CT-175
PROGRAMMING
FUNDAMENTALS

FITNESS AND BODYBUILDING ASSISTANT

PF COURSE PROJECT (CCP)

BUSHRA KHAN CT-215
HURRIYA ALI CT-220
SAEEDIA HASHMI CT-221
MUHAMMAD MAAD-E-KARAB CT-239

PROBLEM STATEMENT:

Staying healthy and physically fit is more important than ever, but keeping track of progress can be difficult without the right tools. This project focuses on creating a Fitness & Bodybuilding Assistant — a simple digital companion designed to guide users throughout their fitness journey.

The system uses key personal details such as weight, target weight, height, age, sex, daily meal count, and the user's fitness goal (muscle gain or weight loss). With this information, the program calculates daily calorie needs, protein requirements, per-meal calorie distribution, and an estimated timeline for reaching the target weight.

This project results in an easy-to-use C console application that handles all essential calculations, provides personalized recommendations, and saves progress for future review. The goal is to give users a practical, data-driven way to monitor their progress and make smarter choices about their fitness routines.

Objectives:

Build a reliable C console program that lets users enter basic personal details along with a fitness goal (either weight gain or weight loss).

- Use a standard formula (Mifflin–St Jeor) to calculate their BMR, then estimate their TDEE based on activity level.
- Give clear calorie and protein targets tailored to the user's goal, along with a simple breakdown for each meal.
- Estimate how long it will take for the user to reach their goal using realistic monthly change rates (about 1 kg/month for gaining weight and 4 kg/month for losing it).
- Save each result to a file called *fitness_data.txt* and automatically show any previously saved progress when the program starts.
- Handle input safely: check for valid meal counts, protect file operations, and prevent unrealistic calorie outputs.
- Present everything with clean, easy-to-follow console messages so users can apply the recommendations in real life.

PROJECT DESCRIPTION:

The application takes the user's physical data and fitness goals as input. Based on these values, it performs several calculations:

- Time estimation to gain or lose the required weight
- Daily caloric requirement based on BMR and TDEE

- Protein intake recommendations suitable for the chosen goal
- Calories per meal, depending on the number of meals per day

After performing these calculations, the program displays the results clearly and saves a detailed record in a file (fitness_data.txt) for future reference.

The core objective is to provide a simple, fast, and effective tool that helps users stay disciplined, monitor their progress, and make better decisions regarding diet and fitness. By automating essential calculations and saving past records, the program helps users stay motivated and committed to achieving their fitness goals.

Concepts Covered in the Course and Their Justification:

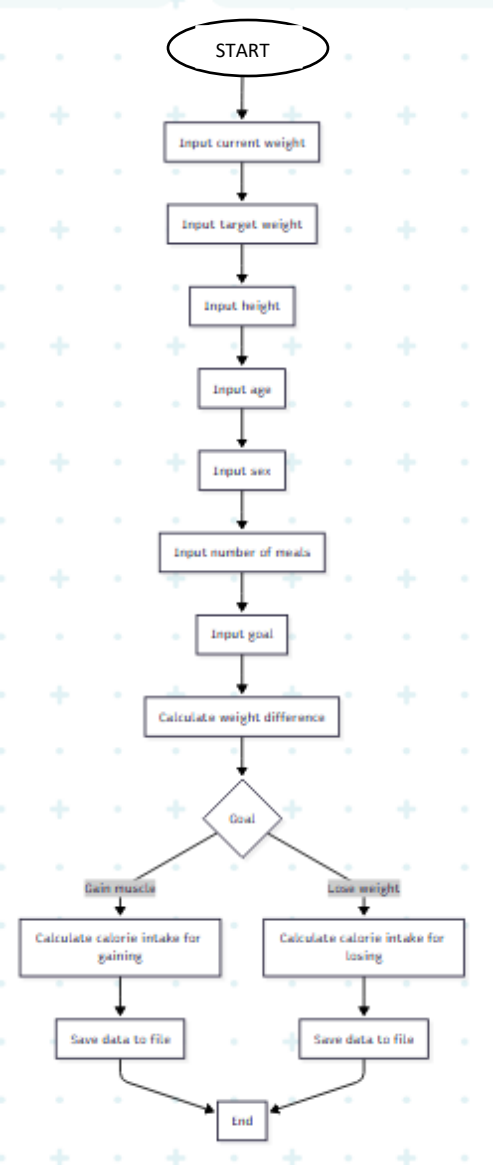
This project incorporates several core programming concepts introduced in the course, each serving a specific purpose in the application:

1. **Variables and Data Types**
   These are used to store essential information such as height, weight, calorie values, protein needs, and time estimates. Using appropriate numeric data types ensures that real-world values are handled accurately throughout the program.
2. **Conditional Statements (if–else)**
   Conditional logic drives important decisions within the application, including:
   - Determining whether the user's goal is weight gain or weight loss
   - Checking if the user has already achieved their target
   - Applying gender-specific BMR formulas and nutritional rules
     These conditions help the program adapt to user-specific scenarios.
3. **Loops**
   Loops are used to read previously saved progress from the file, line by line. This allows the program to display a user's historical records automatically each time it starts, improving usability and continuity.
4. **File Handling (fopen, fgets, fprintf, fclose)**
   The application relies on file handling to store and retrieve progress data:
   - fopen() opens the file for reading or writing
   - fgets() reads past entries
   - fprintf() writes formatted results
   - fclose() safely closes the file and frees resources
     This makes long-term tracking possible and gives the program meaningful practicality.
5. **Input/Output Operations**
   Functions like printf() and scanf() enable smooth interaction between the system and the user, making it easy to enter information and view results in a clear, organized format.

ALGORITHM

1. Start
2. Display the program title.
3. Try to open *fitness_data.txt*.
    a. If it exists, read and show previous progress.
    b. Close the file.
4. Ask the user for:
    a. Current weight
    b. Target weight
    c. Height, age
    d. Sex (M/F)
    e. Meals per day (3–5)
    f. Fitness goal (G/L)
5. Fix meals:
    a. If meals < 3 → set to 3
    b. If meals > 5 → set to 5
6. Calculate weight difference:
7. diff = targetWeight – currentWeight
8. Calculate BMR based on sex.
9. Calculate TDEE = BMR × 1.3.
10. If Goal = Muscle Gain (G):
11. If diff ≤ 0,display “Target reached” and stop.
12. Duration = diff ÷ 1 (1 kg/month).
13. Calculate daily calories (TDEE + 300/400) and limit to 3500.
14. Calculate protein needs.
15. Calories per meal = calories ÷ meals.
16. If Goal = Weight Loss (L):
17. If diff ≥ 0, display “Target reached” and stop.
18. diff = –diff.
19. Duration = diff ÷ 4 (4 kg/month).
20. Calculate daily calories (TDEE – 500/700) and minimum 1200.
21. Calculate protein needs.
22. Calories per meal = calories ÷ meals.
23. Open *fitness_data.txt* in write mode.
24. Save all results.
25. Close the file.
26. Display “Progress saved.”
27. End

FLOWCHART


START
Input current weight
Input target weight
Input height
Input age
Input sex
Input number of meals
Input goal
Calculate weight difference
Goal
Gain muscle
Lose weight
Calculate calorie intake for gaining
Calculate calorie intake for losing
Save data to file
Save data to file
End

