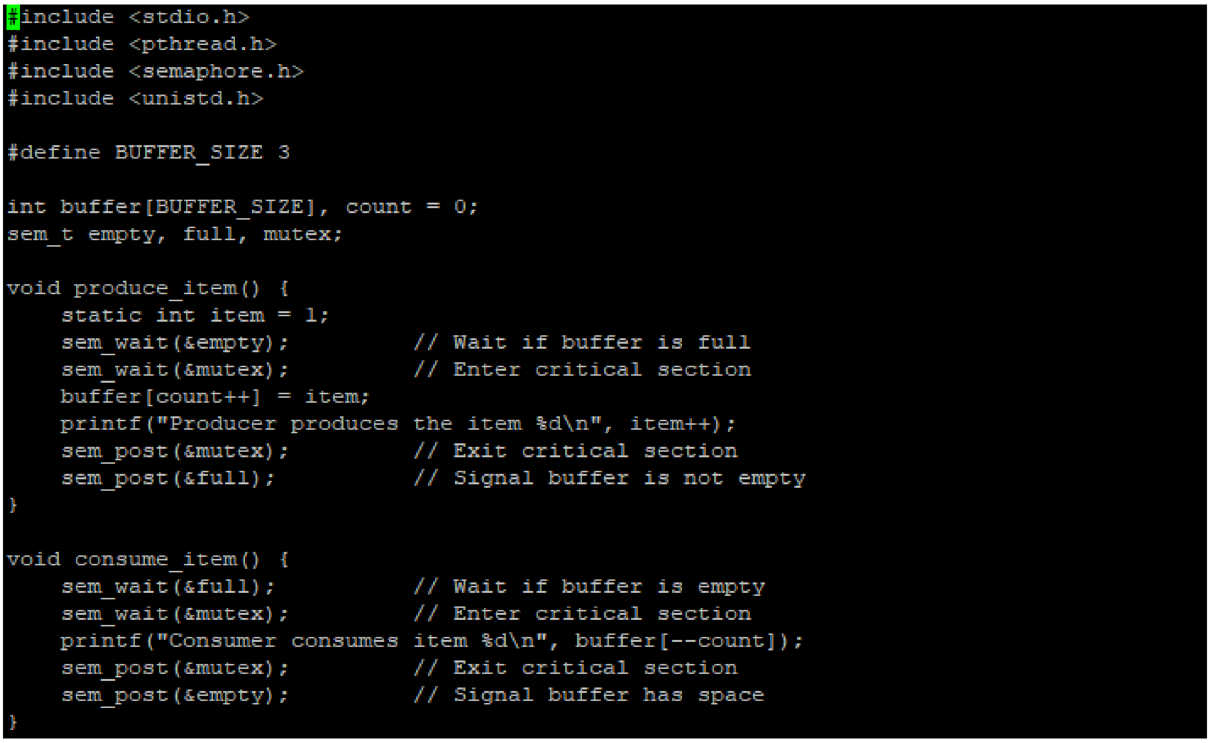
NAME: ARUN MC

ROLL NO: 230701036

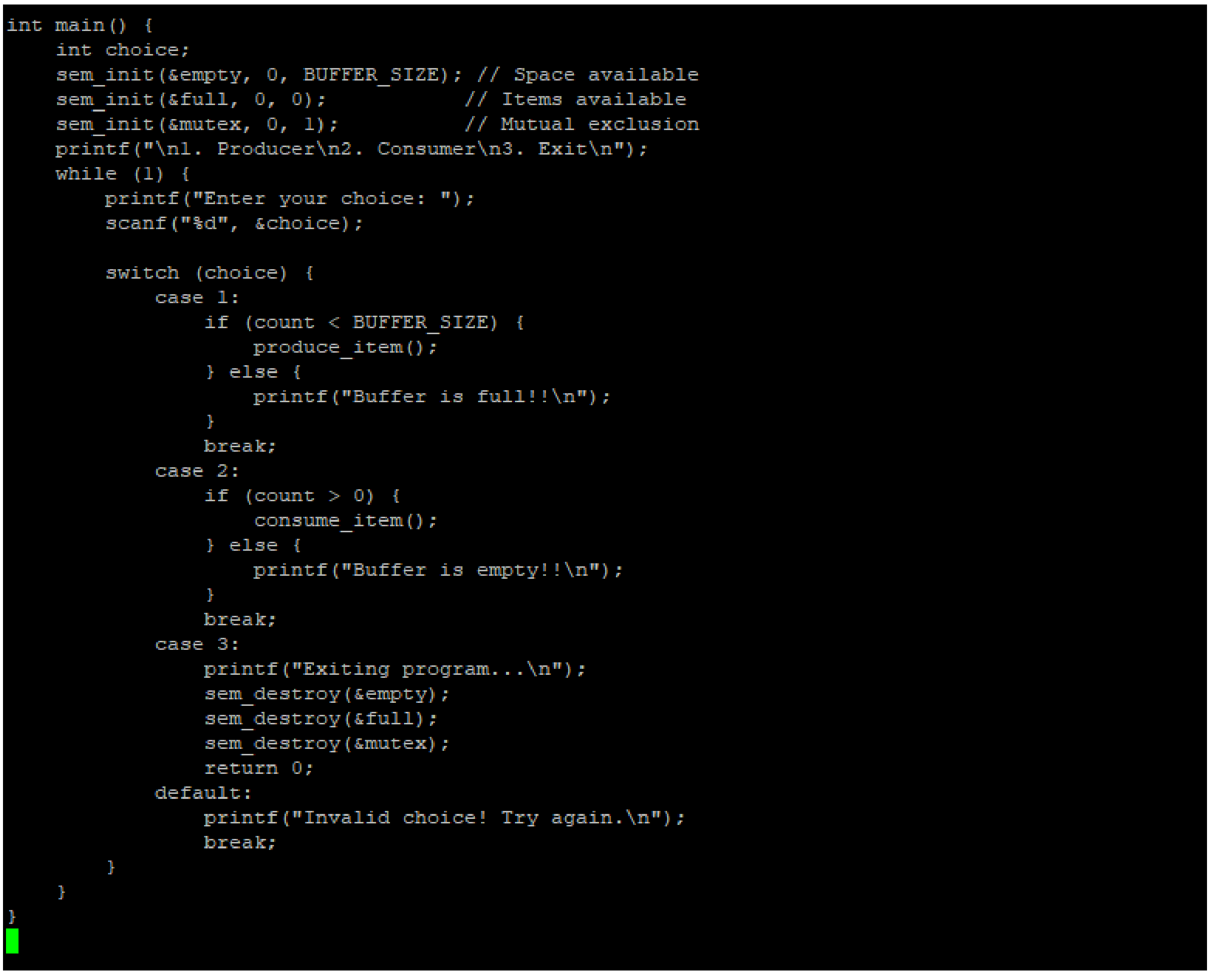
Exp:8 **PRODUCER CONSUMER USING SEMAPHORES**

Aim:

To write a program to implement solution to producer consumer problem using semaphores.
CODE:

