

# Fondamenti di Informatica

Prof. M. Gavanelli, E. Lamma

16 Giugno 2016

## Esercizio 1 (Punti 15 su 31) (1h e 30 min)

In un file binario `alimenti.bin`, sono scritti le giacenze dei prodotti alimentari di un piccolo grossista alimentare. Per ciascun prodotto, il file `alimenti.bin` contiene

- il nome del prodotto (stringa di 50 char),
- il numero di scatole (intero).

Ad esempio, per il prodotto *cracker* si ha:

```
"cracker" 439
```

perché ci sono 439 scatole di cracker.

In un secondo file, di testo, `pranzo.txt`, sono scritti gli alimenti ordinati da un supermercato cliente del grossista. Ad esempio:

```
"cracker" 40
```

```
"pasta" 100
```

```
"tonno" 80
```

se si è ordinato 40 scatole di cracker, 100 di pasta e 80 di tonno.

Si realizzi un programma C, organizzato in almeno tre funzioni, rispettivamente dedicate a:

- a partire dal file `alimenti.bin`, creare un albero binario di ricerca T in memoria centrale che contiene i dati dei prodotti, ordinato in base al nome del prodotto; la **funzioneA** riceve come parametri:
  - il puntatore al file,
  - il puntatore a T (inizializzato a `NULL` nel `main`),più eventuali parametri a scelta, e restituisce il puntatore alla radice dell'albero T;
- *FACOLTATIVO*: stampare l'albero T a video; la **funzioneB** riceve come parametri:
  - il puntatore a T,

più eventuali parametri a scelta, e restituisce `void` ;

- a partire dal file `pranzo.txt` e dall'albero `T` creato, determinare quali alimenti ordinati possono essere consegnati (sono solo quelli ordinati in quantità minore o uguale a quella disponibile presso il grossista), e stampare i loro nomi su un file di uscita `output.txt` da consegnare con i codici sorgente; la **funzioneC** riceve come parametri

- il puntatore al secondo file,
- il puntatore a `T`,

più eventuali parametri a scelta, e restituisce `void`; si noti che la quantità di un prodotto ordinato può essere maggiore di quella disponibile in magazzino; in questo caso il prodotto ordinato non è inserito nella consegna.

**NOTA BENE:** Si consegnino i sorgenti e il file di uscita generato. 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 2 (Punti 3 su 31) (15 min)

**NOTA BENE:** Per questo esercizio si consegna la soluzione in un file `complex.txt`.

Sia data la seguente funzione `fun` che riceve un carattere e una lista non ordinata di caratteri

```
int fun(char i, list L)
{ if (L==NULL) return 0;
  else if (i==L->value) return (1 + fun(i,L->next));
    else
      return fun(i,L->next);
}
```

Si indichi cosa fa la funzione `fun` e se è ricorsiva o iterativa. Se ne valuti anche la complessità asintotica come numero di test `i==L->value`, motivando adeguatamente, ed individuando eventuali casi se necessario.