```c
/* Monitor que gestiona una cola de espera utilizando mutex y var. de cond. */

#define _POSIX_C_SOURCE 199506L
#define _REENTRANT

#include <unistd.h>
#include <pthread.h>
#include <stdio.h>
#include <errno.h>

#include "cola.h"

#define DIM_COLA 10

 /* Cola circular */

 struct colam {
                int ilec;           /* Proxima posicion llena */
                int iesc;           /* Proxima posicion vacia */
                int cdat;           /* Cuenta de datos */
                int buf[DIM_COLA];  /* Buffer de datos */
              } cola;

pthread_mutex_t mcola = PTHREAD_MUTEX_INITIALIZER;
pthread_cond_t no_llena = PTHREAD_COND_INITIALIZER;
pthread_cond_t no_vacia = PTHREAD_COND_INITIALIZER;

/* Insertar dato en la cola */

void pon_dato(int dato)
{
  struct colam *pzona = &cola;

  pthread_mutex_lock(&mcola);

  while(pzona->cdat == DIM_COLA)
  {
    pthread_cond_wait(&no_llena, &mcola);
  }

  /* Aqui ya tiene mutex y sitio */

  pzona->buf[pzona->iesc] = dato;

  /* Actualizacion del indice */

  pzona->iesc++;
  if(pzona->iesc == DIM_COLA) pzona->iesc = 0;

  /* Otro dato mas */

  pzona->cdat++;

  /* Activacion de variables de condicion (seguro que no esta vacia) */

  if(pzona->cdat != DIM_COLA)
  {
    pthread_cond_signal(&no_llena);
  }
  else printf("!cola llena!\n");

  pthread_cond_signal(&no_vacia);

  pthread_mutex_unlock(&mcola);
```



```c
}

/* Extraer un dato de la cola */

void extrae_dato(int *pdato)
{
  int dato;
  struct colam *pzona = &cola;

  pthread_mutex_lock(&mcola);

  while(pzona->cdat == 0)
  {
    pthread_cond_wait(&no_vacia, &mcola);
  }

  /* Aqui ya tiene mutex y sitio */

  dato = pzona->buf[pzona->ilec];

  /* Actualizacion del indice */

  pzona->ilec++;
  if(pzona->ilec == DIM_COLA) pzona->ilec = 0;

  /* Otro dato menos */

  pzona->cdat--;

  /* Activacion de variables de condicion (seguro que no esta llena)*/

  pthread_cond_signal(&no_llena);
  if(pzona->cdat != 0) pthread_cond_signal(&no_vacia);
  else                 printf("!cola vacia!\n");

  pthread_mutex_unlock(&mcola);

  *pdato = dato;
}
```