Programmazione 2008-09 II Compitino 11/3/2009

E' indispensabile consegnare un testo leggibile. Evitare di spezzare una funzione su più pagine. In caso serva si può usare il foglio di protocollo aperto a giornale. Elaborati non leggibili non verranno corretti.

Chi copia perde il diritto di accedere al I appello scritto.

Si tratta di cercare le occorrenze di un carattere y in una lista concatenata i cui nodi hanno tipo struct nodo{char info; nodo* next;};. Vogliamo che la funzione che esegue questa ricerca sia ricorsiva e soddisfi alle seguenti proprietà:

1) F deve comportarsi nel modo seguente:

deve controllare che la lista contenga almeno k nodi con campo informativo uguale a y e solo in questo caso deve stampare le posizioni di questi nodi dall'ultimo al primo (in ordine decrescente), ma attenzione, non deve stampare tutte le posizioni, ma solo 1 si e 1 no, come illustrato dal seguente esempio.

Esempio. *Supponiamo che y appaia nella seconda, terza, quarta, ottava, dodicesima e tredicesima posizione della lista. Se k=7, F non dovrebbe stampare nulla. Se invece k=3, allora F dovrà stampare 12, 4 e 2.*

Attenzione: La selezione dei nodi da stampare deve sempre contenere il nodo contenente y, in posizione minima (quello in posizione 2 nell'esempio). Ovviamente questo determina la selezione di tutti gli altri nodi (nell'esempio, dopo 2 si deve saltare una posizione, cioè 3, e il nodo successivo, cioè 4, è da stampare, poi si salta l'8 e il 12 è da stampare e il 13 non va stampato). L'ordine di stampa dei nodi selezionati è decrescente: 12, 4 e 2.

2) Il prototipo di F deve soddisfare questa specifica: ??? F(nodo*x, int pos, char y, int k, ???), dove x punta al nodo corrente della lista, pos è la posizione del nodo x nella lista (0 se è il primo nodo, 1 se è il secondo e così via), y è il carattere da cercare, k è il numero minimo di occorrenze di y che devono essere presenti nella lista per effettuare la stampa delle posizioni dei nodi che contengono y. I punti interrogativi servono ad indicare che tipo del valore restituito da F è lasciato libero e che altri parametri formali possono venire aggiunti a F.

Attenzione: introdurre parametri passati per riferimento solo se si è certi che sia essenziale farlo.

Si chiede di scrivere con precisione la pre- e soprattutto la post-condizione di F e di scrivere la dimostrazione di correttezza di F secondo il consueto schema induttivo.

MIA SOLUZIONE (ma ne esistono altre):

```
bool F(nodo * x, int pos, char y, int k, bool stampa)
{
 if(!x)
   if(k<=0)
     return true;
   else
     return false;
 if(x->info==y)
   {
     if(F(x->next,pos+1,y,k-1,!stampa))
          {
            if(stampa)
                 cout <<pos<<endl;
            return true;
          }
    else
          return false;
   }
 else
   return F(x->next,pos+1,y,k,stampa);
}
```