18. Write a program for solving the producer consumer problem with the following scenario: The producer should produce data only when the buffer is not full. Data can only be consumed by the consumer if and only if the memory buffer is not empty.

Test Case:

Buffer Size: 3

Consume an item in the beginning and show that the buffer is EMPTY

Produce 4 items and show that the buffer is FULL

Program:

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h> #define BUFFER\_SIZE 3

int buffer[BUFFER\_SIZE]; int count = 0; int in = 0; int out = 0;

pthread\_mutex\_t mutex = PTHREAD\_MUTEX\_INITIALIZER; pthread\_cond\_t empty = PTHREAD\_COND\_INITIALIZER; pthread\_cond\_t full = PTHREAD\_COND\_INITIALIZER;

void \*producer(void \*arg) {

int i;

for (i = 0; i < 4; i++) { pthread\_mutex\_lock(&mutex); while (count == BUFFER\_SIZE) {

printf("Buffer is full, producer is waiting...\n");

pthread\_cond\_wait(&full, &mutex);

}

buffer[in] = i;

printf("Produced item %d\n", buffer[in]); in = (in + 1) % BUFFER\_SIZE; count++;

pthread\_cond\_signal(&empty); pthread\_mutex\_unlock(&mutex);

}

pthread\_exit(NULL);

}

void \*consumer(void \*arg) { int item; while (1) {

pthread\_mutex\_lock(&mutex);

while (count == 0) {

printf("Buffer is empty, consumer is waiting...\n");

pthread\_cond\_wait(&empty, &mutex);

}

item = buffer[out];

printf("Consumed item %d\n", item);

out = (out + 1) % BUFFER\_SIZE;

count--;

pthread\_cond\_signal(&full); pthread\_mutex\_unlock(&mutex);

} } int main() {

pthread\_t producer\_thread, consumer\_thread;

pthread\_create(&producer\_thread, NULL, producer, NULL); pthread\_create(&consumer\_thread, NULL, consumer, NULL);

pthread\_join(producer\_thread, NULL); pthread\_join(consumer\_thread, NULL);

}

Output:

