PAGE NO :
DATE :

Q) C program to implement semaphore

```
#include<stdio.h>
#include< stdlib.h>
#include <pthread.h>
#include <unistd.h>
#define MAX_ITMS 20
# define    BUFFER_SIZE 10

int buffer[BUFFER_SIZE];
int head = 0;
int tail=0;
int cnt=0;
pthread_mutex_t mtx;
pthread_cond_t cond_full;
pthread_cond_t cond_empty;

void enque(int item){
      buffer[tail]=item;
      tail= (tail+1)%. BUFFER_SIZE;
      cnt ++;
  }
void deque(){
     int item= buffer[head]
     head= (head+1)%. BUFFER_SIZE;
     Cnt --;
     return item;
 }
void* producer (void* param){
      int pred_cnt=0;
      while(1){
           int item = rand() % 100;
           pthread_mutex_lock(&mtx);
```

PAGE NO :
DATE :

```
    while(cnt == BUFFER_SIZE) {
        pthread_cond_wait(&cond_empty, &mtx);
    }
    if(prod_cnt >= MAX_ITMS) {
        pthread_mutex_unlock(&mtx);
        pthread_cond_signal(&cond_full);
        break;
    }

    enque(item);
    prod_cnt++;
    printf("~~product~~produced: %d\n", item);
    pthread_mutex_unlock(&mtx);
    pthread_cond_signal(&cond_full);
    sleep(rand()%2);
    ~~for~~ }
    return NULL;
}

void consumer(void* param) {
    int cons_cnt = 0;
    while(1) {
        pthread_mutex_lock(&mtx);
        while(cnt == 0) {
            pthread_cond_wait(&cond_full,
                &mtx);
        }
        if(cons_cnt >= MAX_ITMS) {
            pthread_mutex_unlock(&mtx);
            pthread_cond_signal(&cond_empty);
            break;
        }

        int item = deque();
        cons_cnt++;
        printf("Consumed %d \n", item);
```

```
pthread_mutex_unlock(&mtx);
pthread_cond_signal(&cond_empty);
sleep(rand()%2);
}
return NULL;
}
int main(){
    pthread_t tid_prod, tid_cons;
    pthread_mutex_init(&mtx, NULL);
    pthread_cond_init(&cond_full, NULL
    pthread_cond_init(&cond_empty, NULL
    pthread_create(&tid_prod, NULL, producer
    pthread_create(&tid_cons, NULL, consumer, NU
    pthread_join(tid_prod, NULL);
    pthread_join(tid_cons, NULL);
    printf("production & consumption comple
    pthread_mutex_destroy(&mtx);
    pthread_cond_destroy(&cond_full);
    pthread_cond_destroy(&cond_empty);
    return 0;
}
```

Output

```
produced: 91            consumed: 78
consumed: 91            consumed: 62
produced: 34            consumed: 5
produced: 69            produced: 61
produced: 78            consumed: 81
produced: 62            consumed: 61
produced: 5             produced: 95
consumed: 34            consumed: 95
consumed: 69
produced: 81
```

12/6