**Code:**

#include <sys/types.h> // This is for some system types... I think it's needed for fork().

#include <stdio.h> // For printing stuff like printf,

#include <unistd.h> // This is where fork() comes from, so we need this.

int value = 5; // Declaring a global variable, starting at 5.

int main() {

pid\_t pid; // Variable to store process Id

pid = fork(); // fork() creates a child process. Both processes start here.

if (pid == 0) { // If we're in the child process...

value += 15; // The child adds 15 to `value`. This is only in the child process.

} else if (pid > 0) { // If we're in the parent process...

wait(NULL); // Parent waits for the child process to finish

printf("PARENT: value=%d", value); /\* LINE A \*/

// Parent prints its value. I think it won't change since fork separates memory.

}

exit(0); // Exit the process (both parent and child will get here).

}

**Line A will print:**

PARENT: value=5