NAME :- RIBHU BHUSHAN TIWARI

SAP ID :- 590011166

BATCH :- B2 MCA AI/ML

OPERATING SYSTEM

CODE Q1:-

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

#define BUFFER_SIZE 5
#define NUM_ITEMS 10

int buffer[BUFFER_SIZE];
int in = 0, out = 0;

sem_t empty, full;
pthread_mutex_t mutex;

void *producer(void *arg) {
int item;
for (int i = 0; i < NUM_ITEMS; i++) {
item = rand() % 100;
sem_wait(&empty);
```

```
pthread_mutex_lock(&mutex);

buffer[in] = item;
printf("Producer produced item %d at index %d\n", item, in);
in = (in + 1) % BUFFER_SIZE;

pthread_mutex_unlock(&mutex);
sem_post(&full);
sleep(1);
}
return NULL;
}

void *consumer(void *arg) {
int item;
for (int i = 0; i < NUM_ITEMS; i++) {
sem_wait(&full);
pthread_mutex_lock(&mutex);

item = buffer[out];
printf("Consumer consumed item %d from index %d\n", item, out);
out = (out + 1) % BUFFER_SIZE;

pthread_mutex_unlock(&mutex);
sem_post(&empty);
sleep(1);
}
return NULL;
}

int main() {
```

```
pthread_t prod, cons;

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

pthread_create(&prod, NULL, producer, NULL);
pthread_create(&cons, NULL, consumer, NULL);

pthread_join(prod, NULL);
pthread_join(cons, NULL);

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

return 0;
}
```

OUTPUT Q1:-

```
PS C:\Users\Ribhu\OneDrive\Desktop\OS LAB> cd "c:\Users\Ribh
($?) { .\SemaphoreQ1 }
Producer produced item 41 at index 0
Consumer consumed item 41 from index 0
Producer produced item 67 at index 1
Consumer consumed item 67 from index 1
Producer produced item 34 at index 2
Consumer consumed item 34 from index 2
Producer produced item 0 at index 3
Consumer consumed item 0 from index 3
Producer produced item 69 at index 4
Consumer consumed item 69 from index 4
Producer produced item 24 at index 0
Consumer consumed item 24 from index 0
Producer produced item 78 at index 1
Consumer consumed item 78 from index 1
Producer produced item 58 at index 2
Consumer consumed item 58 from index 2
Producer produced item 62 at index 3
Consumer consumed item 62 from index 3
Producer produced item 64 at index 4
Consumer consumed item 64 from index 4
PS C:\Users\Ribhu\OneDrive\Desktop\OS LAB>
```

Q2 CODE:-

```
#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#include <semaphore.h>

#include <unistd.h>


sem_t rw_mutex, mutex;

int read_count = 0, shared_data = 0;


void *reader(void *arg) {

    int id = *(int *)arg;
```

```
    for (int i = 0; i < 5; i++) {
        sem_wait(&mutex);
        read_count++;
        if (read_count == 1) sem_wait(&rw_mutex);
        sem_post(&mutex);

        printf("Reader %d reads data: %d\n", id, shared_data);
        sleep(1);

        sem_wait(&mutex);
        read_count--;
        if (read_count == 0) sem_post(&rw_mutex);
        sem_post(&mutex);

        sleep(1);
    }
    return NULL;
}

void *writer(void *arg) {
    int id = *(int *)arg;
    for (int i = 0; i < 5; i++) {
        sem_wait(&rw_mutex);

        shared_data++;
        printf("Writer %d updates data to: %d\n", id, shared_data);
        sleep(1);

        sem_post(&rw_mutex);
        sleep(1);
    }
```

```
    return NULL;
}

int main() {
    pthread_t r1, r2, w1;
    int id1 = 1, id2 = 2, id3 = 3;

    sem_init(&rw_mutex, 0, 1);
    sem_init(&mutex, 0, 1);

    pthread_create(&r1, NULL, reader, &id1);
    pthread_create(&r2, NULL, reader, &id2);
    pthread_create(&w1, NULL, writer, &id3);

    pthread_join(r1, NULL);
    pthread_join(r2, NULL);
    pthread_join(w1, NULL);

    sem_destroy(&rw_mutex);
    sem_destroy(&mutex);

    return 0;
}
```

Q2 OUTPUT :-

```
PS C:\Users\Ribhu\OneDrive\Desktop\OS LAB> cd "c
($?) { .\SemaphoreQ2 }
Reader 1 reads data: 0
Reader 2 reads data: 0
Writer 3 updates data to: 1
Reader 1 reads data: 1
Reader 2 reads data: 1
Writer 3 updates data to: 2
Reader 2 reads data: 2
Reader 1 reads data: 2
Writer 3 updates data to: 3
Reader 1 reads data: 3
Reader 2 reads data: 3
Writer 3 updates data to: 4
Reader 1 reads data: 4
Reader 2 reads data: 4
Writer 3 updates data to: 5
PS C:\Users\Ribhu\OneDrive\Desktop\OS LAB>
```

Q3 CODE:-

```
#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

#include <semaphore.h>

#include <unistd.h>
```

```
#define N 5

sem_t forks[N];

void *philosopher(void *num) {
    int id = *(int *)num;

    for (int i = 0; i < 3; i++) {
        printf("Philosopher %d is thinking\n", id);
        sleep(1);

        sem_wait(&forks[id]);
        sem_wait(&forks[(id + 1) % N]);

        printf("Philosopher %d is eating\n", id);
        sleep(2);

        sem_post(&forks[id]);
        sem_post(&forks[(id + 1) % N]);

        printf("Philosopher %d finished eating\n", id);
    }
    return NULL;
}

int main() {
    pthread_t phil[N];
    int id[N];

    for (int i = 0; i < N; i++) sem_init(&forks[i], 0, 1);
```

```
    for (int i = 0; i < N; i++) {
        id[i] = i;
        pthread_create(&phil[i], NULL, philosopher, &id[i]);
    }

    for (int i = 0; i < N; i++) pthread_join(phil[i], NULL);

    for (int i = 0; i < N; i++) sem_destroy(&forks[i]);

    return 0;
}
```

Q3 OUTPUT:-

```
PS C:\Users\Ribhu\OneDrive\Desktop\OS LAB
f ($?) { .\SeamaphoreQ3 }
Philosopher 0 is thinking
Philosopher 1 is thinking
Philosopher 3 is thinking
Philosopher 2 is thinking
Philosopher 4 is thinking
Philosopher 2 is eating
Philosopher 2 finished eating
Philosopher 1 is eating
Philosopher 2 is thinking
Philosopher 0 is eating
Philosopher 1 finished eating
Philosopher 1 is thinking
Philosopher 0 finished eating
Philosopher 0 is thinking
Philosopher 4 is eating
Philosopher 4 finished eating
Philosopher 4 is thinking
Philosopher 3 is eating
Philosopher 3 finished eating
Philosopher 3 is thinking
Philosopher 2 is eating
Philosopher 2 finished eating
Philosopher 2 is thinking
Philosopher 1 is eating
Philosopher 1 finished eating
Philosopher 0 is eating
Philosopher 1 is thinking
Philosopher 0 finished eating
Philosopher 0 is thinking
Philosopher 4 is eating
Philosopher 4 finished eating
Philosopher 4 is thinking
Philosopher 3 is eating
```

```
Philosopher 3 finished eating
Philosopher 3 is thinking
Philosopher 2 is eating
Philosopher 1 is eating
Philosopher 2 finished eating
Philosopher 1 finished eating
Philosopher 0 is eating
Philosopher 0 finished eating
Philosopher 4 is eating
Philosopher 3 is eating
Philosopher 4 finished eating
Philosopher 3 finished eating
PS C:\Users\Ribhu\OneDrive\Desktop\OS LAB>
```