Laboratorio di Linguaggi di Algoritmi e Strutture Dati
Prova di laboratorio - 18 luglio 2002 - turno 1

Dato il tipo `btree` degli alberi binari di ricerca etichettati da interi, come definito nel file `lib.h`:

```
typedef struct cell *btree;

struct cell{
  int info;
  btree left,right;
};
```

1. **(punti: -2 ; 0)** Definire la funzione ricorsiva `int sum(btree t)` che restituisce la somma di tutte le etichette dei nodi dell'albero `t`; non è ammesso né servirsi di strutture dati ausiliarie, né usare le funzioni definite in `ese3.o` ed `ese4.o`.

2. **(punti: 0 ; 1)** Definire la funzione ricorsiva `bool between(btree t,int min,int max)` che restituisce `TRUE` se esiste un nodo di `t` etichettato con un intero `i` tale che `i>=min` e `i<=max`, `FALSE` altrimenti; non è ammesso né servirsi di strutture dati ausiliarie, né usare le funzioni definite in `ese3.o` ed `ese4.o`.

3. **(punti: 0 ; 2)** Senza usare variabili globali, definire la funzione `btree less(btree t,int n)` che restituisce un nuovo albero binario di ricerca che contiene tutte e sole le etichette `i` di `t` tali che `i<n`; l'albero `t` non deve assolutamente essere modificato. È ammesso utilizzare le funzioni definite in `ese3.o`, ma non quelle in `ese4.o`.

4. **(punti: 0 ; 2)** Definire la funzione `btree intersection(btree t1,btree t2)` che restituisce un nuovo albero binario di ricerca che contiene tutte e sole le etichette che appartengono sia a `t1` che a `t2`; gli alberi `t1` e `t2` non devono assolutamente essere modificati. È obbligatorio adottare un algoritmo che tragga vantaggio delle due funzioni definite in `ese4.o` e che abbia la complessità temporale minima. Non è ammesso utilizzare le funzioni definite in `ese3.o`.

Importante: tutte le funzioni devono essere definite anche sull'albero vuoto.