

Tecniche di Programmazione

Esercitazione 9

- Scrivere dei test nel main per verificare che le funzioni scritte siano corrette.
- Manipolare le liste solo tramite le funzioni del tipo astratto (vedere lista.h), non accedere all'implementazione.
- Calcolare il costo asintotico nel caso peggiore di tutte le funzioni realizzate.
- Si consiglia di risolvere gli esercizi sia in maniera iterativa che ricorsiva.

Tipo Lista

Si faccia riferimento al tipo lista fornito tramite i file `lista.h`, `lista.c`.

Esercizio 9.1

Implementare la funzione:

```
TipoLista init(T *vec, int len);
```

che, dato un array in ingresso di lunghezza `len`, restituisca una nuova Lista corrispondente alla sequenza contenuta nell'array.

Esercizio 9.2

Implementare la funzione:

```
TipoLista doubledCopy(TipoLista list);
```

che data una lista di input, restituisca una nuova lista che contiene, per ogni elemento della lista di input, due ripetizioni contigue dello stesso. Se la lista in input è vuota, si restituisca la lista vuota.

Esercizio 9.3

Implementare la funzione:

```
TipoLista multipleSublist( TipoLista list, unsigned int m );
```

che data in input una lista `list`, restituisca una nuova lista che contiene i soli elementi di `list` i cui valori sono multipli di `m`. Se la lista in input è vuota, si restituisca la lista vuota.

Esercizio 9.4

Implementare la funzione:

```
TipoLista subList(TipoLista list, int start, int end);
```

che data in input una lista e due posizioni `start` e `end`, restituisca una nuova Lista che contiene i valori dell'input da `start` a `end`. Il valore in posizione `start` deve essere incluso nell'output mentre quello in posizione `end` no.

Esercizio 9.5

Scrivere una funzione:

```
TipoLista interleave(TipoLista l1, TipoLista l2);
```

che, date due liste in input, ritorni una nuova lista contenente tutti gli elementi di `l1` nelle posizioni pari, e tutti quelli di `l2` nelle posizioni dispari. Si assuma che `l1` e `l2` abbiano la stessa lunghezza.