ASSIGNMENT 9:

Name: Om Prasanna Kulkarni

PRN: 22510034

Batch: S3

Branch: CSE

Topic: Control Flow and GUI

1. Conditional Statements:

a. Write a Python program that checks if a number is even or odd and

prints the result.

Source Code:

number = int(input("Enter a number: "))

if number % 2 == 0:

    print(f"{number} is even.")

else:

    print(f"{number} is odd.")

Output:  


b. Create a program that asks the user for their age and determines if they

are eligible to vote (age &gt;= 18).

Source Code:

age = int(input("Please enter your age: "))

if age >= 18:

    print("You are eligible to vote.")

else:

    print("You are not eligible to vote.")

OUTPUT:



c. Implement a program that compares two numbers and finds the largest

among them.

Source Code:

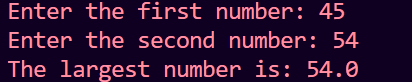
num1 = float(input("Enter the first number: "))

num2 = float(input("Enter the second number: "))

largest = max(num1, num2)

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

OUTPUT:



d. Build a program that determines if a given year is a leap year

(considering leap year rules).

Source Code:

year = int(input("Enter a year: "))

if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):

    print(year, "is a leap year.")

else:

    print(year, "is not a leap year.")

OUTPUT:



2. Loops:

a. Write a Python program that prints all the prime numbers within a given

range using a while loop.

Source Code:

start = int(input("Enter the start of the range: "))

end = int(input("Enter the end of the range: "))

print("Prime numbers in the range", start, "to", end, "are:")

current\_number = start

while current\_number <= end:

    if current\_number > 1:

        is\_prime = True

        for i in range(2, int(current\_number \*\* 0.5) + 1):

            if current\_number % i == 0:

                is\_prime = False

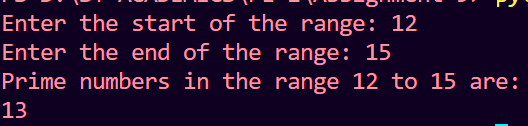
                break

        if is\_prime:

            print(current\_number)

    current\_number += 1

OUTPUT:



b. Create a program that calculates the factorial of a number using a for loop.

Source Code:

n = int(input("Enter a number: "))

factorial = 1

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

    factorial \*= i

print(f"The factorial of {n} is {factorial}")

OUTPUT:



c. Implement a program that prints the Fibonacci sequence up to a specified

number of terms using a for loop.

Source Code:

n = int(input("Enter the number of Fibonacci terms to generate: "))

a, b = 0, 1

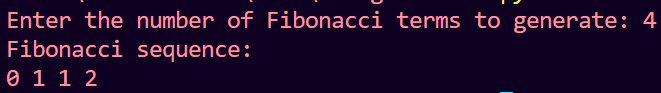
print("Fibonacci sequence:")

for \_ in range(n):

    print(a, end=" ")

    a, b = b, a + b

OUTPUT:



d. Build a program that finds and prints the sum of all even numbers from 1 to

100 using a for loop.

Source Code:

sum\_of\_evens = 0

for number in range(1, 101):

    if number % 2 == 0:

        sum\_of\_evens += number

print("The sum of even numbers from 1 to 100 is:", sum\_of\_evens)

OUTPUT:



3. Functions:

a. Write a function that takes two numbers as arguments and returns their

sum.

Source Code:

def add\_numbers(num1, num2):

    sum\_result = num1 + num2

    return sum\_result

num1 = 5

num2 = 3

result = add\_numbers(num1, num2)

print(f"The sum of {num1} and {num2} is {result}")

OUTPUT:



b. Create a function that checks if a given string is a palindrome.

Source Code:

def is\_palindrome(input\_string):

    clean\_string = input\_string.replace(" ", "").lower()

    return clean\_string == clean\_string[::-1]

input\_string = "racecar"

if is\_palindrome(input\_string):

    print(f"'{input\_string}' is a palindrome.")

else:

    print(f"'{input\_string}' is not a palindrome.")

OUTPUT:



c. Implement a function to calculate the area of a circle when given its

radius.

Source Code:

def calculate\_circle\_area(radius):

    if radius < 0:

        return "Radius should be non-negative."

    else:

        pi\_approximation = 3.14159

        area = pi\_approximation \* (radius \*\* 2)

        return area

radius = 5

area = calculate\_circle\_area(radius)

print(f"The area of a circle with radius {radius} is {area:.2f}")

OUTPUT:



d. Build a program that uses a function to convert a temperature from

Celsius to Fahrenheit and vice versa.

