```
class PC
{ public static void main(String args[])
  {   Q q = new Q();
      new Producer(q);
      new Consumer(q);
      System.out.println("Press Control-C to stop");
  }
}
```

Output:-

Put: 1
Got: 1
Got: 1
Put: 2
Got: 1
Got: 2
Got: 1
Put: 3
Got: 1
Got: 3
Put: 2
Put: 4
Put: 3
Got: 4
Put: 4
Put: 5
Put: 5
Got: 5
Put: 6
Put: 7
Got: 7

9 Program on deadlock

Name: Sai Kowshik CR
USN: 1BM22CS232

Lab 9

```
class A
{ synchronized void foo(B b)
  { String name = Thread.currentThread().getName();
    System.out.println(name + "entered A.foo");
    try
    { Thread.sleep(1000);
    } catch(Exception e) {
        System.out.println("A interrupted");
    }
    System.out.println(name + "trying to call B.last()");
    b.last();
  }
  void last() {
    System.out.println("Inside A.last");
  }
}
class B
{ synchronized void bar(A a) {
    String name = Thread.currentThread().getName();
    System.out.println(name + "entered B.bar");
    try
    { Thread.sleep(1000);
    } catch (Exception e)
    { System.out.println("B interrupted");
    }
    System.out.println(name + "trying to call A.last()");
    a.last();
}
void last()
{ System.out.println("Inside A.last"); }
}
class Deadlock implements Runnable.
{     A a = new A();
```

```
B b = new B();
Deadlock()
{   Thread.currentThread().setName("Main Thread");
    Thread t = new Thread(this, "Racing Thread");
    t.start();
    a.foo(b);
    System.out.println("Back in main thread");
}
public void run()
{   b.bar(a);
    System.out.println("Back in other thread");
}
public static void main(String args[])
{   new Deadlock()
}
}
```

Output:-

```
MainThread entered A.foo.
RacingThread entered B.bar.
MainThread trying to call B.last();
Inside A.last
Back in main thread.
RacingThread trying to call A.last()
Inside A.last
Back in other thread.
```