```
# Author: Madhav Goyal
# Date: 19-10-2025
# Project Title: Daily Calorie Tracker CLI

import datetime   # Used for Bonus Task (timestamp in file saving)

print("================================================")
print("   WELCOME TO THE DAILY CALORIE TRACKER APP   ")
print("================================================")
print("This tool allows you to enter your meals and calorie intake,")
print("calculate total and average calories, compare with a limit,")
print("and optionally save your session in a log file.\n")

# -------------------- Task 2: Input & Data Collection --------------------

# Asking number of meals
num_meals = int(input("How many meals do you want to enter? "))

# Creating empty lists
meals = []
calories = []

# Loop to collect meal names and calories
for i in range(num_meals):
    meal_name = input(f"Enter meal {i+1} name: ")
    calorie_amount = float(input(f"Enter calories for {meal_name}: "))  # Convert
to float

    meals.append(meal_name)
    calories.append(calorie_amount)

# -------------------- Task 3: Calorie Calculations --------------------

total_calories = sum(calories)                  # sum() used
average_calories = total_calories / num_meals   # arithmetic operation

daily_limit = float(input("\nEnter your daily calorie limit: "))  # User input for
comparison

# -------------------- Task 4: Warning System --------------------

if total_calories > daily_limit:                        # Comparison operator >
    status_message = "⚠ Warning: You have exceeded your daily calorie limit!"
else:
    status_message = "☑ Great! You are within your daily calorie limit."

# -------------------- Task 5: Neatly Formatted Output --------------------

print("\n------------------- Daily Calorie Summary -------------------")
print(f"{'Meal Name':<20}\t{'Calories'}")
```

```
print("-------------------------------------------------------------")

for meal, cal in zip(meals, calories):    # zip() used to display paired data
    print(f"{meal:<20}\t{cal}")

print("-------------------------------------------------------------")
print(f"Total Calories:\t\t{total_calories}")
print(f"Average Calories/Meal:\t{average_calories:.2f}")
print("-------------------------------------------------------------")
print(status_message)

# -------------------- Task 6 (Bonus): Save Session Log to File
--------------------

save_choice = input("\nDo you want to save this session to a file? (yes/no):
").lower()

if save_choice == "yes":
    current_time = datetime.datetime.now()  # Get timestamp
    filename = "calorie_log.txt"

    with open(filename, "a") as file:       # Append mode
        file.write("\n=================================================\n")
        file.write(f"Session Date & Time: {current_time}\n")
        file.write("Meal Name\t\tCalories\n")

        for meal, cal in zip(meals, calories):
            file.write(f"{meal:<20}\t{cal}\n")

        file.write("-------------------------------------------------\n")
        file.write(f"Total Calories: {total_calories}\n")
        file.write(f"Average Calories/Meal: {average_calories:.2f}\n")
        file.write(f"Status: {status_message}\n")
        file.write("=================================================\n")

    print(f"☑ Session saved successfully in '{filename}'.")
```