**Riphah University**

**Alisha Ishrat**

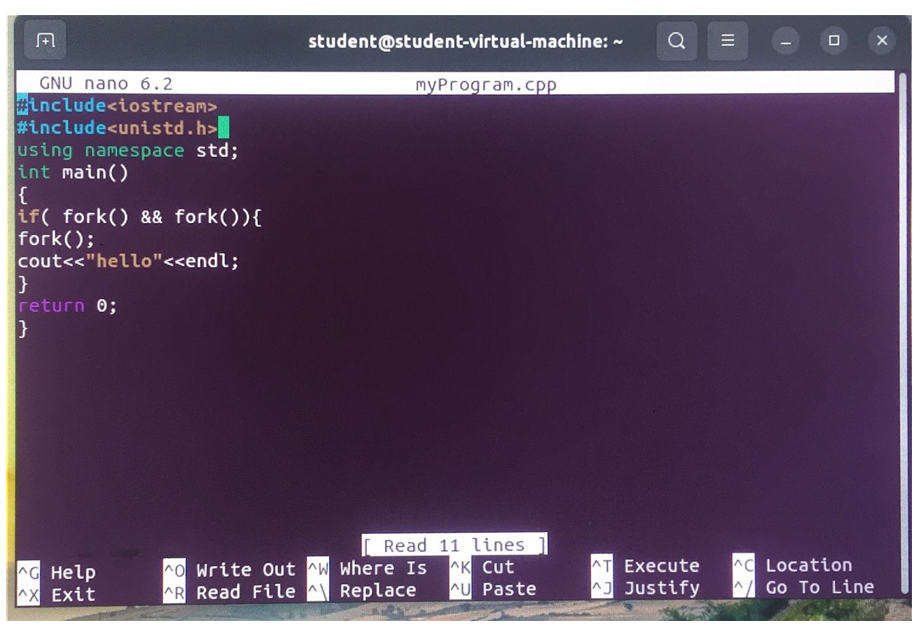
**40851**

**CS-6**

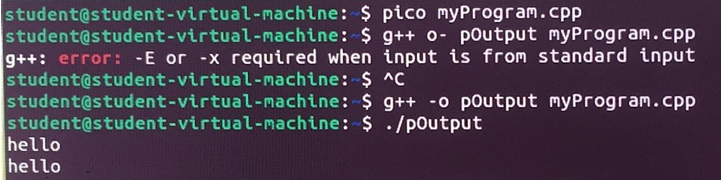
**Operating System**

**Lab: 8**

**Task 1:**

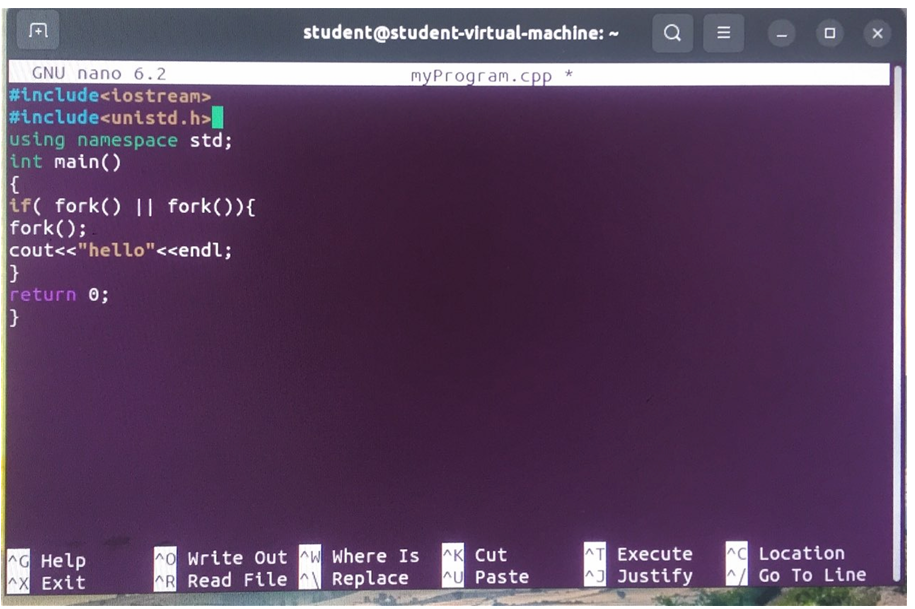


**Output:**

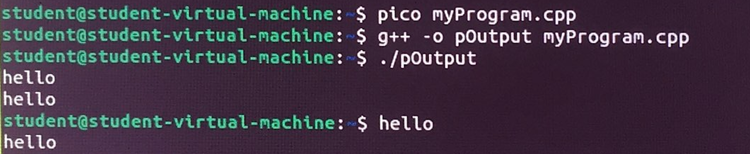
****

**Explanation:**

This C++ program creates multiple processes using the `fork()` system call, which duplicates the running process. Each `fork()` creates a new child process, and both the parent and child processes run the same code. The condition inside the `if` statement ensures that only certain processes (usually the parent) continue to create more processes. Each process that passes through the code prints "hello" to the console. As a result, depending on how many processes are created by the `fork()` calls, "hello" will be printed multiple times.

