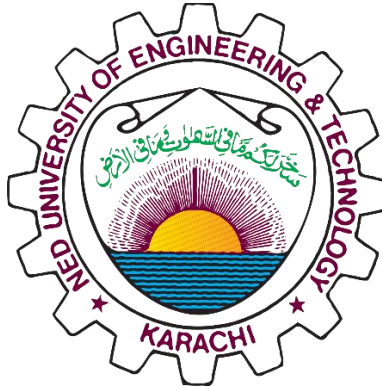


NED UNIVERSITY OF ENGINEERING & TECHNOLOGY, KARACHI



**DEPARTMENT:
COMPUTER SCIENCE & INFORMATION TECHNOLOGY**

**CT-175 PROGRAMMING FUNDAMENTALS
COMPLEX COMPUTING PROBLEM (CCP) REPORT**
<https://github.com/Saad0095/Weekly-Health-Tracker-Advisor>

**PROJECT NAME:
WEEKLY HEALTH TRACKER & ADVISOR**

**SUBMITTED TO:
MISS AMNA**

SUBMITTED BY:
MUHAMMAD SAAD BIN KHALID (CT-25245) (*GROUP LEAD*)
ABDULLAH BIN WAQAR (CT-25246)
HASSAAN AHMED KHAN (CT-25248)
MUHAMMAD SHIBLEE SHAMIM (CT-25233)
MUHAMMAD MUJEEB REHMAN (CT-24200)

SECTION E

Problem Statement:

The user is committed to track their overall health and wellness. To accomplish this, they want to track their core daily habits such as weekly water intake, sleep duration, and workout time. By monitoring these habits, user intends to seek personalized, specialist-like advice and insights that

Introduction & Scope:

The **Weekly Health Tracker & Advisor** is an interactive console-based and web-based application designed to help students track their weekly health. The program collects input on daily water intake, sleep hours, and hours workout from user, calculates averages, and provides advice based on their report.

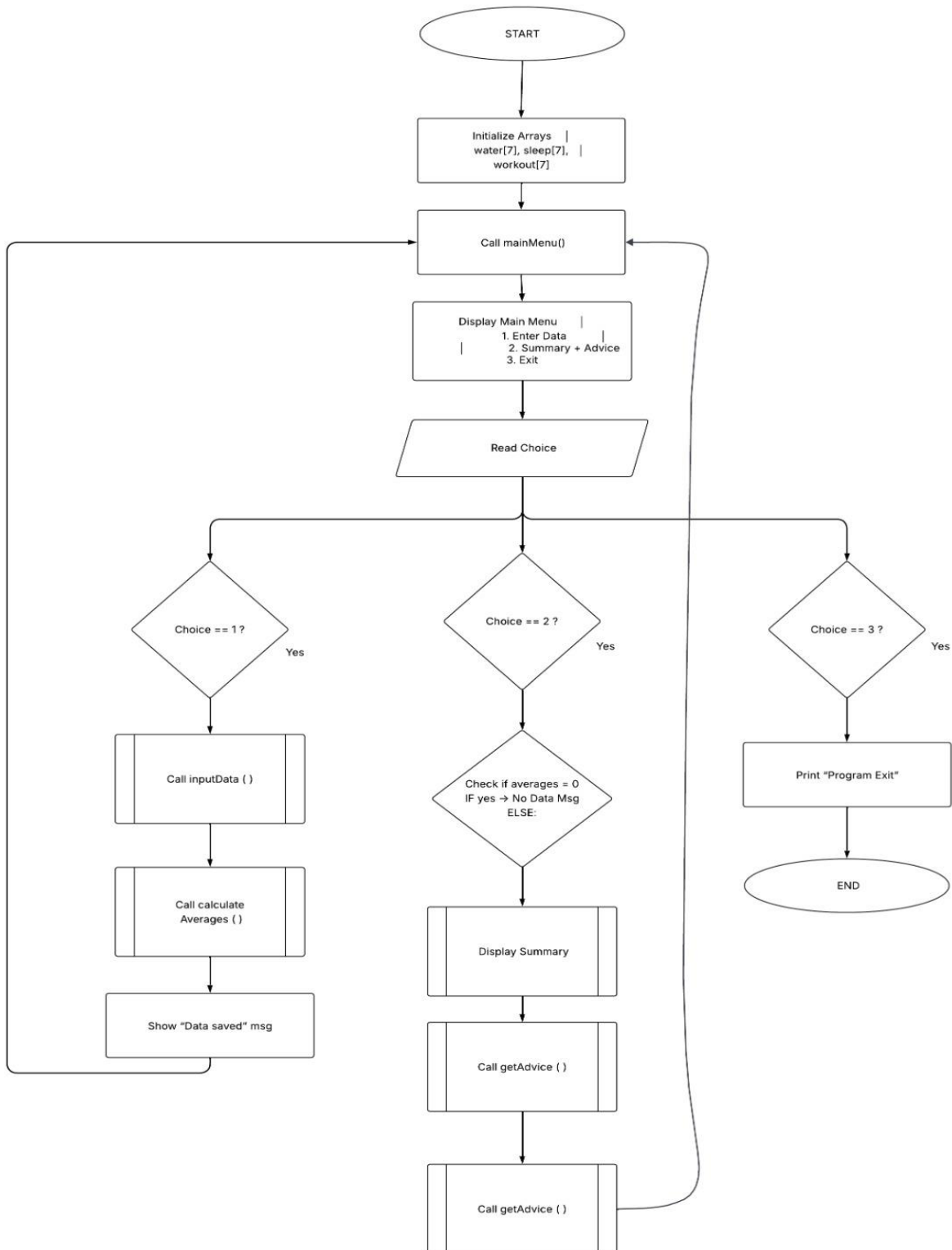
Core Programming Principles:

Our project revolves around the fundamental programming concepts of **arrays, loops, functions, pointers, and conditional statements**. The project's **modular design**, along with **core logical reason** and **structured programming**, enhances the overall efficiency.

Scope:

- Record daily water intake, sleep, and workout for a week.
- Calculate weekly averages and give health advice.
- Display summary in console and UI using charts via Streamlit.
- Give advice based on tracking

Block Diagram of Functions & Methodology:



Algorithm:

Start

1. Initialize Arrays and Constants:

- Define constant DAYS = 7 (for a week).
- Declare arrays:
 - water[DAYS] stores daily water intake.
 - sleep[DAYS] stores daily sleep hours.
 - workout[DAYS] stores daily workout hours.

2. Call Main Menu Function:

- Pass water, sleep, and workout arrays to mainMenu().

Main Menu Function:

3. Display Menu Options:

1. Enter/Update Weekly Data
2. View Summary & Advice
3. Exit Program

4. Input User Choice (choice).

5. Decision Based on Choice:

- If choice = 1:
 - Call inputData() to take daily inputs.
 - Call calculateAverages() to compute weekly averages.
 - Display confirmation message: Data input completed.
- If choice = 2:
 - Check if data exists:
 - If averages are 0, prompt user to enter data first.
 - Else:
 - Call displaySummary() to show daily and average data.
 - Call getAdvice() to provide health recommendations based on averages.
- If choice = 3:
 - Exit program with message Program Exited successfully.
- Else:
 - Display Invalid choice and prompt again.

6. Repeat Menu until user chooses option 3.

Input Data Function (inputData)

7. For Each Day (1 to 7):

- Water Intake:
 - Input water in liters.
 - Validate: $0 \leq \text{water} \leq 6$.
 - If invalid, prompt again.
- Sleep Hours:
 - Input sleep hours.
 - Validate: $0 \leq \text{sleep} \leq 24$.
 - If invalid, prompt again.
- Workout Hours:

- Input workout hours.
- Validate: $0 \leq \text{workout} \leq (24 - \text{sleep})$ to avoid exceeding total hours.
- If invalid, prompt again.

Calculate Averages Function (calculateAverages)

8. For Each Day (1 to 7):

- Sum up water, sleep, and workout arrays.

9. Compute Weekly Average:

- $\text{avgWater} = \text{totalWater} / \text{DAYS}$
- $\text{avgSleep} = \text{totalSleep} / \text{DAYS}$
- $\text{avgWorkout} = \text{totalWorkout} / \text{DAYS}$

Display Summary Function

10. Print Table:

- Day | Water (L/day) | Sleep (hrs/day) | Workout (hrs/day)

11. For Each Day:

- Print values of water[i], sleep[i], workout[i].

12. Print Weekly Averages:

- avgWater, avgSleep, avgWorkout.

Get Advice Function (getAdvice)

13. Compare Averages and Provide Suggestions:

- Water:
 - If $\text{avgWater} < 2.0$, suggest increasing water intake.
 - Else, print Water intake is good!
- Sleep:
 - If $\text{avgSleep} < 8.0$, suggest more sleep.
 - Else, print Sleep hours sufficient!
- Workout:
 - If $\text{avgWorkout} < 0.5$, suggest increasing workout.
 - Else, print Workout routine is good!

End

How This Health Tracker Works?

The App Has Two Parts:

1. C Program (The Brain)

- Handles all the calculations
- Stores your daily health data
- Gives basic health advice
- Works like a simple menu system

2. Streamlit Web App (The Display)

- Shows your data in pretty tables and graphs
- Lets you input data easily with sliders and buttons
- Makes it easy to see your weekly patterns at a glance

Complete Source Code:

○ C Program:

```
#include <stdio.h>

#define DAYS 7 // constant

void inputData(float water[], float sleep[], float workout[]);
void calculateAverages(float water[], float sleep[], float workout[],
                      float *avgWater, float *avgSleep, float *avgWorkout);
void getAdvice(float avgWater, float avgSleep, float avgWorkout);
void displaySummary(float water[], float sleep[], float workout[],
                   float avgWater, float avgSleep, float avgWorkout);
void mainMenu(float water[], float sleep[], float workout[]);

// ** Main Function **

int main() {
    float water[DAYS], sleep[DAYS], workout[DAYS];
    mainMenu(water, sleep, workout);
    return 0;
}

// ** Main Menu **

void mainMenu(float water[], float sleep[], float workout[]) {
    int choice;
    float avgWater = 0, avgSleep = 0, avgWorkout = 0;
    printf("\n\nWelcome to Weekly Health Tracker & Advisor!\n\n");
    do {
        printf("\n--- Main Menu ---\n");
        printf("1. Enter/Update Weekly Data\n");
        printf("2. View Summary & Advice\n");
        printf("3. Exit Program\n");
        printf("Enter your choice (1-3): ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                inputData(water, sleep, workout);
                calculateAverages(water, sleep, workout, &avgWater, &avgSleep,
                                &avgWorkout);
                printf("\n--- Data Input Completed! ---\n");
                printf("Select option 2 to view the results.\n");
                break;
```

```

case 2:
    if (avgWater == 0 && avgSleep == 0 && avgWorkout == 0) {
        printf("\nNo data found. Please select option 1 first.\n");
    } else {
        displaySummary(water, sleep, workout, avgWater, avgSleep, avgWorkout);
        getAdvice(avgWater, avgSleep, avgWorkout);
    }
    break;

case 3:
    printf("\nProgram Exited successfully!\n");
    break;

default:
    printf("\nInvalid choice. Please enter 1, 2, or 3.\n");
    break;
}
} while(choice != 3);
}

// ** Take inputs **

void inputData(float water[], float sleep[], float workout[]) {
    int i;
    printf("\nEnter your water intake per day (in litres): \n");
    for (i = 0; i < DAYS; i++) {
        printf("\nEnter water intake Day %d: ", i + 1);
        scanf("%f", &water[i]);
        if (water[i] < 0 || water[i] > 6) {
            printf("Water intake cannot be negative or more than 6 litres a day!\n");
            i--;
            continue;
        }
    }
}

printf("\nEnter hours you sleep per day (in hrs): \n");
for (i = 0; i < DAYS; i++) {
    printf("\nEnter sleep Day %d: ", i + 1);
    scanf("%f", &sleep[i]);
    if (sleep[i] < 0 || sleep[i] > 24) {
        printf("\n\nEnter hours within 0-24!!");
        i--;
        continue;
    }
}
}

```

```

printf("\nEnter your workout time (in hours): \n");
for (i = 0; i < DAYS; i++) {
    printf("\nEnter workout duration Day %d: ", i + 1);
    scanf("%f", &workout[i]);
    if (workout[i] < 0 || workout[i] > 24 || (24-sleep[i] < workout[i])) {
        printf("\n\nEnter hours within 0-%.0f!!", (24-sleep[i]));
        i--;
        continue;
    }
}
}

// ** Calculates averages **

void calculateAverages(float water[], float sleep[], float workout[],
                      float *avgWater, float *avgSleep, float *avgWorkout) {
    float sumWater = 0, sumSleep = 0, sumWorkout = 0;
    int i;
    for (i = 0; i < DAYS; i++) {
        sumWater += water[i];
        sumSleep += sleep[i];
        sumWorkout += workout[i];
    }
    *avgWater = sumWater / DAYS;
    *avgSleep = sumSleep / DAYS;
    *avgWorkout = sumWorkout / DAYS;
}

// ** Display weekly summary **
void displaySummary(float water[], float sleep[], float workout[],
                   float avgWater, float avgSleep, float avgWorkout) {
    int i;
    printf("\nDays\t\tWater (L/day)\tSleep (hrs/day)\tWorkout (hrs/day)");
    for (i = 0; i < DAYS; i++) {
        printf("\nDay %d:\t\t%.2f\t\t%.2f\t\t%.2f", i + 1, water[i], sleep[i],
              workout[i]);
    }
    printf("\n\nAverage:\t\t%.2f\t\t%.2f\t\t%.2f", avgWater, avgSleep,
          avgWorkout);
}

// ** Displays Advice **
void getAdvice(float avgWater, float avgSleep, float avgWorkout) {
    printf("\n\n\t\t\tHealth Advice");
    if (avgWater < 2.0)

```



```

    printf("\nIncrease your daily water intake to around 2 liters per day.");
else
    printf("\nYour water intake is good!");

if (avgSleep < 8.0)
    printf("\nTry to get at least 8 hours of sleep daily.");
else
    printf("\nYour sleep hours are sufficient!");

if (avgWorkout < 0.5)
    printf("\nIncrease workout time to at least 30 minutes per day.");
else
    printf("\nYour workout routine is good!");
printf("\n");
}

```

○ Streamlit (Python) Code:

```

import streamlit as st
import pandas as pd

# Initialization
if "water" not in st.session_state:
    st.session_state.water = [0.0] * 7
    st.session_state.sleep = [0.0] * 7
    st.session_state.workout = [0.0] * 7

st.set_page_config(page_title="Weekly Health Tracker", layout="wide")
st.title("🌿 Weekly Health Tracker & Advisor")

menu = ["Enter/Update Weekly Data", "View Summary & Advice"]
choice = st.sidebar.selectbox("Menu", menu)

# calculates averages
def calculate_averages():
    avg_water = sum(st.session_state.water) / 7
    avg_sleep = sum(st.session_state.sleep) / 7
    avg_workout = sum(st.session_state.workout) / 7
    return avg_water, avg_sleep, avg_workout

# generates advice
def get_advice(avg_water, avg_sleep, avg_workout):
    advice = []

    if avg_water < 2.0:

```

```

    advice.append("💧 try to drink around 2 liters of water daily.")
else:
    advice.append("💧 good job! your water intake is on track.")

if avg_sleep < 8.0:
    advice.append("🛌 aim for at least 8 hours of sleep per day.")
else:
    advice.append("🛌 your sleep routine looks healthy.")

if avg_workout < 0.5:
    advice.append("🏋️ try adding around 30 minutes of daily workout.")
else:
    advice.append("🏋️ your workout routine looks good.")

return advice

```

Chart/Graphs

```

def progress_bar(value, goal, label):
    pct = min(value / goal, 1.0)
    st.progress(pct)
    st.write(f'{label}: {value:.2f} / {goal}')

```

input validation

```

def has_data():
    return any(v > 0 for v in st.session_state.water + st.session_state.sleep +
st.session_state.workout)

```

** UI **

```

if choice == "Enter/Update Weekly Data":

```

```

    st.header("📝 Enter Your Weekly Data")

```

```

    for i in range(7):
        st.subheader(f'Day {i+1}')

```

```

        water = st.number_input(
            "Water intake (L)",
            min_value=0.0,
            max_value=6.0,
            value=st.session_state.water[i],
            step=0.1,
            key=f"water{i}"
        )

```

```

st.session_state.water[i] = water

sleep = st.number_input(
    "Sleep hours",
    min_value=0.0,
    max_value=24.0,
    value=st.session_state.sleep[i],
    step=0.5,
    key=f"sleep{i}"
)
st.session_state.sleep[i] = sleep

max_workout = 24 - sleep

workout = st.number_input(
    f"Workout hours (max {max_workout:.1f}h)",
    min_value=0.0,
    max_value=float(max_workout),
    value=st.session_state.workout[i]
    if st.session_state.workout[i] <= max_workout else 0.0,
    step=0.25,
    key=f"workout{i}"
)
st.session_state.workout[i] = workout

st.success("✅ weekly data updated!")

elif choice == "View Summary & Advice":

    if not has_data():
        st.warning("⚠️ no data found. please enter your weekly data first!")
    else:
        st.header("📊 Weekly Summary")

        df = pd.DataFrame({
            "Water (L/day)": st.session_state.water,
            "Sleep (hrs/day)": st.session_state.sleep,
            "Workout (hrs/day)": st.session_state.workout
        }, index=[f"Day {i+1}" for i in range(7)])

        st.table(df)

        avg_water, avg_sleep, avg_workout = calculate_averages()

        st.subheader("📈 Averages")

```

```
col1, col2, col3 = st.columns(3)

with col1:
    progress_bar(avg_water, 2.0, "Water")

with col2:
    progress_bar(avg_sleep, 8.0, "Sleep")

with col3:
    progress_bar(avg_workout, 0.5, "Workout")

st.subheader("💡 Health Advice")
for line in get_advice(avg_water, avg_sleep, avg_workout):
    st.write(line)

st.subheader("📊 Weekly Trends")
st.line_chart(df)
```

Sample Output:

C Program (Console):

```
Welcome to Weekly Health Tracker & Advisor!

--- Main Menu ---
1. Enter/Update Weekly Data
2. View Summary & Advice
3. Exit Program
Enter your choice (1-3): 1

Enter your water intake per day (in litres):

Enter water intake Day 1: 5
Enter water intake Day 2: 4
Enter water intake Day 3: 1
Enter water intake Day 4: 8
Enter water intake Day 5: 9
Enter water intake Day 6: 4
Enter water intake Day 7: 5
Enter hours you sleep per day (in hrs):

Enter sleep Day 1: 2
Enter sleep Day 2: 5
Enter sleep Day 3: 8
Enter sleep Day 4: 3
Enter sleep Day 5: 10
Enter sleep Day 6: 14
Enter sleep Day 7: 2

Enter your workout time (in hours):

Enter workout duration Day 1: 0.2
Enter workout duration Day 2: 0.3
Enter workout duration Day 3: 6
```

Enter workout duration Day 4: 2

Enter workout duration Day 5: 0.5

Enter workout duration Day 6: 2

Enter workout duration Day 7: 1

--- Data Input Completed! ---

Select option 2 to view the results.

--- Main Menu ---

1. Enter/Update Weekly Data

2. View Summary & Advice

3. Exit Program

Enter your choice (1-3): 2

Days	Water (L/day)	Sleep (hrs/day)	Workout (hrs/day)
Day 1:	5.00	2.00	0.20
Day 2:	4.00	5.00	0.30
Day 3:	1.00	8.00	6.00
Day 4:	8.00	3.00	2.00
Day 5:	9.00	10.00	0.50
Day 6:	4.00	14.00	2.00
Day 7:	5.00	2.00	1.00
Average:	5.14	6.29	1.71

Health Advice

Your water intake is good!

Try to get at least 8 hours of sleep daily.

Your workout routine is good!

Streamlit UI (Web Interface):

<https://healthtrackerui.streamlit.app/>

Future Add-Ons & Limitations:

Future Add-Ons:

- Track multi-user health progress with features of creating personalized user account.
- BGMI tracking as additional feature.
- File storage to save weekly logs.
- Track multi-week health progress.
- Export weekly reports as PDF.

Limitations:

- No long-term memory or history tracking.
- Does not perform advanced calculations like sleep score, health rating scale in accordance with age and gender of the user.

Contribution:

NAME	ROLL NUMBER	TASK PERFORMED
Muhammad Saad Bin Khalid	CT-25245	Project Planning, UI and Documentation
Abdullah Bin Waqar	CT-25246	Coding and Testing
Hassaan Ahmed Khan	CT-25248	Coding and Upgradation
Muhammad Shiblee Shamim	CT-25233	Module Coding and Documentation
Mujeeb-Ur-Rehman	CT-24200	Module Coding