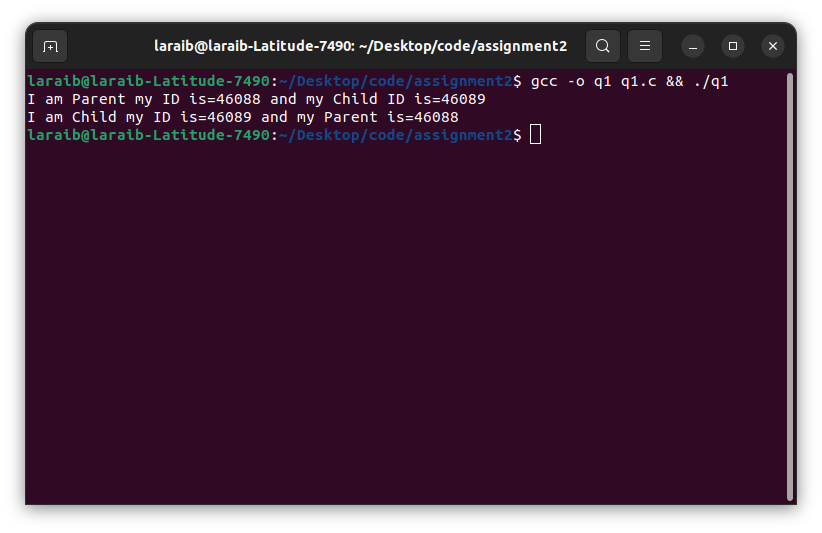
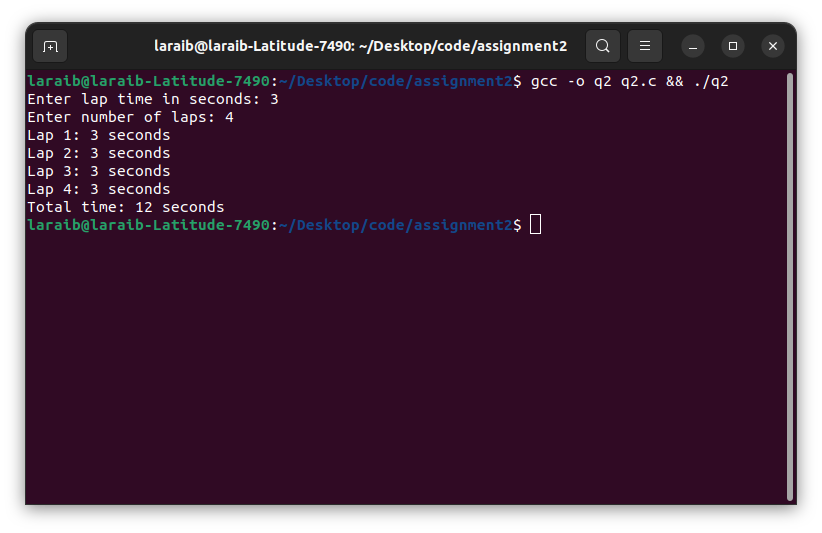
Q1

Q2

Q3

vfork() is a variant of the fork() system call in Unix-like operating systems. It is used to create a new process without copying the address space of the parent process, which is typically done by fork(). The purpose of vfork() is to create a new process quickly and efficiently, by sharing the address space of the parent process with the child process.

The key difference between fork() and vfork() is that fork() creates a copy of the parent process's address space, while vfork() does not. Instead, vfork() creates a new process that shares the same address space as the parent process. This can be more efficient because it avoids the overhead of copying the entire address space.

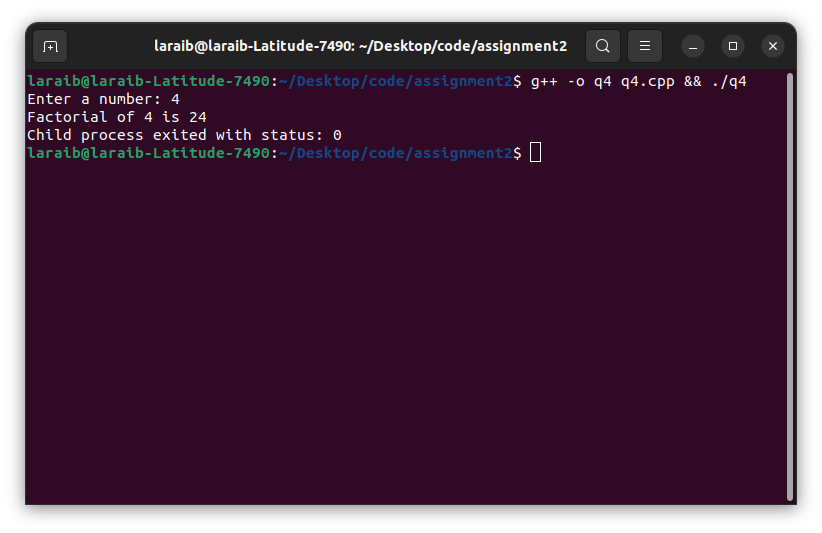
However, since vfork() shares the address space of the parent process, it imposes restrictions on what the child process can do. Specifically, the child process should not modify any data in the parent process's address space until it has called exec() or \_exit(). This is because any modification to the

shared address space will affect the parent process, which could cause unexpected behavior.

