

Esercizio 1 del 3/3/2014 da consegnare corretto entro il 10/3 a mezzanotte

Si legge da "input" gli interi k, y e dim, in cui y può essere un intero qualsiasi, mentre k e dim sono maggiori di 0, dopo di che si costruisce una lista concatenata con dim nodi del tipo seguente:

```
struct nodo{int info1,info2; nodo* next; nodo(int a=0, int c=0, nodo* b=0){info1=a;info2=c; next=b;}};
```

ed in cui nel campo info1 di ciascuno dei dim nodi si deve inserire un valore letto da "input" (ce ne sono dim) mentre in info2 si mette semplicemente la posizione del nodo nella lista, cioè 0,..., dim-1.

A questo punto si richiedono 2 funzioni F1 ed F2 entrambe col seguente prototipo:

nodo* F1/2(nodo*&L, int y, int k, int & v) e che soddisfano le seguenti coppie di pre- e post-condizioni:

PRE_F1=(L è una lista corretta, y e v sono interi qualsiasi, mentre k>0, L=vL)

POST_F1=(se vL contiene meno di k nodi con campo info=y, allora L=vL e F1 restituisce 0, altrimenti, L è la lista ottenuta da vL togliendo da vL gli ultimi (per posizione) k nodi con campo info1=y, mentre F1 restituisce la lista dei nodi tolti da vL in ordine di posizione in vL.....è possibile che sia anche necessario aggiungere qualcosa su v)

PRE_F2=PRE_F1

POST_F2==(se vL contiene meno di k nodi con campo info=y, allora L=vL e F2 restituisce 0, altrimenti, L è la lista ottenuta da vL togliendo da vL gli i primi (per posizione) k nodi con campo info1=y, mentre F2 restituisce la lista dei nodi tolti da vL in ordine di posizione in vL.....è possibile che sia anche necessario aggiungere qualcosa su v)

Viene fornito un main incompleto in cui compaiono le invocazioni a F1 e F2. Dopo ciascuna invocazione il main stampa le liste prodotte dall'invocazione. Si osservi che la lista creata inizialmente (con dim nodi) deve essere duplicata in modo da invocare F1 e F2 con 2 liste diverse sebbene identiche.

Esermpio: Sia L la lista costruita leggendo i dim interi da "input", come segue:

L=(3,0),(2,1),(0,2),(3,3),(0,4),(0,5)(1,6),(0,7),(2,8),(0,9),(3,10), supponiamo anche che k=2 e y=3.

Evidentemente L contiene almeno 2 nodi con info1=3 (ne contiene 3) e quindi sia F1 che F2 modificheranno la lista che ricevono in input.

F1 restituirà col return la lista (3,3),(3,10), mentre L sarà diventata:
(3,0),(2,1),(0,2),(0,4),(0,5)(1,6),(0,7),(2,8),(0,9)

mentre F2 restituirà col return (3,0),(3,3) con L che sarà diventata:
=(2,1),(0,2),(0,4),(0,5)(1,6),(0,7),(2,8),(0,9),(3,10).

Se k fosse 4, entrambe le funzioni restituirebbero 0 lasciando L inalterata. Se k fosse 3, invece le 2 funzioni avrebbero identico output, sia col return che col parametro L.