Lab Assignment-5

Name: Sadman Sharif

ID: 20101107

Section: 02

Course: CSE321

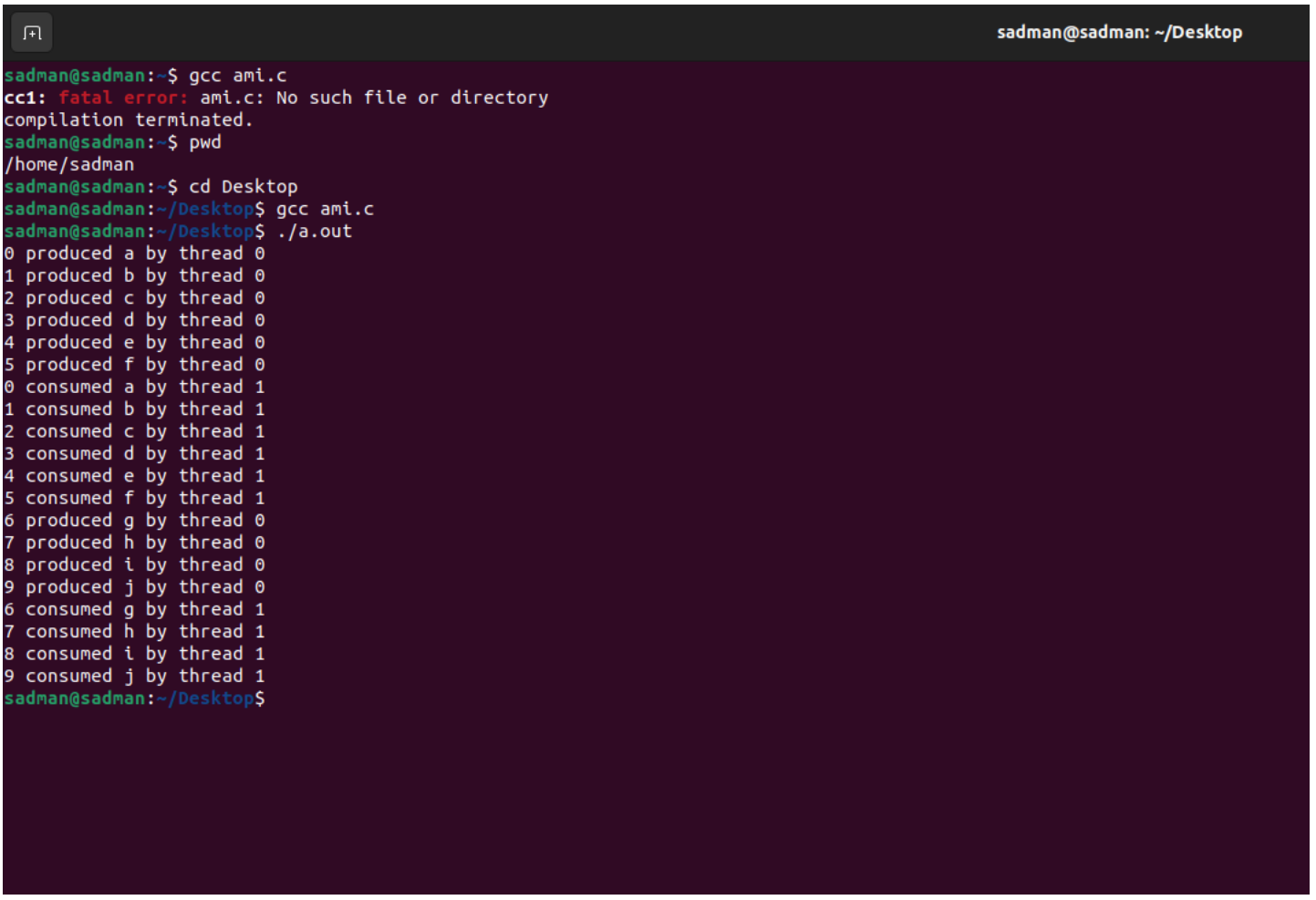
Submission Date: 08/12/2022

Task-1

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

#define Max 10
#define BufLen 6
#define NUMTHREAD 2

sem_t empty;
sem_t full;

