RECIPE RECOMMANDATION SYSTEM

Niya Justine

Roll no – 61

Programming in C

Date – 15/07/2024

**Introduction**

The C programming language is used to develop a system that recommends recipes based on the user preferences with the available ingredients. During the development of such a system we use structure, functions etc. A menu driven program will be suitable for developing such a user-friendly system.

Problem statement: **To create a system that recommends recipes based on user preferences, dietary restrictions and available ingredients.**

The objective of the project is to develop a recipe recommendation system that recommends the recipes to the user based on the requirements of the users. Such a system will be useful for the users to get knowledge of the recipes to be used in case of diets or other nutritional controls they are going through. The system will develop in such a way that the users can add recipes they know such that the system will recommend recipes based on the user preferences and recipes added to the system. It is also developed in such a way that the dietary requirements can also be added along with the recipes we add. The system asks for the available ingredients with which it will provide the recipe which can be prepared.

**System Requirements**

The system requirements for the development of recipe recommendation system includes both hardware and software requirements.

Software Requirements:

1. Integrated Development Environment (IDE) or Text Editors

E.g. Visual Studio Code

2. C Compiler

E.g. GCC (GNU Compiler Collection)

3. Debugger

E.g. Visual Studio Debugger

4. Libraries

E.g. Standard C library etc

5. Operating System

Hardware Requirements:

1. Processor (CPU)

A multi-core processor (intel Core i5/i7) is sufficient.

1. Memory (RAM)
2. Storage
3. Display
4. Input Devices
5. Power Supply
6. Peripheral Devices
7. Networking

**Design and Development**

The logic behind the program is to develop a menu driven program which takes input from user according to the need of the user. The menu contains various options like add recipe, recommend recipe and exit. During the addition of recipes, the dietary restrictions (if any) are also collected from the user. Separate functions are designed to perform each task individually. At first a structure is defined to group various elements in program with different data type together. The user at first adds recipes along with the dietary restrictions (if any) by using the add recipe function in which we mention the ingredients required for its preparation. The program also collects the available ingredients from the user, during the execution of the program to recommend recipes. Based on the available ingredients the program compares it with the ingredients of recipes already added to the program memory by using the add recipe function. If the ingredients match, then the corresponding recipe from the program memory is recommended to the user. It is possible to exit from the program by choosing the option Exit from the menu.

**Pseudocode:**

Define constants:

MAX\_RECIPES = 10

MAX\_INGREDIENTS = 10

MAX\_NAME\_LEN = 50

Declare structure Recipe with fields:

name, ingredients, ingredients\_count, dietary\_restriction

Declare global variables:

recipes[MAX\_RECIPES], recipe\_count = 0

Function add\_recipe:

If recipe\_count >= MAX\_RECIPES: Print "Limit reached", Return

Input new\_recipe fields

Store new\_recipe in recipes[recipe\_count]

Increment recipe\_count

Function recommend\_recipe:

Input available\_count

Input available\_ingredients[available\_count]

For each recipe in recipes:

Calculate match\_count

If match\_count equals recipe's ingredients\_count: Print recipe.name

Main function:

Do:

Print menu, Input option

If option == 1: Call add\_recipe

Else if option == 2: Call recommend\_recipe

Else if option == 3: Print "Exiting", Break

Else: Print "Invalid choice"

Input choice

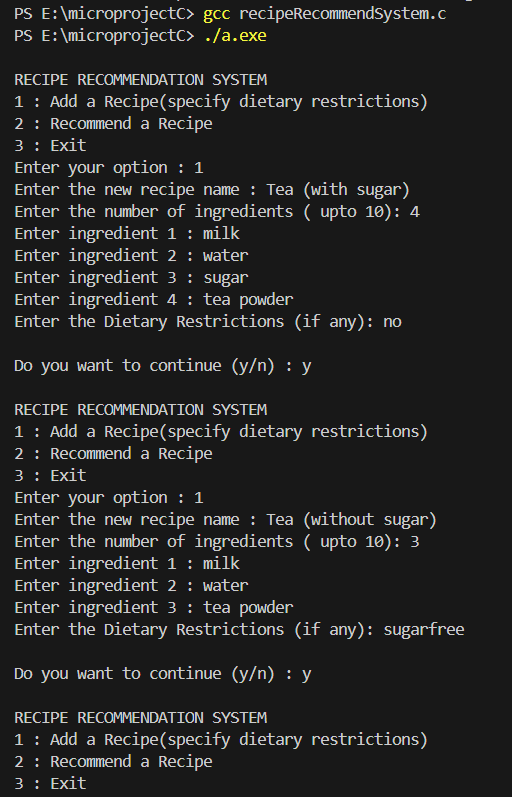
While choice is 'Y' or 'y'

