(x in 'S#@' for x in s), # at least 1 special character

lambda s: 6 <= len(s) <= 16 # length between 6 and 16 characters

]

# Check if the password meets all criteria

if all(criteria(password)):

print("Password is valid")

else:

print("Password is invalid")