FONDAMENTI DI INFORMATICA (II, modulo B, A+B) (2h, punti 30/30) – 22 Giugno 2011

Esercizio 2 – PROGRAMMAZIONE IN C (punti 20)

Un file di testo (FILM.TXT) contiene record relativi a produzioni cinematografiche (si sa che sono al massimo 30). Ogni linea riporta, per ciascun progetto cinematografico, un codice del film (valore numerico intero), nome del regista, budget richiesto (valore numerico intero). Un secondo file testo (PREVENTIVI.TXT) riporta le spese preventivate da ciascuna produzione, sono riportati il codice del film, il nome del produttore, la sua offerta di spesa (valore numerico intero). Per lo stesso progetto cinematografico ci possono essere state più offerte di produttori diversi, **e ve ne è almeno una**.
I nomi (di registra e produttore) sono stringhe di al massimo di 20 caratteri.

Si scriva un programma C organizzato in (almeno tre) funzioni invocate dal main e dedicate rispettivamente a:

a) Per ogni record letto dal primo file, cerchi nel secondo file l'elemento che ha lo stesso codice film e l'offerta di spesa maggiore e restituisca una struttura composta dai campi: codice del film, nome del regista, budget richiesto, nome del produttore (che ha fatto l'offerta maggiore) e la sua offerta. Tale funzione ha come parametri il puntatore al secondo file e restituisce un tipo `struct` da definire in base ai campi indicati e richiesti;

b) Inserisca poi la struttura restituita in un albero binario di ricerca T, ordinato sul campo codice; tale funzione ha come parametri la struttura restituita dal punto precedente, il puntatore all'albero T (più eventuali altri a scelta) e restituisce il puntatore all'albero creato;

c) Stampi ordinatamente in base al codice del film il contenuto dell'albero T su un file di tipo testo (OUTPUT.TXT, da consegnare con i sorgenti ed eseguibile), che riporta, per ciascuna riga, il codice del film, il nome del regista, il budget richiesto, il nome del produttore (che ha fatto l'offerta maggiore) e la sua offerta. Tale funzione ha come parametri il puntatore al file di uscita e il puntatore all'albero T (più eventuali altri a scelta) e restituisce `void`.

d) Per chi svolge il compito A+B: Inserisca in un vettore V ordinato sul campo codice film gli elementi dell'albero T che hanno il campo budget maggiore del campo offerta. Tale funzione ha come parametri il puntatore all'albero T e il vettore V (più eventuali altri a scelta) e restituisce un `int` che rappresenta il numero di elementi inseriti in V (facoltativo: si stampi a video poi il contenuto del vettore).

TEORIA – modulo B

Esercizio n. 1 (punti 6)

Dato il seguente codice in linguaggio C, dove il tipo **`tree`** rappresenta un albero binario di interi e il tipo **`list`** rappresenta una lista di interi (ordinata):

```
int c=0;

int ricerca(int i, list L)
{ if (L!=NULL)
          {if (i==L->value) return 1;          /* (istr. dom.) */
           else return (ricerca(i,L->next));}
  else return 0;
}

void funct(tree T, list L)
{ while (T!=NULL)
           { if (ricerca(T->value, L)) c++;
             funct(T->left, L);
             funct(T->right,L); }
}

void main(void)
{ /* crea T ed L ... */
  funct(T, L);
  printf("%d", c); }
```

a) Si descriva cosa fanno la funzione **`funct`** indicata e la funzione **`ricerca`** e il tipo di algoritmi che realizzano sulle due strutture dati.

b) Si discuta la complessità del codice sapendo che l'albero **`T`** contiene **`M`** elementi e la lista **`L`** contiene **`N`** elementi. Si discuta la **complessità** (in particolare andamento asintotico del numero di esecuzioni complessive del test sottolineato nella funzione **`ricerca`**), in funzione di **`M`** ed **`N`**, individuando, se necessario, caso migliore e peggiore. Si motivino adeguatamente tali casi, se individuati.

Esercizio n. 3 (punti 3)

Domanda su OOP e Java