Source Code:

def celsius\_to\_fahrenheit(celsius):

    return (celsius \* 9/5) + 32

def fahrenheit\_to\_celsius(fahrenheit):

    return (fahrenheit - 32) \* 5/9

choice = input("Choose conversion type (C to F or F to C): ").upper()

if choice == 'C':

    celsius = float(input("Enter temperature in Celsius: "))

    fahrenheit = celsius\_to\_fahrenheit(celsius)

    print(f"{celsius}°C is equal to {fahrenheit}°F.")

elif choice == 'F':

    fahrenheit = float(input("Enter temperature in Fahrenheit: "))

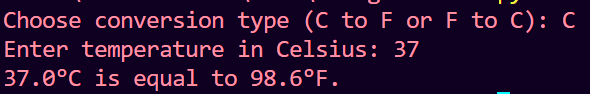
    celsius = fahrenheit\_to\_celsius(fahrenheit)

    print(f"{fahrenheit}°F is equal to {celsius}°C.")

else:

    print("Invalid choice. Please enter 'C' or 'F' for conversion.")

OUTPUT:



4. Combining Control Flow:

a. Write a program that takes a list of numbers and returns the average, sum,

and maximum value.

Source Code:

def calculate\_statistics(numbers):

    if not numbers:

        return None

    total = sum(numbers)

    average = total / len(numbers)

    maximum = max(numbers)

    return average, total, maximum

number\_list = [12, 34, 56, 78, 90, 123]

result = calculate\_statistics(number\_list)

if result is not None:

    average, total, maximum = result

    print(f"Average: {average:.2f}")

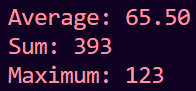
    print(f"Sum: {total}")

    print(f"Maximum: {maximum}")

else:

    print("List is empty.")

OUTPUT:



b. Create a program that generates a random number and asks the user to

guess it, providing feedback if the guess is too high or too low.

Source Code:

import random

random\_number = random.randint(1, 100)

attempts = 0

while True:

    guess = int(input("Guess the number (between 1 and 100): "))

    attempts += 1

    if guess < random\_number:

        print("Too low. Try again.")

    elif guess > random\_number:

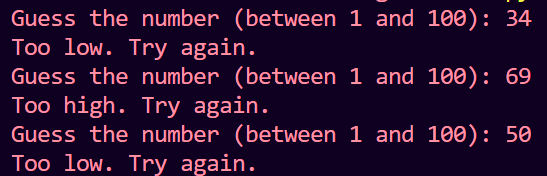
        print("Too high. Try again.")

    else:

        print(f"Congratulations! You guessed the number {random\_number} in {attempts} attempts.")

        break

OUTPUT:



c. Implement a program that calculates the factorial of a number using a

recursive function.

Source Code:

def factorial(n):

    if n == 0:

        return 1

    else:

        return n \* factorial(n - 1)

num = int(input("Enter a non-negative integer: "))

if num < 0:

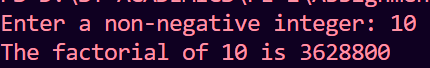
    print("Factorial is not defined for negative numbers.")

else:

    result = factorial(num)

    print(f"The factorial of {num} is {result}")

OUTPUT:



d. Build a program that asks the user for a number and prints all the prime

numbers up to that number.

Source Code:

def is\_prime(num):

    if num <= 1:

        return False

    for i in range(2, int(num \*\* 0.5) + 1):

        if num % i == 0:

            return False

    return True

def print\_primes\_up\_to(limit):

    for number in range(2, limit + 1):

        if is\_prime(number):

            print(number, end=" ")

limit = int(input("Enter a number: "))

if limit < 2:

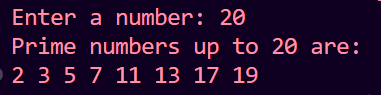
    print("There are no prime numbers less than 2.")

else:

    print("Prime numbers up to", limit, "are:")

    print\_primes\_up\_to(limit)

OUTPUT:



5. Nested Control Flow:

a. Write a program that generates a multiplication table for numbers from 1

to 10.