CODE

```c
#include <stdio.h>
#include <stdlib.h>

int main() {
    float currentWeight, targetWeight, height, age;
    char sex, goal;
    int durationMonths;
    float calorieIntake, proteinIntake;
    float caloriesPerMeal;
    int meals;
    float diff;
    float BMR, TDEE;
    float activityMultiplier;

    FILE *file;

    printf("Fitness & Bodybuilding Assistant (Realistic)\n\n");

    file = fopen("fitness_data.txt", "r");
    if (file != NULL) {
        printf("Previous progress found:\n");
        char line[200];
        while (fgets(line, sizeof(line), file)) {
            printf("%s", line);
        }
        fclose(file);
        printf("\n----------------------------------------\n\n");
    }

    printf("Enter your current weight (kg): ");
    scanf("%f", &currentWeight);

    printf("Enter your target weight (kg): ");
    scanf("%f", &targetWeight);

    printf("Enter your height (cm): ");
    scanf("%f", &height);

    printf("Enter your age: ");
    scanf("%f", &age);

    printf("Enter your sex (M/F): ");
    scanf(" %c", &sex);
```

```c
    printf("How many meals do you eat per day (3-5): ");
    scanf("%d", &meals);
    if (meals < 3) meals = 3;
    if (meals > 5) meals = 5;

    printf("Do you want to gain muscle (G) or lose weight (L)? Enter G/L: ");
    scanf(" %c", &goal);

    diff = targetWeight - currentWeight;

    file = fopen("fitness_data.txt", "w");
    if (file == NULL) {
        printf("Error saving file!\n");
        return 1;
    }

    fprintf(file, "Fitness Progress Record\n\n");
    fprintf(file, "Current Weight: %.1f kg\n", currentWeight);
    fprintf(file, "Target Weight: %.1f kg\n", targetWeight);

    if (sex == 'M' || sex == 'm')
        BMR = 10 * currentWeight + 6.25 * height - 5 * age + 5;
    else
        BMR = 10 * currentWeight + 6.25 * height - 5 * age - 161;

    activityMultiplier = 1.3;
    TDEE = BMR * activityMultiplier;

    if (goal == 'G' || goal == 'g') {
        if (diff <= 0) {
            printf("\nYou already reached or exceeded your target.\n");
            fprintf(file, "Goal: Muscle Gain - Already achieved.\n");
            fclose(file);
            return 0;
        }

        durationMonths = (int)(diff / 1.5);

        if (sex == 'M' || sex == 'm')
            calorieIntake = TDEE + 400;
        else
            calorieIntake = TDEE + 300;
```

```c
    if (calorieIntake > 3500) calorieIntake = 3500;

    if (sex == 'M' || sex == 'm')
        proteinIntake = targetWeight * 2.0;
    else
        proteinIntake = targetWeight * 1.8;

    caloriesPerMeal = calorieIntake / meals;

    printf("\nGoal: Muscle Gain\n");
    printf("Weight Gain Needed: %.1f kg\n", diff);
    printf("Estimated Time: %d months\n", durationMonths);
    printf("Daily Calories: %.0f kcal\n", calorieIntake);
    printf("Protein: %.1f g/day\n", proteinIntake);
    printf("Calories per meal (%d meals): %.0f kcal\n\n", meals, caloriesPerMeal);

    fprintf(file, "Goal: Muscle Gain\nWeight Gain Needed: %.1f kg\nEstimated Time: %d months\nDaily Calories: %.0f kcal\nProtein: %.1f g/day\nCalories per Meal: %.0f kcal\n",
     diff, durationMonths, calorieIntake, proteinIntake, caloriesPerMeal);
}

else if (goal == 'L' || goal == 'l') {
    if (diff >= 0) {
        printf("\nYou are already below or at your target.\n");
        fprintf(file, "Goal: Weight Loss - Already reached.\n");
        fclose(file);
        return 0;
    }

    diff = -diff;
    durationMonths = (int)(diff / 3.0);

    TDEE = BMR * 1.3;

    if (sex == 'M' || sex == 'm')
        calorieIntake = TDEE - 700;
    else
        calorieIntake = TDEE - 500;

    if (calorieIntake < 1200) calorieIntake = 1200;

    if (sex == 'M' || sex == 'm')
        proteinIntake = currentWeight * 1.8;
```



```c
    else
        proteinIntake = currentWeight * 1.6;

    caloriesPerMeal = calorieIntake / meals;

    printf("\nGoal: Weight Loss\n");
    printf("Weight Loss Needed: %.1f kg\n", diff);
    printf("Estimated Time: %d months\n", durationMonths);
    printf("Daily Calories: %.0f kcal\n", calorieIntake);
    printf("Protein: %.1f g/day\n", proteinIntake);
    printf("Calories per meal (%d meals): %.0f kcal\n\n", meals, caloriesPerMeal);

    fprintf(file, "Goal: Weight Loss\nWeight Loss Needed: %.1f kg\nEstimated Time: %d months\nDaily Calories: %.0f kcal\nProtein: %.1f g/day\nCalories per Meal: %.0f kcal\n",
     diff, durationMonths, calorieIntake, proteinIntake, caloriesPerMeal);
}

else {
    printf("\nInvalid choice. Enter G or L.\n");
    fclose(file);
    return 0;
}

fprintf(file, "\nStay consistent and disciplined.\n");
fclose(file);

printf("\nProgress saved in 'fitness_data.txt'.\n");
return 0;
}
```

