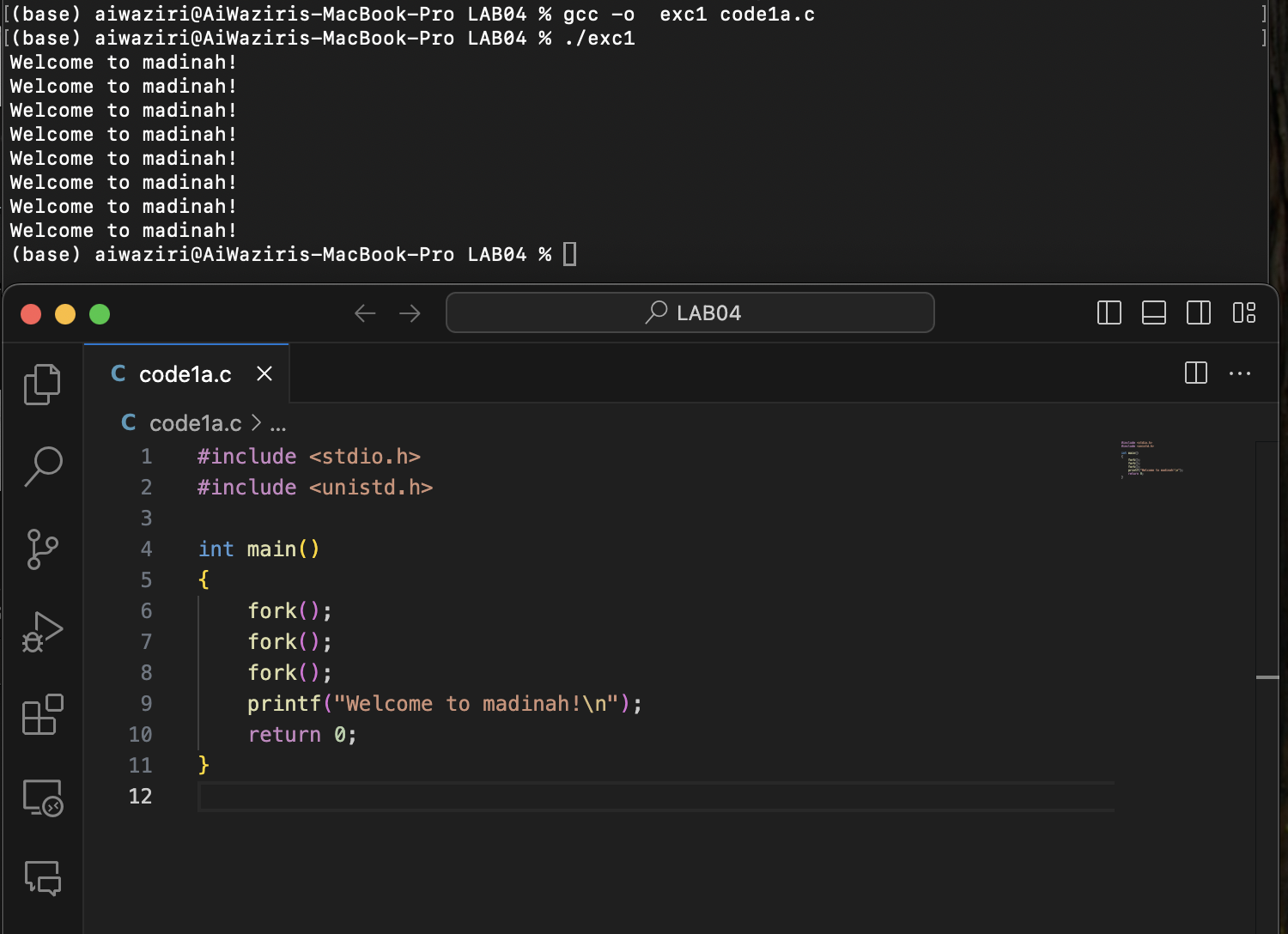
Name: Abubakar Waziri

ID: 4220056

Lab04

Q1a



Q1b

4 processes

Q1c:

A screenshot of a computer program

Description automatically generated

Explanation of Outputs:

* After the first iteration, there are 2 lines of output.
* After the second iteration, there are 4 lines of output.
* In total, there will be (2^2 = 4) lines of output for each iteration, resulting in (2 + 4 = 6) lines of output.

Each process prints its PID, and due to the fork() calls, the number of processes doubles with each iteration of the loop.