int in= 0,pro=0,con=0,out=0;

int buffer[BufLen];

char item ='a';

pthread_mutex_t mutex;

void *producer(void *pno){


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

            sem_wait(&empty);
            pthread_mutex_lock(&mutex);
            buffer[in]= item;
            printf("%d produced %c by thread %d\n",pro,buffer[in],*((int
*)pno));

            in = (in+1) % BufLen;
            pro=pro+1;
            item+= 1;

            pthread_mutex_unlock(&mutex);
            sem_post(&full);
      }

}
```

```
void *consumer(void *cno){

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

            sem_wait(&full);
            pthread_mutex_lock(&mutex);
            int item = buffer[out];

            printf("%d consumed %c by thread %d\n",con,item,*((int
*)cno));

            con=con+1 ;
            out=(out+1)%BufLen;

            pthread_mutex_unlock(&mutex);
            sem_post(&empty);
      }

}

int main(){

      pthread_t p[10],c[10];

      pthread_mutex_init(&mutex, NULL);
      sem_init(&empty,0,BufLen);
      sem_init(&full,0,0);

      int p_ref[1] = {0};
      int c_ref[1] = {1};

      pthread_create(&p[0],NULL, (void *)producer,(void *)&p_ref[0]);

      pthread_create(&c[0], NULL, (void *)consumer,(void *)&c_ref[0]);

      pthread_join(c[0],NULL);
      pthread_join(p[0],NULL);

      pthread_mutex_destroy(&mutex);

      sem_destroy(&empty);
      sem_destroy(&full);

    return 0;
}
```

Task-1 (Output)



Task-2

```
#include <pthread.h>
#include <semaphore.h>
#include <stdio.h>

#define MaxCrops 5
#define warehouseSize 5

sem_t empty;
sem_t full;

int in=0,out=0;
char
crops[warehouseSize]={'R','W','P','S','M'},warehouse[warehouseSize]={'
N','N','N','N','N'};

pthread_mutex_t mutex;

void *Farmer(void *farm)
{
      for(int i=0;i<MaxCrops;i++){
            char item=crops[in];
            sem_wait(&empty);
            pthread_mutex_lock(&mutex);
            warehouse[in]=item;
            printf("Farmer %d: Insert crops %c at %d\n",*((int
*)farm),warehouse[in],in);
            in=(in+1)%warehouseSize;
            pthread_mutex_unlock(&mutex);
            sem_post(&full);
      }
}

void *ShopOwner(void *shop)
{

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

            sem_wait(&full);
            pthread_mutex_lock(&mutex);
            char item = warehouse[out];
            printf("Shop Owner %d: Removes crops %c from %d\n",*((int
*)shop),item,out);
```

```
            warehouse[out]='N';
            out=(out+1)%warehouseSize;
            printf("ShopOwner%d:",*((int*)shop));
            for(int i=0;i<warehouseSize;i++){
                  printf("%c",warehouse[i]);
            }
            printf("\n");

            pthread_mutex_unlock(&mutex);
            sem_post(&empty);
      }

}

int main(){

    pthread_t Far[5],Sho[5];
    pthread_mutex_init(&mutex, NULL);
    sem_init(&empty,0,warehouseSize);
      sem_init(&full,0,0);

    int a[5]={1,2,3,4,5};

    for(int i=0;i<5;i++){
            pthread_create(&Far[i],NULL,(void *)Farmer,(void *)&a[i]);
      }
      for(int i=0;i<5;i++){
            pthread_create(&Sho[i], NULL,(void *)ShopOwner,(void
*)&a[i]);
      }
      for(int i=0;i<5;i++){
            pthread_join(Far[i],NULL);
      }
      for(int i=0;i<5;i++){
            pthread_join(Sho[i],NULL);
      }


    pthread_mutex_destroy(&mutex);
    sem_destroy(&empty);
    sem_destroy(&full);

    return 0;

}
```

Task-2 (Output)