1. FONDAMENTI DI INFORMATICA II e modulo B (tempo 2h) – 8 Marzo 2011

Esercizio n. 1 (punti 20)

Al fine di evitare costosi ricoveri in ospedale, l'azienda ospedaliera ha deciso di mandare le infermiere a casa dei pazienti che hanno bisogno di cure semplici. L'azienda ha un file di testo `visite.txt` che contiene le visite che verranno effettuate durante la settimana (al massimo 100); per ogni visita si hanno:

- Paziente: nome del paziente. Dato di tipo stringa contenente al più 20 caratteri, senza spazi.
- Infermiera: nome dell'infermiera che visiterà il paziente. Dato di tipo stringa con al più 20 caratteri, senza spazi
- Giorno: Numero intero che rappresenta il giorno della settimana in cui avviene la visita (da 1 per lunedì, a 7 per domenica);

Si scriva un programma C che:

a) A partire dal file, crei un albero binario di ricerca T in memoria centrale (supponendo che la memoria heap sia sufficiente) che riporta il contenuto del file, ordinato sul campo Paziente e a parità di paziente ordinato sul campo Giorno; la funzione del punto a) riceve il puntatore a file, e restituisce l'albero creato - un puntatore di tipo tree – (ed eventuali altri parametri a vostra scelta);

b) Accedendo all'albero T, si crei un array V di strutture che riporta, per ogni struttura, solo nome del Paziente e Giorno di visita (si crei V in modo che sia ordinato sul campo Paziente e a parità di Paziente sul campo Giorno); la funzione al punto b) riceve il puntatore T e l'array V (ed eventuali altri parametri a vostra scelta);

a) Accedendo all'array V, determinare qual è il paziente che riceve più visite nella settimana e stamparne nel file `paziente.txt` il nome; la funzione al punto c) riceve l'array e il puntatore al file aperto in uscita.

È possibile utilizzare ***librerie C*** (ad esempio per le stringhe). Nel caso si strutturi a moduli l'applicazione qualunque ***libreria utente*** va riportata nello svolgimento.

Esercizio n. 1 (punti 6)

Dato il seguente codice in linguaggio C, dove il tipo **`tree`** rappresenta un albero binario di ricerca di interi:

```
int member(int i, tree T)
{ while (T!=NULL)
         {if (i==T->value) return 1;         /* (istr. dom.) */
           else
              if (i<T->value) return member(i,T->left);
              else return member(i,T->right);
          }
  return 0;
}

int proc(tree T1, tree T2)
{ int c=0;
  if (T1!=NULL)
          {if (member(t1->value, T2)) c++;
           c=c+proc(T1->left,T2)+ proc(T1->right,T2)
           }
 return c;
}
```

a) Si descriva cosa fanno la funzione **`proc`** indicata e la funzione **`member`**.

b) Si discuta la complessità del codice sapendo che l'albero **`T1`** contiene **`N`** elementi e **`T2`** contiene **`M`** elementi. Si individuino caso migliore e peggiore e li si discuta dal punto di vista della complessità asintotica (in particolare andamento asintotico del numero di esecuzioni complessive del test sottolineato nella funzione **`member`**).

Esercizio n. 3 (punti 4)

Domanda su OOP