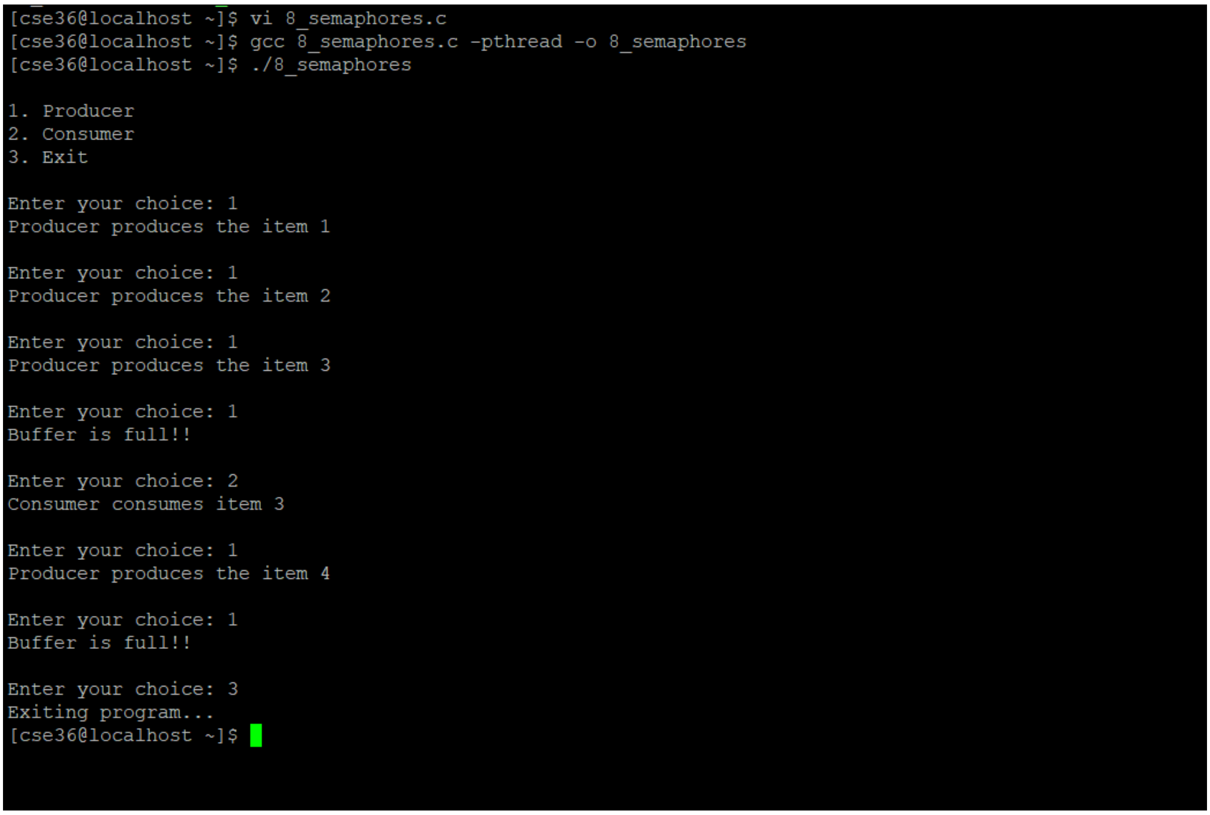


```
#include <stdio.h>
#include <pthread.h>
#include <semaphore.h>
#include <unistd.h>

#define BUFFER_SIZE 3

int buffer[BUFFER_SIZE], count = 0;
sem_t empty, full, mutex;

void produce_item() {
    static int item = 1;
    sem_wait(&empty);          // Wait if buffer is full
    sem_wait(&mutex);          // Enter critical section
    buffer[count++] = item;
    printf("Producer produces the item %d\n", item++);
    sem_post(&mutex);          // Exit critical section
    sem_post(&full);           // Signal buffer is not empty
}

void consume_item() {
    sem_wait(&full);           // Wait if buffer is empty
    sem_wait(&mutex);          // Enter critical section
    printf("Consumer consumes item %d\n", buffer[--count]);
    sem_post(&mutex);          // Exit critical section
    sem_post(&empty);          // Signal buffer has space
}
```



```
int main() {
    int choice;
    sem_init(&empty, 0, BUFFER_SIZE); // Space available
    sem_init(&full, 0, 0);           // Items available
    sem_init(&mutex, 0, 1);          // Mutual exclusion
    printf("\n1. Producer\n2. Consumer\n3. Exit\n");
    while (1) {
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                if (count < BUFFER_SIZE) {
                    produce_item();
                } else {
                    printf("Buffer is full!!\n");
                }
                break;
            case 2:
                if (count > 0) {
                    consume_item();
                } else {
                    printf("Buffer is empty!!\n");
                }
                break;
            case 3:
                printf("Exiting program...\n");
                sem_destroy(&empty);
                sem_destroy(&full);
                sem_destroy(&mutex);
                return 0;
            default:
                printf("Invalid choice! Try again.\n");
                break;
        }
    }
}
```

OUTPUT:



```
[cse36@localhost ~]$ vi 8_semaphores.c
[cse36@localhost ~]$ gcc 8_semaphores.c -pthread -o 8_semaphores
[cse36@localhost ~]$ ./8_semaphores

1. Producer
2. Consumer
3. Exit

Enter your choice: 1
Producer produces the item 1

Enter your choice: 1
Producer produces the item 2

Enter your choice: 1
Producer produces the item 3

Enter your choice: 1
Buffer is full!!

Enter your choice: 2
Consumer consumes item 3

Enter your choice: 1
Producer produces the item 4

Enter your choice: 1
Buffer is full!!

Enter your choice: 3
Exiting program...
[cse36@localhost ~]$
```