

Todos los ficheros deberán comenzar por un comentario con el nombre de quien lo programa y la fecha.

1. Implementar el TAD *Conjunto de cadenas de caracteres*. Usar `Conjunto` como nombre de la clase. Para su implementación se utilizará una lista enlazada. Implementar los siguientes métodos públicos en este TAD:

- Constructor. Inicializa el conjunto a un conjunto vacío
- Constructor de copia
- Destructor
- Sobrecarga del operador de asignación
- Método `incluir` que recibe como parámetro un valor de tipo `string`. Si no hay un elemento con este valor en el conjunto entonces lo añade. En caso contrario no hace nada.
- Método `incluir` que recibe como parámetro un conjunto. Incluye todos los elementos de este conjunto en el conjunto sobre el que se llama al método.
- Método `contiene`. Recibe como parámetro un valor de tipo `string`. Devuelve verdadero si el conjunto contiene este valor. Devuelve falso en caso contrario.
- Método `borrar`. Recibe como parámetro un valor de tipo `string`. Si el conjunto contiene este valor entonces lo elimina del conjunto. No hace nada en caso contrario.
- Método `vaciar`. Elimina todos los valores del conjunto dejando el conjunto vacío.
- Método `elementos`. Devuelve un objeto de la clase `vector` con todos los elementos del conjunto. El orden de los elementos en el vector es indiferente.

Utilizar `conjunto` como nombre para el módulo y `tads` para el espacio de nombres.

2. Implementar el TAD *Lista de conjuntos de cadenas de caracteres*. Utilizar una lista enlazada para implementar este TAD. Usar `ListaC` como nombre de la clase. Los métodos que debe implementar este TAD son:

- Constructor. Inicializa la lista a una lista vacía
- Constructor de copia
- Destructor
- Sobrecarga del operador de asignación
- `void almacenar(unsigned pos, const Conjunto& v);`
Almacena el valor `v` en la posición `pos` de la lista. Si la posición no fuese válida mostraría un mensaje de error por pantalla y no haría nada.
- `void insertar(unsigned pos, const Conjunto& v);`
Inserta el valor `v` en la posición `pos` de la lista. Si la posición no fuese válida mostraría un mensaje de error por pantalla y no haría nada.
- `void borrar(unsigned pos);`
Borrar el valor en la posición `pos` de la lista. Si la posición no fuese válida mostraría un mensaje de error por pantalla y no haría nada.

- Conjunto obtener(unsigned pos) const;
Devuelve el conjunto almacenado en la posición pos de la lista. Si la posición no fuese válida mostraría un mensaje de error por pantalla y devuelve un conjunto vacío.
- Conjunto valores() const;
Devuelve el conjunto formado por la unión de todos los conjuntos de la lista.
- float soporte(const Conjunto& v) const;
Devuelve el porcentaje de conjuntos de la lista que contiene al conjunto v pasado como parámetro.
- void leer(const std::string& fi);
Lee la lista de conjuntos del fichero cuyo nombre se pasa como parámetro. El formato del fichero debe ser una línea por conjunto. Cada línea comenzará por el número de elementos del conjunto seguido por cada una de las cadenas del conjunto separadas por espacios en blanco (suponer que las cadenas del conjunto no contienen espacios en blanco). Este método eliminará cualquier conjunto almacenado en la lista antes de proceder a la lectura del fichero. Si el fichero no se puede leer entonces la lista quedará vacía.
- void escribir(const std::string& fi) const;
Escribe la lista de conjuntos en el fichero cuyo nombre se pasa como parámetro. El formato del fichero debe ser el mismo que para el método leer. Si no se puede escribir el fichero deberá mostrar un mensaje de error por pantalla.

Utilizar listac como nombre para el módulo y tads como espacio de nombres.

3. Implementar el TAD *Pila de conjuntos de cadenas de caracteres* usando una lista enlazada y con los siguientes métodos:

- Constructor. Inicializa la pila a una pila vacía
- Constructor de copia
- Destructor
- Sobrecarga del operador de asignación
- Método apilar. Apila un valor, pasado como parámetro, en la cima de la pila
- Método desapilar. Elimina y devuelve el valor en la cima de la pila. Si la pila estuviese vacía mostraría un mensaje de error por pantalla y detendría el programa
- Método pila vacía. Devuelve verdadero si la pila esta vacía. Falso en caso contrario

Utilizar Pila como nombre de la clase; pila como nombre para el módulo y tads para el espacio de nombres.