#include<pthread.h>

#include<semaphore.h>

#include<unistd.h>

#include<stdio.h>

#include<stdlib.h>

sem\_t e,f,s; // e,f-counting semaphor , s-binary semaphor(as only one (producer/consumer) of them is allowed to access the buffer space)

int buff[5];

int in=0,out=0; // in-index used by producer , out-index used by the consumer.

void\* producer (void\* arg){

sem\_wait(&e); // to decrement empty count

sem\_wait(&s);

int data=rand();

buff[in]=data;

printf("\nProduct %d entered by %lu successfully\n",data,pthread\_self());

in = (in+1)%5;

sem\_post(&s);

sem\_post(&f); // to increment the filled cells count

sleep(5);

return NULL;

}

void\* consumer (void\* arg) {

int conti;

int data;

do{

sem\_wait(&f); // to decrement filled count

sem\_wait(&s);

int data = buff[out];

printf("\nProduct %d consumed by %lu successfully\n",data,pthread\_self());

out = (out+1)%5;

sem\_post(&s);

sem\_post(&e); // to increment the empty cells countsem\_getvalue(&e,&conti);

}while( conti!=5 );

return NULL;

}

void main() {

pthread\_t tid[5] ,ctid;

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

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

sem\_init(&s,0,1);

for(int i=0 ; i<5 ;i++)

{

pthread\_create ( &tid[i] , NULL , producer ,NULL );

}

pthread\_create ( &ctid , NULL , consumer ,NULL );

for(int i=0 ; i<5 ; i++) {

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

}

pthread\_join ( ctid ,NULL );

}

OUTPUT

