NAME: OWOLABI DANIEL

MATRIC NO.: 21CG029915

PROGRAMME: COMPUTER SCIENCE

**Comparison of Prokaryotic and Eukaryotic Cells**

**Prokaryotic Cells:**

1. **Size:** Generally smaller (1-10 μm).
2. **Nucleus:** No true nucleus; DNA is in a nucleoid region.
3. **Organelles:** No membrane-bound organelles.
4. **DNA:** Circular DNA, not enclosed in a membrane.
5. **Reproduction:** Asexual (binary fission).
6. **Examples:** Bacteria and Archaea.

**Eukaryotic Cells:**

1. **Size:** Generally larger (10-100 μm).
2. **Nucleus:** True nucleus enclosed in a nuclear membrane.
3. **Organelles:** Membrane-bound organelles (e.g., mitochondria, Golgi apparatus).
4. **DNA:** Linear DNA enclosed in a nucleus.
5. **Reproduction:** Asexual or sexual (mitosis/meiosis).
6. **Examples:** Animals, plants, fungi, and protists.

(a) Python Program to Calculate Real-Life Size from Microscope Size

def calculate\_real\_size(microscope\_size, magnification):

if magnification <= 0:

raise ValueError("Magnification must be a positive number.")

return microscope\_size / magnification

# Example usage:

microscope\_size = float(input("Enter the size under the microscope (μm): "))

magnification = float(input("Enter the magnification: "))

real\_size = calculate\_real\_size(microscope\_size, magnification)

print(f"Real-life size: {real\_size} μm")

(b) Extended Program with Database Storage

import sqlite3

def initialize\_database():

conn = sqlite3.connect("specimens.db")

cursor = conn.cursor()

cursor.execute("""

CREATE TABLE IF NOT EXISTS specimens (

username TEXT,

microscope\_size REAL,

magnification REAL,

actual\_size REAL

)

""")

conn.commit()

conn.close()

def store\_specimen(username, microscope\_size, magnification, actual\_size):

conn = sqlite3.connect("specimens.db")

cursor = conn.cursor()

cursor.execute("""

INSERT INTO specimens (username, microscope\_size, magnification, actual\_size)

VALUES (?, ?, ?, ?)

""", (username, microscope\_size, magnification, actual\_size))

conn.commit()

conn.close()

initialize\_database()

# Example usage:

username = input("Enter your username: ")

microscope\_size = float(input("Enter the size under the microscope (μm): "))

magnification = float(input("Enter the magnification: "))

actual\_size = calculate\_real\_size(microscope\_size, magnification)

store\_specimen(username, microscope\_size, magnification, actual\_size)

print(f"Real-life size: {actual\_size} μm (stored in database)")

(c) Python-Based GUI (Using Tkinter)

import tkinter as tk

from tkinter import messagebox

import sqlite3

def calculate\_and\_store():

try:

username = username\_entry.get()

microscope\_size = float(microscope\_size\_entry.get())

magnification = float(magnification\_entry.get())

actual\_size = microscope\_size / magnification

actual\_size\_label.config(text=f"Actual size: {actual\_size} μm")

# Store in database

conn = sqlite3.connect("specimens.db")

cursor = conn.cursor()

cursor.execute("""

INSERT INTO specimens (username, microscope\_size, magnification, actual\_size)

VALUES (?, ?, ?, ?)

""", (username, microscope\_size, magnification, actual\_size))

conn.commit()

conn.close()

messagebox.showinfo("Success", "Data stored successfully!")

except ValueError:

messagebox.showerror("Error", "Invalid input. Please enter numbers.")

# Initialize GUI

root = tk.Tk()

root.title("Microscope Size Calculator")

tk.Label(root, text="Username:").pack()

username\_entry = tk.Entry(root)

username\_entry.pack()

tk.Label(root, text="Microscope size (μm):").pack()

microscope\_size\_entry = tk.Entry(root)

microscope\_size\_entry.pack()

tk.Label(root, text="Magnification:").pack()

magnification\_entry = tk.Entry(root)

magnification\_entry.pack()

calculate\_button = tk.Button(root, text="Calculate and Store", command=calculate\_and\_store)

calculate\_button.pack()

actual\_size\_label = tk.Label(root, text="Actual size: ")

actual\_size\_label.pack()

root.mainloop()