# Assignment 04A

*Roll No. 33331*

## Program:

```
//Producer-Consumer Problem
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <semaphore.h>
#include <unistd.h>

#define BUFFER_SIZE 10

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

sem_t empty;
sem_t full;
pthread_mutex_t mutex;

void *producer(void *param) {
    int id = *((int *)param);
    while (1) {
        int item = rand() % 100;
        sem_wait(&empty);
        pthread_mutex_lock(&mutex);

        buffer[in] = item;
        printf("Producer %d produced %d at index %d\n", id, item, in);
        in = (in + 1) % BUFFER_SIZE;
        pthread_mutex_unlock(&mutex);
        sem_post(&full);

        sleep(1);
    }
}

void *consumer(void *param) {
    int id = *((int *)param);
    while (1) {
        sem_wait(&full);
        pthread_mutex_lock(&mutex);

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

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

        sleep(1);
    }
}

int main() {
    int p, c;
    sem_init(&empty, 0, BUFFER_SIZE);
    sem_init(&full, 0, 0);
    pthread_mutex_init(&mutex, NULL);

    printf("Enter number of producers: ");
    scanf("%d", &p);
    printf("Enter number of consumers: ");
    scanf("%d", &c);

    pthread_t producers[p], consumers[c];
    int producer_ids[p], consumer_ids[c];

    for (int i = 0; i < p; i++) {
        producer_ids[i] = i + 1;
        pthread_create(&producers[i], NULL, producer, &producer_ids[i]);
    }

    for (int i = 0; i < c; i++) {
        consumer_ids[i] = i + 1;
        pthread_create(&consumers[i], NULL, consumer, &consumer_ids[i]);
    }

    for (int i = 0; i < p; i++) {
        pthread_join(producers[i], NULL);
    }

    for (int i = 0; i < c; i++) {
        pthread_join(consumers[i], NULL);
    }

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

    return 0;
}
```

## Output :

*Test Case 1:*

Producers = 3, Consumers = 3

```
Producer 1 produced 47 at index 0
Consumer 1 consumed 47 from index 0
Producer 2 produced 52 at index 1
Producer 3 produced 21 at index 2
Consumer 2 consumed 52 from index 1
Producer 1 produced 89 at index 3
Consumer 3 consumed 21 from index 2
Producer 2 produced 34 at index 4
Producer 3 produced 76 at index 5
Consumer 1 consumed 89 from index 3
Producer 1 produced 97 at index 6
Consumer 2 consumed 34 from index 4
Producer 2 produced 65 at index 7
Consumer 3 consumed 76 from index 5
Producer 3 produced 12 at index 8
Consumer 1 consumed 97 from index 6
Consumer 2 consumed 65 from index 7
Consumer 3 consumed 12 from index 8
```

*Test Case 2 :*

Producers = 5, Consumers = 5

```
Producer 1 produced 23 at index 0
Consumer 1 consumed 23 from index 0
Producer 2 produced 45 at index 1
Producer 3 produced 12 at index 2
Consumer 2 consumed 45 from index 1
Producer 4 produced 67 at index 3
Producer 5 produced 33 at index 4
Consumer 3 consumed 12 from index 2
Producer 1 produced 82 at index 5
Consumer 4 consumed 67 from index 3
Producer 2 produced 91 at index 6
Consumer 5 consumed 33 from index 4
Producer 3 produced 77 at index 7
Consumer 1 consumed 82 from index 5
Producer 4 produced 54 at index 8
Consumer 2 consumed 91 from index 6
Producer 5 produced 15 at index 9
Consumer 3 consumed 77 from index 7
Consumer 4 consumed 54 from index 8
Consumer 5 consumed 15 from index 9
```