

## Esercitazioni n.16 – 24/05/2019 – Fondamenti di Informatica

### Esercizio n.1 - Esame 11 Giugno 2015

Sia data la seguente struttura

```
struct elem{
    int info;
    elem* pun;
};
```

Scrivere una funzione che costruisce una nuova lista di elementi di elem, ordinato per valori crescenti del campo info. I valori degli elementi della lista devono essere letti da un file il cui nome è passato come argomento alla funzione. L'operazione ha successo se il file contiene solo interi terminati da un carattere (ad esempio il '.'). In caso di errore nella lettura dei dati in ingresso, la funzione deve restituire una lista vuota.

Per esempio, se il file contiene "numeri.txt": 25 140 6 18 .

La funzione restituisce la lista seguente: 6 -> 18 -> 25 -> 140

Scrivere un main che testi la funzione.

nb.1 Provare a scrivere la funzione nel caso in cui si sfrutti una funzione ausiliaria, e nel caso in cui non si utilizzi.

### Esercizio n.2 - Ripreso da Esame 13 Gennaio 2015

Sia data la seguente struttura

```
struct elem{
    int info;
    elem* pun;
};
```

Scrivere una funzione che ha come argomenti una lista di elementi di tipo elem ed un numero intero N. La funzione elimina dalla lista tutti gli elementi che hanno come campo info un numero intero minore o uguale al numero N.

Per esempio, se la funzione ha come argomenti la lista

2 -> -10 -> 7 -> 0 -> 8 -> 1 e il numero N vale 5, la lista viene modificata come segue:

7 -> 8

Scrivere un main di prova che:

- 1- crea una lista di interi
- 2- stampa la lista di interi
- 3- richiama la funzione passando la lista e un numero a scelta
- 4- stampa la lista modificata dalla funzione

### Esercizio n.3 - Ripreso da Esame 23 Luglio 2015

Sia data la seguente struttura

```
struct elem{
    char lettera;
    elem* pun;
};
```

Scrivere una funzione che, data una stringa di lettere minuscole passata come argomento alla funzione, restituisce una lista di strutture di tipo elem, dove il campo lettera di ciascun elemento contiene una lettera della stringa. La lista deve essere ordinata crescente. Per esempio, se la stringa passata alla funzione è "luna", la funzione restituisce la seguente lista: a -> l -> n -> u

Scrivere un main di prova che:

- 1- Legge una stringa da tastiera
- 2- Chiama la funzione passando la stringa letta
- 3- Stampa la lista creata

Anche qui, scrivere due varianti di funzione, una in cui si usa una funzione ausiliaria, e una dove non si usa.

### Esercizio n.4 - Per casa

Sia data la seguente struttura

```
struct elem{
    char nome[15];
    elem* pun;
};
```

Scrivere una funzione che ha come argomenti una lista di elementi di tipo elem ed un numero intero N. La funzione elimina dalla lista tutti gli elementi che hanno lunghezza stringa minore o uguale al numero N.

Per esempio, se la funzione ha come argomenti la lista

"Casa" -> "Aereo" -> "Grattacielo" -> "Ape" -> "Zucchero"

E il numero passato N = 4, allora la lista viene modificata come segue:

"Aereo" -> "Grattacielo" -> "Zucchero"

Scrivere un main di prova che:

- 1- crea una lista di stringhe (lette da tastiera o messe costanti è indifferente)
- 2- stampa la lista di stringhe
- 3- richiama la funzione passando la lista e un numero a scelta
- 4- stampa la lista modificata dalla funzione

## Esercizio n.5 - Per casa

Sia data la seguente struttura

```
struct elem{
    int info;
    elem* pun;
};
```

Scrivere una funzione che, date due liste L1 ed L2 di elementi di tipo elem passate come argomento alla funzione, restituisce una nuova lista che contiene un elemento per ogni elemento di L1 ed L2 senza duplicati. Non è richiesto uno specifico ordine nella nuova lista.

Per esempio, se la funzione viene chiamata con le liste L1 ed L2

L1: 5 -> 12 -> 9 -> 5

L2: 7 -> 12 -> 3

La funzione, ad esempio usando inserimento in testa restituisce la lista:

L: 7 -> 3 -> 12 -> 9 -> 5

Scrivere un main di prova che testi la funzione