Assessment-3

Winter Sem 2020-21

Name : Jogalekar Ishan Sagar

Reg no : 19BCE2250

Course : CSE2005 - Operating Systems Lab

Slot : L35+L36

Process Synchronization

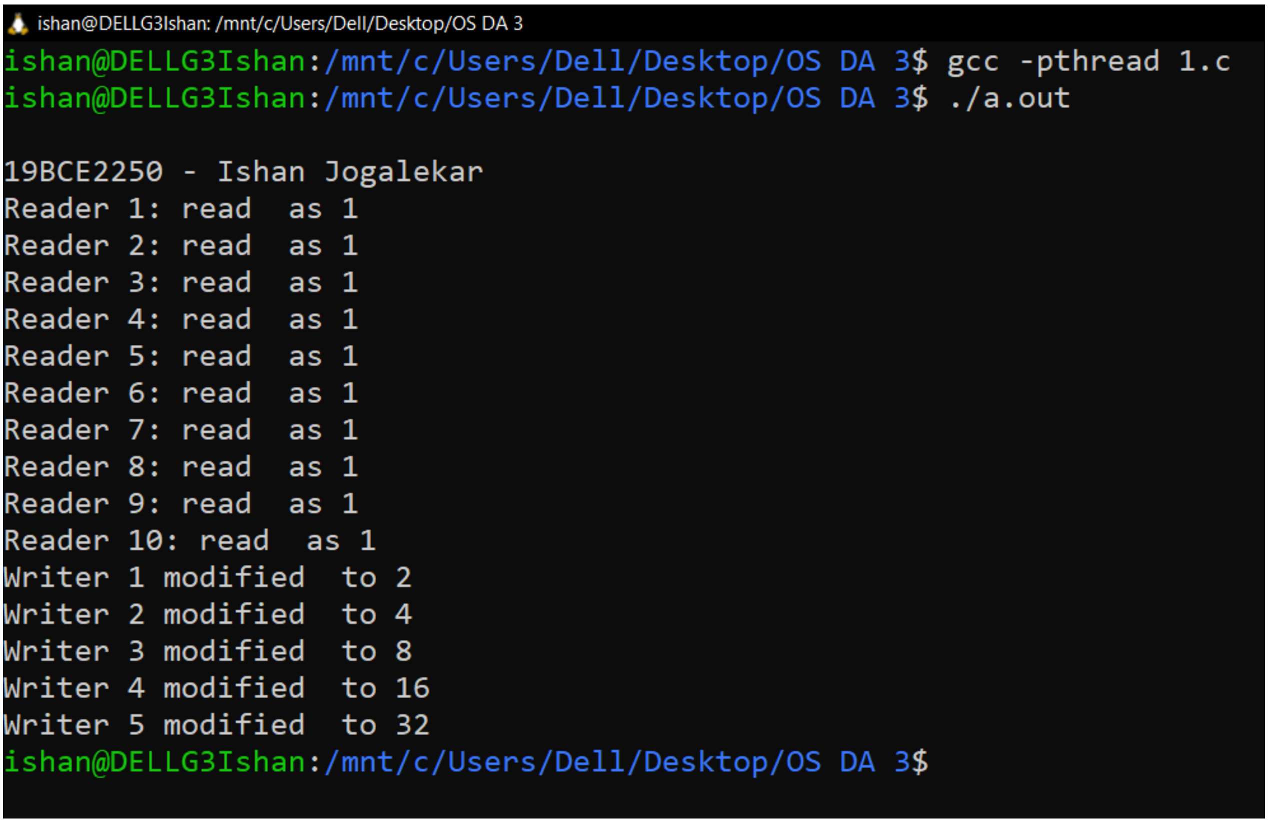
(a) Implement the solution for reader – writer's problem.

Ans :

Code –

```
#include <pthread.h>

#include <semaphore.h>

#include <stdio.h>

sem_t wrt;

pthread_mutex_t mutex;

int cnt = 1;

int numreader = 0;


void *writer(void *wno)

{

    sem_wait(&wrt);

    cnt = cnt*2;

    printf("Writer %d modified  to %d\n",(*((int *)wno)),cnt);
    sem_post(&wrt);

}
void *reader(void *rno)
{

    pthread_mutex_lock(&mutex);
    numreader++;
    if(numreader == 1) {
        sem_wait(&wrt);
    }
    pthread_mutex_unlock(&mutex);

    printf("Reader %d: read  as %d\n",*((int *)rno),cnt);


    pthread_mutex_lock(&mutex);
    numreader--;
    if(numreader == 0) {
        sem_post(&wrt);
    }
    pthread_mutex_unlock(&mutex);
}

int main()
{
        printf("\n19BCE2250 - Ishan Jogalekar\n");

        pthread_t read[10],write[5];
        pthread_mutex_init(&mutex, NULL);
        sem_init(&wrt,0,1);

        int a[10] = {1,2,3,4,5,6,7,8,9,10};

        for(int i = 0; i < 10; i++) {
            pthread_create(&read[i], NULL, (void *)reader, (void *)&a[i]);
        }
        for(int i = 0; i < 5; i++) {
            pthread_create(&write[i], NULL, (void *)writer, (void *)&a[i]);
        }

        for(int i = 0; i < 10; i++) {
            pthread_join(read[i], NULL);
        }
        for(int i = 0; i < 5; i++) {
            pthread_join(write[i], NULL);
        }
       pthread_mutex_destroy(&mutex);
       sem_destroy(&wrt);
       return 0;
}
```

Output :



```
ishan@DELLG3Ishan: /mnt/c/Users/Dell/Desktop/OS DA 3
ishan@DELLG3Ishan:/mnt/c/Users/Dell/Desktop/OS DA 3$ gcc -pthread 1.c
ishan@DELLG3Ishan:/mnt/c/Users/Dell/Desktop/OS DA 3$ ./a.out

19BCE2250 - Ishan Jogalekar
Reader 1: read  as 1
Reader 2: read  as 1
Reader 3: read  as 1
Reader 4: read  as 1
Reader 5: read  as 1
Reader 6: read  as 1
Reader 7: read  as 1
Reader 8: read  as 1
Reader 9: read  as 1
Reader 10: read  as 1
Writer 1 modified  to 2
Writer 2 modified  to 4
Writer 3 modified  to 8
Writer 4 modified  to 16
Writer 5 modified  to 32
ishan@DELLG3Ishan:/mnt/c/Users/Dell/Desktop/OS DA 3$
```

(b) Implement the solution for dining philosopher's problem.

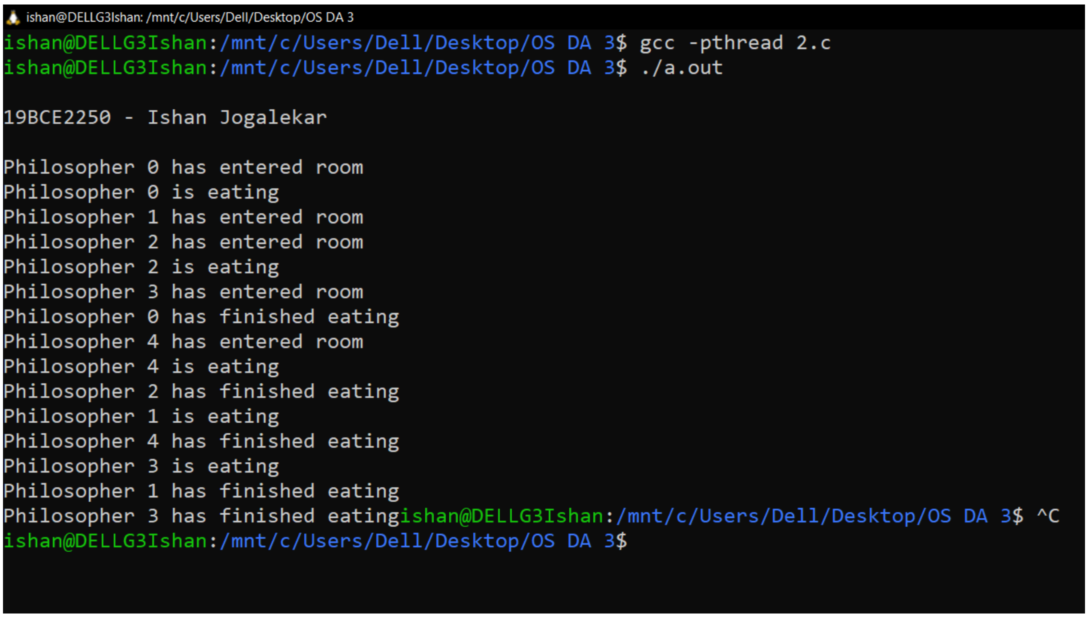
Ans :-

Code –

```
#include<stdio.h>

#include<stdlib.h>

#include<pthread.h>

#include<semaphore.h>

#include<unistd.h>


sem_t room;

sem_t chopstick[5];
```

```
void * philosopher(void *);
void eat(int);
int main()
{
    printf("\n19BCE2250 - Ishan Jogalekar\n");
        int i,a[5];
        pthread_t tid[5];

        sem_init(&room,0,4);

        for(i=0;i<5;i++)
                sem_init(&chopstick[i],0,1);

        for(i=0;i<5;i++){
                a[i]=i;
                pthread_create(&tid[i],NULL,philosopher,(void *)&a[i]);
        }
        for(i=0;i<5;i++)
                pthread_join(tid[i],NULL);
}

void * philosopher(void * num)
{
        int phil=*(int *)num;
        sem_wait(&room);
        printf("\nPhilosopher %d has entered room",phil);
        sem_wait(&chopstick[phil]);
        sem_wait(&chopstick[(phil+1)%5]);

        eat(phil);
        sleep(2);
        printf("\nPhilosopher %d has finished eating",phil);

        sem_post(&chopstick[(phil+1)%5]);
        sem_post(&chopstick[phil]);
        sem_post(&room);
}
void eat(int phil){
        printf("\nPhilosopher %d is eating",phil);}
```

<u>Output</u> :



```
ishan@DELLG3Ishan:/mnt/c/Users/Dell/Desktop/OS DA 3$ gcc -pthread 2.c
ishan@DELLG3Ishan:/mnt/c/Users/Dell/Desktop/OS DA 3$ ./a.out

19BCE2250 - Ishan Jogalekar

Philosopher 0 has entered room
Philosopher 0 is eating
Philosopher 1 has entered room
Philosopher 2 has entered room
Philosopher 2 is eating
Philosopher 3 has entered room
Philosopher 0 has finished eating
Philosopher 4 has entered room
Philosopher 4 is eating
Philosopher 2 has finished eating
Philosopher 1 is eating
Philosopher 4 has finished eating
Philosopher 3 is eating
Philosopher 1 has finished eating
Philosopher 3 has finished eatingishan@DELLG3Ishan:/mnt/c/Users/Dell/Desktop/OS DA 3$ ^C
ishan@DELLG3Ishan:/mnt/c/Users/Dell/Desktop/OS DA 3$
```

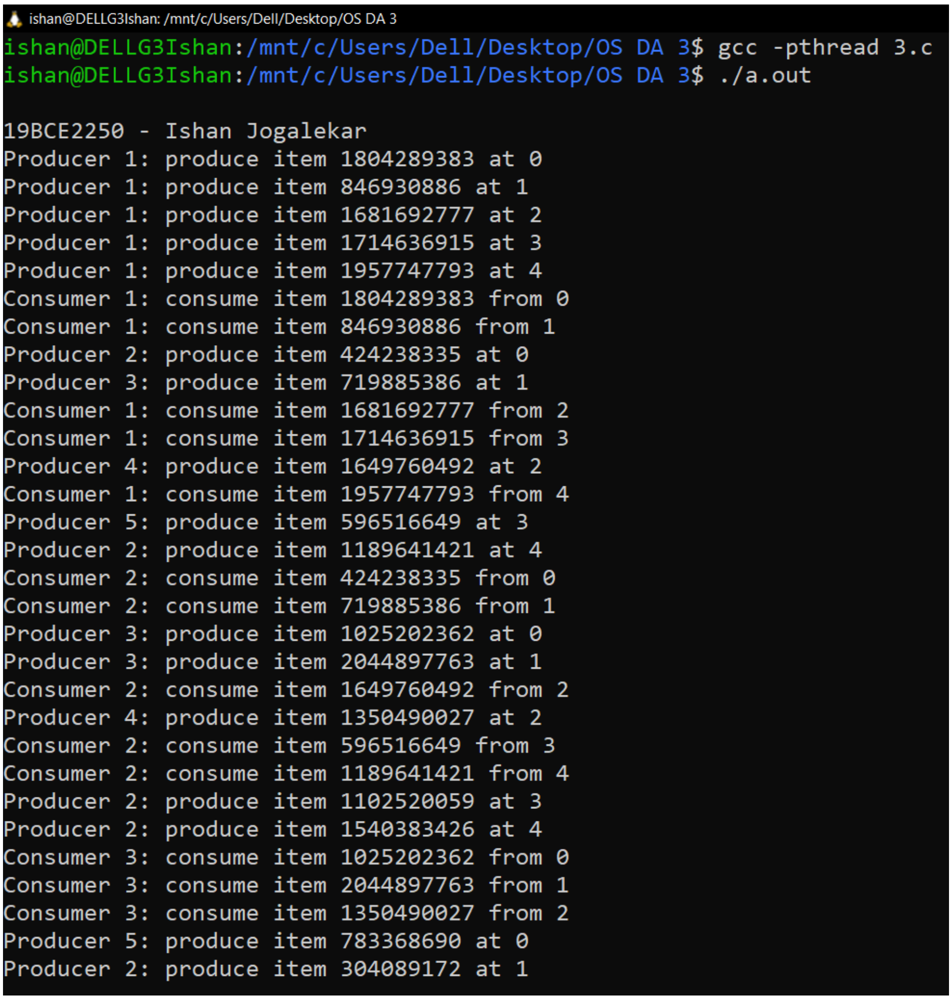
(c) Implement the solution for producer consumer problem

Ans:-

Code –

```
#include <pthread.h>
#include <semaphore.h>
#include <stdlib.h>
#include <stdio.h>
#define MaxItems 5
#define BufferSize 5

sem_t empty;
sem_t full;
int in = 0;
int out = 0;
int buffer[BufferSize];
pthread_mutex_t mutex;

void *producer(void *pno)
{
    int item;
    for(int i = 0; i < MaxItems; i++) {
        item = rand();
        sem_wait(&empty);
        pthread_mutex_lock(&mutex);
        buffer[in] = item;
        printf("Producer %d: produce item %d at %d\n", *((int
*)pno),buffer[in],in);
        in = (in+1)%BufferSize;
        pthread_mutex_unlock(&mutex);
        sem_post(&full);
    }
}
void *consumer(void *cno)
{
    for(int i = 0; i < MaxItems; i++) {
        sem_wait(&full);
        pthread_mutex_lock(&mutex);
        int item = buffer[out];
        printf("Consumer %d: consume item %d from %d\n",*((int
*)cno),item, out);
        out = (out+1)%BufferSize;
        pthread_mutex_unlock(&mutex);
        sem_post(&empty);
    }
}

int main()
{

    printf("\n19BCE2250 - Ishan Jogalekar\n");

    pthread_t pro[5],con[5];
    pthread_mutex_init(&mutex, NULL);
    sem_init(&empty,0,BufferSize);
    sem_init(&full,0,0);

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

    for(int i = 0; i < 5; i++) {
        pthread_create(&pro[i], NULL, (void *)producer, (void *)&a[i]);
    }
    for(int i = 0; i < 5; i++) {
        pthread_create(&con[i], NULL, (void *)consumer, (void *)&a[i]);
    }

    for(int i = 0; i < 5; i++) {
        pthread_join(pro[i], NULL);
    }
    for(int i = 0; i < 5; i++) {
        pthread_join(con[i], NULL);
    }
   pthread_mutex_destroy(&mutex);
    sem_destroy(&empty);
    sem_destroy(&full);
return 0;
  }
```

Output :



```
ishan@DELLG3Ishan:/mnt/c/Users/Dell/Desktop/OS DA 3$ gcc -pthread 3.c
ishan@DELLG3Ishan:/mnt/c/Users/Dell/Desktop/OS DA 3$ ./a.out

19BCE2250 - Ishan Jogalekar
Producer 1: produce item 1804289383 at 0
Producer 1: produce item 846930886 at 1
Producer 1: produce item 1681692777 at 2
Producer 1: produce item 1714636915 at 3
Producer 1: produce item 1957747793 at 4
Consumer 1: consume item 1804289383 from 0
Consumer 1: consume item 846930886 from 1
Producer 2: produce item 424238335 at 0
Producer 3: produce item 719885386 at 1
Consumer 1: consume item 1681692777 from 2
Consumer 1: consume item 1714636915 from 3
Producer 4: produce item 1649760492 at 2
Consumer 1: consume item 1957747793 from 4
Producer 5: produce item 596516649 at 3
Producer 2: produce item 1189641421 at 4
Consumer 2: consume item 424238335 from 0
Consumer 2: consume item 719885386 from 1
Producer 3: produce item 1025202362 at 0
Producer 3: produce item 2044897763 at 1
Consumer 2: consume item 1649760492 from 2
Producer 4: produce item 1350490027 at 2
Consumer 2: consume item 596516649 from 3
Consumer 2: consume item 1189641421 from 4
Producer 2: produce item 1102520059 at 3
Producer 2: produce item 1540383426 at 4
Consumer 3: consume item 1025202362 from 0
Consumer 3: consume item 2044897763 from 1
Consumer 3: consume item 1350490027 from 2
Producer 5: produce item 783368690 at 0
Producer 2: produce item 304089172 at 1
```

(d) The analogy is based upon a hypothetical barber shop with one barber. There is a barber shop which has one barber, one barber chair, and n chairs for waiting for customers if there are any to sit on the chair.

- If there is no customer, then the barber sleeps in his own chair.
- When a customer arrives, he has to wake up the barber.
- If there are many customers and the barber is cutting a customer’s hair, then the remaining customers either wait if there are empty chairs in the waiting room or they leave if no chairs are empty.

Ans :-

Code –

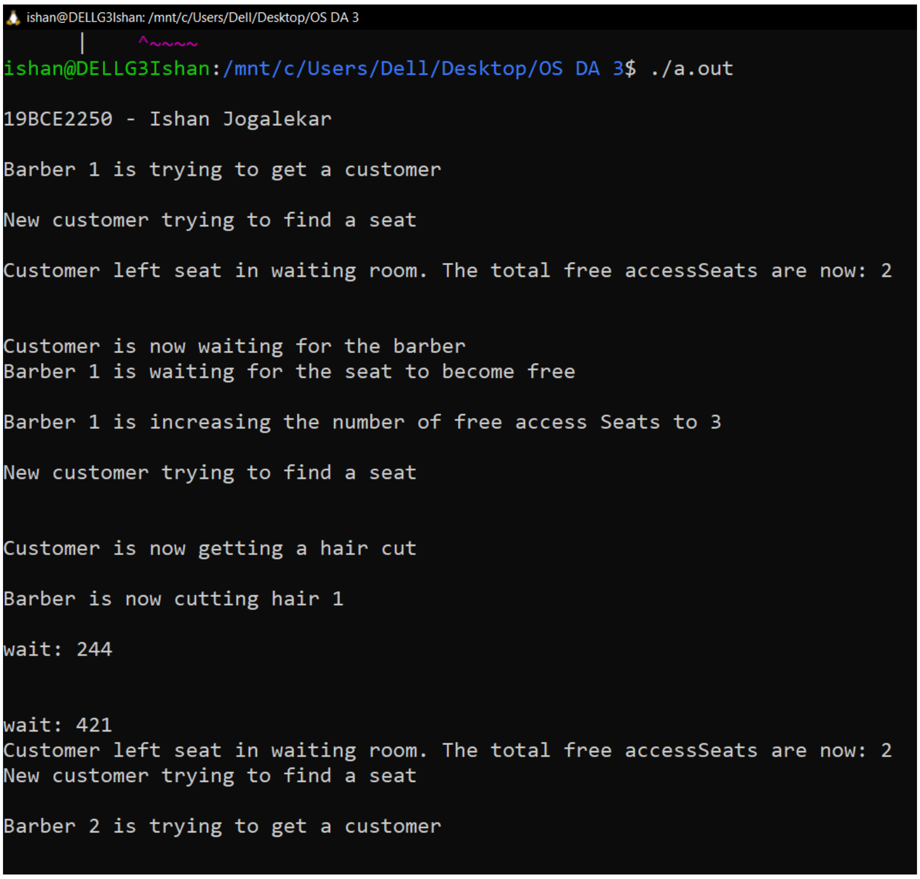
```
#include <stdio.h>
#include <stdlib.h>
#include <sys/stat.h>
#include <time.h>
int accessSeats[2];
int customers[2];
int barber[2];
int freeaccessSeats[2];

void randomWait();
void barber_process();
void customer_process();

void V(int pd[]) {
   int a=1;
   write(pd[1],&a,sizeof(int));
}

void P(int pd[]) {
   int a;
   read(pd[0],&a,sizeof(int));
}

void main() {
   printf("\n19BCE2250 - Ishan Jogalekar\n");
   int i;
```

```
    pipe(accessSeats);
    pipe(customers);
    pipe(barber);
    pipe(freeaccessSeats);

    V(accessSeats);

    int num = 3;
    write(freeaccessSeats[1],&num,sizeof(int));

    if (fork() == 0) {
        srand(time(0)+1);
        barber_process();
        return;
    }

    for (i = 1; i <= 5; i++) {
        if (fork() == 0) {
            srand(time(0)+2*i);
            customer_process();
            return;
        }
    }
    sleep(10);
    printf("\ndone\n\n");
}
```

