```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""

Created on Tue Jun 25 18:09:02 2024
---------------------------------------------------------------------

#### University
#### (####) Principles of Programming
Assessment 1 ----------------------------------------------------------------------

Student Name: Ali Shahbaz Aman
Student ID: #####
Learning Facilitator: ####
Subject Coordinator: ####
---------------------------------------------------------------------

"""

                            # # QUESTION 1

# Inputting customer details
name = input("Please enter your full name: ")
number = input("Please enter your mobile number (in 10 digits): ")
address = input("Please enter your home address: ")
email_id = input("Please enter your email address: ")
dob = input("Please enter your date and month of birth: ")
birth_year = int(input("Please enter your year of birth: "))
age = 2024 - birth_year

if len(number) != 10:
    print("Phone number needs to be 10 digits") #phone number stays in format
if age < 21:
    print("Sorry, the minimum age to register is 21") #age check using birth year
else:
    print("Welcome! Here are your details:","\n", name,"\n", number,"\n",
address,"\n", email_id,"\n", dob)


                            # # QUESTION 2

# # Inputting vehicle measurements
# length = eval(input("Please enter the length of the vehicle in meters: "))
# width = eval(input("Please enter the width of the vehicle in meters: "))
# height = eval(input("Please enter the height of the vehicle in meters: "))

# weight = 100 * (length * width * height) # each cubic meter can hold 100kg, thus
multiplying the volume with 100

# if weight > 5000:
#     print("The maximum weight capacity is 5000 kg") #maximum weight limit
# else:
#     print("The maximum weight the vehicle can carry is", weight, "kg")


                            # # QUESTION 3

# #Inputting distance and speed
# dist = eval(input("Please enter the distance of the route in km: "))
# speed = eval(input("Please enter the average speed of the delivery vehicle in km/
h: "))
```

```
# time = dist/speed # formula for calculating estimated delivery time

# rest_periods = int(time // 15) #calculates rest sessions based on time
# total_time = time + (rest_periods * 8) #adds rest time of 8 hours per session
# print("The estimated delivery time is", total_time, "hours")


                            # # QUESTION 4

# # Input weight (rounded up) and distance
# weight = round(eval(input("Please enter the weight of the products in kg: ")))
# dist = eval(input("Please enter the distance of the delivery in km: "))

# cost_per_km = weight * 0.10 #
# total_cost = (cost_per_km * dist) #formula to calculate total cost

# if  dist > 100: #discount applied
#     print("The total cost of the order is", total_cost * 0.95, "dollars")
# elif total_cost < 30: #minimum delivery charge error
#     print("The minimum delivery charge is 30")
# else:
#     print("The total cost of the order is",total_cost, "dollars")


                            # # QUESTION 5

# #Get the names and distances of the three destinations
# destination1_name = input("Enter the name of destination 1: ")
# destination1_distance = float(input("Enter the distance to destination 1 from the
warehouse (in km): "))

# destination2_name = input("Enter the name of destination 2: ")
# destination2_distance = float(input("Enter the distance to destination 2 from the
warehouse (in km): "))

# destination3_name = input("Enter the name of destination 3: ")
# destination3_distance = float(input("Enter the distance to destination 3 from the
warehouse (in km): "))

# # Determine order of the destinations based on distance
# if destination1_distance >= destination2_distance and destination1_distance >=
destination3_distance:
#     if destination2_distance >= destination3_distance:
#         route = "Warehouse -> " + destination1_name + " -> " + destination2_name
+ " -> " + destination3_name + " -> Warehouse"
#     else:
#         route = "Warehouse -> " + destination1_name + " -> " + destination3_name
+ " -> " + destination2_name + " -> Warehouse"
# elif destination2_distance >= destination1_distance and destination2_distance >=
destination3_distance:
#     if destination1_distance >= destination3_distance:
#         route = "Warehouse -> " + destination2_name + " -> " + destination1_name
+ " -> " + destination3_name + " -> Warehouse"
#     else:
#         route = "Warehouse -> " + destination2_name + " -> " + destination3_name
+ " -> " + destination1_name + " -> Warehouse"
# else:
#     if destination1_distance >= destination2_distance:
```

```
#         route = "Warehouse -> " + destination3_name + " -> " + destination1_name
+ " -> " + destination2_name + " -> Warehouse"
#     else:
#         route = "Warehouse -> " + destination3_name + " -> " + destination2_name
+ " -> " + destination1_name + " -> Warehouse"

# # Display the optimized route
# print("Optimized Route:\n", route)


                            # # QUESTION 6

# # Input weight in pounds
# weight_lbs = eval(input("Please enter the weight of the goods in pounds: "))
# weight_kg = weight_lbs * 0.453592 # Convert pounds to kg

# # Categorize based on weight
# if weight_kg < 10:
#     category = "Lightweight"
#     print("Goods Category:", category)
# elif weight_kg >= 10 and weight_kg <50:
#     category = "Mediumweight"
#     print("Goods Category:", category)
# elif weight_kg >= 50 and weight_kg <120:
#     category = "Heavyweight"
#     print("Goods Category:", category)
# else:
#     # Message to divide boxes if over 120kg
#     print("Item is over 120kg and must be divided into small boxes")


                            # # QUESTION 7

# # List of tracking numbers in conjunction with the status
# tracking_numbers = ["A00001D", "A00002D", "A00003D", "A00004D", "A00005D"]
# status = ["In transit", "Out for delivery", "Delivered", "In transit", "Out for
delivery"]

# # Get the tracking number from the user
# tracking_number = input("Please enter the package's tracking number: ")


# # Check if the tracking number is in the data
# if tracking_number in tracking_numbers:

#     # Get the index of the tracking number
#     index = tracking_numbers.index(tracking_number)

#     # Display the current status of the package
#     print("The current status of the package is:", status[index])

