

Determine if code matches its described intent:



Memory updated

Your code meets the majority of the assignment requirements, but there are a few key points to address:

1. **Array of Characters (Instead of `string`):** You're using arrays of characters, which is correct, and `cin.get()` is used appropriately for input. You've followed the instructions to avoid the `string` class.
2. **File Handling:**
 - You correctly handle reading from and writing to the file using delimiters (`:`). However, the instructions specify the use of the delimiter `|` (not `:`), so you should replace `:` with `|` in file handling operations to comply with the assignment.
3. **Functions:**
 - Your functions are well-structured and adhere to the requirement of having fewer than 30 lines of executable code.
 - There are no global variables used, which follows the requirement.
4. **No Return in the Middle of Loops:**
 - The program correctly avoids using `return` in the middle of loops.
5. **Looping Logic:**
 - Your main decision loop (`decision()` function) is solid. However, Task 6 (display posts from a particular author) isn't fully integrated into the main loop. You should add an option in the menu for displaying posts by a particular author.
6. **Menu & Decision Logic:**
 - You need to update the menu and `decision()` function to handle Task 6 (display posts by a specific author). You can add an extra option like "(e) Display posts by author."
7. **External Data File Loading:**
 - The logic for loading posts from an external file is sound, but consider adding some checks to ensure you don't exceed the array size (in case there are more than 10 entries in the file).
8. **File Input/Output:**
 - Ensure that when you read or write to the file, you're handling edge cases like file errors (e.g., what happens if the file doesn't exist?).

You're on the right track! Just fix the delimiter issue, integrate Task 6 into your menu, and make sure to thoroughly test file operations.



This C++ program allows users to create a collection of important social media posts. Here's a breakdown:

- **Structures:** A `social` structure stores information about a post (author, title, link, text, likes, comments).
- **Functions:**
 - Input/Output functions:
 - `welcome` : Prints a welcome message.
 - `menu` : Displays a menu and gets user input.
 - `readAPost` : Reads user input for a new post and stores it in a `social` structure.
 - `readOne` , `readAll` : Reads posts from the user until the array is full or the user chooses not to add more.
 - `readFileName` : Gets the filename for saving/loading posts.
 - `saveToFile` : Saves all posts in the array to a file.
 - `displayAPost` : Displays information about a single post.
 - `displayAll` : Displays information about all posts in the array.
 - `loadFromFile` : Loads posts from a file into the array.
 - Processing functions:
 - `decision` : Calls the appropriate function based on user input from the menu.
 - `displayAuthor` : Displays posts written by a specific author (entered by the user).
- **Main Function:**
 1. Creates an array of `social` structures to store posts.
 2. Calls functions to perform various actions based on user input from the menu (adding posts, displaying all posts, loading/saving from a file, displaying posts by a specific author).

Key Points:

- The program uses functions to break down tasks into smaller, manageable pieces.
- It demonstrates working with user input, external data files, and arrays of structures.
- Comments are included to explain the purpose of the code.

I hope this summary is helpful!

I've reviewed your code against the assignment requirements. Here's a brief assessment:

1. Structure and basic functionality: Meets requirements
2. File I/O: Implemented correctly
3. Menu interface: Implemented correctly
4. No global variables: Correct
5. Character arrays used instead of string class: Correct
6. Tasks 1-6 implemented: Yes
7. Function size limit (30 statements): Mostly adhered to
8. Use of C++ I/O: Correct
9. Use of ignore after input: Implemented

Areas for potential improvement:

- Consider removing debug print statements (e.g., "The number of posts in...")
- Some functions could be further simplified
- Error handling for file operations could be enhanced

Overall, your code meets the core requirements of the assignment.

 Copy  Retry  



Claude can make mistakes. Please double-check responses.