OUTPUT

```
Enter your current weight (kg): 60
Enter your target weight (kg): 50
Enter your height (cm): 164
Enter your age: 19
Enter your sex (M/F): F
How many meals do you eat per day (3-5): 3
Do you want to gain muscle (G) or lose weight (L)? Enter G/L: L

Goal: Weight Loss
Weight Loss Needed: 10.0 kg
Estimated Time: 3 months
Daily Calories: 1280 kcal
Protein: 96.0 g/day
Calories per meal (3 meals): 427 kcal


Progress saved in 'fitness_data.txt'.

--------------------------------
Process exited after 31.83 seconds with return value 0
Press any key to continue . . .
```

```
Enter your current weight (kg): 80
Enter your target weight (kg): 100
Enter your height (cm): 167
Enter your age: 35
Enter your sex (M/F): M
How many meals do you eat per day (3-5): 4
Do you want to gain muscle (G) or lose weight (L)? Enter G/L: G

Goal: Muscle Gain
Weight Gain Needed: 20.0 kg
Estimated Time: 13 months
Daily Calories: 2576 kcal
Protein: 200.0 g/day
Calories per meal (4 meals): 644 kcal


Progress saved in 'fitness_data.txt'.

--------------------------------
Process exited after 17.84 seconds with return value 0
Press any key to continue . . .
```

Limitations

1. **Accuracy of User Input**
   The program's calculations depend entirely on the information the user provides. Any incorrect or unrealistic inputs—such as wrong weight, age, or activity level—will lead to inaccurate results.
2. **Not a Medical Tool**
   This application offers general fitness guidance, not medical advice. It cannot diagnose health issues or replace professional consultation. Users with medical conditions should seek guidance from certified experts.
3. **Limited Metrics**
   The system focuses on a small set of fitness indicators:
   - Calories
   - Protein intake
   - Weight changes

     It does not measure other important factors like body fat percentage, hydration, muscle mass, or cardiovascular fitness.
4. **Dependence on User Consistency**
   The tool is only effective if users regularly input their data and follow the provided recommendations. Without consistent engagement, progress tracking becomes less meaningful.
5. **Generalized Calculations**
   The program uses standard formulas and assumptions to estimate calories, protein needs, and progress timelines. In reality, factors such as individual metabolism, training intensity, hormone levels, and body composition can influence actual results.

Future Enhancements

Looking ahead, several features could significantly improve the functionality and user experience of the Fitness & Bodybuilding Assistant:

- **Multi-user Accounts and Login System**
  Allow multiple users to save and access their own personalized fitness data.
- **Graphical Progress Tracking**
  Add visual charts that show weekly or monthly trends, making progress easier to understand at a glance.
- **BMI and Body-Fat Calculations**
  Include additional health metrics to give users a more complete picture of their physical condition.
- **Customized Diet and Workout Plans**
  Provide tailored meal suggestions and exercise routines based on user goals and fitness levels.
- **More Precise Activity Level Options**
  Offer adjustable activity categories for better accuracy in TDEE calculations.
- **User-Friendly Interface or Mobile App**
  Transition from a console application to a GUI or mobile version for a smoother, more modern experience.
- **Exportable Progress Reports**
  Allow users to download their progress in formats like PDF or Excel for sharing or long-term tracking.

Conclusion

The Fitness & Bodybuilding Assistant is a simple, beginner-friendly console application that brings together calorie calculations, protein estimates, and progress tracking into one convenient tool. By providing clear, real-time guidance, it helps users understand exactly what they need to do to reach their fitness goals while staying consistent and motivated.

Acting as a digital fitness companion, the program supports users in making informed decisions about their diet, training, and overall lifestyle. Whether the goal is weight loss, muscle gain, or general health improvement, the application offers practical, science-based recommendations tailored to the user's needs.

Ultimately, the Fitness & Bodybuilding Assistant empowers users to take control of their fitness journey, track their progress with confidence, and stay committed to achieving healthier, long-term habits.

MEMBERS CONTRIBUTION:
CODING BY:
Saeedia Hashmi & Maad-e-Karab
REPORT BY:
Bushra Khan & Hurriya Ali