

Amirmortaza Rezaee – 810101429-

“the pseudo code”

Start;

Declare an integer “login” and put its value equal to 0.

Get the command from the user

If the user enters signup, call the sign_up function.

*Sign_up function:

 Get the username and password from the user.

 While the user enters an invalid username:

 Search the usernames list and check if already there is the same username, warn the user and get another username.

 After getting the correct username and password, create a new node in the list of users with the scanned username and password. Then create a node which shows the head of users posts list.

If the user enters login, call the login function.

*Login function:

Get the username and password.

While the user enters an invalid username:

Search the usernames list and find that username, if none of them match, warn the user and get another username.

Point the pointer of current_username to that one.

While the user enters an invalid password:

Warn the user and get another password.

Point the pointer of logged_in_user to that user.

If the user enters the post, call the post function.

*Post function:

Create a new node in the posts linked list.

Get the string of post and put in the allocated memory.

the value of post-id= previous posts id +1.

Point the pointer of user to the logged_in_user.

If the user enters like, call the like function.

*Like function:

Get the username and post_id from the user.

Check the liked posts array of the user.

While the logged in user want to like a post **again**, or enters a wrong input:

Warn the user and get another username and post id.

Add 1 to the number of likes of the post which the user enters its post_id and username.

If the user enters logout, point the pointer of logged_in user to the head node of users linked list.

If the user enters delete, call the delete function.

*Delete function:

Get the post_id from the user.

While the user enters an invalid post_id:

Warn the user and get new post_id.

Point the pointer of previous post to next post(or null if unavailable).

Free the node of current post.

If the user enters info, call the info function.

*Info function:

If the logged_in_user pointer points to the head of users linked list:

Ask the user to login first.

Exit the function.

Print the user's username, password, posts and their details(post_id & likes).

If the user enters find_user, call find_user function.

*Find_user function:

Get the username from the user and check the users list.

While the user enters invalid username:

Ask him to enter another one.

Point the pointer of current_user to the wanted user.

Show current_user's username and posts with each one's post_id & likes.

If the user enters invalid input, warn him and ask him to enter another thing.

End;