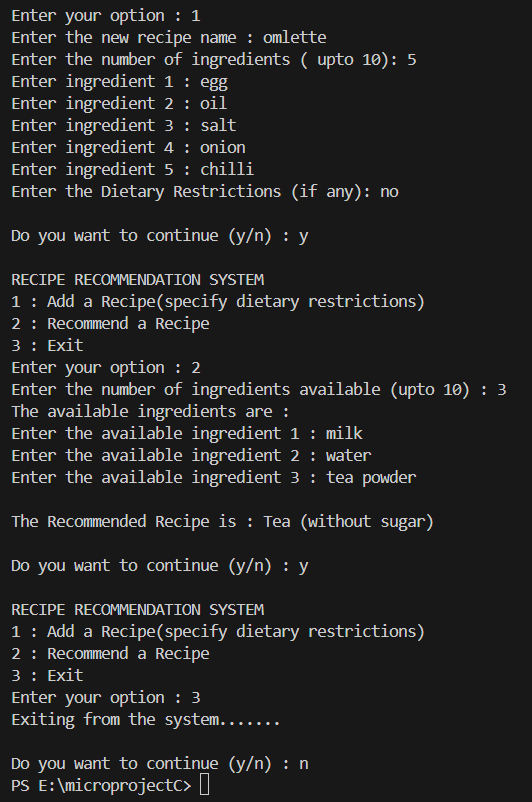
Return 0

**Testing and Results**

The program has various test cases like adding the recipes, recommendation of recipes based on available ingredients, asking for dietary restrictions and exiting from the program.

We have tested various options included in the menu :





While choosing the first option we can enter the ingredients needed for the recipe, then add the dietary restrictions (if any) and then specify the name of the recipe.

While choosing the second option we were asked for the ingredients available with us. Based on this information a recipe is recommended from the data already been collected from the user during the addition of recipes.

While choosing the third option we will exit from the function.

**Conclusion**

The project has been built to make the act of cooking easy by providing the user with recipe ideas. We have tried to provide the user with various options which make the user add the new recipes to the program such that they can be saved to the program and recommend recipes based on the ingredients available. Such a system will be very useful in the kitchen, which encourages the user to prepare various dishes with minimal ingredients. Various dishes with their dietary benefits can be understood by the user.

The scope of the program can be enhanced to such a level by adding the various options which specify the amount of nutrients needed like carbohydrates, proteins, water, vitamins, etc. The quantity of each ingredient to be added, the way of cooking by specifying the steps can also include to make the program more user friendly and useful.

**Appendices**

**Appendix A: Program Structure and Flow**

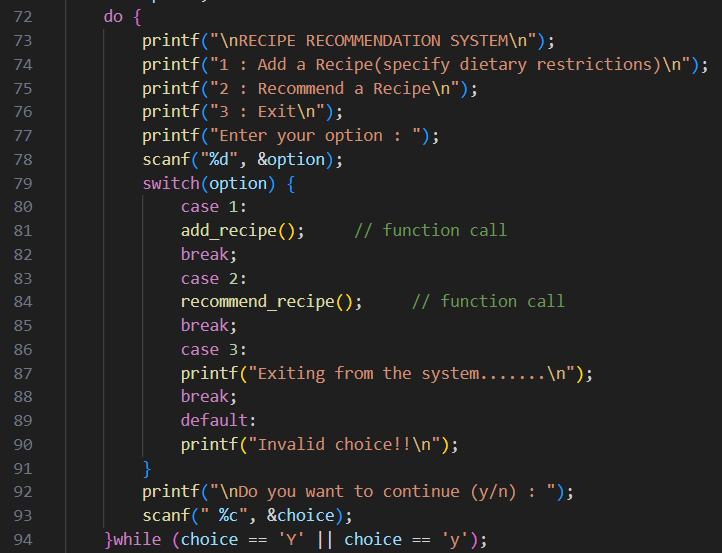
Program structure:

****

A computer screen with text and images

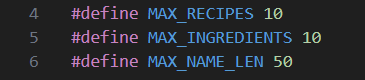
Description automatically generated

Flow of Execution:

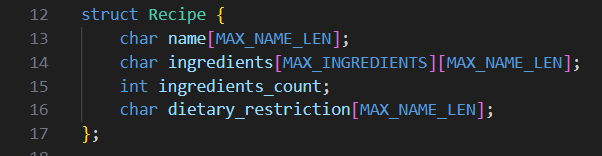


**Appendix-B: Constants and Data Structures**

Constants:

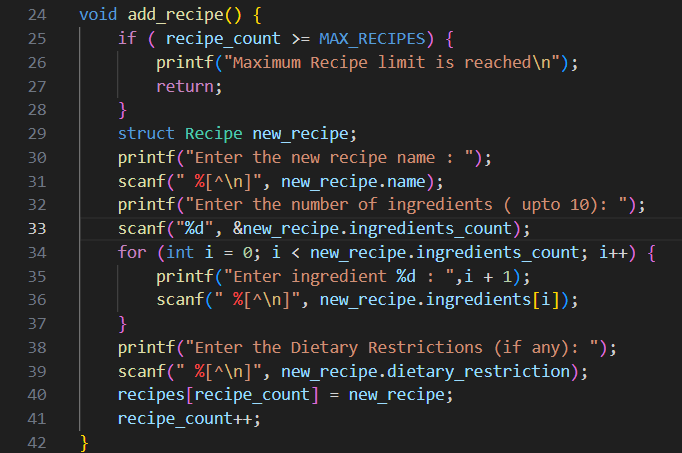


Data Structures:



**Appendix-C: Function Definitions**

add\_recipe Function:



recommend\_recipe Function:



**Appendix-D: Main Function**

