Compito di Programmazione, esercizi di teoria

20 settembre 2016

Cognome, Nome e Matricola:

(1) (2 punti) Data la seguente funzione ricorsiva, inserire appropriate PRE e POST

```
//PRE= ??
int F(Nodo* a) {
if(!a)
  return 0;
if (a->left)    return 1+F(a->left);
if(a->left || a->right) return 2+F(a->left)+F(a->right);
return  3+F(a->left)+F(a->right);
} //POST=??
```

PRE=(a albero corretto)

POST=(viene percorso il cammino più a sinistra possibile, contando 1 i nodi con figlio sinistro, 2 quelli con solo figlio destro e 3 quelli senza figli)

(2) (3 punti) Considerare il seguente programma e spiegate se è corretto o no. E' necessario motivare la risposta ed è utile corredarla con un disegno che spieghi le relazioni delle variabili in memoria. In caso pensiate sia corretto, dite cosa stampa. Risposte non motivate avranno voto 0.

```
int* f(int **p){int b=3,*x=&b; **p=*x; *x=**p; return x; }
main() {int y=5, b=2,*q=&b; *f(&q)=y*2; cout<<y<<b<<*q;}
```

*La funzione f restituisce x che punta alla variabile b che è locale della funzione stessa. Insomma restituisce ad una variabile che viene deallocata alla fine della funzione, quindi si tratta di un dangling pointer. Che x punti a b è dovuto alla dichiarazione int*x=&b; poi **p =*x; assegna 3 all b del main, mentre *x=**p; assegna 3 alla b di f che aveva già R-valore 3. Quindi x continua a puntare a b di f.*

3) (2 punti) Un programma può essere scritto su più file. Supponiamo di avere un programma scritto su due file F1.cpp e F2.cpp. Supponiamo anche che le funzioni scritte su ciascuno dei due file usino uno stesso tipo struttura P ed una stessa funzione f che sono come segue:

```
struct P{int a,b; P* next;};
P*  f(P* x){.....}
```

Spiegare cosa di P e di f deve essere scritto su F1.cpp e su F2.cpp in modo da permettere la compilazione di ciascun file da solo ed anche la compilazione dei 2 file insieme.

Ogni file deve contenere la stessa definizione completa della struct P, mentre la funzione f deve essere definita completamente in un file e i suo prototipo deve apparire nell'altro file prima dell'uso. La maniera standard per ottenere queste cose è di avere un file .h che contiene la def. di P e il prototipo di f. Includendo questo file .h in entrambi i file .cpp si ottiene quanto richiesto. Resta ovviamente che 1 solo dei 2 file .cpp deve contenere la def. completa di f.