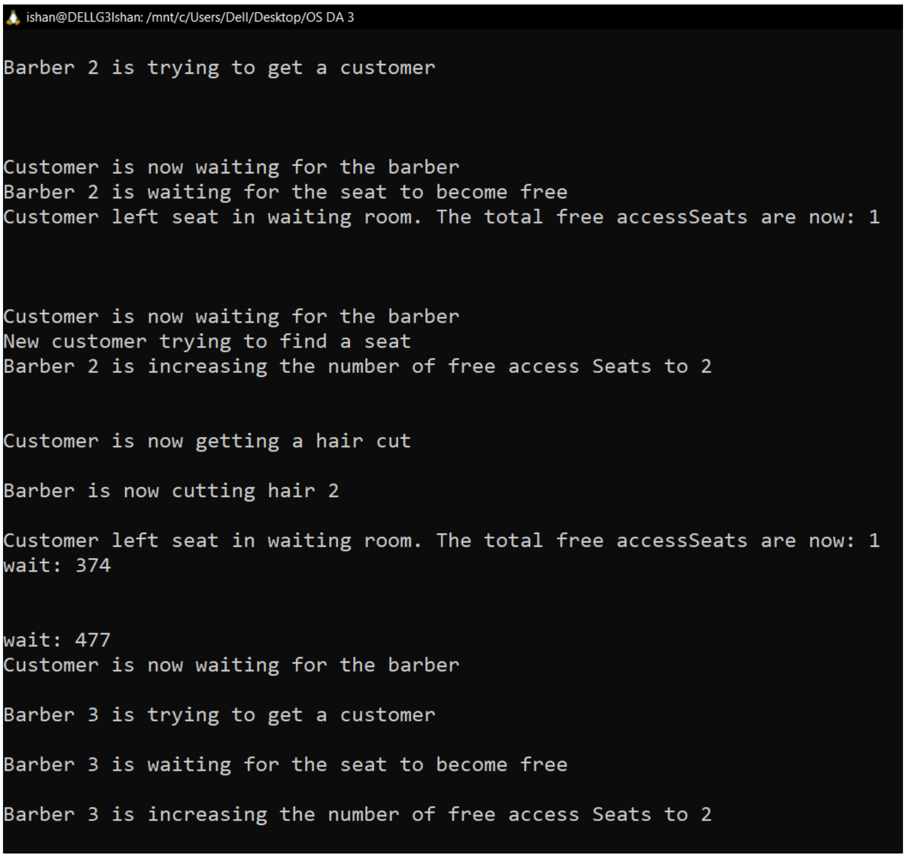
```
void barber_process() {
  int i;
  int num;
  for (i = 1; i <= 10; ++i) {
    printf("\nBarber %d is trying to get a customer\n",i);
    P(customers);
    printf("\nBarber %d is waiting for the seat to become free\n",i);
    P(accessSeats);
    read(freeaccessSeats[0],&num,sizeof(int));
    num++;
    write(freeaccessSeats[1],&num,sizeof(int));
    printf("\nBarber %d is increasing the number of free access Seats to %d\n",i,num);
    V(barber);
    V(accessSeats);
    printf("\nBarber is now cutting hair %d\n",i);
    randomWait();
  }
}

void customer_process() {
  int i;
  int num;
  for (i = 1; i <= 2; ++i) {
    printf("\nNew customer trying to find a seat\n");
    P(accessSeats);
    read(freeaccessSeats[0],&num,sizeof(int));
    if (num > 0)
    {
      num--;
      write(freeaccessSeats[1],&num,sizeof(int));
      printf("\nCustomer left seat in waiting room. The total free
accessSeats are now: %d\n",num);
      V(customers);
      V(accessSeats);
      printf("\nCustomer is now waiting for the barber\n");
      P(barber);
      printf("\nCustomer is now getting a hair cut\n");
    }
    else
    {
      write(freeaccessSeats[1],&num,sizeof(int));
      V(accessSeats);
      printf("\nNo free chairs in waiting room\n");
    }
    randomWait();
  }
}

void randomWait() {
  int delay;
```

```
delay = random() % 500;
printf("\nwait: %d\n", delay);
}
```

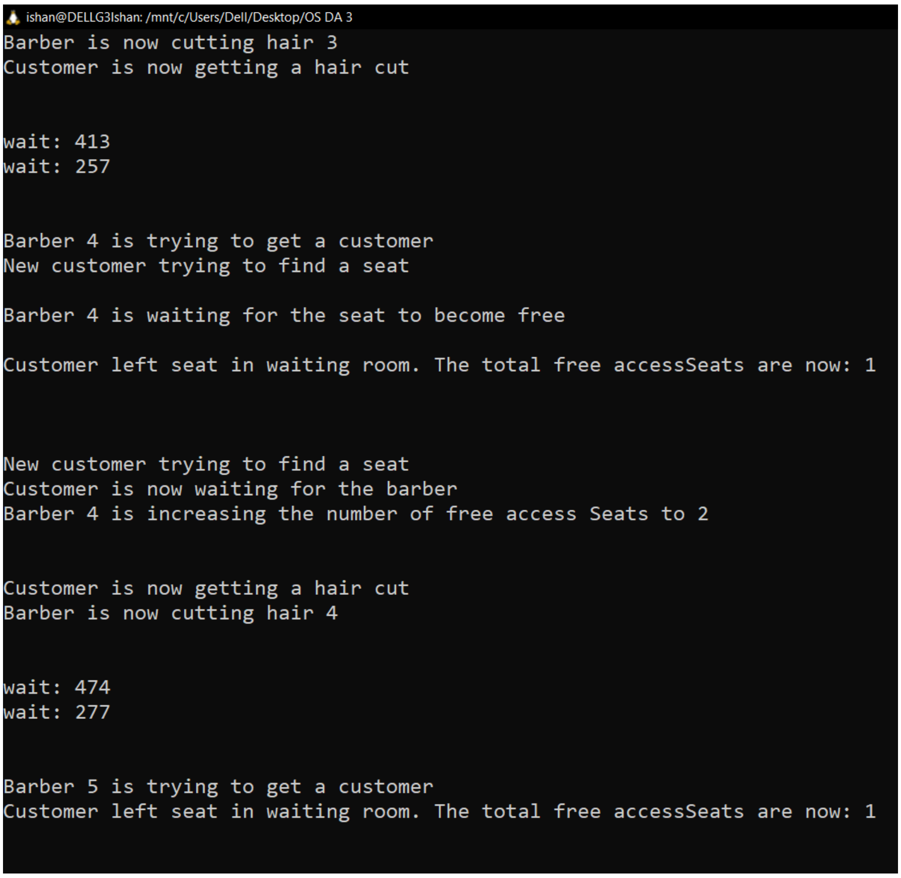
Output :



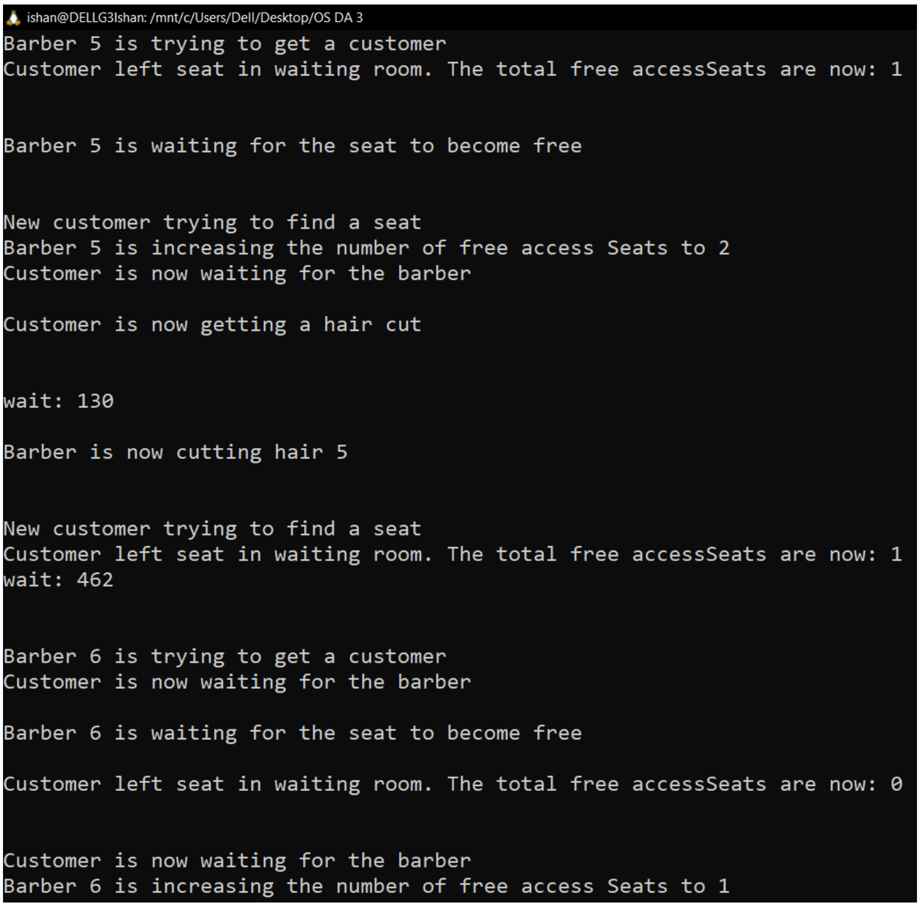
```
      |     ^~~~~
ishan@DELLG3Ishan:/mnt/c/Users/Dell/Desktop/OS DA 3$ ./a.out

19BCE2250 - Ishan Jogalekar

Barber 1 is trying to get a customer

New customer trying to find a seat

Customer left seat in waiting room. The total free accessSeats are now: 2


Customer is now waiting for the barber
Barber 1 is waiting for the seat to become free

Barber 1 is increasing the number of free access Seats to 3

New customer trying to find a seat


Customer is now getting a hair cut

Barber is now cutting hair 1

wait: 244


wait: 421
Customer left seat in waiting room. The total free accessSeats are now: 2
New customer trying to find a seat

Barber 2 is trying to get a customer
```



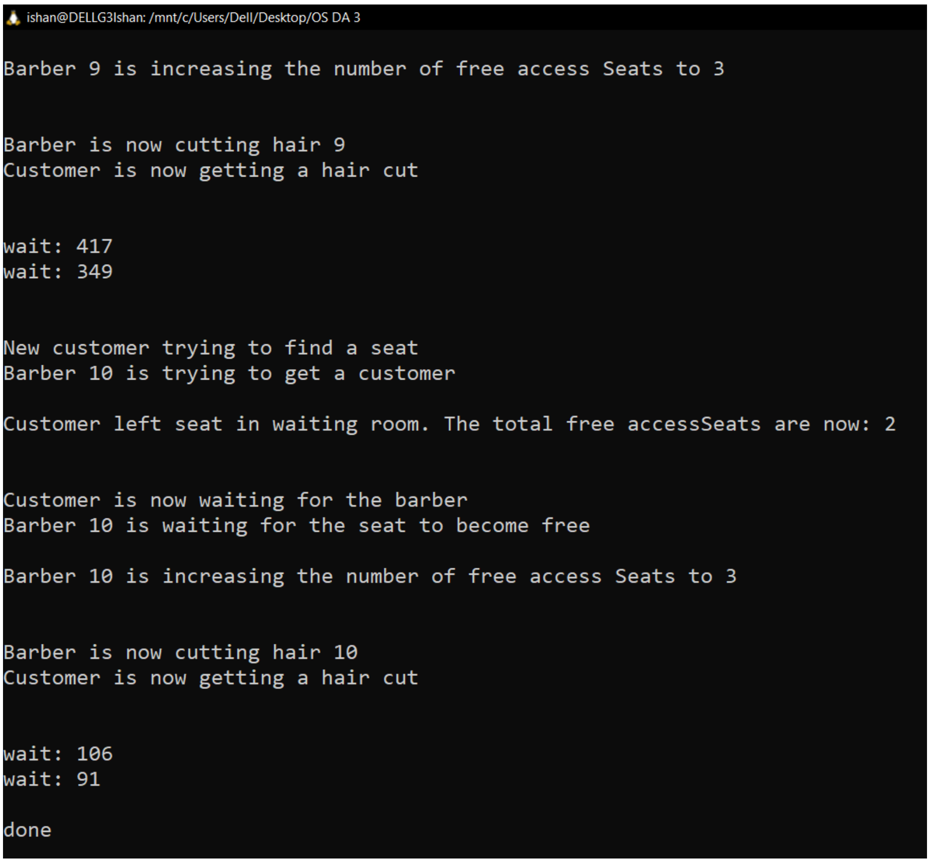
```
Barber 2 is trying to get a customer



Customer is now waiting for the barber
Barber 2 is waiting for the seat to become free
Customer left seat in waiting room. The total free accessSeats are now: 1



Customer is now waiting for the barber
New customer trying to find a seat
Barber 2 is increasing the number of free access Seats to 2


Customer is now getting a hair cut

Barber is now cutting hair 2

Customer left seat in waiting room. The total free accessSeats are now: 1
wait: 374


wait: 477
Customer is now waiting for the barber

Barber 3 is trying to get a customer

Barber 3 is waiting for the seat to become free

Barber 3 is increasing the number of free access Seats to 2
```



```
Barber is now cutting hair 3
Customer is now getting a hair cut


wait: 413
wait: 257


Barber 4 is trying to get a customer
New customer trying to find a seat

Barber 4 is waiting for the seat to become free

Customer left seat in waiting room. The total free accessSeats are now: 1



New customer trying to find a seat
Customer is now waiting for the barber
Barber 4 is increasing the number of free access Seats to 2


Customer is now getting a hair cut
Barber is now cutting hair 4


wait: 474
wait: 277


Barber 5 is trying to get a customer
Customer left seat in waiting room. The total free accessSeats are now: 1
```



```
Barber 5 is trying to get a customer
Customer left seat in waiting room. The total free accessSeats are now: 1


Barber 5 is waiting for the seat to become free


New customer trying to find a seat
Barber 5 is increasing the number of free access Seats to 2
Customer is now waiting for the barber

Customer is now getting a hair cut


wait: 130

Barber is now cutting hair 5


New customer trying to find a seat
Customer left seat in waiting room. The total free accessSeats are now: 1
wait: 462


Barber 6 is trying to get a customer
Customer is now waiting for the barber

Barber 6 is waiting for the seat to become free

Customer left seat in waiting room. The total free accessSeats are now: 0


Customer is now waiting for the barber
Barber 6 is increasing the number of free access Seats to 1
```



```

Barber 9 is increasing the number of free access Seats to 3


Barber is now cutting hair 9
Customer is now getting a hair cut


wait: 417
wait: 349


New customer trying to find a seat
Barber 10 is trying to get a customer

Customer left seat in waiting room. The total free accessSeats are now: 2


Customer is now waiting for the barber
Barber 10 is waiting for the seat to become free

Barber 10 is increasing the number of free access Seats to 3


Barber is now cutting hair 10
Customer is now getting a hair cut


wait: 106
wait: 91

done
```

(e) A pair of processes involved in exchanging a sequence of integers. The number of integers that can be produced and consumed at a time is limited to 100. Write a Program to implement the producer and consumer problem using POSIX semaphore for the above scenario.

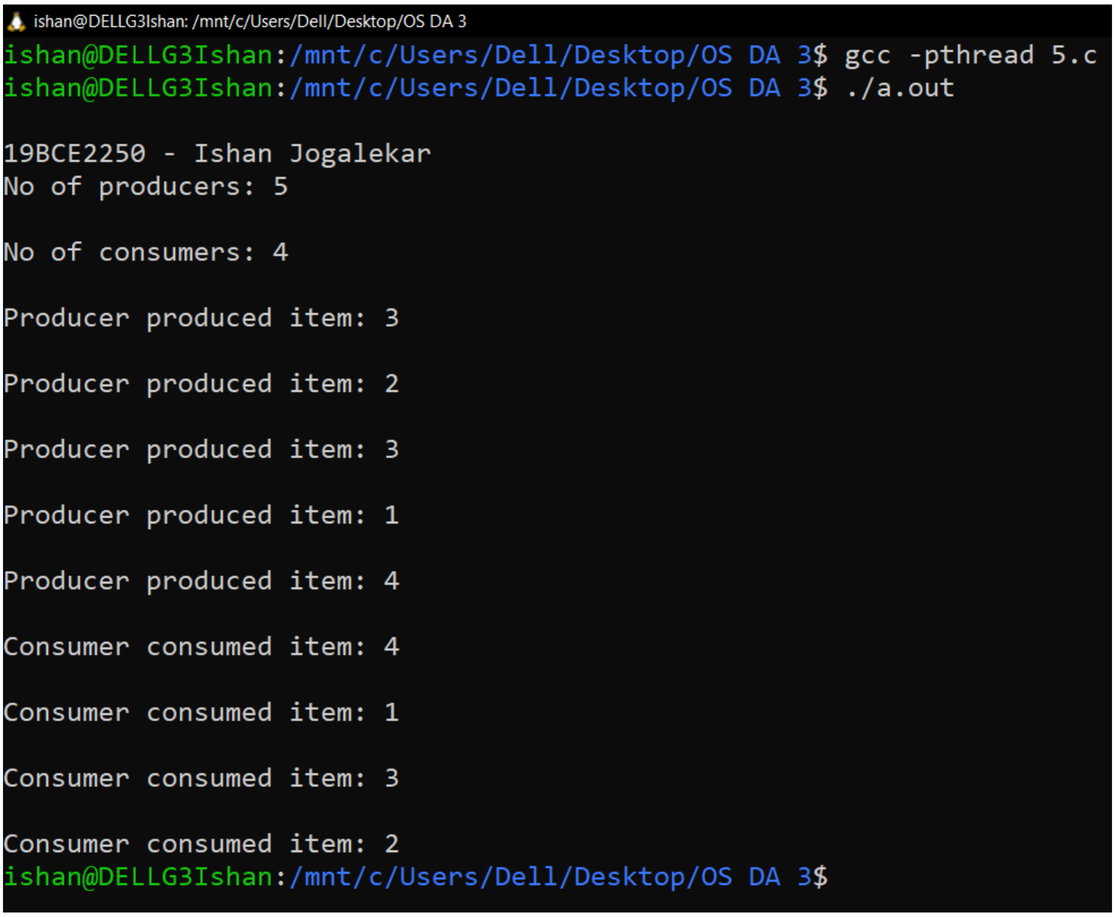
Ans :-

Code –

```
#include<stdio.h>

#include<semaphore.h>

#include<pthread.h>

#include<stdlib.h>

#define buffersize 100

pthread_mutex_t mutex;

pthread_t tidP[100],tidC[100];

sem_t full,empty;

int counter;

int buffer[buffersize];

void initialize()

{

pthread_mutex_init(&mutex,NULL);

sem_init(&full,1,0);

sem_init(&empty,1,buffersize);

counter=0;

}

void write(int item)
```

```
{
buffer[counter++]=item;
}
int read()
{
return(buffer[--counter]);
}
void * producer (void * param)
{
int waittime,item,i;
item=rand()%5;
waittime=rand()%5;
sem_wait(&empty);
pthread_mutex_lock(&mutex);
printf("\nProducer produced item: %d\n",item);
write(item);
pthread_mutex_unlock(&mutex);
sem_post(&full);
}
void * consumer (void * param)
{
int waittime,item;
waittime=rand()%5;
sem_wait(&full);
pthread_mutex_lock(&mutex);
item=read();
printf("\nConsumer consumed item: %d\n",item);
pthread_mutex_unlock(&mutex);
sem_post(&empty);
}
int main()
{
    printf("\n19BCE2250 - Ishan Jogalekar");
    int n1,n2,i;
    initialize();
    printf("\nNo of producers: ");
    scanf("%d",&n1);
    printf("\nNo 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);
exit(0);
}
```

Output :



ishan@DELLG3Ishan: /mnt/c/Users/Dell/Desktop/OS DA 3

```
ishan@DELLG3Ishan:/mnt/c/Users/Dell/Desktop/OS DA 3$ gcc -pthread 5.c
ishan@DELLG3Ishan:/mnt/c/Users/Dell/Desktop/OS DA 3$ ./a.out

19BCE2250 - Ishan Jogalekar
No of producers: 5

No of consumers: 4

Producer produced item: 3

Producer produced item: 2

Producer produced item: 3

Producer produced item: 1

Producer produced item: 4

Consumer consumed item: 4

Consumer consumed item: 1

Consumer consumed item: 3

Consumer consumed item: 2
ishan@DELLG3Ishan:/mnt/c/Users/Dell/Desktop/OS DA 3$
```