

## Esercizio 1 del 23 Aprile 2018 (Ricerca lineare incerta)

Scrivere un programma che legga da cin una sequenza di interi fino alla sentinella -1 e che, mentre legge gli interi, costruisca una lista concatenata composta da nodi il cui campo info contenga gli interi letti. Successivamente, il programma legge un intero  $n > 0$  a cui seguono  $n$  interi ognuno dei quali va cercato nella lista concatenata costruita prima. La ricerca di un valore  $x$  può andare a buon fine, nel qual caso andrà stampata la frase "il valore  $x$  è in posizione  $m$ " oppure non avrà successo, nel qual caso andrà stampata la frase "il valore  $x$  non si trova nella lista". La posizione di un nodo in una lista concatenata è 0 per il primo nodo, poi 1, 2, eccetera (insomma come per gli elementi di un array).

La lista concatenata da costruire consiste di nodi del seguente tipo struttura nodo:  
`struct nodo {int info; nodo *next;};`

In pratica l'esercizio richiede di scrivere le seguenti 3 funzioni ricorsive:

- a) Una funzione ricorsiva `leggi`, che legge interi da cin fino alla sentinella -1 e produce una lista concatenata che contiene i valori letti (ma che non contiene il -1 finale). La funzione deve seguire la seguente specifica.

PRE=(cin contiene  $x_1..x_{n-1}$  con  $n \geq 0$ )

nodo\* `leggi()`

POST=(restituisce una lista concatenata con  $n$  nodi in cui il campo info del nodo 0 sia  $x_1$ , quello del nodo 1 sia  $x_2$ , eccetera)

- (b) Una funzione ricorsiva `fai_ricerche` che legge gli  $n$  valori e li ricerca nella lista restituita da `leggi`.

PRE=( $L$  è una lista ben formata, cin contiene  $n$  valori, con  $n \geq 0$ , che sono valori da cercare in  $L$ )

void `fai_ricerche(nodo*L, int n)`

POST=(ricerca in  $L$  gli  $n$  valori che sono in cin e per ognuno stampa la frase che si applica)

- (c) `fai_ricerche`, per eseguire la ricerca in  $L$  di un dato valore  $x$ , usa una funzione ricorsiva `ricerca` che segue la seguente specifica:

PRE=( $L$  è una lista ben definita)

int `ricerca(nodo*L, int x)`

POST(se esiste un nodo di  $L$  con campo info= $x$  allora restituisce la sua posizione in  $L$ , altrimenti restituisce -1)

**Correttezza:** fare le prove di correttezza delle 3 funzioni rispetto alle pre e postcondizioni date.

**NOTA:** una lista  $L$  è ben formata se  $L=0$  (lista vuota) oppure  $L$  consiste di un nodo con campo info definito e il cui campo next punta ad una lista ben formata.