A. (8 points) Repeat assignment 5 part 2, except that you shall now use a TCP/IP socket for communicating between the processes instead of a pipe.

Use the following socket functions in their default mode. You may use the `man` command in your Linux virtual machine for information about the parameters:

CLIENT	**SERVER**
socket() – opens a socket (similar to pipe())	socket()
connect() – connects to a server	bind() – assigns a particular port number to the server listen() – listens to connection requests from clients accept() – accepts a connection from client
read() – reads a buffer from the socket, just as in file or pipe reading	write() – writes a buffer to the socket, just as in file or pipe writing
close() – closes the socket	close()

You shall use sockets of type SOCK_STREAM and assign the parent (consumer) as the client and the child (producer) as the server.

Insert an initial random wait (1 to 5 seconds) at the child process (but not the parent) prior to it starting to listen and accept connections.

The parent process (client) may thus fail to connect if it tries to do so before the child process (server) has started to listen (which is after the random wait). As such, you should insert a loop in the parent that repeatedly attempts to connect, waiting 100 ms between attempts, till it succeeds eventually.