

Un `Bosco` è formato da `n` file di alberi. Le file sono numerate a partire da 1. Non si possono né aggiungere né togliere file. Ogni fila può mantenere un numero illimitato di alberi. Un albero è identificato dal suo tipo che consiste in una stringa di lunghezza inferiore o uguale a 30 caratteri. Le operazioni che possono essere effettuate su un `Bosco` sono le seguenti:

- **`InizializzaBosco(b, n);`**

Costruttore che inizializza un bosco `b` di `n` file. Tutte le file sono vuote.

- **`piantaAlbero(b, k, s, i)`**

Funzione che aggiunge `k` alberi di tipo `s` nella fila `i` del bosco `b`. In caso di errore nei parametri di ingresso, la funzione lascia il bosco inalterato.

- **`eliminaAlbero_vecchio(b, s, i)`**

Funzione che elimina l'albero di tipo `s` piantato meno di recente nella fila `i` del bosco `b`. La funzione restituisce **`true`** se l'operazione ha successo, **`false`** altrimenti.

- **`stampaBosco(b)`**

Operatore di uscita per il `Bosco`. L'uscita ha il seguente formato: numero della fila racchiuso fra parentesi quadre seguito dall'identificatore degli alberi presenti nella fila. Gli alberi sono visualizzati nell'ordine in cui sono stati piantati (dal meno recente al più recente) separati dalla `','`. L'uscita seguente corrisponde ad un `Bosco` con 5 file, dove la prima fila contiene due alberi (l'albero piantato più di recente è un Platano), la terza fila contiene tre alberi, l'albero piantato meno di recente è un Abete. Tutte le altre file sono vuote.

[1] Pino, Platano

[2]

[3] Abete, Abete, Pino

[4]

[5]

- **`eliminaTutti(b)`**

Funzione che elimina tutti gli alberi da tutte le file del bosco `b`.

Mediante il Linguaggio C++, realizzare il tipo `Bosco` definito dalle precedenti specifiche utilizzando le strutture. Implementare le funzioni e gestire le eventuali situazioni di errore.