CS354
Lab 1
Shafay Haq
Professor Sonia Fahmy

1) The remove method has a mutex which is a semaphore with a count of 1, hence it only allows one process to run the remove code at a time.

2) Consumers and producers continue in a sort of concurrent way, if there are items to insert, and items to remove, then the remove and insert methods continue however, the mutex prevents these concurrent processes from changing the buffer and other shared variables at the same time.

3) They do insert in contiguous locations since the global shared counter head is used, it is modified mutually exclusively. And removed in FIFO order from contiguous locations.

4)

normal conditions equal priority of 20

Producer A: created 44 items
Producer B: created 44 items
Producer C: created 44 items
Producer D: created 44 items
Producer E: created 44 items

Consumer a: deleted 2 items from producer A, deleted 31 items from producer B, deleted 22 items from producer C, deleted 0 items from producer D, deleted 0 items from producer E
Consumer b: deleted 42 items from producer A, deleted 13 items from producer B, deleted 0 items from producer C, deleted 0 items from producer D, deleted 0 items from producer E
Consumer c: deleted 0 items from producer A, deleted 0 items from producer B, deleted 22 items from producer C, deleted 33 items from producer D, deleted 0 items from producer E
Consumer d: deleted 0 items from producer A, deleted 0 items from producer B, deleted 0 items from producer C, deleted 11 items from producer D, deleted 44 items from producer E

The shared buffer contains: 0 items from producer A, 0 items from producer B, 0 items from producer C, 0 items from producer D, 0 items from producer E

Producer Priority 20 > Consumer Priority 1

Producers have a priority of 20 and consumers have a priority of 1 this means the process manager will prioritize producing at a rate of 44 productions per process to consuming 55 processes per process, in the results below it only consumes 4 items per process per producer.
This shows a decrease in the trend of items being consumed since the producers are accessing the buffer before and more often than consumers. So each consumer consumed on average more of each produced item since there were more items to consume of each.

Producer A: created 44 items

Producer B: created 44 items
Producer C: created 44 items
Producer D: created 44 items
Producer E: created 44 items

Consumer a: deleted 29 from producer A, 26 from producer B, 0 from producer C, 0 from producer D, 0 from producer E

Consumer b: deleted 5 from producer A, 18 from producer B, 32 from producer C, 0 from producer D, 0 from producer E

Consumer c: deleted 5 from producer A, 0 from producer B, 12 from producer C, 38 from producer D, 0 from producer E

Consumer d: deleted 5 from producer A, 0 from producer B, 0 from producer C, 6 from producer D, 44 from producer E

The shared buffer contains: 0 items from producer A, 0 items from producer B, 0 items from producer C, 0 items from producer D, 0 items from producer E

cons prior 20 > prod prior 1

In this scenario consumers had higher priority than producers, the distribution of items consumed was very uniform for every consumed item, ie. 11.
This happens since each consumer had a higher priority than producers, everytime a producer went to sleep all the consumers consumed a certain amount of items, and all these items produced by a producer were consumed before more were produced. The buffer was accessed by consumers more often than producers, but since consumers are waiting for producers, it was fair access of the buffer since all consumers accessed buffer for same amount of time, due to same priority, and producers had time to produce their items.

Producer A: created 44 items
Producer B: created 44 items
Producer C: created 44 items
Producer D: created 44 items
Producer E: created 44 items

Consumer a: deleted 11 items from producer A, deleted 11 items from producer B, deleted 11 items from producer C, deleted 11 items from producer D, deleted 11 items from producer E

Consumer b: deleted 11 items from producer A, deleted 11 items from producer B, deleted 11 items from producer C, deleted 11 items from producer D, deleted 11 items from producer E

Consumer c: deleted 11 items from producer A, deleted 11 items from producer B, deleted 11 items from producer C, deleted 11 items from producer D, deleted 11 items from producer E

Consumer d: deleted 11 items from producer A, deleted 11 items from producer B, deleted 11 items from producer C, deleted 11 items from producer D, deleted 11 items from producer E

The shared buffer contains: 0 items from producer A, 0 items from producer B, 0 items from producer C, 0 items from producer D, 0 items from producer E

cons unique higher priority increments of ((i+1)*5)

Consumers had higher priority that was unique, the last consumer had highest priority, increments were of multiples of 5.
Buffer access was less fair since consumers had higher unique priorities and hence some consumers accessed the buffer more often than others and consumed more.

Producer A: created 44 items
Producer B: created 44 items
Producer C: created 44 items
Producer D: created 44 items
Producer E: created 44 items

Consumer a: deleted 44 items from producer A, deleted 11 items from producer B, deleted 0 items from producer C, deleted 0 items from producer D, deleted 0 items from producer E

Consumer b: deleted 0 items from producer A, deleted 33 items from producer B, deleted 22 items from producer C, deleted 0 items from producer D, deleted 0 items from producer E

Consumer c: deleted 0 items from producer A, deleted 0 items from producer B, deleted 22 items from producer C, deleted 33 items from producer D, deleted 0 items from producer E

Consumer d: deleted 0 items from producer A, deleted 0 items from producer B, deleted 0 items from producer C, deleted 11 items from producer D, deleted 44 items from producer E

The shared buffer contains: 0 items from producer A, 0 items from producer B, 0 items from producer C, 0 items from producer D, 0 items from producer E

prods unique higher priority increments of ((i+1)*5)

Producers had higher priority that was unique, the last producer had highest priority, increments were of multiples of 5.
Buffer access was less fair since producers had higher unique priorities and hence some producers accessed the buffer more often than others and produced more.

Producer A: created 44 items
Producer B: created 44 items
Producer C: created 44 items
Producer D: created 44 items
Producer E: created 44 items

Consumer a: deleted 7 items from producer A, deleted 0 items from producer B, deleted 9 items from producer C, deleted 39 items from producer D, deleted 0 items from producer E

```
Consumer b: deleted 6 items from producer A, deleted 0 items from producer B, deleted 0 items from
producer C, deleted 5 items from producer D, deleted 44 items from producer E
Consumer c: deleted 28 items from producer A, deleted 27 items from producer B, deleted 0 items from
producer C, deleted 0 items from producer D, deleted 0 items from producer E
Consumer d: deleted 3 items from producer A, deleted 17 items from producer B, deleted 35 items
from producer C, deleted 0 items from producer D, deleted 0 items from producer E

The shared buffer contains: 0 items from producer A, 0 items from producer B, 0 items from producer
C, 0 items from producer D, 0 items from producer E
```