```c
#include <stdlib.h>
#include <stdio.h>
#include <pthread.h>
#include <math.h>

#define N 10
#define LOOPS 10
#define PAUSE 100000
#define BUSYWAIT() {int x; for(x=0; x<PAUSE; x++) sqrt( (double)x
 ); }

/* Lock */
pthread_mutex_t mutex = PTHREAD_MUTEX_INITIALIZER;
pthread_mutex_t db = PTHREAD_MUTEX_INITIALIZER;

/* shared data */
char buffer[N];
int result = 0, rc = 0;

/* Threads to run */
void reader(void); void writer(void);

int main(int argc, char *argv[])
{
        int idx;
        pthread_t t1, t2, t3, t4, t5, t6, t7;

        for(idx=0; idx<N; idx++) buffer[idx]=1;

        pthread_create(&t1, NULL,
                (void *)reader , (void *) NULL);
        pthread_create(&t2, NULL,
                (void *)reader , (void *) NULL);
        pthread_create(&t3, NULL,
                (void *)writer , (void *) NULL);
        pthread_create(&t4, NULL,
                (void *)reader , (void *) NULL);
        pthread_create(&t5, NULL,
                (void *)reader , (void *) NULL);
        pthread_create(&t6, NULL,
                (void *)reader , (void *) NULL);
        pthread_create(&t7, NULL,
                (void *)reader , (void *) NULL);

        pthread_join(t1,NULL); pthread_join(t2,NULL);
        pthread_join(t3,NULL); pthread_join(t4,NULL);
        pthread_join(t5,NULL); pthread_join(t6,NULL);
        pthread_join(t7,NULL);

        printf("Result = %d\n", result);
        return 0;
}
```

```
void writer(void){
        int item, i=LOOPS, idx;

        while (i--) {
                BUSYWAIT();

                /* item to write to db */
                item = 1 + (int)(20.0 * rand()/(RAND_MAX+1.0));

                pthread_mutex_lock(&db);
                idx = 1 + (int)(10.0 * rand()/(RAND_MAX+1.0));
                buffer[idx] = item;
                printf("WRITER: rc=%d\n", rc);
                pthread_mutex_unlock(&db);
        }
}

void reader(void){
        int item, i=LOOPS, idx;

        while (i--) {
                pthread_mutex_lock(&mutex);
                rc = rc + 1;
                if( rc == 1 ) pthread_mutex_lock(&db);
                pthread_mutex_unlock(&mutex);

                idx = 1 + (int)(10.0 * rand()/(RAND_MAX+1.0));
                item = buffer[idx];
                BUSYWAIT();
                printf("READER: rc=%d\n", rc);

                pthread_mutex_lock(&mutex);
                rc = rc - 1;
                if( rc == 0 ) pthread_mutex_unlock(&db);
                /* Consume item */
                result += item;
                pthread_mutex_unlock(&mutex);

        }
}
```