1.

Semaphore s1 = 0, s2 = 0;

```
Client {
 makeOrder();
 up(s1);
 down(s2);
 provideRequiredInformation();
 up(s1);
 down(s2);
 receiveProduct();
}
```

```
Shopping {
 down(s1);
 receiveOrder();
 requireAdditionalInformation();
 up(s2);
 down(s1);
 verifyReceivedInformation();
 deliverProduct();
 up(s2);
}
```

2. semaphore hydro = 0, oxy = 0, hydro_2 = 2, oxy_1 =1;

```
P1{
  down(oxy_1);
  create1Oxygen();
  up(oxy);
}
```

```
P2{
   down(hydro_2);
   create1Hydrogen();
   up(hydro);
}
```

```
P3{
   down(hydro);
   down(hydro);
   down(oxy);
   create1H20();
   up(hydro_2);
   up(hydro_2);
   up(oxy_1);
}
```

5.

a.

Trong trường hợp tệ nhất, mỗi tape drive sẽ giữ 1 và sẽ cùng chờ 1 resourse nữa => n = 6 => deadlock

b.

Giảm số process (tape drive) xuống còn 5 (6 – 1 = 5) có 5 tape drives thì mình sẽ dư 1 cái tape disks, cái tape disks còn dư lại sẽ đưa cho 1 tape drives bất kì để chạy, khi xong nó sẽ trả ra và cho lần lượt các tape drives còn lại chạy.

=> no deadlock

6.

Trong trường hợp tệ nhất:

- P1 giữ 2 và chờ 1

- P2 giữ 0 và chờ 1

- P3 giữ 1 và chờ 1

- P4 giữ 1 và chờ 1

R(min) = 2 + 0 + 1 + 1 = 4 => deadlock

Khi đó ta sẽ tăng só resource R lên 4 + 1 = 5, một process sẽ được thực thi, sau khi xong thì sẽ trả 1 resource đó lại cho các process khác tiếp tục chạy => no deadlock