```
import java.io.*;

import java.net.*;


public class Consumer

{

   public static void main(String args[])throws IOException, InterruptedException

   {

      Socket s=new Socket("localhost",7000);

      BufferedReader sc = new BufferedReader(new InputStreamReader(System.in));


      //Streams

      PrintStream out = new PrintStream(s.getOutputStream());

      BufferedReader in = new BufferedReader(new InputStreamReader(s.getInputStream()));


      while(true){

         System.out.println("Want to consume?");

         String console_inp=sc.readLine();


         if(console_inp.equalsIgnoreCase("Yes")){

                  out.println("CONSUME");


                  String item=in.readLine();


                  System.out.println("Consumer consumed - "  + item);

         }
```

```
        }

    }
}

import java.io.*;
import java.net.*;
import java.util.PriorityQueue;
import java.util.Queue;

public class Monitor
{
        public static Queue<String> item_q = new PriorityQueue<String>();
        static int capacity = 2;

        public static void print_q(){
                System.out.println("---Queue elements----");

                for(String s : item_q) {
                    System.out.print(s.toString()+" | ");
                    }
                System.out.println("\n-------------");
        }
```

```
      // Function called by producer thread
public static void produce(String value) throws InterruptedException
{

       // producer thread waits while list
       // is full
       while (item_q.size()==capacity)
          Thread.sleep(5000);;

       System.out.println("Producer produced-"
                         + value);

       // to insert the jobs in the list
       //synchronized(item_q){
               item_q.add(value);
               print_q();
               System.out.println("Lock with producer");
               Thread.sleep(5000);
       //}

       // notifies the consumer thread that
       // now it can start consuming
      // notify();
```

```
            // makes the working of program easier
            // to  understand
            Thread.sleep(5000);



    }


// Function called by consumer thread


    public static String consume() throws InterruptedException
    {


            // consumer thread waits while list
            // is empty
            while (item_q.size()==0)
                Thread.sleep(5000);;


            //to retrive the first job in the list
            String val=null;
           // synchronized(item_q){
                val = item_q.poll();
                System.out.println("Lock with consumer");
                Thread.sleep(5000);
            //}
            System.out.println("Consumer consumed-"
                                + val);

            print_q();

            // and sleep

            Thread.sleep(5000);


            return val;


    }




    public static void main(String args[])throws IOException, InterruptedException

    {

            ServerSocket s=new ServerSocket(7000);

      BufferedReader sc = new BufferedReader(new InputStreamReader(System.in));


      //Accept producer

      Socket ss1=s.accept();


      // Create producer thread

      Thread producer = new Thread(new Runnable()

      {

            PrintStream out = new PrintStream(ss1.getOutputStream());

          BufferedReader in = new BufferedReader(new InputStreamReader(ss1.getInputStream()));


          @Override
```

```
   public void run()
   {
     while(true){
                String item=null;
                                                      try {
                                                                item = in.readLine();
                                                      } catch (IOException e) {
                                                                // TODO Auto-generated catch block
                                                                e.printStackTrace();
                                                      }
                try {
                                                                produce(item);
                                                      } catch (InterruptedException e) {
                                                                // TODO Auto-generated catch block
                                                                e.printStackTrace();
                                                      }

                out.println("PRODUCE");
     }
   }
});

producer.start();

//Accept consumer
```

```
Socket ss2=s.accept();


// Create consumer thread

Thread consumer = new Thread(new Runnable()

{

      PrintStream out = new PrintStream(ss2.getOutputStream());

    BufferedReader in = new BufferedReader(new InputStreamReader(ss2.getInputStream()));


    @Override

    public void run()

    {

      while(true){

                try {

                                                in.readLine();

                                        } catch (IOException e) {

                                                // TODO Auto-generated catch block

                                                e.printStackTrace();

                                        }

                String item=null;

                                        try {

                                                item = consume();

                                        } catch (InterruptedException e) {

                                                // TODO Auto-generated catch block

                                                e.printStackTrace();

                                        }
```

```
                    out.println(item);
             }
          }
      });

      consumer.start();



  }
}
import java.io.*;
import java.net.*;

public class Producer
{
  public static void main(String args[])throws IOException, InterruptedException
  {
      Socket s=new Socket("localhost",7000);
      BufferedReader sc = new BufferedReader(new InputStreamReader(System.in));

      //Input Output Streams
      PrintStream out = new PrintStream(s.getOutputStream());
      BufferedReader in = new BufferedReader(new InputStreamReader(s.getInputStream()));
```

```
        while(true){

            System.out.println("Want to produce?");

            String console_inp=sc.readLine();


            if(console_inp.equalsIgnoreCase("Yes")){

                    String item=sc.readLine();


                    out.println(item);


                    in.readLine();

                    System.out.println("Producer produced - "  + item);


            }




        }


    }

}
```