README for Question 2

Inter-Process Communication:-

The following question has implemented the functionality of IPC using three different techniques- Using Unix Domain Sockets, Using FIFO Pipes, and Using Shared Memory. Every type has two variants, P1 and P2, for reading and writing.
Firstly in every P1 variant, we create a function that randomly generates 50 strings of the same length. Then we print all the strings. When 5 strings have been printed the second program sends an acknowledgment to the first in the form of the highest index of all the 5 strings received.

SOCKETS:

An AF_UNIX socket has been used For IPC.
Functions used: connect(), then read() and write() in P1 and P2 accordingly.

SHARED MEMORY:

Using the shared memory structure shmseg which sends the strings using a buffer array in packets of 5 strings. Then copying the generated random string into the buffer. We have to do sleep(3) in start of P2 as the writing process will still be faster than reading from P2.
Functions used: shmget() to get the shmid , shmat().

FIFO:

Using FIFO pipes for transferring of strings. Firstly making the FIFO from mkfifo(). Then similarly sending the strings by write() and read() in P2.
Functions used: mkfifo() , open() , close() , read() , write().