Midterm Project

LAB #7
SECTION #4

Nicholas Erickson

10/7/2021

Problem

Create a program that offers the user to take one of 3 different quizzes. Users can take multiple quizzes Program then grades quizzes and shows average score overall at the end.

Analysis

Requirements require the quiz selection be ‘randomly’ selected based on user inputting numbers between 1 - 9. (however this only increases ‘randomness’ from 1 in 3 to 1 in 9). Then create a quiz function that prints a question, asks for user input, and compares their answer to see if it’s correct. If correct, add points, otherwise move to the next question. If the user did not get a 100% on the current quiz, their total score as well as attempt for that quiz is recorded in separate variables, but then their current points are reset and they’re forced to retake the quiz until they receive a 100%. After the user gets a perfect score, print how many attempts it took them along with their average score over all attempts. Then, give the user the option to continue to take another quiz. Next, the quiz is randomly selected based on first seed input from the very start, based on assignment instructions. If the user decides to quit, print out the total attempts taken on all quizzes, as well as the average score over all quizzes.

Design

Followed instructions to seed the rand function with user input to make it ‘random’. Scanned user input and fed the input into the srand(). The rand() function then picked a number between 1 and 3 which chose the quiz. See screenshot 1.

Decided to create an infinite while loop that would use the break; function to exit once the user says they no longer want to quiz. Within the while loop, there are 3 if statements that check for what number quizChoice is equal to (quizChoice is picked by the rand function). The quizzes were the easiest part of the lab - print out questions and scan for user input. Then compared char and compared strings with strcmp to see if user input matched the correct answer. Awarded points for that current quiz attempt for each answer that was correct. At the end of the quiz, checked if the user had gotten a 100% by comparing current points to 15 (max score). Users current score as well as attempts are added to outside variables and then the current score, current attempts, and quizChoice is reset to zero. If the user did not get a perfect score, the user is forced to retake the quiz. Once the user has gotten a perfect score, the program asks for input if they’d like to continue or not. See screenshot 2.

When the user gets a perfect score the if loop for the current quiz ends and drops down to another if loop. See screenshot 3. If the user says no to taking another quiz, it prints

the total quiz attempts taken as well as the total overall score and then breaks from the while loop, ultimately ending the program. If the user says yes, it calls on the rand function to once again select another quiz.

Testing

Bulk of testing was done by getting the first quiz to work. Simply reran the program multiple times until I liked the formatting of the questions as well as made sure the different answer inputs worked (true/false, multiple choice, fill in the blank). After that, I used lots of brute force trying different ways to add scores. Wasn't much thought beforehand on how to keep score, it more so just worked by accidental discovery. Once the first quiz was working, added in the simple if statement that checked what quizChoice was and then copied and pasted 2 more of the entire quiz program for a total of 3 quizzes. Then filled in quizzes with real questions and answers.

Comments

Realized I could've used functions when I finished to shorten the mess, but saw nowhere in instructions requiring it. The rand function for choosing quiz is horribly messy. If you finish the first quiz and decide to take another, the next quiz is chosen using the rand function. Only did this because of instructions. Causes an issue to where you can be given the same 10 times in a row due to the rand function picking a number for you. Could've added a feature to make it so you couldn't take the same quiz twice, but the rand function had already annoyed me so I decided to not mess with it. Nowhere in requirements mentions it can't give the same quiz multiple times in a row, so I stuck with it. Makes more sense for the user to choose the quiz they want.

Screenshots

1

```
    printf("\e[1mQuiz Time\e[0m\n");
    printf("\nPlease enter a number between 1 - 9 to randomly choose a quiz!\n");
    printf("Input: ");
    scanf("%d", &randSeedInput);

    //Seeding random quiz selection
    srand(time(0));
    quizChoice = rand() % (3) + 1;

    // Loop that gives and grades quizzes
    while(1) {

        // Quiz 1
        if (quizChoice == 1) {
```

2.

```
        printf("\n\e[1mQuestion 3\e\[0m\n");
        printf("Fill in the blank: The _____ function outputs text to the screen.\n");
        printf("Answer: ");
        scanf(" %s", &userStringInput);
        if (strcmp("printf", userStringInput) == 0) {
            printf("Correct!\n");
            currentQuizScore = currentQuizScore + 5;
        } else {
            printf("Wrong!\n");
        }
        quizAttempts = quizAttempts + 1;
        totalScores = totalScores + currentQuizScore;
        globalQuizAttempts = globalQuizAttempts + 1;
        globalTotalScores = globalTotalScores + currentQuizScore;
        if (currentQuizScore == 15) {
            printf("\nYou got all questions correct!\n");
            printf("It took you %d attempt(s) with an average score of %.2lf%%\n", quizAttempts, (totalScores / (15 * quizAttempts)) * 100);
            quizComplete = 1;
        } else {
            printf("\nQuiz failed! Try again!\n");
        }
    currentQuizScore = 0;
    }
    quizComplete = 0;
    totalScores = 0;
    quizAttempts = 0;
    quizChoice = 0;
```

3.

```
// Asks user if they want another quiz or quit. If quit, shows total attempts and overall score.
printf("Would you like to take another quiz? Please type y or n: ");
scanf(" %c", &continueQuiz);
if (continueQuiz == 'y') {
    quizChoice = rand() % (3) + 1;
} else {
    printf("\nYou took a total of %.0lf quizzes with an overall average of %.2lf%%!", globalQuizAttempts, (globalTotalScores / (15 * globalQuizAttempts)
    break;
}
```