

Laboratorio di Linguaggi di Algoritmi e Strutture Dati

Prova di laboratorio, turno 1, 18 giugno 2002

Si consideri il tipo `list` delle liste di interi realizzate con puntatori semplici senza elemento fittizio (detto anche header), come definito nel file `lib.h`:

```
typedef struct cell *list;

struct cell{
    int info;
    list next;
};
```

1. (**punti 0**) Si definisca la funzione `bool is_sorted(list l)` che restituisce `TRUE` se la lista `l` è ordinata (secondo l'ordine crescente), `FALSE` altrimenti.
2. (**punti 1**) Si definisca la funzione `list sort(list l)` che restituisce la lista `l` ordinata (secondo l'ordine crescente), utilizzando l'algoritmo di **merge sort**.

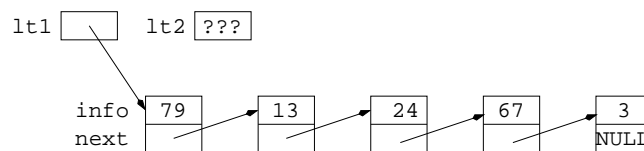
Nota bene: qualsiasi altro tipo di algoritmo, anche se corretto, verrà valutato 0 punti!

La funzione deve esclusivamente utilizzare le celle della lista `l` senza allocarne delle nuove e deve essere definita usando le seguenti due funzioni di `lib.o`:

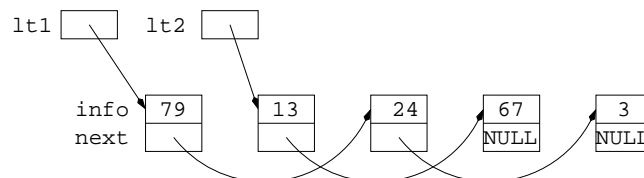
- `list split(list l1)` restituisce la lista contenente gli elementi di posto pari di `l1` e lascia in `l1` solo gli elementi di posto dispari; nessuna nuova cella viene allocata. Per ulteriori spiegazioni si veda il punto 3.
 - `list merge(list l1, list l2)` restituisce la lista ordinata ottenuta fondendo assieme le due liste ordinate `l1` e `l2`, come accade nell'algoritmo di **merge sort**; nessuna nuova cella viene allocata. Per ulteriori spiegazioni si veda il punto 4.
3. (**punti 2**) Si definisca la funzione `list split(list l1)` utilizzata al punto 2. Come già specificato, la funzione restituisce la lista contenente gli elementi di posto pari di `l1` e lascia in `l1` solo gli elementi di posto dispari, **senza allocare nuove celle**.

Se la lista `l1` è vuota o ha un solo elemento viene restituita la lista vuota e `l1` rimane invariata.

Nel caso in cui invece `l1` abbia più di un elemento, a titolo di esempio, si consideri la variabile `lt1` contenente la seguente lista:



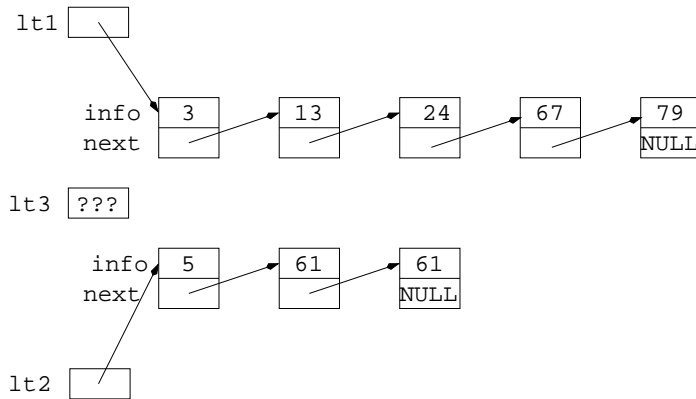
Dopo l'istruzione `lt2=split(lt1)`, si ottiene la seguente configurazione:



4. **(punti 2)** Si definisca la funzione `list merge(list l1, list l2)` utilizzata al punto 2. Come già specificato, la funzione restituisce la lista ordinata ottenuta fondendo assieme le due liste ordinate `l1` e `l2` **senza allocare nuove celle**.

Se la lista `l1` è vuota viene restituita la lista `l2` e, viceversa, se la lista `l2` è vuota viene restituita la lista `l1`.

Nel caso in cui invece sia `l1` che `l2` non siano vuote, a titolo di esempio, si considerino le variabili `lt1` e `lt2` contenenti le seguenti liste ordinate:



Dopo l'istruzione `lt3=merge(lt1,lt2)`, si ottiene la seguente configurazione:

