Consumer Program

```
#include<stdio.h>
#include<unistd.h>
#include<pthread.h>
#include<semaphore.h>
#include<stdlib.h>
#define maxsize 20
typedef struct
{
        int in,out;
        int list[maxsize];
        sem_t full;
        sem_t emp;
        pthread_mutex_t lock;
}Itemlist;
Itemlist A;
int item,size;
void *prod(void *arg);
void *cust(void *arg);
void init();
void main()
{
        int np,nc,i;
        init();
        pthread_t pd[5],cus[5];
        printf("\nEnter no of producers ");
        scanf("%d",&np);
        printf("\nEnter no of customers ");
        scanf("%d",&nc);
        printf("\nhow many items to be produced ");
        scanf("%d",&size);
        for(i=0;i<np;i++)
        {
                int *arg=malloc(sizeof(int *));
                *arg=i;
                pthread_create(&pd[i],NULL,prod,arg);
                printf("\nproducer thread %d is created",i+1);
        }
        for(i=0;i<nc;i++)
        {
                int *arg=malloc(sizeof(int *));
                *arg=i;
                pthread_create(&cus[i],NULL,cust,arg);
                printf("\ncustomer thread %d is created",i+1);
        }
        for(i=0;i<np;i++)
        {
                pthread_join(pd[i],NULL);
                printf("\nproducer thread %d is finished",i+1);
        }
        for(i=0;i<nc;i++)
```

```
	{
		pthread_join(cus[i],NULL);
		printf("\ncustomer thread %d is finished",i+1);
	}
}
void *prod(void *arg)
{
	int i=*(int *)arg;
	while(item<size+1)
	{
		sem_wait(&A.emp);
		pthread_mutex_lock(&A.lock);
		printf("\nproducer %d has produced item %d ",i+1,item);
		A.list[(A.in++)%maxsize]=item++;
		pthread_mutex_unlock(&A.lock);
		sem_post(&A.full);
		sleep(2);
	}

}
void *cust(void *arg)
{
	int i=*(int *)arg;
	while(1)
	{
		sem_wait(&A.full);
		pthread_mutex_lock(&A.lock);
		printf("\ncustomer %d purchased item %d ",i+1,A.list[(A.out++)%maxsize]);
		pthread_mutex_unlock(&A.lock);
		sem_post(&A.emp);
	}

}
void init()
{
	A.in=0;A.out=0;
	sem_init(&A.full,0,0);
	sem_init(&A.emp,0,maxsize);
	item=1;
	pthread_mutex_init(&A.lock,NULL);
}
```

OUTPUT:

```
pl-lab@pllab-OptiPlex-3000:~$ gcc consumer.c
pl-lab@pllab-OptiPlex-3000:~$ ./a.out

Enter no of producers 4

Enter no of customers 4

how many items to be produced 1
```

```
producer thread 1 is created
producer thread 2 is created
producer 1 has produced item 1
producer thread 3 is created
producer thread 4 is created
customer thread 1 is created
customer thread 2 is created
customer 2 purchased item 1
customer thread 3 is created
customer thread 4 is created
producer thread 1 is finished
producer thread 2 is finished
producer thread 3 is finished

pl-lab@pllab-OptiPlex-3000:~$ gcc consumer.c
pl-lab@pllab-OptiPlex-3000:~$ ./a.out

Enter no of producers 4

Enter no of customers 4

how many items to be produced 2

producer thread 1 is created
producer thread 2 is created
producer 1 has produced item 1
producer thread 3 is created
producer 3 has produced item 2
producer 2 has produced item 3
producer thread 4 is created
customer thread 1 is created
customer thread 2 is created
customer 1 purchased item 1
customer 1 purchased item 2
customer 1 purchased item 3
customer thread 3 is created
customer thread 4 is created
producer thread 1 is finished
producer thread 2 is finished
producer thread 3 is finished
```