NAME: Kaustubh Gajanan Indulkar
TE-IT 25027 (35026)

ASSIGNMENT 4

Problem Statement: Thread synchronization using counting semaphores. Application to demonstrate: producer-consumer problem with counting semaphores and mutex.

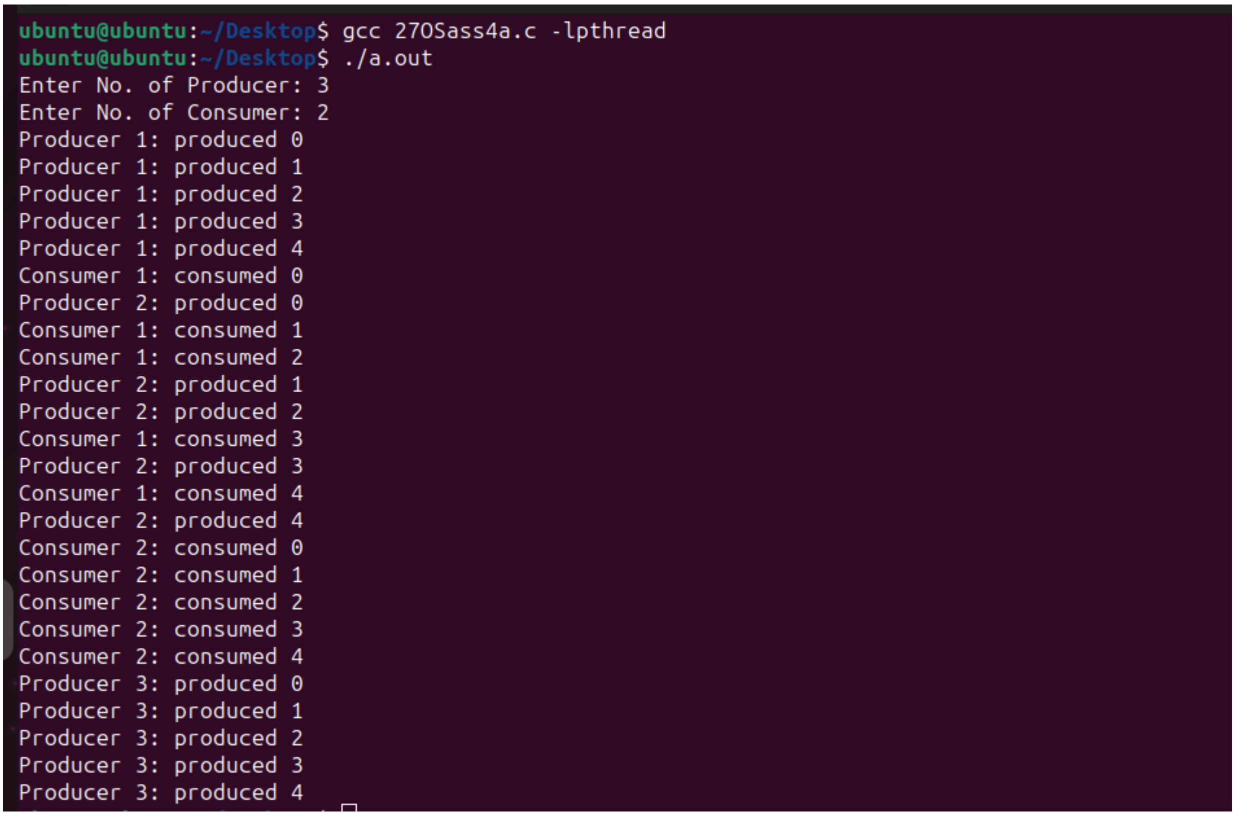
Producer-Consumer

```
#include <pthread.h>
#include <semaphore.h>
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>

#define MaxItems 5 // Maximum items a producer can produce or a consumer can consume
#define BufferSize 5 // Size of the buffer
#define MaxSize 5
sem_t empty;
sem_t full;
int in = 0;
int out = 0;
int buffer[BufferSize];
pthread_mutex_t mutex;

void *producer(void *pno)
{
        int item;
        for(int i = 0; i < MaxItems; i++) {
        item = rand(); // Produce an random item
        sem_wait(&empty);
        pthread_mutex_lock(&mutex);
        buffer[in] = item;
        printf("Producer %d: produced %d\n", *((int *)pno),in);
        in = (in+1)%BufferSize;
        pthread_mutex_unlock(&mutex);
        sem_post(&full);
        }
}
void *consumer(void *cno)
```

```
{
    for(int i = 0; i < MaxItems; i++) {
    sem_wait(&full);
    pthread_mutex_lock(&mutex);
    int item = buffer[out];
    printf("Consumer %d: consumed %d\n",*((int *)cno), out);
    out = (out+1)%BufferSize;
    pthread_mutex_unlock(&mutex);
    sem_post(&empty);
    }
}

int main()
{
    int pn,cn;
    pthread_t pro[5],con[5];
    pthread_mutex_init(&mutex, NULL);
    sem_init(&empty,0,BufferSize);
    sem_init(&full,0,0);
    printf("Enter No. of Producer: ");
    scanf("%d",&pn);
    printf("Enter No. of Consumer: ");
    scanf("%d",&cn);
    int cArr[MaxSize];
    int pArr[MaxSize];
    for(int i = 0; i < pn; i++) {
        pArr[i]=i+1;
        pthread_create(&pro[i], NULL, (void *)producer, (void *)&pArr[i]);
        sleep(2);//optional by omkar
    }
    for(int i = 0; i < cn; i++) {
        cArr[i]=i+1;
        pthread_create(&con[i], NULL, (void *)consumer, (void *)&cArr[i]);
        sleep(2);//optional by omkar
    }

    for(int i = 0; i < 5; i++) {
    pthread_join(pro[i], NULL);
    }

    for(int i = 0; i < 5; i++) {
    pthread_join(con[i], NULL);
    }

    pthread_mutex_destroy(&mutex);
```

NAME: Kaustubh Gajanan Indulkar
TE-IT 25027 (35026)

```
        sem_destroy(&empty);
        sem_destroy(&full);

        return 0;

}
```



```
ubuntu@ubuntu:~/Desktop$ gcc 27OSass4a.c -lpthread
ubuntu@ubuntu:~/Desktop$ ./a.out
Enter No. of Producer: 3
Enter No. of Consumer: 2
Producer 1: produced 0
Producer 1: produced 1
Producer 1: produced 2
Producer 1: produced 3
Producer 1: produced 4
Consumer 1: consumed 0
Producer 2: produced 0
Consumer 1: consumed 1
Consumer 1: consumed 2
Producer 2: produced 1
Producer 2: produced 2
Consumer 1: consumed 3
Producer 2: produced 3
Consumer 1: consumed 4
Producer 2: produced 4
Consumer 2: consumed 0
Consumer 2: consumed 1
Consumer 2: consumed 2
Consumer 2: consumed 3
Consumer 2: consumed 4
Producer 3: produced 0
Producer 3: produced 1
Producer 3: produced 2
Producer 3: produced 3
Producer 3: produced 4
```