

PROGRAMMAZIONE II– CANALE P-Z
Prova in Laboratorio del 12/07/02

La prova dura 1h 30m.

Supponiamo di avere definito i tipi

```
struct nodo {  
    int val;  
    struct nodo *sn, *dx;  
};  
typedef struct nodo NODO;  
typedef NODO *ALBBIN;
```

per rappresentare alberi binari.

Esercizio 1. Supponiamo di rappresentare gli alberi in forma prefissa in base alle seguenti regole

```
<albero> ::= ' | ' <val> <albero> <albero>  
           ',.'
```

la prima regola, specifica come scrivere un albero non vuoto, la seconda corrisponde ad un albero vuoto. Un esempio di albero è pertanto

```
| 3 | 4 . | 5 . . | 2 . .
```

Scrivere la funzione

```
ALBBIN leggialbero();
```

che legge un albero in notazione prefissa e crea il corrispondente albero binario.

Esercizio 2. Scrivere una funzione

```
LISTA verificapesi(ALBBIN t);
```

che verifica se l'albero gode della seguente proprietà: per ogni nodo, la somma dei pesi del sottoalbero sinistro è uguale alla somma dei pesi del sottoalbero destro. La funzione dovrà ritornare il puntatore alla radice di un sottoalbero per cui non è verificata la proprietà pur essendo vera per i suoi figli, o NULL altrimenti.

Esercizio 3. Scrivere la funzione

```
void stampaalbero(ALBBIN);
```

che stampa l'albero binario nella notazione prefissa precedentemente specificata.

Esercizio 4. Utilizzando le funzioni precedenti, scrivere un programma che

1. legge un albero in forma prefissa;
2. determina se la proprietà dei pesi è verificata;
3. stampa l'albero vuoto se la proprietà è verificata, o un sottoalbero che non verifica la proprietà pur essendo la proprietà vera per i suoi figli.