```
#include<stdio.h>
#include<semaphore.h>
#include<pthread.h>
#include<stdlib.h>
#define buffersize 100

pthread_mutex_t m;
pthread_t tidp[20],tidc[20];
sem_t full,empty;
int counter=0;
int buffer [buffersize];
void initialize()
{
pthread_mutex_init(&m,NULL);
sem_init(&full,1,0);
sem_init(&empty,1,buffersize);
}
void write(int item)
{
buffer[counter++]=item;
}
int read()
{
return (buffer[--counter]);
}
void * producer (void *p)
{
int waittime;
int item=rand();
sem_wait(&empty);
pthread_mutex_lock(&m);
printf("\n\np[%d] item is produced by producer",item);
write(item);
pthread_mutex_unlock(&m);
sem_post(&full);
}
void * consumer (void *c)
{
int waittime;
int item=rand();
sem_wait(&full);
pthread_mutex_lock(&m);
item=read();
printf("\n\nC[%d] item is consumed by consumer",item);
pthread_mutex_unlock(&m);
sem_post(&empty);
}
int main()
{
int n1,n2,i;
initialize();
printf("\n  Enter the no of Producer: ");
scanf("%d",&n1);
printf("enter the no of consumers: ");
scanf("%d",&n2);
for(i=0;i<n1;i++)
pthread_create(&tidp[i],NULL,producer,NULL);
for(i=0;i<n2;i++)
pthread_create(&tidc[i],NULL,consumer,NULL);
for(i=0;i<n1;i++)
pthread_join(tidp[i],NULL);
for(i=0;i<n2;i++)
pthread_join(tidc[i],NULL);
printf("\n");
exit(0);
}
```

```
OUTPUT:-
[root@localhost Documents]# gcc Producer.c -lpthread
[root@localhost Documents]# ./a.out

  Enter the no of Producer: 5
enter the no of consumers: 5


p[1804289383] item is produced by producer

p[846930886] item is produced by producer

C[846930886] item is consumed by consumer

p[1714636915] item is produced by producer

p[1957747793] item is produced by producer

C[1957747793] item is consumed by consumer

C[1714636915] item is consumed by consumer

p[719885386] item is produced by producer

C[719885386] item is consumed by consumer

C[1804289383] item is consumed by consumer
[root@localhost Documents]#
```