Lab 6

In this lab we will implement double-linked lists and a "command history".

You should continue to create atomic issues and make atomic commits referencing the issues. You should continue using a milestone (Lab6 in this case) to keep together the issues related to this lab. I will expect to see a number of issues and corresponding commits to match them, within that milestone.

Basic steps

1. You should have already decided which of the two partner's GitHub account to use.
2. If you have not switched roles with your partner for a while, this is a good time to do so.
3. You need to first bring your version of the lab up-to-date by fetching from the remote that is linked to MY project. This is a bit tedious. Only ONE of the two persons needs to do this.
    - Use "git remote –v" to see the remote branches. There should be one named mainSource and pointing to skiadas/WebAppsLabs.
    - We will start by fetching the newest version of that, with: "git fetch mainSource". It should tell you about a new branch, Lab6.
    - Create a new branch with "git checkout –b Lab6 ––track mainSource/Lab6".
    - Push this repository to your fork, by "git push origin Lab6".
    - Set your local branch to in the future update your repository by: "git branch ––set–upstream Lab6 origin/Lab6".
    - To make sure this is set up properly, do "git branch –vv". You should see [origin/Lab6] next to the Lab6 branch line.
4. Check out the correct branch, via git checkout Lab6.
5. One person did steps 3 and 4, here is what the other person needs to do after that, on their repository:
    - Use "git fetch origin" to grab the changes your partner just made.
    - Create a new branch to follow the new lab by "git branch Lab6 origin/Lab6".
    - Switch to the new branch via "git checkout Lab6".
6. In the GitHub issues page for your project, switch to the Milestones tab and create a Lab6 milestone. Use that instead of a label for all issues you create related to Lab6.
7. In the Settings page you can set the "Default branch" for the application. Set this to the Lab6 branch. This should make it so that your pushed commits close any issues that they mention via "Close #n".
8. Open the README.md file there, it will contain the instructions on what you need to do.
9. When you are ready to submit simply email me a link to your project and the SHA of the commit that contains your final submission.

10. You should use the issues page to track your progress. Use the Lab6 milestone to track the relevant issues. I will review those issues to look at your work.
11. Needless to say it, but you are NOT allowed to look at other people's forks of the project and their issues/solutions.
12. I expect you to do the coding using pair programming.