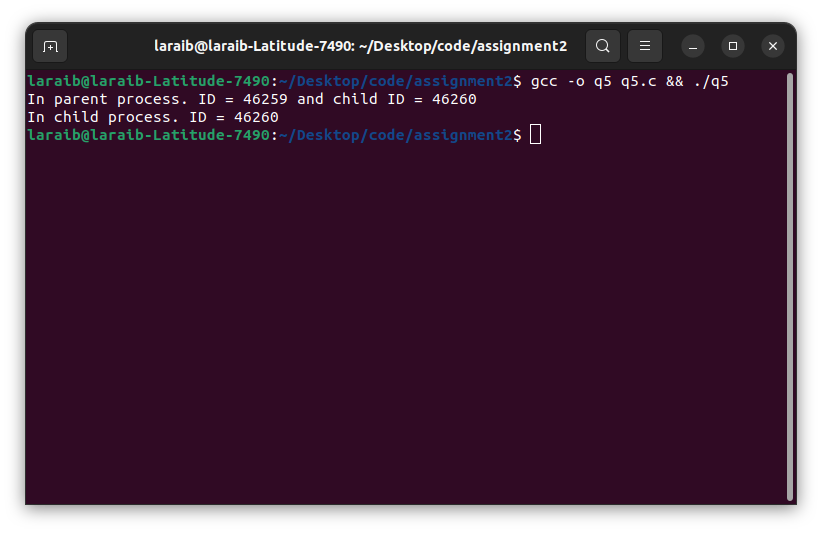
Overall, vfork() is useful when a new process needs to be created quickly and efficiently, and when the child process does not need to modify the parent's address space. However, fork() is typically the safer and more flexible option, since it creates a complete copy of the parent process's

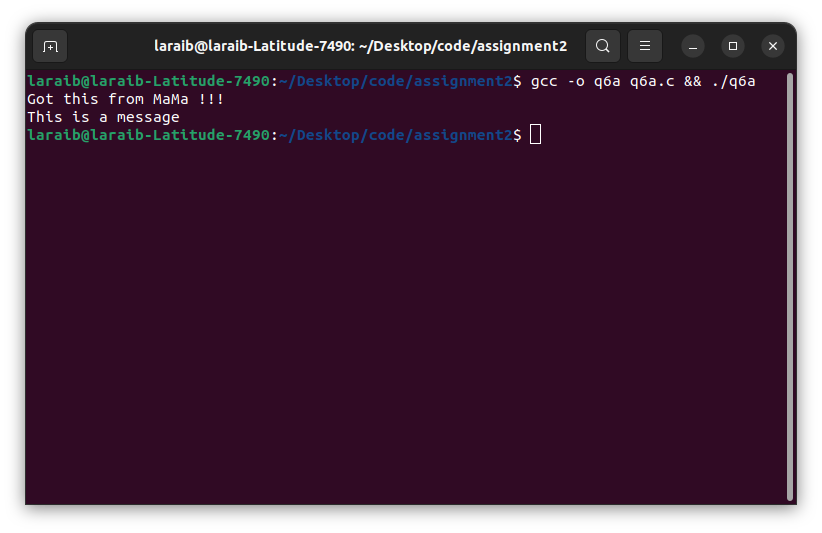
address space for the child process.

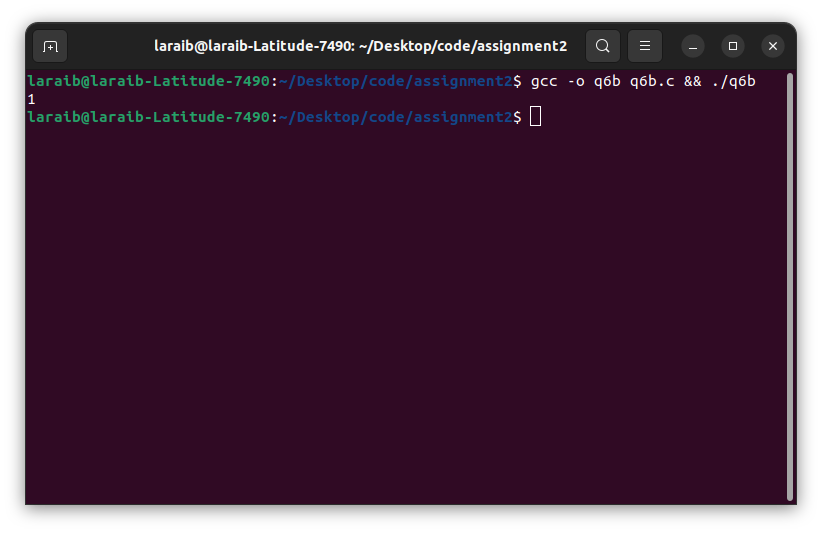
Q4



Q5

Q6a

Q6b



Q7

