**PA1 Report**

I couldn’t create the binary tree using execvp call so I did small part of the PA1.

* In Node struct, I declared my required variables such as left or right, num1, left and right node.
* inOrderTraversal function is for traversing the function from leftmost to rightmost node.
* newNode function creates a new node.
* In main function, I first check if I have 4 arguments such as current depth, max depth, left-right and ./treePipe. If the user enters less or more arguments, an appropriate error message will be printed to the console and exits the program.
* If maximum depth is 0, parent forks a child process and sends num1 and num2 to the child process using pipe named **fd**. Child process execute left operation on these numbers using execvp system call. Then, child process sends the result back to parent process. After that, it prints the appropriate messages to terminal and exit.
* If the maximum depth is 1, I forked the left child first. From the parent process, I sent num1 to the child process using parent-child pipe named **fd**. Then, I forked again from the left child process for finding the result. Its communication is different than the parent-child because it is parent-grandchild communication. For parent-grandchild communication, I used **grandchildparent** pipe. When the result is computed, grandchild sent this result to parent and then parent (root node) calls execvp on itself for evaluating the result again using fork system call. After that, I handled the right child which has the same steps, the only difference is the computation.