Voting System with File Encryption

This is a C program for an online voting system. It has a main function and several helper functions.

The main function first prompts the user for their username and password. If the user enters the correct login credentials, they can access the voting system. If they enter the username "admin" and password "admin123", the program decrypts the data file.

The user can then choose to vote or exit the system. If the user chooses to vote, they are prompted to enter their ID number. If their ID number is found in the voter database, they are eligible to vote and can choose from a list of candidates. The user's vote is recorded and the voter record is updated to reflect that they have voted.

After the user has finished voting or has chosen to exit the system, the program writes the vote results to a file and encrypts the file.

There is also a helper function, **encryptDecryptFile**, that can be used to encrypt or decrypt the vote results file.

1. **main()** function: This function is the entry point of the program. It starts by prompting the user to enter their username and password. If the credentials are correct, the user can proceed to vote, and the program records their vote and updates their voter record. If the credentials are incorrect, the program exits. After all eligible voters have voted, the program writes the results to a file and encrypts it.
2. **register\_voters()** function: This function initializes the **voters** array with some dummy data.
3. **check\_eligibility(char \*id)** function: This function checks if the voter with the given **id** is eligible to vote. If the voter is not found in the **voters** array or has already voted, the function returns 0; otherwise, it returns 1.
4. **update\_voter\_record(char \*id)** function: This function updates the voter record of the voter with the given **id** to indicate that they have already voted.
5. **record\_vote(int option, int \*vote\_counts)** function: This function records the vote of the voter for the selected candidate and updates the vote counts.
6. **write\_results\_to\_file(int \*vote\_counts)** function: This function writes the vote counts to a file named "results.txt".
7. **encryptDecryptFile(int mode)** function: This function takes an integer parameter **mode**, which determines whether to encrypt or decrypt the "results.txt" file. It opens the file, reads each character, and applies a bitwise XOR operation with a predefined key **KEY** to encrypt or decrypt the file.

Overall, the program simulates a simple online voting system that requires the user to log in with their credentials before casting their vote. The program also ensures that each voter can only vote once and stores the vote counts in a file that is encrypted for security.