Source Code:

start = 1

end = 10

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

    print(f"Multiplication table for {i}:")

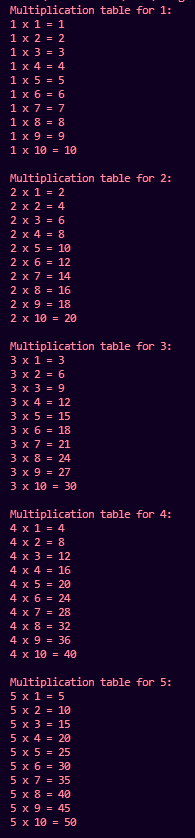
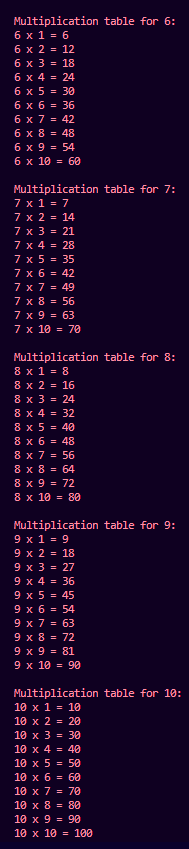
    for j in range(1, 11):

        product = i \* j

        print(f"{i} x {j} = {product}")

    print()

OUTPUT:

b. Create a program that simulates a simple ATM, allowing users to check

their balance, withdraw money, and deposit money.

Source Code:

balance = 0

def check\_balance():

    print(f"Your balance is ${balance:.2f}")

def withdraw\_money(amount):

    global balance

    if amount > balance:

        print("Insufficient funds.")

    else:

        balance -= amount

        print(f"You have withdrawn ${amount:.2f}. Your new balance is ${balance:.2f}")

def deposit\_money(amount):

    global balance

    balance += amount

    print(f"You have deposited ${amount:.2f}. Your new balance is ${balance:.2f}")

while True:

    print("\nWelcome to the Simple ATM")

    print("1. Check Balance")

    print("2. Withdraw Money")

    print("3. Deposit Money")

    print("4. Exit")

    choice = input("Enter your choice (1/2/3/4): ")

    if choice == '1':

        check\_balance()

    elif choice == '2':

        amount = float(input("Enter the amount to withdraw: $"))

        withdraw\_money(amount)

    elif choice == '3':

        amount = float(input("Enter the amount to deposit: $"))

        deposit\_money(amount)

    elif choice == '4':

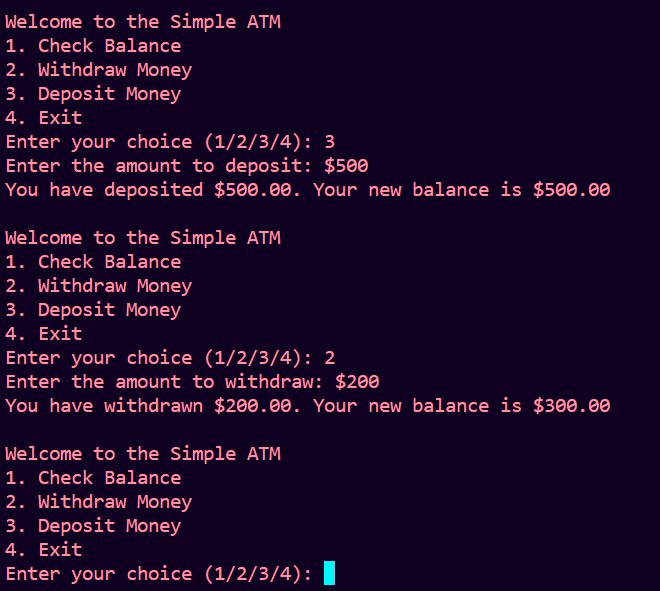
        print("Thank you for using the ATM. Goodbye!")

        break

    else:

        print("Invalid choice. Please choose 1, 2, 3, or 4.")

OUTPUT:



c. Implement a program that asks the user for their age and grade, and then

provides a personalized message based on their responses.

