Task 1:

Implement a system that tracks login attempts using a **stack**. If a user enters the wrong password 3 times consecutively, they should be locked out. After a successful login or if the account is locked, the stack resets.

Steps:

- Use a stack to store failed login attempts.
- Every time a user enters an incorrect password, push the failed attempt onto the stack.
- If the stack reaches 3 failed attempts, lock the account.
- Upon a successful login or lockout, clear the stack.

Example:

Input:

- User enters password 'wrong1' -> failed attempt
- User enters password 'wrong2' -> failed attempt
- User enters password 'correct' -> successful login, stack resets

Input:

- User enters password 'wrong1' -> failed attempt
- User enters password 'wrong2' -> failed attempt
- User enters password 'wrong3' -> failed attempt
- Stack has 3 failed attempts, lock user account.

Task 2:

Implement a system that uses a linked list to maintain a whitelist of trusted URLs. The program will check if a URL is in the whitelist before allowing the user to visit it. If the URL is not in the list, it should be flagged as potentially dangerous.

Steps:

Use a linked list to store URLs (trusted websites).

Every time a user tries to visit a website, search the linked list for that URL.

If the URL exists in the list, allow the user to visit the site.

If the URL is not found in the linked list, flag it as a potential phishing attempt.

Implement functionality to add and remove trusted websites from the list.

Example:

Input:

- Whitelisted sites: [google.com, github.com, wikipedia.org]
- User tries to visit: 'google.com' -> Safe, allow access.
- User tries to visit: 'phishingsite.com' -> Not in whitelist, flag as phishing attempt.

Operations:

- Add new site to whitelist: facebook.com
- Remove site from whitelist: wikipedia.org