**COURSE TITLE: PF LAB**

**INSTRUCTOR NAME: SIR ADEEL KHALID**

**SUBMITTED BY:**

**RAMESA SOHAIL FA24B1-CS-004**

**YESHRA NOOR FA24B1-CS-047**

**RAFIA MARYAM FA24B1-CS-060**

**BCS1-A**

**Report on Math Problem Game Code**

# Problem

The program is a simple math game where the user solves random math problems. It keeps track of the user's score and saves their details in a file. The user can play as many times as they want.

# Introduction

This program, written in C, demonstrates basic programming concepts such as:

* Structuring code with functions
* Handling user input/output
* Random number generation
* File handling
* Implementing a scoring system

The game provides an engaging way to interact with programming logic while enhancing problem-solving skills.

# Explanation

## Code Structure

The program is organized into several components:

* *User Information Structure:*
  + The struct UserInfo holds the user's name, age, and score.
* *Functions:*
  + getUserInfo(): Prompts the user to enter their name and age.
  + saveUserInfo(): Saves the user's name, age, and final score in a file named user\_info.txt.
  + generateMathProblem(): Creates random math problems, checks the user's answer, and returns whether the answer is correct.
* *Main Function:*
  + Orchestrates the flow by using these functions to manage user data, generate math problems, and handle replay logic.

## Detailed Process

* *Initialization:*
  + Random numbers are initialized using srand(time(NULL)) to ensure randomness in math problem generation.
* *Collecting User Information:*
  + getUserInfo() gathers the user's name and age via console input.
* *Game Loop:*
  + The program generates math problems using generateMathProblem().
  + Supported operations include addition, subtraction, multiplication, and division.
  + The user answers each question, and their score increments if correct.
  + The user can choose to play multiple rounds.
* *Saving Results:*
  + After the game ends, the user’s name, age, and final score are saved to a file for record-keeping.

## Key Features

* **Dynamic Questions**: Random numbers and operators ensure each problem is unique.
* **Replay Option**: Users can play multiple rounds.
* **Score Tracking**: Keeps count of correct answers.
* **Data Persistence**: User details and scores are saved to a file.

# Result

* *User Interaction:* 
  + The program successfully engages the user by generating random math problems and evaluating their responses.
* *Data Recording:* 
  + User information is saved in user\_info.txt for future reference.
* *Score Tracking:* 
  + The scoring mechanism provides feedback on the user's performance