Source Code:

age = int(input("Please enter your age: "))

grade = int(input("Please enter your grade: "))

if age >= 18:

    if grade >= 9:

        print("You are old enough and in a high enough grade to graduate soon. Make the most of your education!")

    else:

        print("You are old enough to make important decisions about your future. Keep working hard in school.")

else:

    if grade < 9:

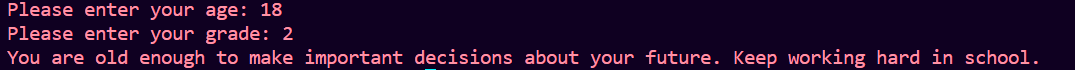
        print("You have plenty of time to learn and grow. Enjoy your school years.")

    else:

        print("You're young, but you're in a higher grade. Keep up the good work in school!")

*# Optional: You can add more personalized messages based on specific age and grade ranges as needed.*

OUTPUT:



6. Write a Python program to create GUI based student registration system

using tkinter.

a. Create 4 labels and corresponding textboxes for accepting Name,

address, Mobile number and email id.

b. Create radio button for accepting gender of student.

c. Use list box for accepting student’s state (add 5 state names)

d. Add checkbutton to select hostel requirement.

e. Display all the details on console and in Messagebox.

Source Code:

import tkinter as tk

from tkinter import messagebox

def submit\_registration():

    name = name\_entry.get()

    address = address\_entry.get()

    mobile\_number = mobile\_entry.get()

    email = email\_entry.get()

    gender = gender\_var.get()

    selected\_state = state\_listbox.get(state\_listbox.curselection())

    hostel = hostel\_var.get()

    print("Name:", name)

    print("Address:", address)

    print("Mobile Number:", mobile\_number)

    print("Email ID:", email)

    print("Gender:", gender)

    print("State:", selected\_state)

    print("Hostel Requirement:", "Yes" if hostel else "No")

    info\_message = f"Name: {name}\nAddress: {address}\nMobile Number: {mobile\_number}\nEmail ID: {email}\nGender: {gender}\nState: {selected\_state}\nHostel Requirement: {'Yes' if hostel else 'No'}"

    messagebox.showinfo("Registration Details", info\_message)

root = tk.Tk()

root.title("Student Registration System")

name\_label = tk.Label(root, text="Name:")

name\_label.pack()

name\_entry = tk.Entry(root)

name\_entry.pack()

address\_label = tk.Label(root, text="Address:")

address\_label.pack()

address\_entry = tk.Entry(root)

address\_entry.pack()

mobile\_label = tk.Label(root, text="Mobile Number:")

mobile\_label.pack()

mobile\_entry = tk.Entry(root)

mobile\_entry.pack()

email\_label = tk.Label(root, text="Email ID:")

email\_label.pack()

email\_entry = tk.Entry(root)

email\_entry.pack()

gender\_label = tk.Label(root, text="Gender:")

gender\_label.pack()

gender\_var = tk.StringVar()

male\_radio = tk.Radiobutton(root, text="Male", variable=gender\_var, value="Male")

male\_radio.pack()

female\_radio = tk.Radiobutton(root, text="Female", variable=gender\_var, value="Female")

female\_radio.pack()

state\_label = tk.Label(root, text="State:")

state\_label.pack()

state\_listbox = tk.Listbox(root, selectmode=tk.SINGLE)

state\_listbox.pack()

states = ["State 1", "State 2", "State 3", "State 4", "State 5"]

for state in states:

    state\_listbox.insert(tk.END, state)

hostel\_var = tk.BooleanVar()

hostel\_checkbutton = tk.Checkbutton(root, text="Hostel Requirement", variable=hostel\_var)

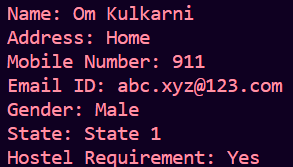
hostel\_checkbutton.pack()

submit\_button = tk.Button(root, text="Submit", command=submit\_registration)

submit\_button.pack()

root.mainloop()

OUTPUT:



7. Write a program that reads a text file and creates another file that is identical

except that every sequence of consecutive blank spaces is replaced by a

single space. (perform write operation on other file)

Source Code:

input\_file\_name = "input.txt"

output\_file\_name = "output.txt"

try:

    with open(input\_file\_name, 'r') as input\_file:

        input\_content = input\_file.read()

    modified\_content = ' '.join(input\_content.split())

    with open(output\_file\_name, 'w') as output\_file:

        output\_file.write(modified\_content)

    print(f"Consecutive spaces removed and saved to {output\_file\_name}")

except FileNotFoundError:

    print(f"File '{input\_file\_name}' not found.")

except Exception as e:

    print(f"An error occurred: {e}")

OUTPUT:



