**Practical No :- 01**

**AIM :- Implement Producer and Consumer Synchronization using Shared Memory and Semaphores**

**INPUT:-**

import threading

import time

# Shared Memory variable

CAPACITY = 10

buffer = [-1 for \_ in range(CAPACITY)]

in\_index = 0

out\_index = 0

# Declaring Semaphores

mutex = threading.Semaphore(1) # Only 1 thread can access the buffer at a time

empty = threading.Semaphore(CAPACITY) # Initially, all slots are empty

full = threading.Semaphore(0) # No slots are full initially

counter = 0

# Producer Thread Class

class Producer(threading.Thread):

def run(self):

global CAPACITY, buffer, in\_index, counter, mutex, empty, full

items\_produced = 0

while items\_produced < 20:

empty.acquire()

mutex.acquire()

counter += 1

buffer[in\_index] = counter

print("Producer produced:", counter)

in\_index = (in\_index + 1) % CAPACITY

mutex.release()

full.release()

time.sleep(1)

items\_produced += 1

# Consumer Thread Class

class Consumer(threading.Thread):

def run(self):

global CAPACITY, buffer, out\_index, mutex, empty, full

items\_consumed = 0

while items\_consumed < 20:

full.acquire()

mutex.acquire()

item = buffer[out\_index]

print("Consumer consumed item:", item)

out\_index = (out\_index + 1) % CAPACITY

mutex.release()

empty.release()

time.sleep(2.5)

items\_consumed += 1

# Creating Threads

producer = Producer()

consumer = Consumer()

# Starting Threads

producer.start()

consumer.start()

# Waiting for threads to complete

producer.join()

consumer.join()

**Output :-**

Producer produced: 1

Consumer consumed item: 1

Producer produced: 2

Producer produced: 3

Consumer consumed item: 2

Producer produced: 4

Producer produced: 5

Consumer consumed item: 3

Producer produced: 6

Producer produced: 7

Producer produced: 8

Consumer consumed item: 4

Producer produced: 9

Producer produced: 10

Consumer consumed item: 5

Producer produced: 11

Producer produced: 12

Producer produced: 13

Consumer consumed item: 6

Producer produced: 14

Producer produced: 15

Consumer consumed item: 7

Producer produced: 16

Producer produced: 17

Consumer consumed item: 8

Producer produced: 18

Consumer consumed item: 9

Producer produced: 19

Consumer consumed item: 10

Producer produced: 20

Consumer consumed item: 11

Consumer consumed item: 12

Consumer consumed item: 13

Consumer consumed item: 14

Consumer consumed item: 15

Consumer consumed item: 16

Consumer consumed item: 17

Consumer consumed item: 18

Consumer consumed item: 19

Consumer consumed item: 20