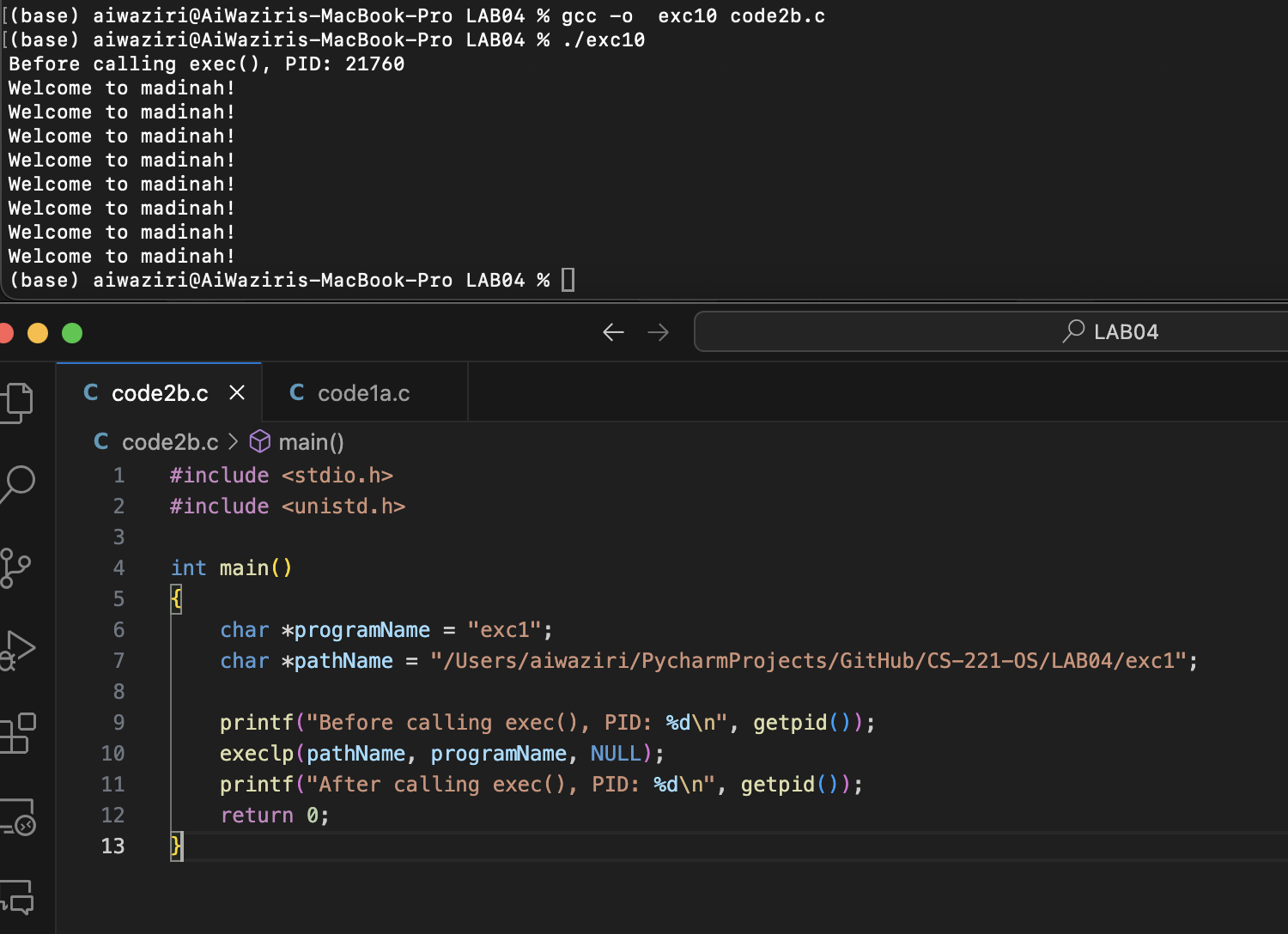
Q2a

A screenshot of a computer program

Description automatically generated

The message "After calling exec()" is not displayed because ***execlp***successfully replaces the current process with the ***ps*** command.

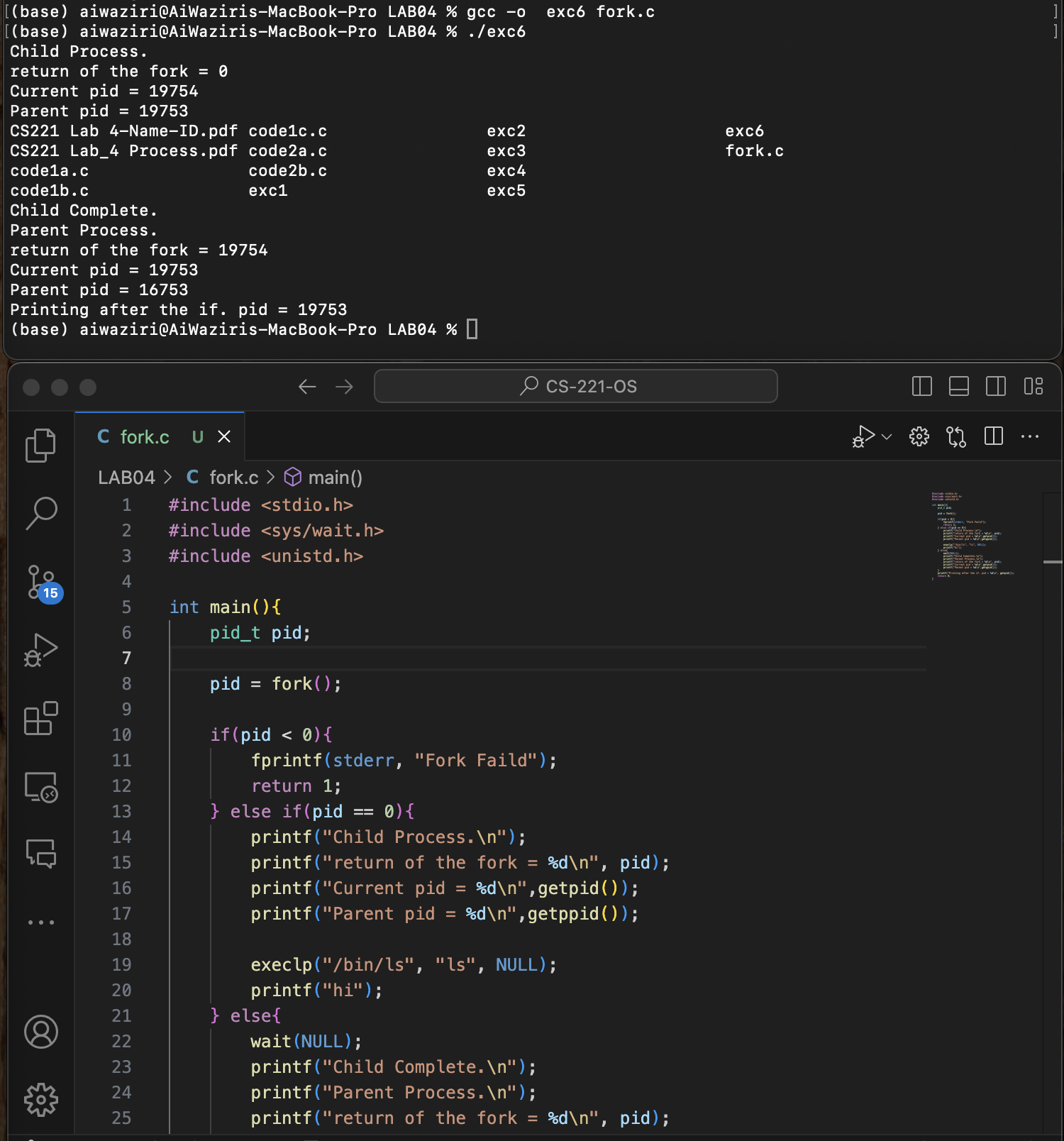
Q2b



**Explanation**

* The ***getpid()*** function is used to get the process ID of the current process.
* The ***execlp***function replaces the current process image with a new process image specified by the given path *(./code2a).*
* If ***[execlp](vscode-file://vscode-app/Applications/Visual%20Studio%20Code.app/Contents/Resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)***is successful, it does not return to the original program; instead, the new program *(./code2a)* starts executing.
* The process ID remains the same because ***[execlp](vscode-file://vscode-app/Applications/Visual%20Studio%20Code.app/Contents/Resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)***does not create a new process; it replaces the current process image with the new one.
* Therefore….. the PID before and after calling ***[execlp](vscode-file://vscode-app/Applications/Visual%20Studio%20Code.app/Contents/Resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)*** will be the same, but the "After calling exec()" message will not be displayed if ***[execlp](vscode-file://vscode-app/Applications/Visual%20Studio%20Code.app/Contents/Resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)***is successful.

Q3a



**Explanation:**

When fork() is called, it creates a new child process. If fork() returns 0, the child process executes, printing its process ID (PID) and its parent's PID, then replaces its memory space with the ls command using execlp(), which lists the directory of the contents.

The printf("hi"); statement is never executed because execlp() replaces the process image. If fork() returns a positive value, the parent process waits for the child to complete using wait(NULL);, then prints a message indicating the child has completed, along with its own PID and the child's PID.

**In general:** The code shows how to create a child process, execute a command within it, and synchronize with the parent process.

Q3b

A screenshot of a computer program

Description automatically generated

YES:

because the parent process no longer waits for the child process to complete before continuing its execution. This results in the parent and child processes running concurrently. Consequently, the parent process may print its messages before the child process finishes executing the ls command, leading to interleaved output.

Q4

A screenshot of a computer program

Description automatically generated

because the parent and child processes have separate copies of the global variable value, and each process modifies its own copy independently after the fork().