Shared Git Repo Exercise

February 4, 2025

1 Exercise Setup

Follow these steps to begin:

1. You can work alone or in smaller groups (2-4 people)
2. One member create a repository on GitHub called L05-Exercise. All group members can clone (`git clone [your-repo-url]`) and later push to this repository. Each of you can try different tasks in section 2
3. Move into the folder after cloning
4. Verify you have the latest code with `git pull` and confirm you are on the 'main' branch
5. Create your own branch: `git checkout -b your-macID`
6. Create a new directory: `mkdir your-macID`
7. Move into the directory: `cd your-macID`
8. Complete one or more of the tasks in Section 2 in a new file called `str-exercise.c` (Look at section 3 for a template)
9. Your source code must be in L05-Exercise/your-macID/str-exercise.c (otherwise there will be conflicts later)
10. After completion, check that the code compiles and executes, and output makes sense.
11. Now you can add, commit, and push your work (you are pushing to your own branch, so it does not impact others in the repo):

```
git add str-exercise.c
git commit -m "Completed string tasks"
git push -u origin your-macID

```

12. Create a Pull Request (PR) to the main branch on GitHub. If working in groups, you can check each others PRs and approve/merge or leave comments, suggesting changes. If working alone, you can show me, or approve and merge yourself.
13. Switch to the main branch (`git checkout main`), pull the new changes with `git pull`, then compile and execute your code.

2 Programming Tasks

Implement one or more of these functions in your `str-exercise.c` file:

- `int has_exact_n_chars(const char *str, int n)`
  - Returns 1 if string has exactly n characters (before null terminator)
- `int contains_char(const char *str, char c)`

  - Returns 1 if character appears anywhere in string
- `int starts_with(const char *str, const char *sub)`
  - Returns 1 if string begins with the substring
- `int ends_with(const char *str, char c)`
  - Returns 1 if last character matches input character
- `int exactly_one_occurrence(const char *str, char c)`
  - Returns 1 if character appears exactly once
  - Hint: Use both `strchr` and `strrchr`

3 Code Structure Template

Listing 1: str-exercise.c Structure

```c
#include <stdio.h>
#include <string.h>

// Implement your functions here

int main() {
    // Test has_exact_n_chars
    printf("'Hello' has 5 chars? %d\n",
           has_exact_n_chars("Hello", 5));

    // Test contains_char
    printf("'Apple' contains 'z'? %d\n",
           contains_char("Apple", 'z'));

    // Add more test cases...

    return 0;
}
```