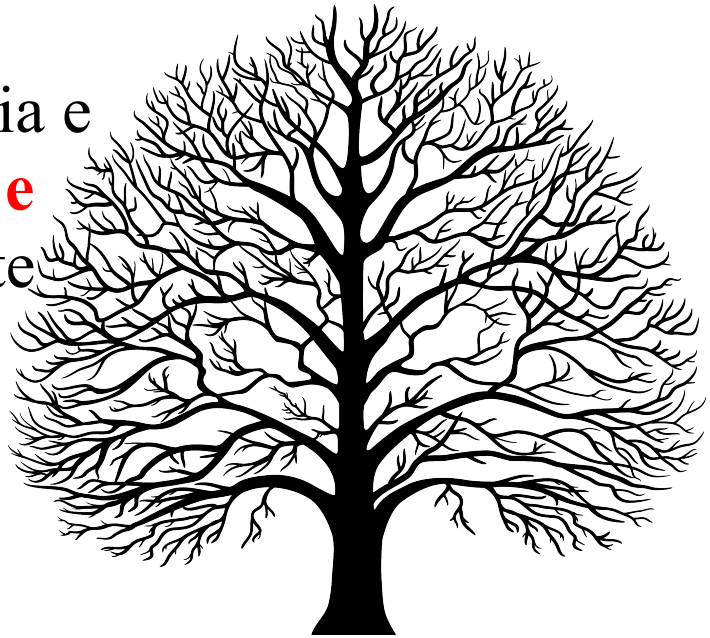
```
class Nodo {
private:
  Nodo(char c='*', Nodo* s=0, Nodo* d=0): info(c), sx(s), dx(d) {}
  char info;
  Nodo* sx;
  Nodo* dx;
};
class Tree {
public:
  Tree(): root(0) {}
  Tree(const Tree&); // dichiarazione costruttore di copia
private:
  Nodo* root;
};
```

Gli oggetti della classe Tree rappresentano alberi binari ricorsivamente definiti di char. Si ridefiniscano assegnazione, costruttore di copia e distruttore di Tree come **assegnazione, copia e distruzione profonda**. Scrivere esplicitamente eventuali dichiarazioni friend che dovessero essere richieste da tale definizione.



Definire una classe **Vettore** i cui oggetti rappresentano array di interi. Vettore deve includere un costruttore di default, una operazione di concatenazione che restituisce un nuovo vettore v1+v2, una operazione di append v1.append(v2), l'overloading dell'uguaglianza, dell'operatore di output e dell'operatore di indicizzazione. Deve inoltre includere il costruttore di copia profonda, l'assegnazione profonda e la distruzione profonda.