

PART A: Write a C program lab3a.c with the following requirements:

- In the main program, invoke the fork() system call. The child process must display the Process ID and the parent ID 30 times (with a gap of 1 sec between each printing). This should be followed by the appropriate exit() system call.
- The parent process must **wait** for the child process to complete execution and then print the status of the child's termination using the bit manipulation macros in sys/wait.h (WIFEXITED(), WIFSIGNALED() etc.)
- You are required to check the output in two scenarios and print appropriate messages related to the execution status of the child process (terminated/signaled)
 - When the child process completes its execution normally (Prints 30 times with one sec gaps)
 - When you kill the child process while it is running using `$ kill -9 pid` from the terminal (pid of the child process can be obtained using `$ ps -u` (You need to output the signal no of the signal associated with the termination of the child process)

✕

PART B: Write a C program lab3b.c with the following requirements:

- In the main program, invoke two successive fork() system calls, this should create 3 process (two child processes and one grand child process)
- You are required to print "Welcome to COM 8567" **from the grandchild process only** along with the process id of the grandchild process.

Submission:

Upload two files: lab3a.c and lab3b.c