#include<stdio.h>

#include<pthread.h>

#include<semaphore.h>

#include<stdio.h>

#include<unistd.h>

void \*producer(void \* thread);

void \*consumer(void \* thread);

int count=0,in=0,out=0,a[5];

sem\_t full;

sem\_t empty;

pthread\_mutex\_t mutex;

int main()

{

int i,j,p,c;

pthread\_t pid[10],cid[10];

pthread\_mutex\_init(&mutex,0);

sem\_init(&full,0,0);

sem\_init(&empty,0,5);

printf("\nEnter no of producers:");

scanf("%d",&p);

printf("\nEnter no of consumers:");

scanf("%d",&c);

for(i=0;i<p;i++)

pthread\_create(&pid[i],NULL,(void\*)producer,&i);

for(i=0;i<c;i++)

pthread\_create(&cid[i],NULL,(void\*)consumer,&i);

for(i=0;i<p;i++)

pthread\_join(pid[i],NULL);

for(i=0;i<c;i++)

pthread\_join(cid[i],NULL);

sem\_destroy(&full);

sem\_destroy(&empty);

pthread\_mutex\_destroy(&mutex);

}

void \*producer(void \* thread)

{

int t=\*(int \*)thread;

while(1)

{

sem\_wait(&empty);

pthread\_mutex\_lock(&mutex);

if(count>=5)

printf("\nBuffer is full");

else

{

a[in]=rand()%100;

printf("\nproducer %d produced:%d",t,a[in]);

in=(in+1)%5;

count++;

}

pthread\_mutex\_unlock(&mutex);

sem\_post(&full);

sleep(1);

}

pthread\_exit(0);

}

void \*consumer(void \*thread)

{

int t=\*(int \*)thread;

while(1)

{

sem\_wait(&full);

pthread\_mutex\_lock(&mutex);

if(count<=0)

printf("\nBuffer is empty");

else

{

printf("\nconsumer %d consumed:%d",t,a[in]);

out=(out+1)%5;

count--;

}

pthread\_mutex\_unlock(&mutex);

sem\_post(&empty);

sleep(1);

}

pthread\_exit(0);

}