

Esercizio 2 – PROGRAMMAZIONE IN C (punti 20)

Un file binario (IMPIANTI.DAT) contiene record relativi a progetti di impianti termo-idraulici. Ogni record riporta, per ciascun progetto di impianto, un codice impianto (intero), nome del progettista, budget previsto (intero). Si sa che all'interno di tale file vi sono i dati di al massimo 30 impianti.

Un secondo file binario (OFFERTE.DAT) riporta le offerte ricevute: per ciascun codice impianto, sono riportati il codice impianto, il nome dell'offerente e l'offerta (intero). Per lo stesso progetto di impianto ci possono essere state più offerte.

I nomi (di progettista 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:

- a) Caricare il contenuto del primo file in un vettore V in memoria centrale, ordinare tale vettore sul campo codice impianto e stampare a video il contenuto di V. Per realizzare l'ordinamento, si utilizzi la funzione di libreria `qsort`; si ricorda il prototipo della `qsort`:

```
void qsort(void *base, size_t n, size_t width, int (*fcmp)(void *e1, void *e2));
```

La funzione al punto a) riceve come parametri il puntatore al file e il vettore V, più eventuali parametri a scelta, e restituisce `void`;

- b) Scandendo il secondo file, per ciascuna offerta letta, cercare in V l'elemento che ha lo stesso codice impianto dell'offerta (realizzare tale ricerca come ricerca binaria, o come ricerca sequenziale, a propria scelta). Nel caso di ricerca con successo, inserire poi in un albero binario di ricerca T, anch'esso ordinato sul campo codice, il codice impianto, il nome del progettista, il budget, il nome dell'offerente e la sua offerta. Si inseriscano nell'albero solo le offerte minori del budget previsto per l'impianto. Tale funzione ha come parametri il puntatore al secondo file, il vettore V e il puntatore all'albero T (più eventuali altri a scelta) e restituisce il puntatore all'albero creato;
- c) Stampare ordinatamente in base al codice impianto 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 impianto, il nome del progettista, il budget, il nome dell'offerente 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`.

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

int funct(tree T, list L)
{ int c=0;
  while (L!=NULL)
    { if (ricerca(L->value, T)) c++;
      L=L->next;    }
  return c;
}
```

- Si descriva cosa fanno la funzione **funct** indicata e la funzione **ricerca** e il tipo di algoritmi che realizzano.
- 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 asintotica** (in particolare andamento asintotico del numero di esecuzioni complessive del test sottolineato nella funzione **ricerca**), individuando, se necessario, caso migliore e peggiore. Si motivino adeguatamente tali casi, se individuati.

Esercizio n. 3 (punti 3)

Domanda su OOP