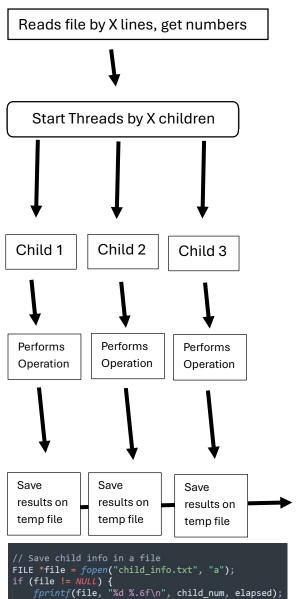
Programming Assignment 1 Report

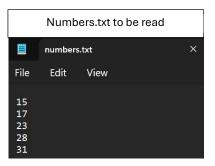
**Anthony Carvalho** 

Thales Moraes

This program will read numbers from a separate txt file and use those numbers in 4 different child processes, which are created by using "fork()". Each child will then find all prime numbers from 0 up to a given number from the txt file. When the process is completed, each child will display a message stating the task is done and how long it took to finish the task. After all child processes finish their tasks, the parent process then displays the final message and, as an extra, we created a leaderboard of the top 3 fastest child processes.

With this assignment, we were able to learn how child processes are created and how related functions work, such as fork() and wait(). The biggest challenges were creating and deleting a temp file to store the times for each process, as well as figuring out the formatting for the leaderboard.





Starts threads

Creating 4 child processes.

```
Operations Consists of finding prime numbers  
Child 1 (PID: 1523808) finding primes up to 15:
2 3 5 7 11 13
Child 2 (PID: 1523809) finding primes up to 17:
2 3 5 7 11 13 17
Child 3 (PID: 1523810) finding primes up to 23:
2 3 5 7 11 13 17 19 23
Child 4 (PID: 1523811) finding primes up to 28:
2 3 5 7 11 13 17 19 23

// Function to check if a number is prime int is_prime(int num) {
    if (num <= 1) return 0;
    for (int i = 2; i * i <= num; i++) {
        if (num % i == 0) return 0;
    }
    return 1;
}
```

Read and display results and leaderboard, delete temp file

```
// Display the leaderboard
display_leaderboard();

// Clean up
remove("child_info.txt");
```