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

int buffer[5], size = 5, index1 = 0, a;
sem_t E, F;
pthread_mutex_t S;
pthread_t con[5], pro[5];

int insert(int item)
{
        if(index1 < size)
        {
                buffer[index1] = item;
                index1++;
        }
        else
        {
                printf("\nBuffer Overflow");
        }
}

int delete()
{
        if(index1 > 0)
        {
                index1--;
        }
        else
        {
                printf("\nBuffer Underflow");
        }
}

void *producer()
{
        sleep(2);
        sem_wait(&E);
        a = rand()/100000000;//  You can read a value using any input functions and insert
it
        pthread_mutex_lock(&S);
        insert(a);
        pthread_mutex_unlock(&S);
        printf("\nProducer produces item : %d",a);
        sem_post(&F);
}

void *consumer()
{
        sem_wait(&F);
        pthread_mutex_lock(&S);
        delete();
        pthread_mutex_unlock(&S);
        printf("\nConsumer consumes item : %d\n",a);
        sem_post(&E);
}

int main()
{
        int i;
        sem_init(&E,1,size);
        sem_init(&F,1,0);
```

```
        pthread_mutex_init(&S,NULL);

        for(i=0; i<size; i++)
        {
                pthread_create(&pro[i], NULL, producer, NULL);
                pthread_create(&con[i], NULL, consumer, NULL);
        }

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