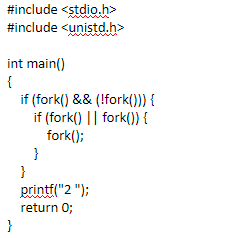
3. Guess the output for the following program.



**Solution**: 2222222

**Explanation**:

1. The first fork() creates a child process.
2. In the parent process of the first fork(), fork()&&!fork() evaluates to false because the first fork() returns a non-zero value (child's PID), and (!fork()) also returns false because the second fork() in this condition returns a non-zero value in the child process. So, the parent doesn't fork again.
3. However, in the child process of the first fork(), fork() && (!fork()) evaluates to true because the first fork() returns 0 (child process) and (!fork()) returns true because the second fork() returns 0 in this child process. So, this child process forks again.
4. Now, we have two processes running concurrently: one is the parent process of the first fork(), and the other is the child process of the first fork().
5. In the child process of the first fork(), fork() || fork() evaluates to true because the first fork() returns 0 (child process), and the second fork() returns a non-zero value (child's PID). So, this child process forks again.
6. Now, we have two child processes running concurrently: one is the child of the first fork(), and the other is the child of the child of the first fork().
7. Both child processes reach the final fork() statement and fork again.
8. Adding up all, there are a total of 7 processes. Hence due to the printf(“2”), 2222222 is the output