# CODE

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <time.h>**

**struct UserInfo {**

**char name[100];**

**int age;**

**int score;**

**};**

**void getUserInfo( struct UserInfo\* user) {**

**printf("Enter your name: ");**

**gets(user->name);**

**printf("Enter your age: ");**

**scanf("%d", &user->age);**

**}**

**void saveUserInfo(struct UserInfo\* user) {**

**FILE\* file ;**

**file= fopen("user\_info.txt", "w");**

**if (file != NULL) {**

**fprintf(file, "Name: %s\nAge: %d\nScore: %d\n", user->name, user->age, user->score);**

**fclose(file);**

**} else**

**printf("Error opening file");**

**}**

**int generateMathProblem()**

**{ int num1 = rand() % 100;**

**int num2 = rand() % 100;**

**int operator = rand() % 4;**

**int answer;**

**switch (operator)**

**{ case 0:**

**printf("What is %d + %d?\n", num1, num2);**

**answer = num1 + num2;**

**break;**

**case 1:**

**printf("What is %d - %d?\n", num1, num2);**

**answer = num1 - num2;**

**break;**

**case 2:**

**printf("What is %d \* %d?\n ", num1, num2);**

**answer = num1 \* num2;**

**break;**

**case 3:**

**printf("What is %d / %d?\n", num1, num2);**

**answer = num1 / num2;**

**break;**

**}**

**int userAnswer;**

**scanf("%d", &userAnswer);**

**if (userAnswer == answer) {**

**printf("Correct!\n");**

**return 1;**

**} else {**

**printf("Sorry, that's incorrect. The answer was %d.\n", answer);**

**return 0;**

**}**

**}**

**int main() {**

**srand(time(NULL));**

**struct UserInfo user;**

**getUserInfo(&user);**

**int correctAnswers = 0;**

**int totalQuestions = 0;**

**char playAgain = 'y';**

**while (playAgain == 'y' || playAgain == 'Y') {**

**if (generateMathProblem())**

**{**

**correctAnswers++;**

**}**

**totalQuestions++;**

**printf("Do you want to play again? (y/n): ");**

**scanf(" %c", &playAgain);**

**}**

**user.score = correctAnswers;**

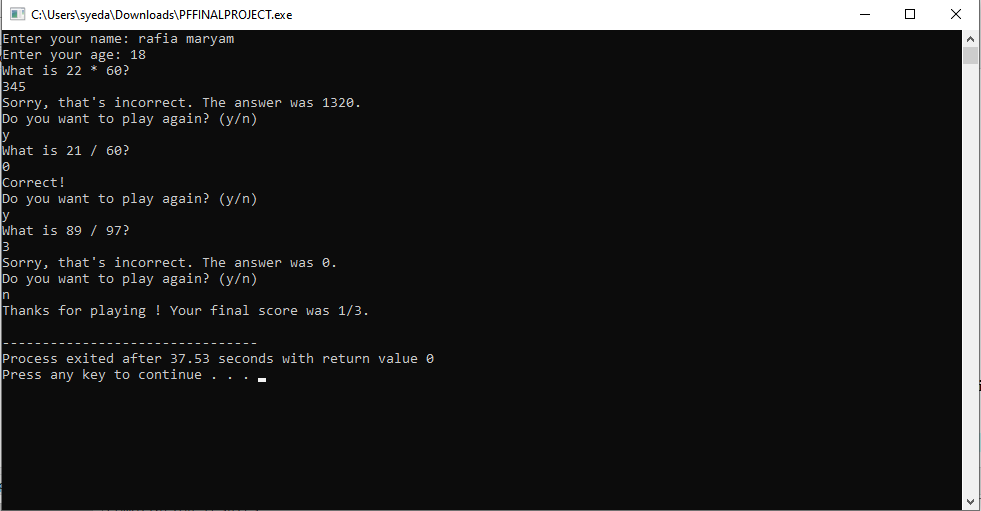
**saveUserInfo(&user);**

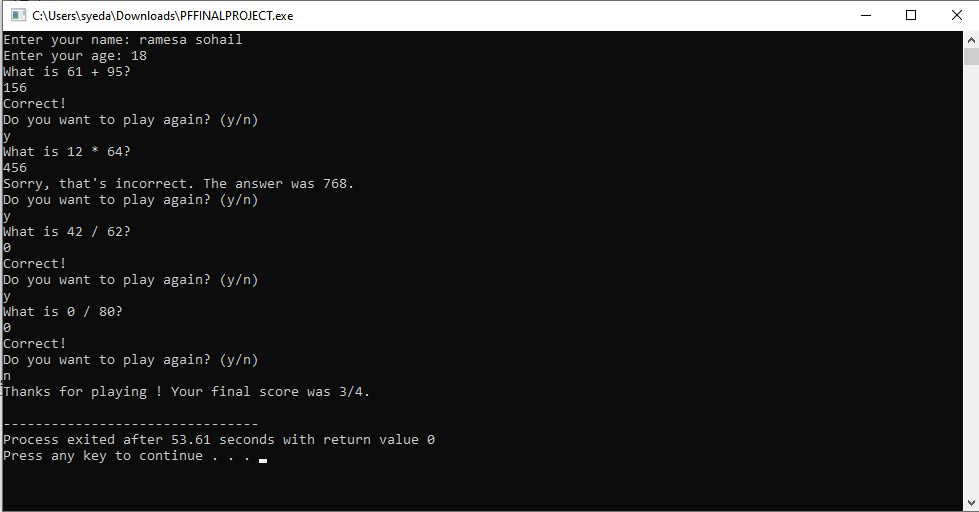
**printf("Thanks for playing ! Your final score is %d/%d.", correctAnswers, totalQuestions);**

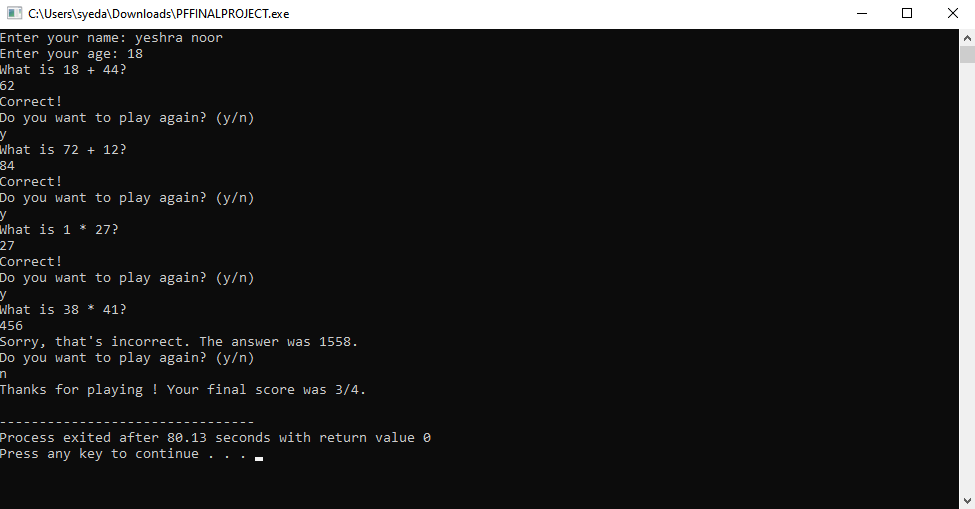
**return 0;**

**}**

# Sample Output







# Conclusion

This program is a practical demonstration of programming fundamentals, offering both educational value and entertainment. It shows how to:

* Collect and process user input
* Generate and evaluate random math problems
* Save data to a file.