Name : Bramha Nimbalkar

Roll no : 7

SRN : 202100381

ASSIGNMENT 4

Problem : Implement in C language solution of Producer-consumer problem using

i) Semaphore

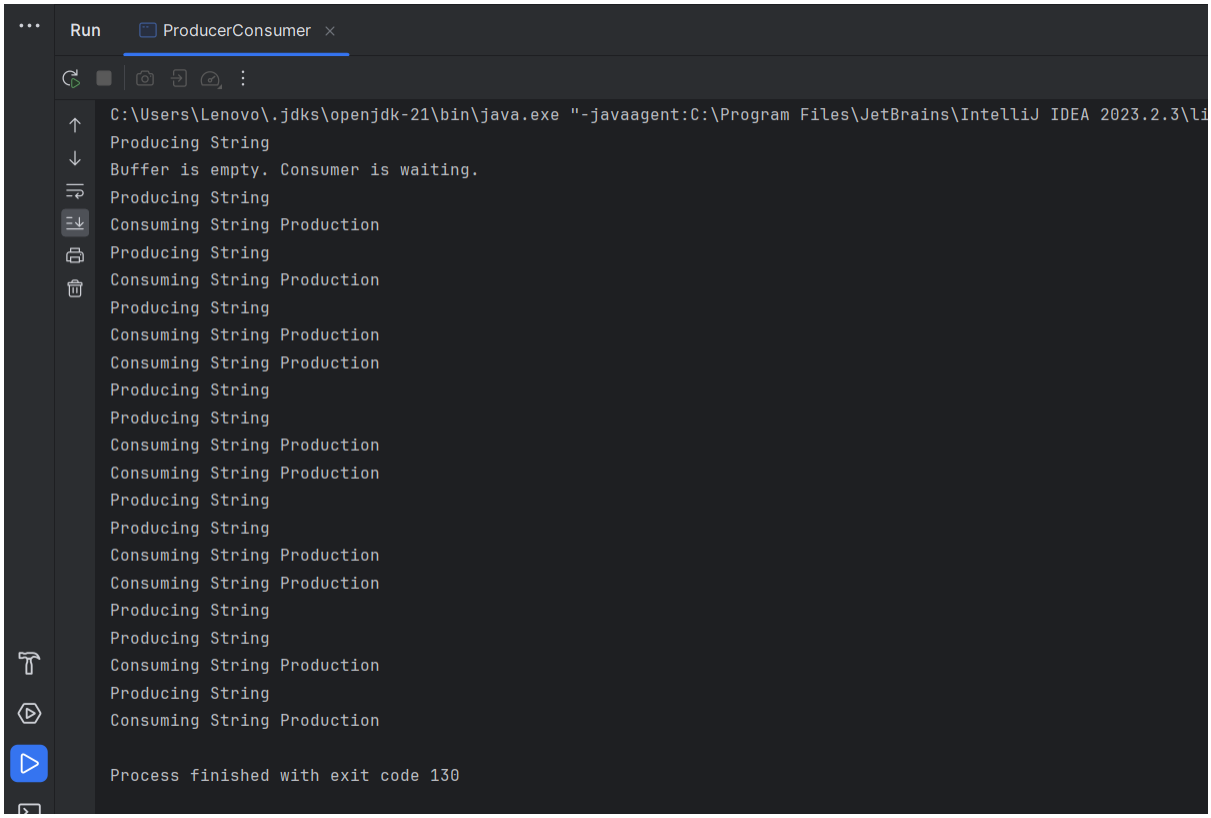
ii) Mutex

```
import java.util.concurrent.Semaphore;

public class ProducerConsumer {
    private static Semaphore bFull = new Semaphore(0);
    private static Semaphore Mutex = new Semaphore(1);
    private static int n = 10;
    private static Semaphore bEmpty = new Semaphore(n);
    private static String[] buffer = new String[n];
//    private static int producerIndex = 0;
    private static int Index = 0;

    private static int Indexc = 0;
    private static int count = 0;

    public static void main(String[] args) {
        Thread producerThread = new Thread(new Producer());
        Thread consumerThread = new Thread(new Consumer());

        producerThread.start();
        consumerThread.start();

        try {
            producerThread.join();
            consumerThread.join();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
```

```
    static class Producer implements Runnable {
        @Override
        public void run() {
            while (true) {
                try {

                    if (count != n) {
                        bEmpty.acquire();
                        Mutex.acquire();
                        buffer[Index] = "Production";
                        Index = (Index + 1) % n;
//                          System.out.println(Index);
                        System.out.println("Producing String");
                        count++;
                        bFull.release();
                        Mutex.release();
                        Thread.sleep(1000);
                    } else {
                        System.out.println("Buffer is full. Producer is
waiting.");
                        Thread.sleep(1000);
                    }

                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        }
    }

    static class Consumer implements Runnable {
        @Override
        public void run() {
            while (true) {
                try {

                    if (count != 0) {
                        bFull.acquire();
                        Mutex.acquire();
//                          bufferIndex-=1;
//                          Index = Index - 1;
                        String data = buffer[Indexc];
                        Indexc = (Indexc + 1) % n;
//                          System.out.println(Indexc);
                        System.out.println("Consuming String " + data);
                        count--;
                        bEmpty.release();
                        Mutex.release();
                        Thread.sleep(1000);
                    } else {
                        System.out.println("Buffer is empty. Consumer is
waiting.");
//                          Mutex.release();
                        Thread.sleep(1000);
                    }


                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
```

```
            }
        }
}
```

Output :


Run
ProducerConsumer
C:\Users\Lenovo\.jdks\openjdk-21\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2023.2.3\li
Producing String
Buffer is empty. Consumer is waiting.
Producing String
Consuming String Production
Producing String
Consuming String Production
Producing String
Consuming String Production
Consuming String Production
Producing String
Producing String
Consuming String Production
Consuming String Production
Producing String
Producing String
Consuming String Production
Consuming String Production
Producing String
Producing String
Consuming String Production
Producing String
Consuming String Production
Process finished with exit code 130