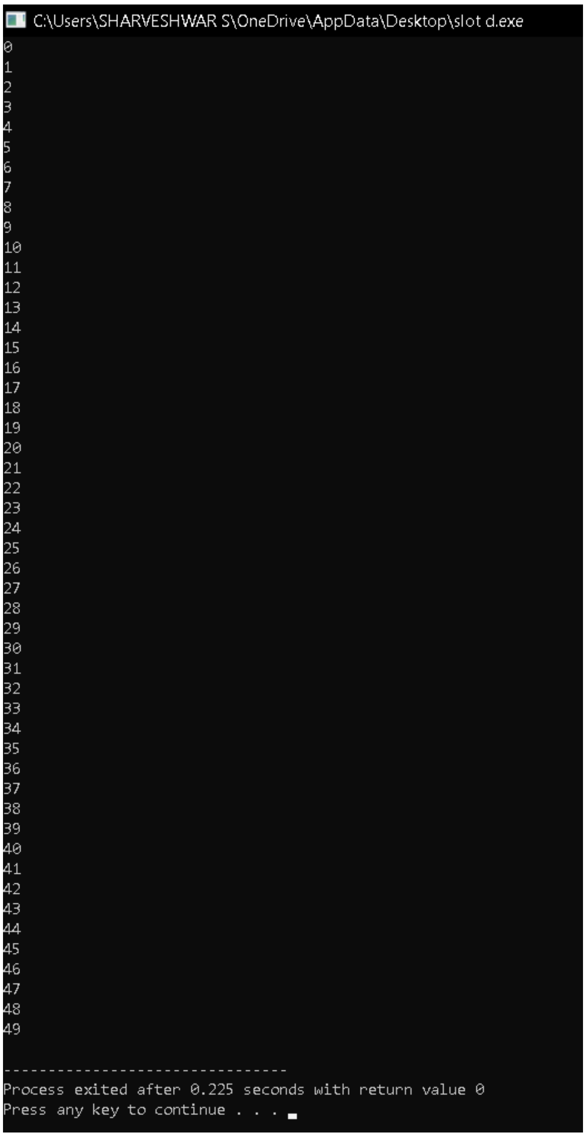
```
#include <stdio.h>

#include <pthread.h>

#include <semaphore.h>


#define BUFFER_SIZE 10


int buffer[BUFFER_SIZE];

int fill = 0;

int use = 0;


sem_t empty;

sem_t full;

pthread_mutex_t mutex;


void put(int value) {

    buffer[fill] = value;

    fill = (fill + 1) % BUFFER_SIZE;

}


int get() {

    int tmp = buffer[use];

    use = (use + 1) % BUFFER_SIZE;

    return tmp;

}


void *producer(void *arg) {

    int i;

    for (i = 0; i < 50; i++) {

        sem_wait(&empty);

        pthread_mutex_lock(&mutex);

        put(i);
        pthread_mutex_unlock(&mutex);
        sem_post(&full);
    }
    return NULL;
}

void *consumer(void *arg) {
    int i;
    for (i = 0; i < 50; i++) {
        sem_wait(&full);
        pthread_mutex_lock(&mutex);
        int tmp = get();
        pthread_mutex_unlock(&mutex);
        sem_post(&empty);
        printf("%d\n", tmp);
    }
    return NULL;
}

int main() {
    pthread_t producer_thread, consumer_thread;

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

    pthread_create(&producer_thread, NULL, producer, NULL);
    pthread_create(&consumer_thread, NULL, consumer, NULL);

    pthread_join(producer_thread, NULL);
    pthread_join(consumer_thread, NULL);
    sem_destroy(&empty);

    sem_destroy(&full);

    pthread_mutex_destroy(&mutex);


    return 0;

}
```



```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49

--------------------------------
Process exited after 0.225 seconds with return value 0
Press any key to continue . . .
```