Exp:

Code:

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#define BUFFER\_SIZE 5

int buffer[BUFFER\_SIZE];

int in = 0, out = 0, count = 0;

pthread\_mutex\_t mutex = PTHREAD\_MUTEX\_INITIALIZER;

pthread\_cond\_t full = PTHREAD\_COND\_INITIALIZER;

pthread\_cond\_t empty = PTHREAD\_COND\_INITIALIZER;

void \*producer(void \*arg) {

int item;

for (int i = 0; i < 10; i++) {

item = rand() % 100; // Simulate producing an item

pthread\_mutex\_lock(&mutex);

while (count == BUFFER\_SIZE) {

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

}

buffer[in] = item;

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

count++;

printf("Produced: %d\n", item);

pthread\_cond\_signal(&full);

pthread\_mutex\_unlock(&mutex);

}

pthread\_exit(NULL);

}

void \*consumer(void \*arg) {

int item;

for (int i = 0; i < 10; i++) {

pthread\_mutex\_lock(&mutex);

while (count == 0) {

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

}

item = buffer[out];

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

count--;

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

pthread\_cond\_signal(&empty);

pthread\_mutex\_unlock(&mutex);

}

pthread\_exit(NULL);

}

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);

return 0;

}

Output:

