## // Consumer Producer Problem

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <semaphore.h>
#define max 5
#define buffers 5
sem_t empty;
sem_t full;
int in = 0,out=0;
int buffer[buffers];
pthread_mutex_t mutex;

void *producer(void *p){
        int item;
    for(int i = 0; i < max; i++) {
        item = rand();
        sem_wait(&empty);
        pthread_mutex_lock(&mutex);
        buffer[in] = item;
        printf("In producer %d: Insert Item %d at %d\n", *((int *)p),buffer[in],in);
        in = (in+1)%buffers;
        pthread_mutex_unlock(&mutex);
        sem_post(&full);
    }
}
void *consumer(void *c)
{
    for(int i = 0; i < max; i++) {
        sem_wait(&full);
        pthread_mutex_lock(&mutex);
        int item = buffer[out];
        printf("In Consumer %d: Remove Item %d from %d\n",*((int *)c),item, out);
        out = (out+1)%buffers;
        pthread_mutex_unlock(&mutex);
        sem_post(&empty);
    }
}


int main()
{
    pthread_t produce[5],consume[5];
```

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

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

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

	for(int i = 0; i < 5; i++) {
		pthread_join(produce[i], NULL);
	}
	for(int i = 0; i < 5; i++) {
		pthread_join(consume[i], NULL);
	}

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

	return 0;
}
```

## // Reader-Writer Problem

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <semaphore.h>
sem_t wrt;
pthread_mutex_t mutex;
int count = 1,numr=0;

void *reader(void *rno)
{
	pthread_mutex_lock(&mutex);
	numr++;
	if(numr == 1) {
		sem_wait(&wrt);
	}
	pthread_mutex_unlock(&mutex);
	printf("Reader %d: read count as %d\n",*((int *)rno),count);

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

void *writer(void *wno)
{
	sem_wait(&wrt);
	count = count*2;
	printf("Writer %d modified count to %d\n",(*((int *)wno)),count);
	sem_post(&wrt);
}

int main()
{

	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;
}
```