

Ejercicio cola

Para el próximo concierto de Lisa se habilitó un sistema de venta anticipada de entradas. Cada solicitud de compra puede ser representada por el siguiente registro:



```
typedef struct solicitud {  
    bool pago_aprobado;  
    bool es_miembro_club_fans;  
    char beneficios_utilizados[MAX_BENEFICIOS][MAX_NOMBRE_BENEFICIO];  
    int cantidad_beneficios_utilizados;  
} solicitud_t;
```

Se tiene un TDA cola que contiene todas las solicitudes recibidas por el sistema de venta. Se pide:

1. Crear una función que reciba una cola de solicitudes y devuelva la primera solicitud exitosa de la cola sin modificar su contenido.
2. Crear una función que reciba un puntero a la cola cargada con las solicitudes y devuelva el porcentaje de solicitudes que resultaron exitosas.
3. Crear una función que quite de la cola las solicitudes rechazadas manteniendo el orden.

Aclaración

Una solicitud es considerada **exitosa** si:

- El pago fue aprobado.
- Utilizó 4 o más beneficios.
- Uno de los beneficios utilizados es "AccesoAnticipado".

```

/* Pre condiciones:
 * Post condiciones: Crea una cola vacía. Devuelve NULL en caso de error.
 */
ticketera_t* ticketera_crear();

/* Pre condiciones: ticketera debe apuntar a una cola creada con
`ticketera_crear`.
 * Post condiciones: Encola una solicitud. Devuelve true si pudo encolar,
 *                  false en caso contrario.
 */
bool ticketera_encolar(ticketera_t* ticketera, solicitud_t
nueva_solicitud);

/* Pre condiciones: ticketera debe apuntar a una cola creada con
`ticketera_crear`.
 * Post condiciones: Desencola el primer elemento de ticketera.
 *                  Devuelve true si se pudo desencolar,
 *                  false si la cola estaba vacía.
 */
bool ticketera_descolar(ticketera_t* ticketera);

/* Pre condiciones: ticketera debe apuntar a una cola creada con
`ticketera_crear`.
 * Post condiciones: Devuelve el primer elemento de ticketera sin
desencolarlo.
 *                  0 una solicitud inválida en caso de error o cola
vacía.
 */
solicitud_t ticketera_primero(const ticketera_t* ticketera);

/* Pre condiciones: ticketera debe apuntar a una cola creada con
`ticketera_crear`.
 * Post condiciones: Devuelve true si ticketera está vacía,
 *                  false en caso contrario.
 */
bool ticketera_esta_vacia(const ticketera_t* ticketera);

/* Pre condiciones: ticketera debe apuntar a una cola creada con
`ticketera_crear`.
 * Post condiciones: Destruye ticketera y libera la memoria.
 */
void ticketera_destruir(ticketera_t* ticketera);

```