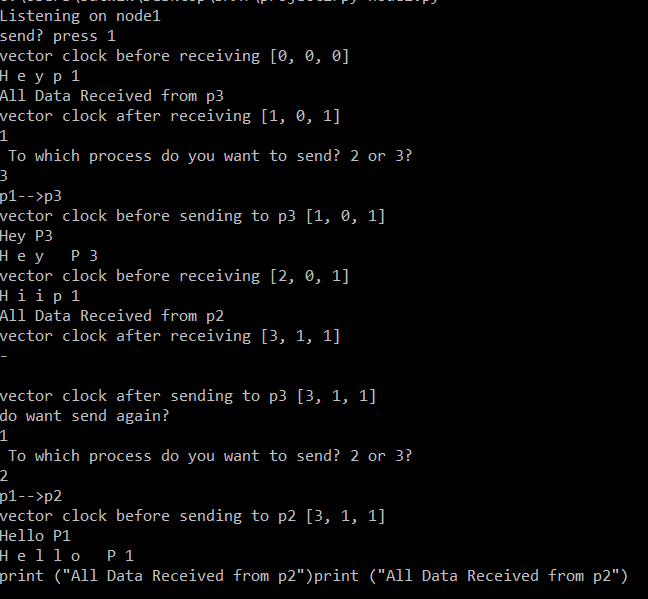
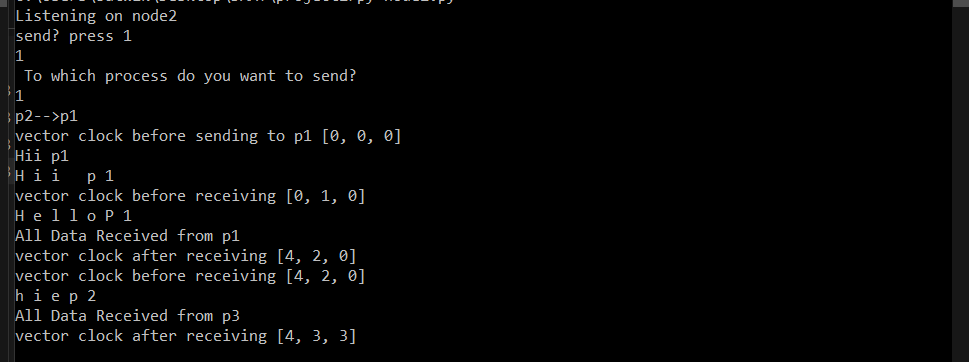
**Report**

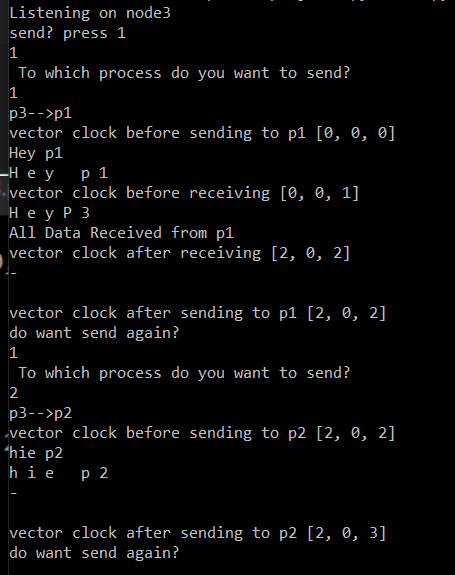
      I have neither given nor received unauthorized assistance on this work

In this project, we have implemented a vector clock using socket programming in python. We have taken three nodes as node1, node2, and node3 each having its client and server. Each node is a client and a server. They communicate using sockets and have unique port numbers. Each node consists of two threads, the first thread takes care of sending messages to other processes or nodes and the other thread takes care of receiving messages. The second thread is started first and then the first thread so that the process starts listening if any other process wants to send a message and then concurrently sends messages. After a process sends a message, it will print its vector clock before and after sending the message and before and after receiving a message.

**Figure 1: Node1- sending and receiving data**

****

**Figure 2: Node2- sending and receiving data**

****

**Figure 3: Node3- sending and receiving data**

**Things Learnt:**

By implementing this project, we understood how vector clock is used in distributed systems, how a connection is made between client and server using sockets, and the use of threading.

**Challenges Faced:**

In the beginning, we got issues with threading and communication between the nodes for message transfer. Also, we got stuck with the updating part of the vector clock after sending and receiving the message as we need to figure out which process has sent the message. Then we attached a header to the message itself based on which server can figure out the process number.

**References:**

<https://www.geeksforgeeks.org/multithreading-python-set-1/>

**Signed**: Jayaram Sivalanka Venkata - 1002081584

Srinivasa Sai Satwik Chakkirala - 1002057355                 **Date**:  10th October 2022