Assignment No .4 - A

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <semaphore.h>
#include <unistd.h>
  int *buffer;
int in = 0, out = 0;
int BS;
int item = 0;
sem_t empty, full;
pthread_mutex_t mutex;
void *producer(void *arg)
{
  int id = *((int *)arg);
  while (1)
  {
     item++;
     sem_wait(&empty);
     pthread_mutex_lock(&mutex);
     buffer[in] = item;
     printf("Producer %d produced item: %d at index %d\n", id, item,
          in);
     in = (in + 1) % BS;
     pthread_mutex_unlock(&mutex);
     sem_post(&full);
     sleep(1);
  }
  return NULL;
}
void *consumer(void *arg)
{
  int id = *((int *)arg);
  while (1)
  {
     sem_wait(&full);
     pthread_mutex_lock(&mutex);
out);
  }
  int item = buffer[out];
printf("Consumer %d consumed item : %d at index %d\n", id, item,
out = (out + 1) % BS;
pthread_mutex_unlock(&mutex);
sem_post(&empty);
sleep(1);
return NULL;
}
int main()
{
  int num_producers, num_consumers;
  printf("Enter buffer size: ");
  scanf("%d", &BS);
  printf("Enter number of producers: ");
  scanf("%d", &num_producers);
  printf("Enter number of consumers: ");
  scanf("%d", &num_consumers);
  buffer = (int *)malloc(sizeof(int) * BS);
  pthread_t producers[num_producers], consumers[num_consumers];
  int prod_ids[num_producers], cons_ids[num_consumers];
  sem_init(&empty, 0, BS);
  sem_init(&full, 0, 0);
  pthread_mutex_init(&mutex, NULL);
  for (int i = 0; i < num_producers; i++)
  {
    prod_ids[i] = i + 1;
    pthread_create(&producers[i], NULL, producer, &prod_ids[i]);
  }
  for (int i = 0; i < num_consumers; i++)
  {
    cons_ids[i] = i + 1;
    pthread_create(&consumers[i], NULL, consumer, &cons_ids[i]);
    for (int i = 0; i < num_producers; i++)
    {
      pthread_join(producers[i], NULL);
    }
  }
  for (int i = 0; i < num_consumers; i++)
  {
    pthread_join(consumers[i], NULL);
  }
  pthread_mutex_destroy(&mutex);
  sem_destroy(&empty);
  sem_destroy(&full);
  free(buffer);
  return 0;
}
```

<u>Output:-</u>

```
Enter buffer size: 7
Enter number of producers: 2
Enter number of consumers: 3
Producer 1 produced item: 1 at index 0
Producer 2 produced item: 2 at index 1
Consumer 1 consumed item : 1 at index 0
Consumer 2 consumed item : 2 at index 1
Producer 1 produced item: 3 at index 2
Producer 2 produced item: 4 at index 3Consumer 1 consumed item : 3 at index 2
Consumer 3 consumed item : 4 at index 3
Producer 2 produced item: 5 at index 4
Producer 1 produced item: 6 at index 5
Consumer 1 consumed item : 5 at index 4
Consumer 2 consumed item : 6 at index 5
Producer 2 produced item: 7 at index 6
Producer 1 produced item: 8 at index 0
Consumer 1 consumed item : 7 at index 6
Consumer 2 consumed item : 8 at index 0
Producer 2 produced item: 9 at index 1
Consumer 2 consumed item : 9 at index 1
Producer 1 produced item: 10 at index 2
Consumer 3 consumed item : 10 at index 2
Producer 2 produced item: 11 at index 3
Producer 1 produced item: 12 at index 4
Consumer 1 consumed item : 11 at index 3
Consumer 3 consumed item : 12 at index 4
Producer 2 produced item: 13 at index 5
Consumer 3 consumed item : 13 at index 5
Producer 1 produced item: 14 at index 6
Consumer 1 consumed item : 14 at index 6
Producer 2 produced item: 15 at index 0
Consumer 2 consumed item : 15 at index 0
Producer 1 produced item: 16 at index 1
Consumer 1 consumed item : 16 at index 1
Producer 2 produced item: 17 at index 2
Consumer 2 consumed item : 17 at index 2
Producer 1 produced item: 18 at index 3
Consumer 1 consumed item : 18 at index 3
Producer 2 produced item: 19 at index 4
Producer 1 produced item: 20 at index 5
```