


1. Utilizando una zona de memoria compartida entre un proceso y procesos hijos (2 o más) verifique la necesidad de un acceso controlado a esa zona. (Por ejemplo, haga que cada proceso hijo repetidamente escriba un patrón propio en la memoria compartida y que el proceso padre repetidamente lea la zona, comprobando que el patrón se corresponda en forma íntegra a alguno de los procesos hijo).
2. Utilizando una región de memoria compartida, implemente una cola circular y utilice un semáforo para la sincronización de acceso, otro para mantener la cantidad de entradas libres y como recurso de bloqueo para procesos productores y un tercero para mantener la cantidad de elementos en la cola y como recurso de bloqueo para procesos consumidores.

Implemente los servicios:



```
// Inicializa (debe residir en un segmento de shared memory)
void QueueInit(Queue_t *pQ);

// Destruye el contenedor, liberando recursos
void QueueDestroy(Queue_t *pQ);

// Agrega un Nuevo elemento. Bloquea si no hay espacio
void QueuePut(Queue_t *pQ, int elem);

// Remueve y retorna un elemento, bloquea si no hay elementos
int QueueGet(Queue_t *pQ);

// recupera la cantidad de elementos en la cola
int QueueSize(Queue_t *pQ);
```

3. Utilizando los servicios implementados en el problema 1, implemente:

- Un proceso que cree e inicialice la cola circular.
- Un proceso que agregue elementos a la cola
- Un proceso que remueva elementos
- Un proceso que monitoree el estado de la cola