Practical 6

Name: Rahul Baser

Roll No.: A4-75

Aim: Producer and Consumer problem

Code:

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


#define MaxItems 5
#define BufferSize 5

sem_t empty;
sem_t full;
int in = 0;
int out = 0;
int buffer[BufferSize];
pthread_mutex_t mutex;

void initialize(){
    sem_init(&empty,0,BufferSize);
    sem_init(&full,0,0);
}



void *producer(void *pno)
{
    int item;
    for(int i = 0; i < MaxItems; i++)
    { item = rand()/1000000;
    sem_wait(&empty);
```

```c
        pthread_mutex_lock(&mutex);
        buffer[in] = item;
        printf("Producer Insert Item %d at %d\n", buffer[in],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 Remove Item %d from %d\n", item, out);
        out = (out+1)%BufferSize;
        pthread_mutex_unlock(&mutex);
        sem_post(&empty);
        }
}

int main(){

        initialize();
        pthread_t pro[5],con[5];
        pthread_mutex_init(&mutex, NULL);

        int a[5] = {10, 20, 30, 40, 50};

        for(int i = 0; i < 1; i++) {
        pthread_create(&pro[i], NULL, (void *)producer, (void
*)&a[i]);
        }
        for(int i = 0; i < 1; i++) {
        pthread_create(&con[i], NULL, (void *)consumer, (void
*)&a[i]);
        }

        for(int i = 0; i < 1; i++) {
        pthread_join(pro[i], NULL);
        }
```
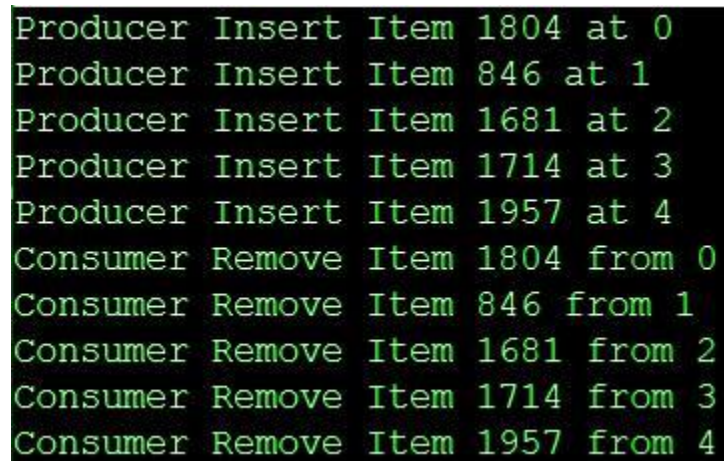
```
    for(int i = 0; i < 1; i++) {
    pthread_join(con[i], NULL);
    }

    return 0;

}
```

Output:

```
Producer Insert Item 1804 at 0
Producer Insert Item 846 at 1
Producer Insert Item 1681 at 2
Producer Insert Item 1714 at 3
Producer Insert Item 1957 at 4
Consumer Remove Item 1804 from 0
Consumer Remove Item 846 from 1
Consumer Remove Item 1681 from 2
Consumer Remove Item 1714 from 3
Consumer Remove Item 1957 from 4
```