

1. The operation *enter* in the `barrierObj` object is the entry point of the barrier, whenever a new process calls this method it checks if there are already 3 processes waiting, *i.e. the current process calling it is the fourth one*, if so all the waiting processes are released, otherwise the current process waits.

In the main program I created 4 processes that trigger that barrier. Upon successful execution you should see print statements with info about what is happening as required in the exercise description.

**Related files:** `1-barrier.m` (source code)  
`1-barrier.txt` (output)

2. I implemented a `sharedBuffer` object which provides two operations `produce` and `consume` and has an internal buffer (data) that keeps the items produced. The `produce` operation is used by the producer and fills up the `sharedBuffer` with data as long as it has space, and the `consume` operation is used by the consumer to read that data as long as it has something there to read. Access to the data array is synchronized via those operations. Upon successful execution you should see print statements with info about what is happening as required in the exercise description.

**Related files:** `2-prod_cons.m` (source code)  
`2-prod_cons.txt` (output)

3. I made modifications in the provided `kilroy` program to calculate the potential speed (nodes per second) of the program to visit nodes. I actually managed to run this on planet lab in different nodes, just had to run as `sudo` because Emerald was complaining about different user ids whenever I contacted another Emerald node, running as `sudo` made every process have user id 0 and it worked.

**Nodes used to run this program:**

Same site: `ple3.planet-lab.eu` (master), `ple4.planet-lab.eu`, `ple44.planet-lab.eu`,  
`ple43.planet-lab.eu` - Paris, France.

Different sites: *ple4.planet-lab.eu (master) (Paris, France)*, *planetlab1.xeno.cl.cam.ac.uk (Cambridge, UK)*, *pl1.uni-rostock.de (Rostock, Germany)*, *planet-lab-node2.netgroup.uniroma2.it (Rome, Italy)*

**Related files:** 3-kilroy.m (source code)  
3-kilroy-samesite.txt (output of running this program on the same site)  
3-kilroy-differentsites.txt (output of running this program in sites at least 500km apart)