

Laboratorio

4 Maggio 2010

Risolvete il seguente esercizio, prestando particolare attenzione alla formattazione dell'input e dell'output, in quanto la correzione è automatica. Per consegnare un elaborato dovete fornire il codice sorgente attraverso il comando `./consegna` che avete nella vostra home directory. Il comando deve essere utilizzato nel seguente modo:

```
/tmp/consegna fileSorgente.c numEx
```

ricordando che il percorso deve essere specificato a partire dalla vostra home directory e `numEx` deve essere un identificativo numerico (es. 1).

*Il comando **consegna** può essere utilizzato molteplici volte, per cui è possibile sovrascrivere la propria soluzione per un dato esercizio. Di tutte le consegne per un dato esercizio, viene corretta soltanto l'ultima.*

*Il file da consegnare deve contenere nelle prime righe un commento *C* che specifica il vostro Nome, Cognome e Numero di Matricola. Per esempio:*

```
/*  
  Nome: Alan  
  Cognome: Turing  
  Matricola: 193700  
*/
```

File non contenenti tali informazioni NON saranno ritenuti validi.

Nota: *E' possibile consultare i manuali (in inglese) contenenti la spiegazione di funzionamento e la sintassi di funzioni di libreria utilizzando il comando **man**, ad esempio:*

```
man strlen
```

Esercizio 1

Si implementi un programma che dato un albero binario semplice in input, restituisca in output il numero di foglie che esso contiene. Il programma dovrà leggere da input la descrizione dell'albero così fatta.

Sulla prima riga si trova il numero N di nodi dell'albero. I nodi si suppongono essere numerati da 0 a $N - 1$ compresi, dove 0 è la radice. Sulla seconda riga si trovano $N - 1$ interi p_1, p_2, \dots, p_n dove p_i rappresenta il numero del padre dell' i -esimo nodo. Si noti che il padre p_0 di 0 non viene descritto. L'output consiste in un unico intero, seguito da un a capo: il numero di foglie contenute nell'albero. Si può sempre supporre che $N \geq 2$.

Esempio

Input

```
6
0 1 2 0 2
```

Output

```
3
```

In questo caso le foglie sono 3, 4, e 5.

Esercizio 2

Scrivere un programma che legga da tastiera una sequenza di N interi distinti positivi e li inserisca in un albero binario di ricerca (senza ribilanciamento) secondo l'ordine dato. Il programma entra poi in un ciclo infinito nel quale legge un intero i da tastiera e lo cerca nell'albero. Se l'input è -1 il programma si ferma senza dare ulteriore output o fare ricerche nell'albero. Altrimenti, se $i \geq 0$ si trova nell'albero stampa la profondità a alla quale l'elemento si trova (contando da 0) altrimenti stampa -1.

L'input è formattato nel seguente modo: nella prima riga si trova la lunghezza della sequenza e nella successiva si trovano gli interi che compongono la sequenza. Seguono poi le varie richieste, una per riga. L'output è formato da una riga per ognuna delle operazioni di ricerche fatte, riportando la profondità (o -1) a cui si trova ogni elemento.

Esempio

Input

13
51 5 78 23 12 46 78 64 60 70 86 83 92
51
22
23
-1

Output

0
-1
2