# else:
#     # Display an error message if the tracking number is not found
#     print("Tracking number not found. Please try again.")


                            # # QUESTION 8

# # Initialize empty lists to store delivery destinations and times
```

```
# destinations = []
# delivery_times = []

# # Ask user to input the destination and delivery time for 3 deliveries
# dest_1 = input("Enter the destination of delivery 1: ")
# del_time1 = float(input("Enter the delivery time for destination 1 (in hours):
"))
# destinations.append(dest_1)
# delivery_times.append(del_time1)

# dest_2 = input("Enter the destination of delivery 2: ")
# del_time2 = float(input("Enter the delivery time for destination 2 (in hours):
"))
# destinations.append(dest_2)
# delivery_times.append(del_time2)

# dest_3 = input("Enter the destination of delivery 3: ")
# del_time3 = float(input("Enter the delivery time for destination 3 (in hours):
"))
# destinations.append(dest_3)
# delivery_times.append(del_time3)

# # Calculate the average delivery time
# avg_del_time = (del_time1 + del_time2 + del_time3) / 3

# print("\nDelivery Statistics:")
# print("Average delivery time:", round(avg_del_time), "hours")


# # Find the fastest and slowest delivery times with destinations
# if del_time1 <= del_time2 and del_time1 <= del_time3:
#     print("Fastest delivery:", dest_1, "with", del_time1, "hours")
# elif del_time2 <= del_time1 and del_time2 <= del_time3:
#     print("Fastest delivery:", dest_2, "with", del_time2, "hours")
# else:
#     print("Fastest delivery:", dest_3, "with", del_time3, "hours")

# if del_time1 >= del_time2 and del_time1 >= del_time3:
#     print("Slowest delivery:", dest_1, "with", del_time1, "hours")
# elif del_time2 >= del_time1 and del_time2 >= del_time3:
#     print("Slowest delivery:", dest_2, "with", del_time2, "hours")
# else:
#     print("Slowest delivery:", dest_3, "with", del_time3, "hours")


                            # # QUESTION 9

# # Get the names and license numbers of the drivers
# driver1_name = input("Enter the name of driver 1: ")
# driver1_license = input("Enter the license number of driver 1 (in the format XXX-
XXX): ")
# if len(driver1_license,) != 7: # Ensure licence format
#     print("License number needs to be in the correct format")

# driver2_name = input("Enter the name of driver 2: ")
# driver2_license = input("Enter the license number of driver 2 (in the format XXX-
XXX): ")
# if len(driver2_license,) != 7: # Ensure licence format
#     print("License number needs to be in the correct format")
```

```
# driver3_name = input("Enter the name of driver 3: ")
# driver3_license = input("Enter the license number of driver 3 (in the format XXX-
XXX): ")
# if len(driver3_license,) != 7: # Ensure licence format
#     print("License number needs to be in the correct format")

# # Get the number of deliveries available
# num_deliveries = int(input("Enter the number of deliveries available: "))

# # Assign deliveries to each driver
# deliveries_per_driver = num_deliveries // 3 # Dividing deliveries beteen drivers
# remaining_deliveries = num_deliveries % 3 # Remaining Deliveries for driver 3
# # Assigning drivers their deliveries
# driver1_deliveries = deliveries_per_driver
# driver2_deliveries = deliveries_per_driver
# driver3_deliveries = deliveries_per_driver + remaining_deliveries

# # Display the delivery schedule
# print("Delivery Schedule:")
# print("License Number:", driver1_license, driver1_name, "has",
driver1_deliveries, "deliveries")
# print("License Number:", driver2_license, driver2_name, "has",
driver2_deliveries, "deliveries")
# print("License Number:", driver3_license, driver3_name, "has",
driver3_deliveries, "deliveries")


                            # # QUESTION 10

# # Initialize the lists to store the account information
# names = ["Ali Shahbaz", "Aragorn", "Legolas"]
# emails = ["alishahbaz@email.com", "aragorn@email.com", "legolas@email.com"]
# passwords = ["password1", "password2", "password3"]
# newsletters = [True, False, True]

# # Get user's email address and password
# email = input("Enter your email address: ")
# password = input("Enter your password: ")

# # Validate the user's email address and password
# if email in emails:
#     index = emails.index(email)
#     if passwords[index] == password:
#         # User is valid, allow them to update their account settings
#         print("Welcome, " + names[index])

#         # Update personal information
#         new_name = input("Enter your new name: ")
#         if new_name != "":
#             names[index] = new_name

#         new_email = input("Enter your new email address: ")
#         if new_email != "":
#                 emails[index] = new_email

#         new_password = input("Enter your new password: ")
#         if new_password != "":
#             passwords[index] = new_password
```

```
#         # Update communication preferences
#         newsletter_choice = input("Do you want to subscribe to our newsletter?
(y/n): ")
#         if newsletter_choice.lower() == "y":
#             newsletters[index] = True
#         elif newsletter_choice.lower() == "n":
#             newsletters[index] = False

#         print("Account settings updated successfully!")
#     else:
#         print("Invalid password. Please try again.")
# else:
#     print("Invalid email address. Please try again.")
```