**Task 2:  
**

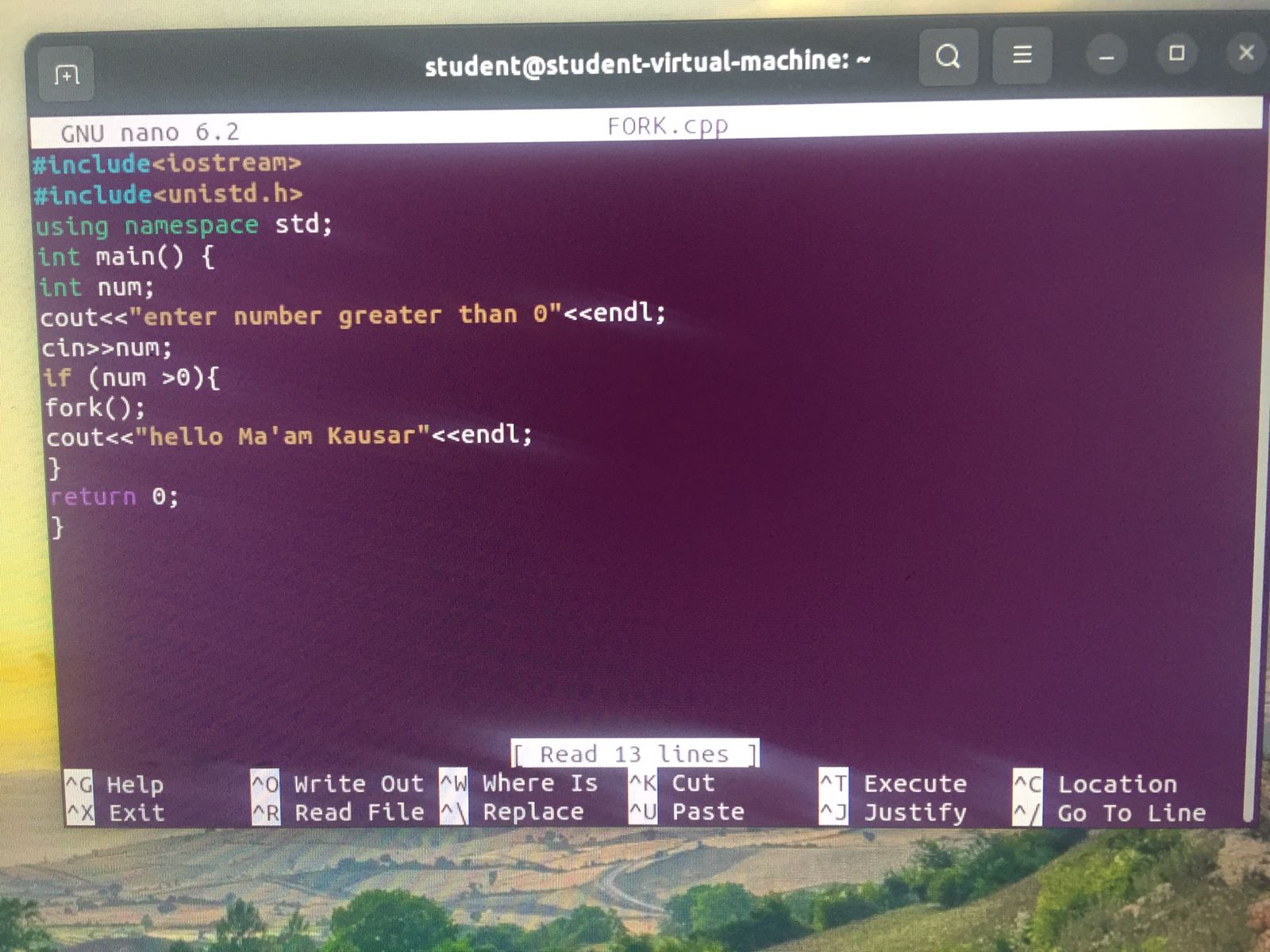
**Output:**



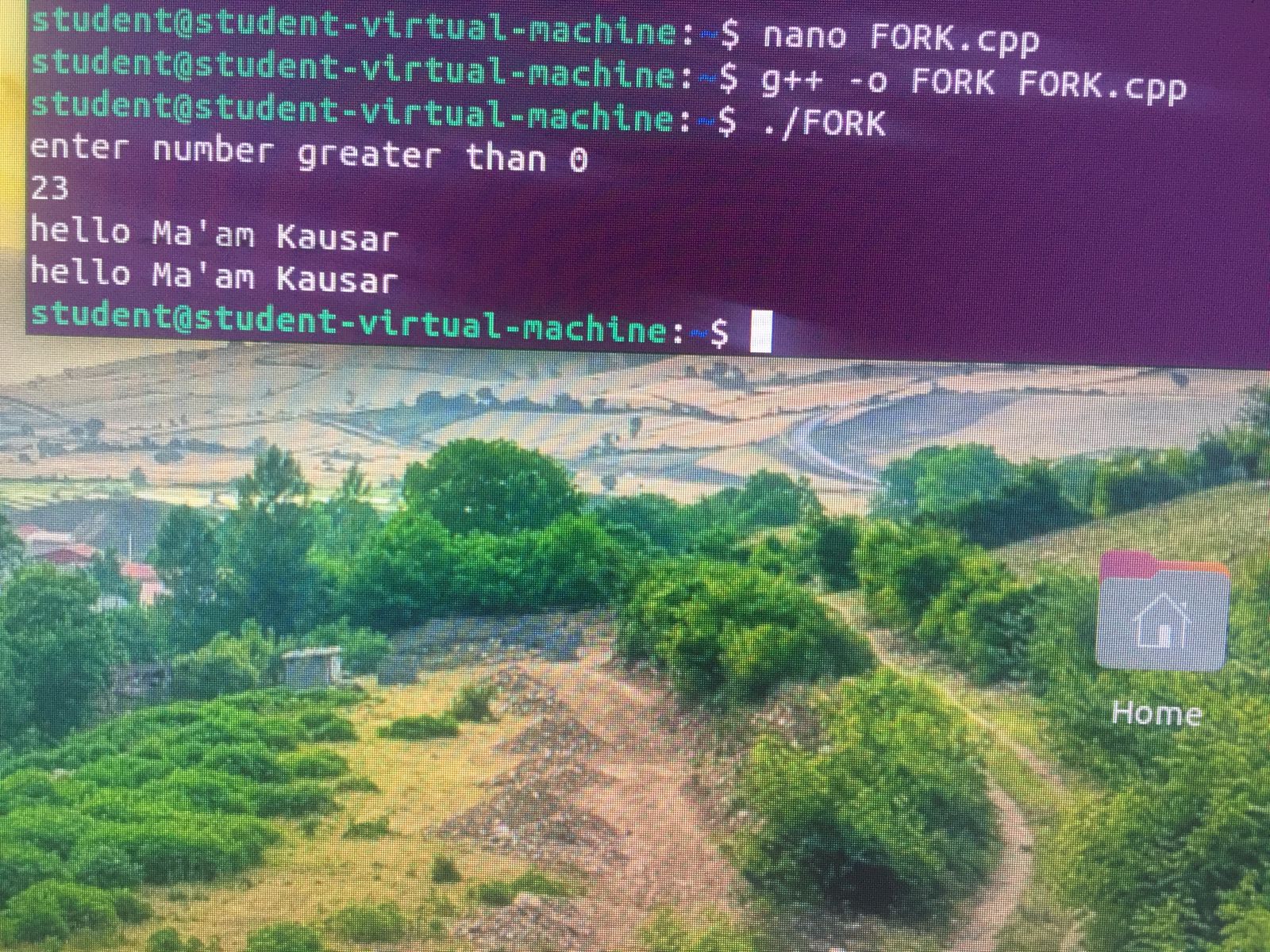
**Explanation:**

This C++ program uses `fork()` to create multiple processes, with the condition inside the `if` statement using a logical OR (`||`), meaning if either of the two `fork()` calls successfully creates a child process, the condition will be true. When true, it triggers another `fork()` inside the block, potentially creating more processes. Each process that reaches the `cout` statement prints "hello" to the console. As a result, multiple processes are created, and "hello" can be printed several times, depending on how many processes are spawned through the `fork()` calls.

**Task 3:**



**Output:**



**Explanation:**

The provided C++ code demonstrates the use of the fork() system call to create a child process. This system call creates a new process that is an exact copy of the parent process. In the given code, the fork() call is executed only if the user enters a number greater than 0. If this condition is met, both the parent and child processes will print the message "hello Ma'am Kausar". This is because both processes are executing the same code after the fork() call.