Name: Shangirne Kharbanda

Registration Number: 20BAI1154

LAB – 7

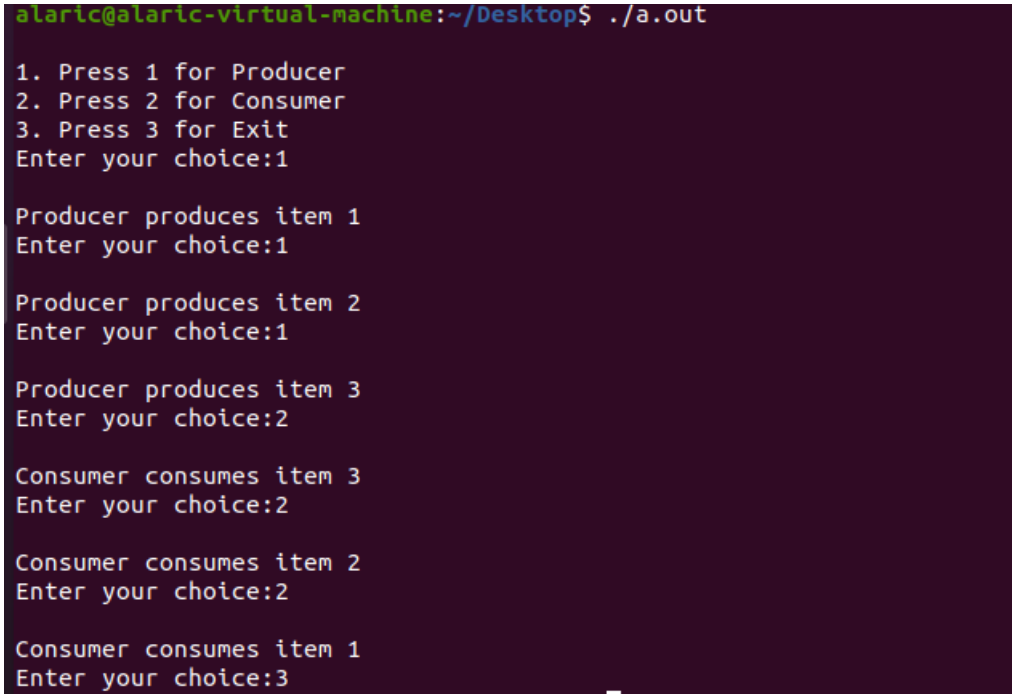
Producers Consumers Problem

Code:

```c
#include <stdio.h>
#include <stdlib.h>
int mutex = 1;
int full = 0;
int empty = 10, x = 0;
void producer()
{
        --mutex;
        ++full;
        --empty;
        x++;
        printf("\nProducer produces item %d",x);
        ++mutex;
}
void consumer()
{
        --mutex;
        --full;
        ++empty;
        printf("\nConsumer consumes "
               "item %d",
               x);
        x--;
        ++mutex;
}
int main()
{
        int n, i;
        printf("\n1. Press 1 for Producer"
               "\n2. Press 2 for Consumer"
               "\n3. Press 3 for Exit");
        for (i = 1; i > 0; i++) {
                printf("\nEnter your choice:");
                scanf("%d", &n);
                switch (n) {
                case 1:
                        if ((mutex == 1)
                                && (empty != 0)) {
                                producer();
                        }
                        else {
                                printf("Buffer is full!");
                        }
```

```
43                      }
44                      break;
45
46              case 2:
47                      if ((mutex == 1)
48                              && (full != 0)) {
49                              consumer();
50                      }
51                      else {
52                              printf("Buffer is empty!");
53                      }
54                      break;
55              case 3:
56                      exit(0);
57                      break;
58              }
59      }
60 }
61
```

Output:



Readers Writers Problem

Code:

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

sem_t wrt;
pthread_mutex_t mutex;
int cnt = 1;
int numreader = 0;

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

}
void *reader(void *rno)
{

    pthread_mutex_lock(&mutex);
    numreader++;
    if(numreader == 1) {
        sem_wait(&wrt);
    }
    pthread_mutex_unlock(&mutex);

    printf("Reader %d: read cnt as %d\n",*((int *)rno),cnt);

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

int main()
{

    pthread_t read[10],write[10];
    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 < 10; 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 < 10; i++) {
        pthread_join(write[i], NULL);
    }

    pthread_mutex_destroy(&mutex);
    sem_destroy(&wrt);

    return 0;

}

```

Output:

```
alaric@alaric-virtual-machine:~/Desktop$ gcc readers_writers.c -lpthread
alaric@alaric-virtual-machine:~/Desktop$ ./a.out
Reader 1: read cnt as 1
Reader 4: read cnt as 1
Reader 5: read cnt as 1
Reader 7: read cnt as 1
Reader 3: read cnt as 1
Reader 2: read cnt as 1
Reader 6: read cnt as 1
Reader 8: read cnt as 1
Reader 9: read cnt as 1
Reader 10: read cnt as 1
Writer 2 modified cnt to 2
Writer 3 modified cnt to 4
Writer 4 modified cnt to 8
Writer 5 modified cnt to 16
Writer 8 modified cnt to 32
Writer 6 modified cnt to 64
Writer 1 modified cnt to 128
Writer 10 modified cnt to 256
Writer 9 modified cnt to 512
Writer 7 modified cnt to 1024
```