CSE 102 Assignment 3 - Function and String

You will implement a text-based **game** where players attempt challenges like unscrambling anagrams, decrypting Caesar ciphers, and guessing hidden words.

Description

1. **Game Overview**
    - Players will attempt three challenges sequentially: **Anagram Challenge**, **Caesar Cipher Challenge**, and **Word Guessing Challenge**.
    - After each challenge, the game should display a success message or the correct answer if the player fails.
    - Track the player's score across all challenges. The final score is shown at the end of the game.
2. **Menu Display**
    - Implement a `show_menu` function that displays the game options to the player.
    - The menu should include options to start the game or exit. If the player selects "Exit," the game should terminate.
3. **Anagram Challenge**
    - **Definition:** An anagram is a word or phrase created by rearranging the letters of another word or phrase, using each letter exactly once. For example, "listen" is an anagram of "silent," and "night" is an anagram of "thing."
    - Create three global arrays containing three different words, such as `"listen"`, `"earth"`, and `"binary"`.
    - Randomly select one of these words for each game session, scramble it, and prompt the player to guess the original word.
    - Track incorrect attempts (up to a maximum of 3), and update the player's score based on the number of remaining attempts. If the player runs out of attempts, display the correct answer and move on to the next challenge.
4. **Caesar Cipher Challenge**
    - **Definition**: A Caesar cipher is an encryption technique where each letter in the plaintext is shifted to a certain number of places in the alphabet. For example, with a shift of 3, 'A' would be encrypted to 'D', 'B' to 'E', and so on. To decrypt, the letters are shifted back by the same number.
    - Create three global arrays containing three phrases, such as `"There is a secret code"`, `"Attack at dawn"`, and `"Meet me at the park"`.
    - Randomly select one phrase and encrypt it using a Caesar cipher with a fixed shift (e.g., 3).
    - Display the encrypted phrase to the player and prompt them to decrypt it by entering the original phrase.

    - Implement two functions for Caesar cipher manipulation:
        - `caesar_encrypt`: Encrypts a given text with a specified shift.
        - `caesar_decrypt`: Decrypts a given text with a specified shift.
    - Track incorrect attempts (up to 3) and update the player’s score accordingly. If attempts are exhausted, reveal the correct phrase.
5. **Word Guessing Challenge**
    - Create three global arrays with three words, such as `"program"`, `"network"`, and `"science"`.
    - Randomly select one word, then display a hint to the player (the first three letters of the word).
    - Prompt the player to guess the full word within a maximum of 3 attempts.
    - The user has three options for each attempt:
        - **Write Answer**: Directly enter the complete word they think is correct.
        - **Check Substring**: Enter a substring to check if it exists within the word. If correct, it outputs "Yes"; otherwise, "No".
        - **Check Length**: Enter the length of the word to see if it matches. If correct, it outputs "Yes"; otherwise, "No".
    - Each utility function (Check Substring, Check Length) can be used **only once** during the challenge. Attempting to use a utility function twice will display the message: “Error: utility function already used.”
    - For each utility function used, **2 points will be deducted** from the total score, unless the maximum attempts are reached and the user fails to guess the word—in which case, the score will be **0** and no additional deductions will apply.
    - Track incorrect attempts and adjust the score based on remaining attempts. Display the correct word if the player reaches the attempt limit without success.
6. **Score Calculation**
    - Players earn points based on the remaining number of attempts for each challenge and successful completion of the challenge:
        - **10 points** for each remaining attempt.
        - **10 points** for successful completion
    - If a player fails a challenge, their score for that challenge is 0.
    - If a player completes all challenges successfully, a bonus of **5 points** will be awarded.
    - At the end of the game, display the player’s total score.
7. **Restriction**
    - **No `string.h`**: You are **not allowed** to use the C standard `string.h` library in this assignment.
    - **Implement Required Functions**: Your game will require common string functions like `strlen`, `strcmp`, `strcpy`, and `substr`. You must implement these 4 functions yourself. Make sure to use these functions wherever possible.

Notes

- **Function Usage**: Write clear, reusable functions for each major or repeated task (i.e. each game, random word/phrase generation). Use separate functions for scrambling, Caesar encryption/decryption, scoring, displaying the menu, etc.
- **Random Selection**: Use the `rand()` function to select random words or phrases for each challenge.
- **Global Variables**: Define global arrays for the anagram words, Caesar phrases, and guessing words.

Sample Output

```
Welcome to the Game World!
1. Start Game
2. Exit
Enter your choice: 1

Starting Anagram Challenge...
Scrambled word: rinyab
Your guess: brainy
Incorrect! Try again.
Your guess: binary
Correct! You solved it in 2 attempt(s).

Starting Caesar Cipher Challenge (Shift: 3)…
Encrypted phrase: wkhuh lv d vhfuhw frgh
Your guess: there is a safe code
Incorrect! Try again.
Your guess: there is a secret code
Correct! You decrypted it in 2 attempt(s).

Starting Word Guessing Challenge...
Hint: net_____
Select an option: 1. Write Answer 2. Check Substring 3. Check Length
2
Enter substring: sun
No
3
Enter length: 7
Yes
3
Error: Utility function already used.
1
Enter your guess: network
Correct! You guessed it in 1 attempt(s) using two utility functions.

Game Over!
Your total score: 71 points
```

Marks Distribution:

Task No.	Task	Marks
1	Game Setup & Menu	5
2	Anagram Challenge	15
3	Caesar Cipher Challenge	20
4	Word Guessing Challenge	25
5	Score Calculation	10
6	Custom String Function Implementation	15
8	Code Structure (proper use of functions, indentation, etc)	10
	Total Marks	100

Submission Guidelines:

- ➔ Name your C file as ***<your_id>.c*** (***2305xxx.c***) and upload the file in Moodle.
- ➔ **Deadline: Saturday, 16 November 2024 11:00 PM.**
- ➔ **Do not copy from your classmate. In this case, both of you will get -100%. Also, do not copy code from any website or ChatGPT. You will get -200% in this case.**