

# 1. FONDAMENTI DI INFORMATICA II (tempo 2h) – 11 Marzo 2009

## Esercizio n. 1 (punti 7)

Dato il seguente codice in linguaggio C, dove il tipo **tree** rappresenta un albero binario di interi: e **ordins** la funzione di inserimento in un albero binario di ricerca:

```
#include <stdio.h>
#define N 10
int vet[N]= {1,2,3,4,5,6,7,8,9,10};

int ricerca(int e, int V[], int i, int j)
{ int m=(i+j)/2;
  int trovato=0;
  while ((i<=j) && (!trovato))
    { if (e==V[m]) trovato=1;
      else if (e<V[m]) j=m-1;
        else i=m+1;
      m=(i+j)/2; }
  return trovato;
}

int visita(tree T)
{ if (T==NULL)return 0;
  else
    if (ricerca(T->value, vet, 0, N-1))
      return 1+visita(T->left)+visita(T->right);
    else return visita(T->left)+visita(T->right);
}

void main(void)
{ int i, t=0;
  tree T =NULL;

  scanf("%d",&i);
  while (i>0)
    { T=ordins(i,T);
      scanf("%d",&i);
    }

  printf("%d",visita(T));
}
```

- Si indichi cosa fa questo frammento di programma e le funzioni **ricerca** e **visita** in particolare;
- Se ne indichi la complessità in termini di numero di test condizionali dell'istruzione **while** (test sottolineato) nella funzione **ricerca**, ipotizzando che da input siano stati dati **M** valori che sono stati inseriti nell'albero **T**, in funzione di **M** ed **N**. Discutere il caso migliore e il caso peggiore.

## **Esercizio n. 2 (punti 23)**

Una polisportiva che gestisce corsi di nuoto per bambini mantiene i dati sui corsi in un file binario, CORSI.DAT, che memorizza per ciascun corso la denominazione (stringa, ad esempio “cavallucci”, “delfini”, ...), l’orario espresso da due interi (ore e minuti), il nome dell’istruttore (stringa), il numero massimo di bambini previsti per il corso (intero).

Un secondo file binario contiene i dati dei bambini iscritti (ISCRITTI.DAT); per ciascun bambino iscritto si memorizzano nome e cognome (stringhe), nome del corso (stringa) ed orario (ore e minuti, come interi).

Si chiede di scrivere un programma C che chiami tre funzioni (da definire) dedicate rispettivamente a:

- a) creare una lista L in memoria centrale, a partire dal file CORSI.DAT e ISCRITTI.DAT, che memorizzi per ciascun corso la sua denominazione, l’orario (ore e minuti), il nome dell’istruttore, numero massimo di bambini previsti per il corso e il numero effettivo di bambini partecipanti (questo ultimo determinato in base al contenuto del file ISCRITTI.DAT); non si richiede che L sia ordinata su alcun campo;
- b) a partire da L, produrre una seconda lista L2 (ordinata in senso crescente in base al numero di iscritti) in cui sono inseriti i corsi di L che hanno un numero di iscritti maggiore del massimo previsto.
- c) stampi in modo ricorsivo su un file di tipo testo TROPPI.TXT il contenuto della lista L2.

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.

## 2. FONDAMENTI DI INFORMATICA II (tempo 2h) – 11 Marzo 2009

### Esercizio n. 1 (punti 7)

Dato il seguente codice in linguaggio C, dove il tipo **tree** rappresenta un albero binario di interi: e **ordins** la funzione di inserimento in un albero binario di ricerca:

```
#include <stdio.h>
#define N 10
int vet[N]= {3,4,5,7,1,2,6,8,9,10};

int trova(int e, int V[], int i, int j)
{ int trovato=0;
  while ((i<=j) && (!trovato))
    if (e==V[i]) trovato=1;
    else i++;
  return trovato;
}

int visita(tree T)
{ if (T==NULL)return 0;
  else
    if (trova(T->value, vet, 0, N-1))
      return 1+visita(T->left)+visita(T->right);
    else return visita(T->left)+visita(T->right);
}

void main(void)
{ int i, t=0;
  tree T =NULL;

  scanf("%d",&i);
  while (i>0)
    { T=ordins(i,T);
      scanf("%d",&i);
    }

  printf("%d",visita(T));
}
```

- c) Si indichi cosa fa questo frammento di programma e le funzioni **trova** e **visita** in particolare;
- d) Se ne indichi la complessità in termini di numero di test condizionali dell'istruzione **while** (test sottolineato) nella funzione **ricerca**, ipotizzando che da input siano stati dati **M** valori che sono stati inseriti nell'albero **T**, in funzione di **M** ed **N**. Discutere il caso migliore e il caso peggiore.

## Esercizio n. 2 (punti 23)

Si stanno organizzando delle escursioni scolastiche di una scuola. A tale scopo nel file binario GITE.DAT sono memorizzati per ciascuna escursione la destinazione (stringa), la data espressa da due interi (giorno e mese), il nome dell'accompagnatore (stringa), il numero massimo di bambini previsti per l'escursione (intero).

Un secondo file binario contiene i dati degli alunni iscritti alle escursioni (ISCRITTI.DAT); per ciascun alunno iscritto si memorizzano nome e cognome (stringhe), destinazione dell'escursione (stringa) e data (giorno e mese, come interi).

Si chiede di scrivere un programma C che chiami tre funzioni (da definire) dedicate rispettivamente a:

- d) creare una lista L in memoria centrale, a partire dal file GITE.DAT e ISCRITTI.DAT, che memorizzi per ciascuna escursione la sua destinazione, la data (giorno e mese), il nome dell'accompagnatore, il numero massimo di alunni previsti per l'escursione e il numero effettivo di alunni partecipanti (questo ultimo determinato in base al contenuto del file ISCRITTI.DAT); non si richiede che L sia ordinata su alcun campo;
- e) a partire da L, produrre una seconda lista L2 (ordinata in senso crescente base al numero di iscritti) in cui sono inserite le escursioni di L che hanno un numero di iscritti maggiore del massimo previsto.
- f) stampi in modo ricorsivo su un file di tipo testo TANTI.TXT il contenuto della lista L2.

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.