

**Esercizio 2 – PROGRAMMAZIONE IN C (punti 20)**

Un file binario (PROGETTI.DAT) contiene record relativi a progetti edili. Ogni record riporta un codice (intero), nome del committente, budget previsto (intero). Un secondo file binario (OFFERTE.DAT) riporta le offerte ricevute: per ciascun codice progetto, sono riportati il codice, il nome dell'offerente e l'offerta (intero). Per lo stesso progetto ci possono essere state più offerte.

I nomi (di committente e offerenti) 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:

- 1) Scandendo i due file, produrre una lista L, ordinata sul campo codice, che riporta per ciascun progetto, il codice del progetto, il nome del committente, il nome dell'offerente e la sua offerta. Si inseriscano in lista solo le offerte minori del budget previsto per il progetto. Tale funzione ha come parametri i due puntatori a file e il puntatore alla lista L (più eventuali altri a scelta) e restituisce il puntatore alla lista creata;
- 2) A partire da L, determinare qual è il progetto che ha l'offerta minore e stamparne a video i dati (codice, nome committente, nome offerente, offerta). Tale funzione ha come parametri il puntatore alla lista L (più eventuali altri a scelta) e restituisce void;
- 3) Scandendo la lista L, determinare quante offerte ha fatto la ditta "CESA" e stampare tale valore a terminale. Tale funzione ha come parametri il puntatore alla lista L, la stringa che rappresenta il nome della ditta ("CESA"), (più eventuali altri a scelta) e restituisce un valore intero.

## TEORIA – modulo B

### Esercizio n. 1 (punti 7)

Dato il seguente codice in linguaggio C, dove il tipo **list** rappresenta una lista di interi, e **tree** un tipo albero binario (di ricerca). Le variabili **L** di tipo **list** e **T** di tipo **tree** siano state create leggendo da input valori che sono stati inseriti poi nelle strutture dati in modo che siano rispettivamente una lista ordinata e un albero binario di ricerca.

```
#include <stdio.h>
. . .
list crea_lista(list);
tree crea_tree(tree);

list L=NULL;
tree T=NULL;

int same(list l, tree t)
{ int trovato=0;
  tree aux=t;
  while (l!=NULL)
    { t=aux;
      trovato=0;
      while ((t!=NULL) || (!trovato))
        if (l->value==t->value) trovato=1;
        else if (l->value <= t->value) t=t-> left;
            else t=t->right; }
      if (trovato) l=l->next;
      else l=NULL;
    }
  return trovato;
}

void main(void)
{ L=crea_lista(L);
  T=crea_tree(T);

  printf("%d", same(T,L));
}
```

- Si indichi cosa fa questo frammento di programma e la funzione **same** in particolare;
- Se ne indichi la complessità in termini di numero di esecuzioni del test condizionale sottolineato, ipotizzando che da input siano stati inseriti N valori in lista e M valori nell'albero e che gli N valori in lista siano presente anche nell'albero T. Si discutano caso migliore e caso peggiore.

### Esercizio n. 3 (punti 3)

Domanda di teoria su OOP.