Report - Programming Assignment - CS3510

<u>main():</u>

The main function gets the input for the variable 'n' from the user and checks if its positive or not.

If it’s not, it prints that the given number is not positive and stops the execution. This is done since

the Collatz sequence of numbers makes sense only for positive integers. If the input is positive the main function makes a function call to print_collatz(n).

<u>print_collatz (int n):</u>

The function first declares a variable pid of type pid_t to store the process id of the current process.

It then calls the function fork() and stores the output in pid.

Call to fork() splits the program flow into 2 versions namely the parent and child version.

Both execute the same lines of code following the fork() like except that the parent version has pid = pid of child process and the child version has the pid = 0 .So we typically split the part of program following fork() into an if block for pid>0 and pid == 0 so that we can decide what to perform based on if it’s the parent or child version.

In this case for the child version, we perform the printing the collatz numbers

We print n first. We then switch the value of n as n/2 and 3*n+1 for ‘n’ even or odd respectively and print its changed value. We repeat the above process while n>1.

This prints the collatz sequence of numbers of a given integer n.

For the parent version we print that we are waiting for the child execution to complete and execute the wait() statement which waits till the execution of child is complete.

If pid is neither >0 nor ==0 then it implies a negative pid. This means that fork() has failed to create a child and it conveys it by returning a negative value we stored in pid. So, we have a branch for such cases where we print that creation of child was unsuccessful and return to main.

Finally, the control reaches back to main where the print command "Back to main" gets executed twice since there are two versions of the program running parallelly.