

Se desea implementar el tipo `Conjunto` que represente un conjunto de números enteros usando la siguiente estructura:

```
struct Conjunto{  
    int max_elem;  
    int num_elem;  
    int * elementos;  
};
```

Los elementos del conjunto se almacenan en un vector dinámico cuya dirección de memoria está almacenada en `elementos`. La memoria reservada viene dada por el entero `max_elem` (el máximo número posible de enteros almacenados) y `num_elem` indica el cardinal del conjunto, es decir, el número de elementos que actualmente forman parte del mismo (siempre menor que `max_elem`). Además, el vector con los elementos se encuentra ordenado de forma ascendente, y después de cada modificación sobre la estructura se debe conservar dicha ordenación. Se pide implementar las siguientes funciones:

- Una función que inicialice una variable de tipo de dato `Conjunto` a un conjunto con 0 elementos, pero con 10 componentes reservadas (`max_elem` es 10).
- Una función que inicialice una variable de tipo de dato `Conjunto` utilizando un vector (array) de enteros (no necesariamente ordenados).
- Una función que inicialice una variable de tipo de dato `Conjunto` utilizando un vector dinámico de enteros (no necesariamente ordenados).
- Una función que libere la memoria reservada en un tipo de dato `Conjunto`.
- Una función para determinar si un entero se encuentra en un conjunto.
- Una función para incorporar un nuevo elemento a un conjunto (en un conjunto no hay elementos repetidos).
- Una función para eliminar un elemento de un conjunto.
- Una función que devuelva la unión de dos conjuntos
- Una función que devuelva la intersección de dos conjuntos.
- Una función que doble la capacidad de almacenamiento de una variable de tipo `Conjunto`. Es decir, que el vector dinámico `elementos` tenga el doble de componentes, pero conserve los valores de la componentes utilizadas.
- Una función que cambie la capacidad de almacenamiento de una variable de tipo `Conjunto` a la mitad, si `num_elem` es menor que un cuarto de `max_elem`. Es decir, que el vector dinámico `elementos` tenga la mitad de componentes, pero conserve los valores de la componentes utilizadas.
- Utiliza las dos funciones anteriores para modificar la función de añadir elementos, de manera que se puedan utilizar aunque el vector dinámico esté lleno. De igual manera, modifica la función de eliminar elementos para no malgastar memoria.